Robust 3D Human Pose Estimation from Single Images or Video Sequences

Chunyu Wang, Yizhou Wang, Zhouchen Lin, *Fellow, IEEE* and Alan L. Yuille

Abstract—We propose a method for estimating 3D human poses from single images or video sequences. The task is challenging because: (a) many 3D poses can have similar 2D pose projections which makes the lifting ambiguous, and (b) current 2D joint detectors are not accurate which can cause big errors in 3D estimates. We represent 3D poses by a sparse combination of bases which encode structural pose priors to reduce the lifting ambiguity. This prior is strengthened by adding limb length constraints. We estimate the 3D pose by minimizing an L_1 norm measurement error between the 2D pose and the 3D pose because it is less sensitive to inaccurate 2D poses. We modify our algorithm to output K 3D pose candidates for an image, and for videos, we impose a temporal smoothness constraint to select the best sequence of 3D poses from the candidates. We demonstrate good results on 3D pose estimation from static images and improved performance by selecting the best 3D pose from the K proposals. Our results on video sequences also show improvements (over static images) of roughly 15%.

Index Terms—3D human pose estimation, sparse basis, anthropomorphic constraints, L_1-norm penalty function

1 Introduction

Human pose estimation is an important problem in computer vision which has received much attention because many applications require human poses as inputs for further processing [1] [2] [3] [4]. Representing human motion by poses is arguably better than using low-level features [5] because it is more interpretable and compact [3].

In recent years there has been much progress in estimating 2D poses from images [6] [7] [8] [9] and videos [10] [11] [3]. A 2D pose is typically represented by a set of body joints [6] [12] or body parts [7] [8]. Then a graphical model is formulated where the graph node corresponds to a joint (or body part) and the edges between the nodes encode spatial relations. This can be extended to video sequences [10] [11] [3] to explore the temporal cues for improving performance. Nevertheless, it seems more natural to represent humans in terms of their 3D poses because this is invariant to viewpoint and the spatial relations between joints are simpler.

But estimating 3D poses from a single image is difficult for many reasons. Firstly, it is an under-constrained problem because we are missing depth information and many 3D poses can give rise to similar 2D poses after projection into the image plane. In short, there are severe ambiguities when "lifting" 2D poses to 3D. Secondly, estimating 3D poses requires first estimating the 2D joint locations in images which can make mistakes and can result in bad estimates for some joints. All these issues can degrade 3D pose estimation if not dealt with carefully. Thirdly, the situation becomes even worse if the camera parameters are unknown which is typically the case in real applications. Hence we must estimate the 3D pose and camera parameters jointly [13] which leads to non-convex formulations which is difficult.

1.1 Method Overview

We present an overview of our approach illustrating our main contributions. This builds on, and gives a more detailed description of, our preliminary work [14] which estimated 3D poses from a single image. Our new contributions include extending the work to output K candidate 3D poses, to improve our 3D pose estimates by post-processing, and to estimate 3D poses from videos exploiting temporal cues.

We break our approach down into five components described below. These are: (i) the 3D pose prior, (ii) the measurement error between the 2D pose and the 3D projection, (iii) the inference algorithm for estimating 3D pose and camera parameters using the alternate direction method (ADM), (iv) our method for outputting K candidate 3D poses, and (v) the extension to video sequences.

1.1.1 The Prior for 3D Poses

We represent 3D poses by a linear combination of basis functions. This is partly motivated by earlier work [13] which used PCA to estimate the bases from a 3D dataset. By contrast, we learn the bases by imposing a sparsity constraint. This implies that for a typical 3D pose only a small number of the basis coefficients will be non-zero. We argue that sparse bases are more natural than PCA for representing 3D poses because the space of 3D poses is highly non-linear. The sparsity requirement puts a strong prior on the space of 3D poses.

- *Chunyu Wang is with Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, Beijing 100871, P.R.China. E-mail: wangchunyu@pku.edu.cn*
- *Yizhou Wang is with Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, Beijing 100871, P.R. China, and the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai 200240, P.R.China. E-mail: Yizhou.Wang@pku.edu.cn*
- *Zhouchen Lin is with Key Laboratory of Machine Perception (MOE), School of EECS, Peking University, Beijing 100871, P.R. China, and the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University, Shanghai 200240, P.R. China. E-mail:zlin@pku.edu.cn*
- *Alan L. Yuille is Bloomberg Distinguished Professor of Cognitive Science and Computer Science at Johns Hopkins University, Baltimore, MD.*

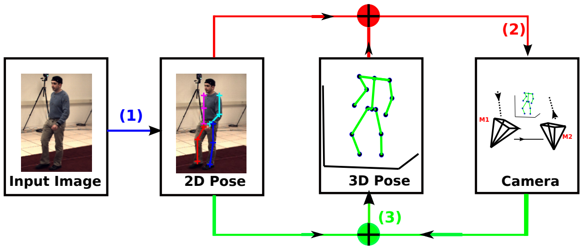


Fig. 1. **Method overview.** (1) On a test image, we first estimate the 2D joint locations and obtain an initial 3D pose by the mean pose in the training data. This initializes an alternating direction method which recursively alternates the two steps (i.e. steps 2 and 3). (2) Estimate the the camera parameters from the 2D pose and current estimate of the 3D pose. (3) Re-estimate the 3D pose using the 2D pose and the current estimates of the camera parameters. The algorithm converges when the difference of the estimates is small.

But this prior needs to be strengthened because unrealistic 3D poses can still have sparse representations.

To strengthen the sparsity prior we build on previous work on anthropomorphic constraints [15] [16] which shows that the limb length ratios of people are similar and can be exploited to estimate 3D pose (but when used by themselves anthropomorphic constraints have ambiguities). We were motivated to use anthropomorphic constraints by observing that many of the unrealistic 3D poses often violate them. Hence we supplement the sparse basis representation with hard limb length ratio constraints to discourage incorrect poses.

1.1.2 Robust Measurement Error: L_1-norm

The difficulty of 3D pose estimation is that we frequently get large errors, or outliers, in the positions of some 2D joints. We use an L_1 norm to compute the measurement error between the detected 2D poses and the projections of the 3D poses. We argue that this is better than using the standard L_2 norm because the L_1 is much more robust to large errors, or "outliers", in the 2D pose estimates. The greater robustness of the L_1 norm is well-known in the statistics literature[17].

1.1.3 Algorithm to minimize the objective function

We formulate an objective function by combining the measurement error with the sparsity penalty and the anthropomorphic constraints. Our inference algorithm minimizes this objective function to jointly estimate the 3D pose and the camera parameters. We first initialize the 3D pose and then estimate the camera parameter and the 3D pose alternatively (with the other fixed). See Fig. 1. The estimations (for 3D pose and camera separately) are done using the alternating direction method (ADM) which yields a fast algorithm capable of dealing with the constraints.

1.1.4 Multiple candidate proposals and selection

For a single image, our best estimate of the 3D pose is usually good [14] but not perfect. There are two main reasons for this. Firstly, the problem is highly non-convex so our estimation algorithm can get trapped in a local minimum. Secondly, our basis functions are learnt from 3D pose datasets of limited size, which may cause some errors.

To address this issue, we modify our approach to output a set of K 3D poses, where K takes a default value of eight. Our experiment shows that one of our top eight candidates is typically very close to the groundtruth, but the best candidate may not be the one that minimizes our objective function, see Fig. 4. We show that we can improve performance by a second stage where we select the candidate that best satisfies the anthropomorphic constraints.

1.1.5 Selecting the Best Pose: Temporal Smoothness

If we have a video then we can obtain candidate 3D proposals for each frame and select them by imposing temporal smoothness. This assumes that the 3D pose does not change much between adjacent frames. We select the 3D pose by minimizing an objective function which imposes temporal consistency and agreement with the 3D pose priors.

In summary, the main novel contributions of this paper are the use of sparsity to obtain a prior for 3D poses which can effectively reduce the 3D pose lifting ambiguities. Our method for supplementing this with anthropomorphic constraints is also novel (but different forms of limb length constraints have been explored in history [16]). The use of the L_1 norm to penalize measurement errors is new for this application (but well-known in the statistics literature [17]). Our use of the ADM algorithm to impose non-linear constraints is novel for this application. Our work on estimating the K best poses builds on prior work, e.g., [18], but they do not extend this to 3D poses and video sequences. The first part of this work (single 3D pose estimation) was first presented in our preliminary work [14] but in less detail.

The paper is organized as follows: We first review related work in section 2. Section 3 and section 4 describe the details of image/video based pose estimation, respectively. The basis learning method is discussed in section 5. Sections 6 and 6.4 give the experiment results. We conclude in section 7. Appendix A presents the optimization method.

2 Related Work

2.1 Related Work on 3D Pose Estimation

Existing work on 3D pose estimation can be classified into four categories by their inputs. The first class takes images and camera parameters as inputs. We only list a few of them here due to space limitations. Please see [19] for a more comprehensive overview. Lee *et al.* [20] first parameterize the body parts by truncated cones. Then they optimize the rotations of body parts to minimize the silhouette discrepancy between the model projections and the image by a sampling algorithm. The most challenging factor for 3D pose estimation from a single camera is the twofold 'forwards/backwards flipping' ambiguity for each body part which leads to an exponential number of local minima. Rehg, Morris and Kanade [21] comprehensively analyze the ambiguities and propose a two-dimensional scaled prismatic model for figure registration which has fewer ambiguity problems. Sminchisescu and Triggs [22] propose to apply inverse kinematics to systematically explore the complete set of configurations which shows improved performance over the baselines. Then in a later work, they [23] propose to reduce the number of local minima by building 'roadmaps' of nearby minima linked by transition pathways which are found by searching for the codimension-1 saddle points. Simo-Serra *et al.* [24] first estimate the 2D joint locations and model each joint by a Gaussian distribution. Then they propagate the uncertainty to the 3D pose space and sample a set of 3D skeletons there. They learn a SVM to resolve the ambiguity by selecting the most feasible skeleton. In a later work [25], they propose to detect the 2D and 3D poses simultaneously by first sampling the 3D poses from a generative model then reweighting the samplers by a discriminative 2D part detector model. They repeat the process until convergence.

The second class uses manually labelled body joints in multiple images as inputs. The use of multiple images eliminates much of the ambiguity of lifting 2D to 3D. Valmadre *et al.* [26] first apply rigid structure from motion to estimate the camera parameters and the 3D poses of the torsos (which are assumed to be rigid), and then requires human input to resolve the depth ambiguities for non-torso joints. Similarly, Wei *et al.* [27] propose "rigid body" constraints to remove the ambiguity. They assume that the pelvis and the left and right hip joints form a rigid structure, and require that the distance between any two joints on the rigid structure remain unchanged. They estimate the 3D poses by minimizing the discrepancy between the 3D pose projections and the 2D joint detections without violating the "rigid body" constraints.

The third class takes the joints in a single image as inputs. For example, Taylor [15] assumes the limb lengths are known and calculates the relative depths of the limbs. Barron and Kakadiaris [28] extend this idea by estimating the limb length parameters. Both approaches [15] [28] suffer from sign ambiguities. Pons-Moll, Fleet and Rosenhahn [29] propose to tackle the ambiguities by semantic pose attributes. These attributes represent Boolean geometric relationships between body parts which can be directly inferred from image data using a structured SVM model. They sample multiple poses from the distribution and select the best one by the attributes. Ramakrishna *et al.* [13] represent a 3D pose by a linear combination of PCA bases. They greedily add the most correlated basis into the model and estimate the basis coefficients by minimizing an L_2-norm error between the projection of the 3D pose and the 2D pose. They also enforce a constraint on the sum of the limb lengths of the 3D poses. This constraint is weak because the individual limb lengths are not necessarily correct, even if the sum is. Akhter and Black [30] propose to learn an even more strict prior, i.e. *pose-conditioned joint angle limits* from a large motion capture dataset. They also use sparse bases to represent poses. But different from ours, they do not use the robust reconstruction loss term neither the limb lengths constraints.

The fourth class [31] [32] [33] [34] [35] [36] [37] requires only a single image or image features. Mori *et al.* [31] match a test image to the stored exemplars, and transfer the matched 2D pose to the test image. They lift the 2D pose to 3D by [15]. Gregory *et al.* [33] propose to learn a set of hashing functions that efficiently index the training 3D poses. Bo *et al.* [38] use twin Gaussian Process to model the correlations between images and 3D poses. Elgammal *et al.* [32] learn a view-based silhouette manifold by Locally Linear Embedding (LLE) and the mapping function from the manifold to 3D poses. Agarwal *et al.* [34] present a method to recover 3D poses from silhouettes by direct nonlinear regression of the joint angles from the silhouette shape descriptors. These approaches do not explicitly estimate camera parameters and require a lot of training data from different viewpoints in order to generalize to other datasets. Ionescu, Carreira and Sminchisescu [9] propose to simulate the Kinect systems to first label the image pixels and then regress the 3D joint locations from the derived features. In [39], the authors apply deep networks to regress 3D human poses and 2D joint detections in images together under a multi-task framework. In [36], the authors propose a univeral network to regress the pixelwise segmentations, 2D poses and the 3D poses. The authors in [37] propose an dual-source approach to combine the 2D and 3D pose estimation datasets which improves the results when using only one data source. In a recent work [35], the authors propose to first detect the 2D poses in an image and then fit a 3D human shape model by minimizing the projection errors.

Our method only requires a single image as inputs. Unlike [31] [32] [33] [34], we explicitly estimate the camera parameters which reduces dependence on training data. Our method is similar to [13] but there are five differences: (i) we do not require human intervention. We obtain the 2D joint locations by applying a 2D pose detector [6] instead of by manual labeling; (ii) we use the L_1-norm penalty instead of the L_2-norm because it is more robust [17] to inaccurate 2D poses; (iii) we enforce eight limb length constraints, which is more effective than the sum of the limb lengths; (iv) we add an explicit L_1-norm regularization term on the basis coefficients in our formulation to encourage sparsity; while they greedily add a limited number of bases into the model. They need to re-estimate the basis coefficients every time a new basis is added; (v) We learn the bases on training data which combines all the actions while their approach splits the training data into classes, applies PCA to each class and finally combines the principal components as bases. Our approach is easier to generalize to other datasets because it does not require people to manually split the training data.

2.2 Related Work on M-best Models

Meltzer *et al.* [40] observe that maximum a posteriori (MAP) estimates often do not agree with the groundtruth. There are several reasons accounting for this phenomenon. Firstly, the algorithm computing the MAP estimate may get stuck in a local minimum. Secondly, the model itself is only an approximation and may depend on parameters which are learnt inaccurately from a small training dataset.

To address the problem, several work [41][42][43][44][3] propose to compute multiple 3D human poses for post-processing. For example, Sminchisescu and Jepson [41] present a mixture smoother for non-linear dynamical systems which can accurately locate multiple trajectories. They use dynamic programming or maximum a posteriori for picking the final solutions. This is similar to ours except that our work focuses on how to generate multiple solutions for a single static image. Kazemi *et al.* [44] first generate a set of high scoring 2D poses and then reorder them by training a rank-svm using a more complicated scoring function. Batra *et al.* [43] generate a *diverse* set of proposal solutions progressively by augmenting the energy function with a term measuring the similarity to previous solutions. Similarly, Park and Ramanan [42] propose an iterative method for computing M-best 2D pose solutions from a part model that do not overlap, by iteratively partitioning the solution space into M sets and selecting the best solution from each set. But it doesn't deal with 3D poses neither the videos as our method.

Our work for producing multiple solutions has two modules: candidates generation and candidate selection. In terms of computing multiple solutions, our work is related to [43] which also generates diverse solutions under the framework of Markov Random Fields. However, our work differs from [43] in terms of that we propose a novel diversity term which can be naturally integrated into our 3D pose estimation model. It is also related to [42]. But in [42], generating multiple solutions is natural because of the tree structure based inference (each root node location results in a solution). So the focus of [42]lies in how to select M-best from them. In contrast, our work focuses on obtaining multiple solutions by adding a diversity term. In terms of selecting the best candidate, our use of the limb length constraints is novel. However, the use of temporal coherence cues and dynamic programming has been explored extensively in previous work [3] [41].

3 Pose Estimation From a Single Image

This section describes how our algorithm can estimate candidate solutions for the 3D pose and camera parameters from a single image. The input is the estimates of the 2D pose produced by a state-of-the-art detection algorithm [45].

The first two sections describe the material that was first presented in our preliminary work which outputs a single 3D pose estimate [14] while including additional details. The third section describes an extension which outputs multiple 3D poses and selects the best of these candidates.

3.1 The 3D Pose Representation

We describe the 2D and 3D pose representations and the camera model in section 3.1.1, the measurement error in section 3.1.2, the sparse linear combinations of bases in section 3.1.3, the anthropomorphic constraint in section 3.1.4 and the camera parameter estimation in section 3.1.5.

3.1.1 The Representation and Camera Projection

We represent 2D and 3D poses by n joint locations $x \in \mathbb{R}^{2n}$ and $y \in \mathbb{R}^{3n}$, respectively. These can be expressed in matric forms by $X \in \mathbb{R}^{2\times n}$ and $Y \in \mathbb{R}^{3\times n}$ respectively, where the i_{th} column are the 2D and 3D locations of the i_{th} joint. We assume that the 2D and 3D poses have already been mean-centered, i.e. the mean value of each row of X and Y is zero.

We assume that people are not close to the camera which enables us to use a weak perspective camera model. The camera projection matrix is denoted by $M_0 = \begin{pmatrix} m_1^T \\ m_2^T \end{pmatrix} \in \mathbb{R}^{2\times 3}$ where $m_1^T m_2 = 0$. The scale parameters have been implicitly considered in the m_1 and m_2. In other words, $\|m_1\|$ is not necessarily to be one. Then the 2D projection x of a 3D pose y is given by: $x = My$, where $M = I_n \otimes M_0$, in which I_n is an identity matrix and $\otimes$ is the Kronecker product operator.

3.1.2 The Measurement Error: L_1 or L_2 norm

The measurement error quantifies the difference between the 2D pose and the projection of the 3D pose. In this paper we consider two different *measurement errors*, specified by the L_1 and L_2 norms respectively:

$$||x - My||_1, \quad L_1 \text{ norm} \quad \text{and} \quad ||x - My||_2, \quad L_2 \text{ norm.} \tag{1}$$

The L_2-norm is the most widely used measurement error in the computer vision literature. But, as discussed earlier, there can be large errors, or outliers, in the 2D pose estimation due to occlusion and other factors. Fig. 2 gives an example of an "outlier" measurement. The right foot location (estimated by [6]) is very inaccurate and biases the 3D estimate to the wrong solution if the L_2 norm is used, while the L_1 norm gives a better estimate. Hence we prefer to use the L_1 norm because it is more robust to outliers [17]. In the experimental section we show that the L_1 norm gives better results.

3.1.3 Sparse Linear Combination of Bases

We represent a 3D pose y as a linear combination of a set of bases $B = \{b_1, \cdots, b_k\}$, *i.e.*, $y = \sum_{i=1}^k \alpha_i b_i + \mu$ (or $y = B\alpha + \mu$), where the α are the basis coefficients and μ is the mean pose. The bases and the mean pose are learned from a dataset of 3D poses as described in Section 5.

Combining this with the measurement error gives a penalty:

$$\|x - M(B\alpha + \mu)\|_1 . \tag{2}$$

In addition, we apply an L_1 sparsity penalty $\theta \|\alpha\|_1$ on the coefficients α so that typically only a few bases are activated for each 3D pose. This is aimed to reduce the effective dimension of the 3D pose space. Although human poses are highly variable geometrically it is clear that they do not form a linear space so not all linear combination of bases should be allowed. In fact researchers have shown that 3D poses can be modelled by a low dimensional non-linear space [46]. This leads to an objective penalty, which combines the measurement error with the sparsity prior:

$$\min_{\alpha} \quad \|x - M(B\alpha + \mu)\|_1 + \theta \|\alpha\|_1 \tag{3}$$

where $\theta > 0$ is a parameter which balances the projection error and the sparsity penalty. The sparsity penalty can be thought of as a *sparsity prior* on the set of 3D poses when combined with the requirement that each pose is a linear sum of bases. In our experiments, we set the parameter θ by cross-validation. More specifically, it is set to be 0.01 in all experiments.

There is, however, a problem with using Eq. (3) by itself. We have observed that there are 3D configurations y for which the objective function takes small values, but which are unlike human poses. Hence the *sparsity prior* is not sufficient and needs to be strengthened.

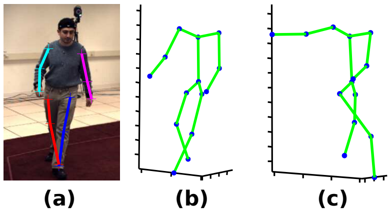


Fig. 2. **(a)** The estimated 2D joint locations where the right foot location is inaccurate. **(b-c)** are the estimated 3D poses using the L_1-norm and L_2-norm projection error, respectively. Using L_2-norm biases the estimation to a completely wrong pose. In contrast, using L_1-norm returns a reasonable pose which does not have obvious errors despite the right foot joint. See Section 3.1.2

3.1.4 *Anthropomorphic Constraints*

It is known that the limb length ratios of different people are similar in spite of the differences in their heights. This is sometimes called an anthropomorphic, or structural, prior and has been explored to decrease the ambiguities for human pose estimation [15] [16] [13]. By itself it is not sufficient because it allows sign ambiguities and cannot distinguish, for example, between an arm pointing forward or backward.

We use the anthropomorphic prior to strengthen our sparsity prior. This requires formulating it as a anthropomorphic constraint which we can incorporate into our objective function. More specifically, we require that the lengths of the eight limbs of a 3D pose should comply with certain proportions. The eight limbs are the Left Upper Arm (LUA), Right Upper Arm (RUA), Left Lower Arm (LLA), Right Lower Arm (RLA), Left Upper Leg (LUL), Right Upper Leg (RUL), Left Lower Leg (LLL), and Right Lower Leg (RLL), respectively. These limb proportions are computed from the statistics of the poses in the training dataset (they are independent of individual subjects). We now proceed with the formulation.

We define a joint selection matrix $E_j = [0, \cdots, I, \cdots, 0] \in \mathbb{R}^{3\times 3n}$, where the j_{th} block is an identity matrix of dimension 3×3 and the other blocks are zeros. We can verify that the product of E_j and y returns the 3D location of the j_{th} joint in pose y. Let $C_i = E_{i_1} - E_{i_2}$. Then $\|C_i y\|_2^2$ is the squared length of the i_{th} limb whose ends are the i_1-th and i_2-th joints.

We normalize the squared limb length of right lower leg to one and compute the average squared lengths of the other seven limbs (say L_i) correspondingly from the training data. We propose the following constraints $\|C_i (B\alpha + \mu)\|_2^2 = L_i$.

3.1.5 *Robust Camera Estimation*

Another component of the approach is to estimate the camera parameters M_0 given the estimated 3D pose y and the corresponding 2D pose x by minimizing the L_1-norm projection error. Ideally the equality relationship between the 2D and 3D poses: $X = M_0 Y$ should hold, where $M_0 = \begin{pmatrix} m_1^T \\ m_2^T \end{pmatrix}$ is the projection matrix of a weak perspective camera. Note that the scale parameters have been implicitly considered in the m_1 and m_2. So we propose to estimate the camera parameters m_1 and m_2 by solving the following problem:

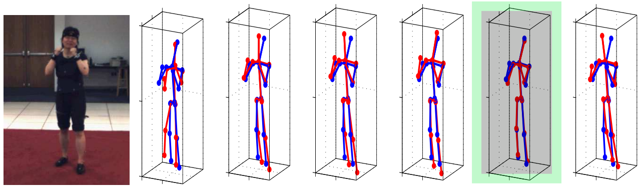


Fig. 3. Top-six 3D pose estimations of a sample image. The plots in blue and red are the ground-truth and estimated 3D poses, respectively. The fifth estimation is the best among the candidates. See section 3.3.

$$\min_{m_1, m_2} \left\| X - \begin{pmatrix} m_1^T \\ m_2^T \end{pmatrix} Y \right\|_1, \quad \text{s.t.} \quad m_1^T m_2 = 0. \tag{4}$$

3.2 The Inference Algorithm: Estimating the 3D Pose and the Camera Parameters

Given the discussions above, we obtain the complete objective function which depends on both the basis coefficients and the camera parameters:

$$\begin{aligned} \min_{\alpha, M} \quad & \|x - M(B\alpha + \mu)\|_1 + \theta \|\alpha\|_1 \\ \text{s.t.} \quad & \|C_i (B\alpha + \mu)\|_2^2 = L_i, i = 1, \cdots, 8 \\ & m_1^T m_2 = 0, \end{aligned} \tag{5}$$

where $M = I_n \otimes M_0$ and $M_0 = \begin{pmatrix} m_1^T \\ m_2^T \end{pmatrix}$. We minimize the objective function by alternating between M (with α fixed) and α (with M fixed). We first initialize the 3D pose to be the mean pose of the training dataset and optimize M. Then with the estimated M we optimize the basis coefficients α. Both the basis coefficient estimation and camera parameter estimation problems are not convex because of the quadratic equality constraints. We solve the problem by using an alternating direction method (ADM) [47]. Briefly, we define an augment Lagrangian function which contains primal variables (the 3D pose coefficients and the camera parameters) and dual variables (Lagrange multipliers which enforce the equality constraints). The ADM updates the variables by extremizing an augmented Lagrangian function with respect to the primal and dual variables alternately. Although there is no guarantee of global optimum, we almost always obtain reasonably good solutions (see section 3.3 for how we address this non-convexity issue by producing multiple solutions).

3.3 Producing Several Candidate Poses

The objective function in Eq. (5) is non-convex in M and α so we cannot guarantee that our algorithm has found the global minimum. This motivates us to extend our approach to output a diverse set of K solutions. After that we describe how to select the best 3D pose from these candidates.

We require that the K 3D poses $\mathcal{Y}$ returned by the model are dissimilar to each other to avoid redundancy. We use the

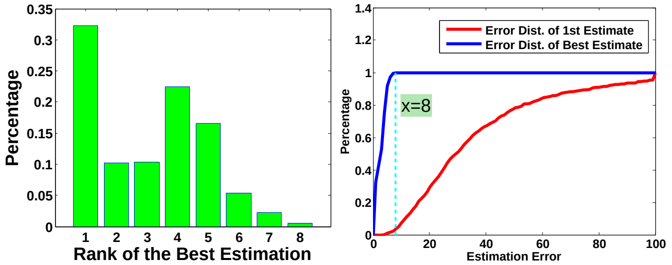


Fig. 4. All results are based on the HumanEva dataset for three subjects. Left figure: The rank distribution of the best poses among the eight candidates. Right figure: The error distribution when choosing the first candidate vs. choosing the best candidate (by oracle). X-axis is the average joint error of the three subjects and the y-axis represents the percentage of cases whose errors are smaller than X. The estimation error units are millimetres (mms). See Section 3.3.

squared Euclidean distance between two poses y_i and y_j as the dissimilarity measure, i.e., $\triangle(y_i, y_j) = \|y_i - y_j\|^2$. Note that we use L_2-norm rather than the L_1-norm here because two poses are dissimilar even when only one joint of the two poses are dissimilar. To estimate the K_{th} candidate pose y_K given the 2D pose x and the first $K-1$ poses $\{y_i \mid i = 1, \cdots, K-1\}$, we solve an augmented minimization problem which minimizes a linear combination of the original objective function and the y_K's similarity to the existing poses:

$$\begin{aligned} \min_{\alpha} \quad & \|x - M(B\alpha + \mu)\|_1 + \theta_1 \|\alpha\|_1 + \theta_2 \sum_{i=1}^{K-1} \|B\alpha + \mu - y_i\|^2 \\ \text{s.t.} \quad & \|C_i(B\alpha + \mu)\|_2^2 = L_i, i = 1, \cdots, 8, \end{aligned} \tag{6}$$

where $\theta_2 \le 0$ is a parameter which balances the loss term and the similarity term. The problem can be solved by a small modification of our original ADM based optimization algorithm. We set the parameter θ_1 the same as the θ in the previous model. The parameter θ_2 is set by cross-validation. In particular, it is set to be -0.1 in all experiments.

We now select the best 3D pose from the set of K candidates. We observe that the 3D pose estimate that minimizes the objective function in Eq. (5) is sometimes not the best solution. Fig. 3 shows an example where the fifth candidate rather than the first one is the best among the six candidates. Fig. 4 (left) shows the rank distributions of the best pose among the candidates. We can see from the left most green bar that, for only about 30% of testing samples, the first estimate is the best estimate (closest to the groundtruth). In other words, the best one is not in the first position (rank one) for nearly 70% of the cases. The right figure shows the estimation error distributions of selecting the first pose vs. selecting the best pose (using oracle) from the candidates. We can see that the performance can be significantly improved if we can select the "correct" estimate from the candidates.

Why is the best solution (as evaluated by groundtruth) not always the 3D pose that minimizes the objective function? This may occur because of the limitations of the objective function. But it can also happen that because of the nature of our ADM algorithm the anthropomorphic constraints have not fully been enforced. Hence the 3D pose candidates may partially violate the anthropomorphic constraints. Fig. 7 (blue bars) shows that the limb length errors for the eight limbs are not zero although they are small. This motivates selecting the candidate 3D pose which best satisfies the anthropomorphic constraints. We show, in the experimental section, that this improves our results.

4 Pose Estimation From A Video

Now suppose we have a video sequence as input. This gives another way to improve 3D pose estimation using our K best candidates. For each image frame, we estimate K candidates and use temporal information to select the best sequence.

We define an objective function which encourages similarity among the 3D poses in adjacent frames [48] [3] and uses unary terms similar to those defined for static images. We compute the Euclidean distance between two neighboring poses as the pairwise term $\triangle(y_i, y_j)$. This term discourages sharp pose changes which can be helpful for difficult images especially when their neighbouring estimations are accurate.

Suppose a video sequence consists of T frames. For each frame $I^t, t = 1, \cdots, T$, we first obtain the K-best estimations $y_j^t, j = 1, \cdots, K$ using the proposed method. Then we infer the best pose $y_{j_t}^t, 1 \le j_t \le K$ for each frame by minimizing an objective function:

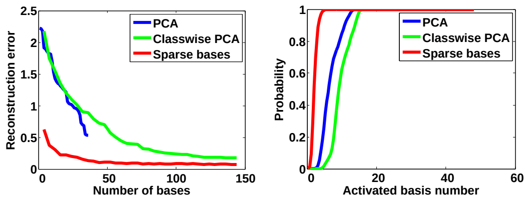
$$j^* = \min_{(j_1, \cdots, j_T)} \sum_{t=1}^{T} f(y_{j_t}^t) + \theta_3 \sum_{t=1}^{T-1} \triangle(y_{j_t}^t, y_{j_{t+1}}^{t+1}). \tag{7}$$

Here $f(.)$ measures how well a 3D pose obeys the anthropomorphic constraints (unary term) while the $\triangle(,)$ function measures the differences between adjacent poses (the pairwise term), $\theta_3 \ge 0$ specifies the trade off between the unary term and the pairwise term, which is set by cross validation.

We use dynamic programming to estimate j^* by minimizing the objective function in Eq. (7). This exploits the one-dimensional nature of the problem and is efficient since we only enforce temporal smoothness between adjacent time frames. Experiments show that this simple extension can yield improvements in the 3D pose estimation results.

4.1 The Unary Term

We investigated two candidate unary terms in this work. The first is the measurement error between the projected 3D pose and the estimated 2D pose. We found this was not effective because the measurement errors of the top K candidates are very similar. The reason is that incorrect 3D poses can have small measurement errors because the camera parameters compensate (i.e., bad 3D poses with bad camera parameters can still give small projection/measurement errors). Secondly, we measured how well the 3D pose satisfies the anthropomorphic constraints. More precisely, we computed the absolute difference between the estimated limb length and the mean limb length L obtained during training. This was effective and we used the second method in our experiment. Note that



(a) (b)

Fig. 5. Comparison of the three basis learning methods. **(a)** 3D pose reconstruction errors using different number of bases. In this experiment, the 3D poses are normalized so that the length of the right lower leg is one.**(b)** Distribution of the number of activated bases for representing a 3D pose. The y-axis is the percentage of the cases whose number of activated bases is less than x. See section 5.2.

we also use anthropomorphic constraints when selecting the best solution for a single image from a set of K proposals, i.e. we use these constraints for both static images and video sequences to select from a set of candidates.

5 Basis Learning

We now describe how we learn the bases using sparse coding in experiments. We compare with two most popular basis learning methods including PCA and classwise PCA.

5.1 Our Basis Learning Method

We learn the bases from a set of of 3D skeletons $\mathbb{Y} = [y_1, \cdots, y_l]$ by optimizing the empirical cost function:

$$f_l(B) \triangleq \frac{1}{l}\sum_{i=1}^{l} c(y_i, B). \tag{8}$$

$B = [b_1, \cdots, b_k] \in \mathbb{R}^{3n \times k}$ is the basis dictionary to be learned with each column representing a pose basis, and $c(.,.)$ is a loss function such that $c(y, B)$ is small if B represents the pose y well. Also $c(.,.)$ imposes sparsity so that each pose is typically represented by only a small number of bases. Note that over-complete dictionaries with $k \geq 3n$ are allowed. As previous work, e.g., see [49], and consistent with our objective function (Eq.(3)) the loss function $c(y, B)$ is given by:

$$c(y, B) = \frac{1}{2}\|y - B\alpha\|^2 + \theta\|\alpha\|_1 \tag{9}$$

To prevent B from being arbitrarily large we constrain its columns, i.e. the norms of each basis, to have an L_2-norm less than or equal to one.

$$\begin{aligned} \min_{B,\alpha} \quad & \frac{1}{l}\sum_{i=1}^{l}\frac{1}{2}\|y_i - B\alpha_i\|^2 + \lambda\|\alpha_i\|_1 \\ \text{s.t.} \quad & b_j^T \cdot b_j \leq 1, \forall j = 1, \cdots, k \end{aligned} \tag{10}$$

This problem is not convex with respect to B and α but convex with respect to each of the two variables when the other is fixed. We use [49] to solve this optimization problem which alternates between the two variables.

5.2 Other Basis Learning Methods

We compare our approach with two classic basis learning methods proposed in the literature. The first method [46] applies PCA to the training motion capture data and uses the principal eigenvectors as the bases. Note that the maximum number of bases is limited by the dimension of the poses ($3n$ in our case) because the eigenvectors are orthogonal to each other. As discussed in [13], it is problematic to directly apply PCA to the poses for all 3D actions because PCA is most suitable to data which comes from a single-mode Gaussian distribution. Usually this is a very strong assumption. Hence Ramakrishna *et al.* [13] split the training dataset into different classes using the action labels and assume that the data of each action class follows the Gaussian distribution. They apply PCA on each class, and combine the principal components in each class as the bases. We name this approach *classwise PCA*. Since the bases learned for different classes are learned separately, there may be redundancy when compared with those jointly learned bases.

We evaluate the three different basis learning methods (i.e. PCA, classwise PCA and sparse coding) in a 3D pose reconstruction setting. We reconstruct each 3D pose y by solving an L_1-norm regularized least square problem:

$$\min_{\alpha} \quad \frac{1}{2}\|y - B\alpha\|^2 + \lambda\|\alpha\|_1. \tag{11}$$

We compute the reconstruction error $\|y - B\alpha\|_2$ as the evaluation metric. The average reconstruction errors using different number of bases are shown in Fig. 5 (a). Note that the maximum number of bases for PCA and classwise PCA methods is 36 (which is the dimension of a 3D pose) and 144 (36 * 4 classes), respectively. The reconstruction error of PCA bases is the largest (slightly above 0.5) because the poses do not follow Gaussian distribution and the number of PCA bases is small. Although the reconstruction errors of classwise PCA bases gradually decrease as more bases are introduced, they are still larger than those of the sparse bases. One of the main reasons might be that the classwise PCA method does not encourage basis sharing between action classes. Hence the bases might contain redundancy. In contrast, the sparse bases are shared between classes as they are learned from the training data of different action classes together. In addition, Fig. 5 (b) shows that fewer bases are activated using sparse bases. This also justifies their representative power.

6 Evaluation on Single Images

We conduct two types of experiments to evaluate our approach. The first type is synthetic where we assume the groundtruth 2D joint locations are known and recover the 3D poses. We systematically evaluate: (i) the influence of the three factors in the model, i.e. the L_1-norm measurement error, the anthropomorphic constraints and the L_1-norm sparsity regularization on the basis coefficients; (ii) the influence of the 2D pose accuracy; (iii) the influence of the relative human-camera angles; (iv) the generalization capabilities of the learned bases. The second type of experiments is real: we estimate the 2D joint locations by running a state-of-the-art 2D pose detector

[45] and then estimate the 3D poses. We compare our method with the state-of-the-art ones [13] [24] [50]. We also observe that our approach can refine the original 2D pose estimations by projecting the inferred 3D pose back to the images.

We use 12 body joints, *i.e.,* the left and right shoulders, elbows, hands, hips, knees and feet, for quantitative evaluation, which is consistent with the 2D pose detector [45]. We learn 200 bases for all experiments and approximately 14 bases are activated for representing a 3D pose.

6.1 The Datasets

We evaluate our approach on two benchmark datasets: the HumanEva [51]and the H3.6M [52] datasets. Following the previous work, e.g.,[24], we use the walking and jogging actions of three subjects for evaluation (the fourth subject is withheld by the authors) and learn the bases on the training subset of the poses independently for each action. We report results on the validation sequences. The H3.6M dataset [52] includes 11 subjects performing 15 actions, such as eating, posing and walking. We use the data of subjects S1, S5, S6, S7 and S8 for training and the data of S9 and S11 for testing.

6.2 Synthetic Experiments: Known 2D Poses

We assume the 2D poses x are known and recover the 3D poses y from x. We use mean 3D joint error [51] as evaluation metric which is the average error over all joints. The results are reported using unit of millimetres (mms). All synthetic experiments are on the HumanEva dataset.

6.2.1 Necessity of Basis Representation

We first discuss the necessity of the basis representation. We design a baseline which represents a 3D pose by the nearest neighbor. Intuitively, we treat all training poses as bases and represent a pose by its nearest neighbor. Since the training poses should approximately satisfy the limb length constraints, we remove those constraints. We also remove the sparsity term because only one basis will be activated. The 3D pose which can minimize the L_1-norm projection error is the final estimate. The mean square error on the HumanEva dataset for this method is about 72mms while the result for our proposed method is about 40mms. We think the main reason for the degraded performance is because the training poses differ from the testing poses and the nearest neighbor method does not have the capability to represent the unseen poses.

6.2.2 Influence of the Three Factors

We evaluate the influence of the three factors in the proposed method: the robust L_1-norm measurement error, the anthropomorphic constraints and the sparsity regularization term.

We compare our approach with seven baselines. The first is symbolized as L2S which uses the L_2-norm error function and Sparsity term on the basis coefficients. The second baseline is L1A which uses the L_1-norm measurement error function and Anthropomorphic constraints. The remaining baselines are symbolized as L2, L2A, L2AS, L1 and L1S whose meanings can be similarly understood by their names. We solve the non-convex optimization problems in L2A and L2AS by the ADM method used to solve our method (L1AS). To solve the optimization problems in the other baselines, we use CVX, a package for solving convex programs [53].

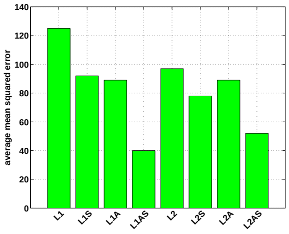


Fig. 6. 3D pose estimation errors of the baselines and our method (L1AS). The units for estimation errors are mms.

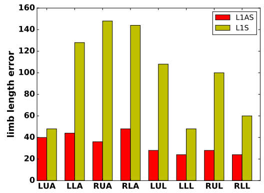


Fig. 7. Average limb length error of the L1AS and L1S.

Fig. 6 shows the results on the HumanEva dataset. First, the four baselines without the sparsity term (*i.e.,* L1, L1A, L2 and L2A) achieve much larger estimation errors than those with the sparsity term (*i.e.,* L1S, L1AS, L2S and L2AS). The results demonstrate that the bases encode the priors in human poses and can prevent overfitting to 2D poses— given enough bases, the 2D projection error could always be decreased to zero but the resulting 3D pose might still have large errors. Using sparse bases helps prevent it from happening.

Second, enforcing the eight limb length constraints further improves the performance, *e.g.,* L2AS outperforms L2S and our approach outperforms L1S. Fig. 7 shows that the limb lengths of the estimated poses are more accurate by enforcing the anthropomorphic constraints. Third, using the L_1-norm reconstruction error outperforms L_2-norm, *e.g.,* L1AS is better than L2AS and our approach is slightly better than L2AS. However, the difference is small because the poses in this experiment are accurate which does not reveal the potential influences of the L_1-norm penalty function. It is interesting to see that L1 is worse than L2. The reason is that ignoring the anthropomorphic constraints and the sparsity term will result in implausible 3D poses. In this case, using the robust term will tolerate inconsistent matches between the 3D pose projections and the 2D poses (which are accurate in this experiment).

6.2.3 Influence of Inaccurate 2D Poses

We evaluate the robustness of our approach to inaccurate 2D pose estimations. We generate outlier 2D poses by adding seven levels (magnitudes) of noises to the accurate 2D poses to simulate 2D pose estimation errors. In particular, for each 2D pose, we randomly select a body joint, generate a random 2D spatial shift orientation, and add the corresponding transformation (of a certain magnitude) to the selected joint. The magnitude of the $i_{th} (1 \leq i \leq 7)$ level of noises is $8i$ pixels.

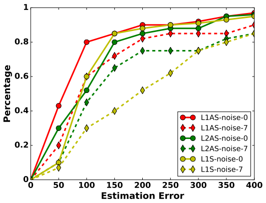


Fig. 8. Results when different levels of noises are added to 2D poses. The x-axis is the estimation error and the y-axis is the percentage of cases where the estimation error is less than the corresponding x value.

Fig. 8 demonstrates the results for L1AS, L2AS and L1S on the HumanEva dataset. We do not report the results for the other five baselines because the estimation errors are very large compared to these three. We can see that L1AS performs much better than the other two. For example, the estimation errors are smaller than 100 for 60% of data for L1AS even when the largest (7th) level of noises are added. In contrast, this number is decreased to only 40% for L2AS. The results verify that L_1-norm is more robust to inaccurate 2D poses than L_2-norm. L1S performs worst among the three which also shows the importance of anthropomorphic constraints especially when the 2D joint locations are inaccurate.

6.2.4 Influence of Human-Camera Angles

The degree of ambiguities in 3D pose estimation depends on the relative angle between human and camera. Generally speaking, ambiguity is largest when people face the cameras and is the smallest when people turn sideways. We quantitatively evaluate the approach's disambiguation ability on various human-camera angles. We synthesize ten virtual cameras of different panning angles and project the 3D poses to 2D using the virtual cameras. Then we estimate the 3D poses from the 2D projections in each camera and compare the estimation results. More specifically, we first transform the 3D poses into a local coordinate system, where the x-axis is defined by the line passing the two hips, the y-axis is defined by the line of spine and the z-axis is the cross product of the x-axis and y-axis. Then we rotate the 3D poses around y-axis by a particular angle, ranging from 0 to 180, and project them to 2D by a weak perspective camera. Note that the y-axis is the axis where most viewpoint variations happen in real world images. Hence we only report the performance for the y-axis rotations for space limitations.

Fig. 9 shows that the average estimation errors of the method proposed in [13] increase quickly as human moves from profile (90 degrees) towards frontal pose (0 degree). However, our approach is more robust against viewpoint changes due to the structural prior imposed by the sparse bases and the strong limb length constraints.

6.2.5 Influence of Camera Parameter Estimation

Fig. 10 (left) shows the estimation errors of the camera rotation angles (*i.e.,* yaw, pitch and roll). We can see that the errors are small for most cases. Fig. 10 (right) shows the 3D pose estimation errors using the estimated cameras and ground truth cameras, respectively. Note that when using ground truth camera parameters, we do not update them in each iteration. We can see that camera estimation results can affect the 3D pose estimation to some extent.

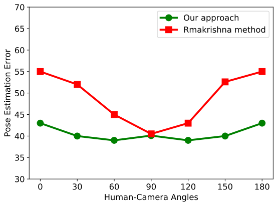


Fig. 9. 3D pose estimation errors when the human-camera angle varies from 0 to 180 degrees. We compare with Ramakrishna's method[13]. See Section 6.2.4.

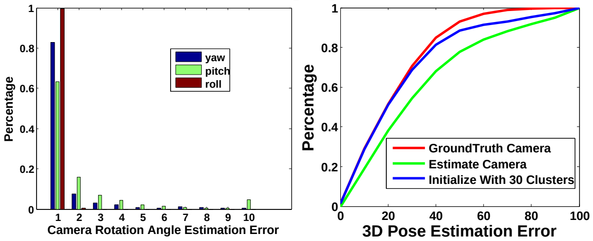


Fig. 10. Left figure: Error distribution of the estimated Camera rotation angles. The units are degrees. Right figure: 3D pose estimation errors when camera parameters are (1) set by ground truth, (2) estimated by initializing the 3D pose with mean pose, or (3) estimated by initializing the 3D pose with 30 cluster centers for parallel optimization (but only the best result is reported). The y-axis is the percentage of the cases whose estimation error is less than x. The units for x are mms. See section 6.2.5.

The initialization of the 3D pose influences the 3D pose estimation accuracy — more accurate 3D pose initializations can improve the final result. So we cluster the training poses into 30 finer clusters and initialize the 3D pose with each of the centers respectively. We optimize the 3D poses for the 30 initializations in parallel and keep the one which is closest to ground truth. The performance can be further improved using finer initializations as shown in Fig. 10.

6.2.6 Generalization Capabilities of the Bases

We conduct three types of experiments to validate the generalization capabilities of our approach: (1) cross-subject, (2) cross-action and (3) cross-datasets experiments.

For the cross-subject experiment (on the HumanEva dataset including all the six actions), we use the leave-one-subject-out criteria, i.e., training on the two subjects and testing on the remaining one. The average error is about 43.2mms. This is comparable with the previous experiment setup (the result is 40mm) when training and testing on all subjects.

Similarly for the cross-action experiment (on the HumanEva dataset), we use the leave-one-action-out criteria. In this experiment, we use the sequences of all the six actions provided in the dataset in addition to the walking and jogging sequences. In particular, we train on the poses of the five actions and test on the remaining one. We repeat the above process for all the six configurations and report the average estimation error. In this experiment, The average estimation error is about 48.4mms which is slightly higher than training/testing on the same actions. This slight performance degradation is reasonable as the bases are learned from different actions which might have different sets of poses.

For the cross-datasets experiment, we train on the H3.6M dataset (including all actions) and test on the HumanEva dataset. The average estimation error is about 57.3mms. The estimation error is larger than the previous one and the reason might be because the two datasets have slightly different annotations. For example, the hip joints might correspond to slightly different parts of the human body. Another reason might be because the two datasets have different sets of actions. But overall, this is still a reasonable performance.

6.3 Real Experiments: Unknown 2D Poses

We first estimate the 2D joint locations in the test images by running a 2D pose detector [45]. Then we estimate the 3D poses from the 2D joint locations. We compare our method with the state-of-the-arts in section 6.3.1. We also observe that by projecting the estimated 3D poses and camera parameters to 2D, we can actually improve the 2D pose estimation results. This is stated in section 6.3.2.

6.3.1 Comparison to the State-of-the-arts

We compare our approach with the state-of-the-art ones [24] [50] [25] on the HumanEva and the H3.6M datasets. Table 1 shows the mean squared errors and the standard deviations. Note that the results are not directly comparable because of different experiment setups. First, it is fair to compare the results of ours (using 2D detector [6]) with the methods [25] [24] as they use the same 2D pose detectors. Second, our method using the state-of-the-art 2D pose detector outperforms our method using [6] which is mainly due to the improvement from the 2D joint location estimation. Method [38] achieves similar performance as our method by assuming that the silhouettes are known. Table 2 shows the results on the H3.6M dataset. We can see that our method achieve comparable performance as the state-of-the-arts.

6.3.2 Evaluation on 2D Pose Estimation

We observe that projecting the estimated 3D pose by the camera parameters can improve the original 2D pose estimations. The reason is that the sparse bases and the anthropometric constraints could bias the estimated 3D pose to a correct configuration in spite of the errors in 2D joint locations. In experiments, we project the estimated 3D poses to 2D and compare it with the original 2D pose estimation [6] and [13]. For [13], we project its estimated 3D pose to 2D image.

TABLE 1
Real experiment on the HumanEva dataset: comparison with the state-of-the-art methods [24] [50]. We present results for both walking and jogging actions of all three subjects and camera C1. The numbers in each cell are the mean 3D joint errors and standard deviation, respectively. We use the unit of millimeter as in [24] and [50]. The length of the right lower leg is about 380 mm. See Section 6.3.1.

Walking	S1	S2	S3	Average
Ours (2D[6])	54.3 (16.2)	43.5 (14.9)	67.4 (10.3)	55.06
Ours (2D[45])	40.3 (17.4)	37.6 (14.5)	**37.4** (18.3)	38.43
[25]	65.1 (17.4)	48.6 (29.0)	73.5 (21.4)	62.4
[24]	99.6 (42.6)	108.3 (42.3)	127.4 (24.0)	111.76
[50]	89.3	108.7	113.5	103.83
[38]	**38.2** (21.4)	**32.8** (23.1)	40.2 (23.2)	**37.06**

Jogging	S1	S2	S3	Average
Ours (2D[6])	54.6 (10.7)	43.3 (12.1)	34.4 (10.2)	44.1
Ours (2D[45])	**39.7** (9.7)	36.2 (7.8)	38.4 (27.8)	**38.1**
[25]	74.2 (22.3)	46.6 (24.7)	**32.2** (17.5)	51.0
[24]	109.2 (41.5)	93.1 (41.1)	115.8 (40.6)	106.03
[38]	42.0 (12.9)	**34.7** (16.6)	46.4 (28.9)	41.03

We report the results using two criteria. The first is the probability of correct pose (PCP) — an estimated body part is considered correct if its segment endpoints lie within 50% of the length of the ground-truth segment from their annotated location as in [6]. The second criterion is the Euclidean distance between the estimated 2D pose and the groundtruth in pixels as in [24]. Table 3 shows the estimation accuracy on each of the eight body parts and the overall accuracy. We can see that our approach performs the best on six body parts. In particular, we improve over the original 2D pose estimators by about 0.03 (0.741 vs. 0.714) using the first PCP criteria. Our approach also performs best using the second criterion.

6.4 Evaluation on the Videos

We now evaluate the influence of integrating the temporal consistency into our model. In particular, we quantitatively investigate the influence of the two factors (*i.e.,* the unary term and pairwise term) in the Video-Based Pose Estimation (VBPE) method on the HumanEva dataset. We divide the long videos into short snippets with each snippet having five frames. We also experiment with other length choices but it does not make much difference unless it has fewer than three frames or more than ten frames which will degrade the performance. The balancing parameter between the unary and pairwise terms is set by cross-validation. In particular, in our experiment, this is set to be -0.01 on the HumanEva dataset.

Fig. 11 shows the advantages of VBPE (green line) over the single Image-Based-Pose-Estimation (IBPE, red line) method. First, the VBPE outperforms the IBPE which verifies that estimating poses on videos by considering both the new unary term and the pairwise term can improve the performance. The average estimation error of VBPE is decreased by about 15% compared with IBPE. Secondly, relying on the pairwise term alone defined on the temporal consistency (magenta line) offers some benefits over IBPE. It improves the estimation results for images having large estimation errors (between 40mm and 60mm). Third, using only the unary term defined on limb length (blue) provides larger gains. This observation shows the importance of the anthropomorphic constraints. It also suggests that the optimizer for IBPE could possibly get trapped in local optimum and return a 3D pose that does not well satisfy the anthropomorphic measurements.

TABLE 2
Real experiment on the H3.6M dataset: comparison with the state-of-the-art methods.

	Directions	Discussion	Eating	Greeting	Phoning	Photo	Posing	Purchases
LinKDE [52] *	115.79	113.27	99.52	128.80	113.44	183.09	131.01	144.89
Li et al. [54]	-	136.88	96.94	124.74	-	168.68	-	-
Tekin et al. [55]	102.39	158.52	87.95	126.83	118.37	185.02	114.69	107.61
Zhou et al. [56]	87.36	109.31	87.05	103.16	116.18	143.32	106.88	99.78
SMPLify [35]	**62.0**	**60.2**	**67.8**	**76.5**	**92.1**	**77.0**	**73.0**	**75.3**
Ours (2D detector [45])	90.34	117.56	86.02	110.98	123.48	154.90	100.49	97.34
	Sitting	SittingDown	Smoking	Waiting	WalkDog	Walking	WalkTogether	Average
LinKDE [52] *	160.92	172.98	114.00	138.95	180.56	131.15	146.14	138.30
Li et al. [54]	-	-	-	-	132.17	69.97	-	-
Tekin et al. [55]	136.15	205.65	118.21	146.66	128.11	65.86	77.21	125.28
Zhou et al. [56]	124.52	199.23	107.42	118.09	114.23	79.39	97.70	113.01
SMPLify [35]	**100.3**	**137.3**	**83.4**	**77.3**	**79.7**	86.8	**81.7**	**82.3**
Ours (2D detector [45])	130.58	200.67	130.56	110.29	123.98	64.89	87.98	115.34

* The results are obtained on the testing dataset making the method not directly comparable to ours.

TABLE 3
2D pose estimation results. We report: (1) the Probability of Correct Pose (PCP) for the eight body parts and the whole pose, (3) and the Euclidean distance between the estimated 2D pose and the groundtruth in pixels.

	PCP									Pixel Diff.
	LUA	LLA	RUA	RLA	LUL	LLL	RUL	RLL	Overall	
Yang *et al.* [6]	0.751	**0.416**	0.771	**0.286**	0.857	0.825	0.910	0.894	0.714	109
Ramakrishna *et al.* [13]	0.792	0.383	0.722	0.241	0.906	0.829	0.890	0.849	0.702	62
Ours	**0.829**	0.376	**0.800**	0.245	**0.955**	**0.861**	**0.963**	**0.902**	**0.741**	**55**

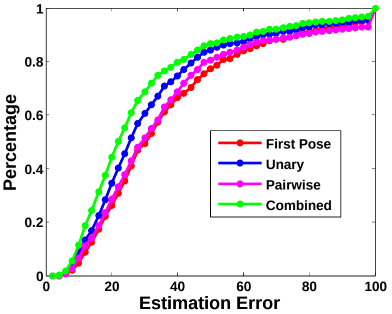


Fig. 11. The 3D pose estimation results on videos. The results using unary term, pairwise term and both of the two terms are reported. The red line shows the result on static images. The error units are mms. See section 6.4.

7 Conclusion

We address the problem of estimating 3D human poses from a single image or a video sequence. We first tackle the ambiguity of "lifting" a 2D pose to 3D by proposing a sparse basis based representation of 3D poses and anthropomorphic constraints. Second, we use an L_1-norm measurement error which makes the approach robust to inaccurate 2D pose estimates. Third, the problem of local optimum is alleviated by generating several probable but diverse solutions and selecting the correct one using temporal consistency cues.

Appendix A
Optimization

We sketch the major steps of ADM for solving our pose estimation (Eq. (6)) and camera parameter estimation (Eq. (4)) problems. The k and l are the number of iterations.

A.1 3D Pose Estimation

Given the currently estimated camera parameters M and the detected 2D pose x, we estimate the 3D pose by solving the following L_1 minimization problem using ADM:

$$\begin{aligned} \min_{\alpha} \quad & \|x - M(B\alpha + \mu)\|_1 + \theta_1 \|\alpha\|_1 + \theta_2 \sum_{i=1}^{K-1} \|B\alpha + \mu - y_i\|^2 \\ \text{s.t.} \quad & \|C_i(B\alpha + \mu)\|_2^2 = L_i, i = 1, \cdots, m \end{aligned} \tag{12}$$

We introduce two auxiliary variables β and γ and rewrite Eq. (12) as:

$$\begin{aligned} \min_{\alpha,\beta,\gamma} \quad & \|\gamma\|_1 + \theta_1 \|\beta\|_1 + \theta_2 \textstyle\sum_{i=1}^{K-1} \|B\alpha + \mu - y_i\|^2 \\ \text{s.t.} \quad & \gamma = x - M(B\alpha + \mu), \quad \alpha = \beta, \\ & \|C_i(B\alpha + \mu)\|_2^2 = L_i, i = 1, \cdots, m. \end{aligned} \tag{13}$$

The augmented Lagrangian function of Eq. (13) is:

$$\begin{aligned} \mathcal{L}_1(\alpha, \beta, \gamma, \lambda_1, \lambda_2, \eta) = \|\gamma\|_1 + \theta_1 \|\beta\|_1 + \\ \theta_2 \textstyle\sum_{i=1}^{K-1} \|B\alpha + \mu - y_i\|^2 + \\ \lambda_1^T[\gamma - x + M(B\alpha + \mu)] + \lambda_2^T(\alpha - \beta) + \\ \tfrac{\eta}{2}\left[\|\gamma - x + M(B\alpha + \mu)\|^2 + \|\alpha - \beta\|^2\right] \end{aligned}$$

where λ_1 and λ_2 are the Lagrange multipliers and $\eta > 0$ is the penalty parameter. ADM is to update the variables by minimizing the augmented Lagrangian function w.r.t. the variables α, β and γ alternately.

A.1.1 Update γ

We discard the terms in $\mathcal{L}_1$ which are independent of γ and update γ by:

$$\gamma^{k+1} = \underset{\gamma}{\text{argmin}} \|\gamma\|_1 + \frac{\eta_k}{2}\left\|\gamma - \left[x - M(B\alpha^k + \mu) - \frac{\lambda_1^k}{\eta_k}\right]\right\|^2$$

which has a closed form solution [57].

A.1.2 Update β

We drop the terms in $\mathcal{L}_1$ which are independent of β and update β by:

$$\beta^{k+1} = \underset{\beta}{\text{argmin}} \|\beta\|_1 + \frac{\eta_k}{2\theta}\left\|\beta - \left(\frac{\lambda_2^k}{\eta_k} + \alpha^k\right)\right\|^2$$

which also has a closed form solution [57].

A.1.3 Update α

We dismiss the terms in $\mathcal{L}_1$ which are independent of α and update α by:

$$\begin{aligned} \alpha^{k+1} = \arg\min_{\alpha} \quad & z^T W z \\ \text{s.t.} \quad & z^T \Omega_i z = 0, \quad i = 1, \cdots, m \end{aligned} \tag{14}$$

where $z = [\alpha^T \quad 1]^T$, W=
$$\begin{pmatrix} B^T M^T M B + I + \frac{2\theta_2(K-1)}{\eta_k} B B^T & 0 \\ 2\left[\left(\gamma^{k+1} - x + M\mu + \frac{\lambda_1^k}{\eta_k}\right)^T M B - \beta^{k+1} + \frac{\lambda_2^k}{\eta_k} + D\right] & 0 \end{pmatrix}$$
and $\Omega_i = \begin{pmatrix} B^T C_i^T C_i B & B^T C_i^T C_i \mu \\ \mu^T C_i^T C_i B & \mu^T C_i^T C_i \mu - L_i \end{pmatrix}$. $D = \frac{2\theta_2}{\eta}\sum_{i=1}^{K-1}(\mu - y_i)B$.

Let $Q = zz^T$. Then the objective function becomes $z^T W z = \text{tr}(WQ)$ and Eq. (14) is transformed to:

$$\begin{aligned} \min_{Q} \quad & \text{tr}(WQ) \\ \text{s.t.} \quad & \text{tr}(\Omega_i Q) = 0, \quad i = 1, \cdots, m, \\ & Q \succeq 0, \quad \text{rank}(Q) \leq 1. \end{aligned} \tag{15}$$

We still solve problem (15) by the alternating direction method [57]. We introduce an auxiliary variable P and rewrite the problem as:

$$\begin{aligned} \min_{Q,P} \quad & \text{tr}(WQ) \\ \text{s.t.} \quad & \text{tr}(\Omega_i Q) = 0, \quad i = 1, \cdots, m, \\ & P = Q, \quad \text{rank}(P) \leq 1, \quad P \succeq 0. \end{aligned} \tag{16}$$

Its augmented Lagrangian function is:

$$\mathcal{L}_2(Q, P, G, \delta) = \text{tr}(WQ) + \text{tr}(G^T(Q - P)) + \frac{\delta}{2}\|Q - P\|_F^2$$

where G is the Lagrange Multiplier and $\delta > 0$ is the penalty parameter. We update Q and P alternately.

- Update Q:

$$Q^{l+1} = \underset{\substack{\text{tr}(\Omega_i Q) = 0, \\ i = 1, \cdots, m}}{\text{argmin}} \quad \mathcal{L}_2(Q, P^l, G^l, \delta_l). \tag{17}$$

  This is a constrained least square problem and has a closed form solution.

- Update P: We discard the terms in $\mathcal{L}_2$ which are independent of P and update P by:

$$P^{l+1} = \underset{\substack{P \succeq 0, \\ \text{rank}(P) \leq 1}}{\text{argmin}} \quad \left\|P - \tilde{Q}\right\|_F^2 \tag{18}$$

  where $\tilde{Q} = Q^{l+1} + \frac{2}{\delta_l}G^l$. Note that $\left\|P - \tilde{Q}\right\|_F^2$ is equal to $\left\|P - \frac{\tilde{Q}^T + \tilde{Q}}{2}\right\|_F^2$. Then (18) has a closed form solution by the lemma A.1.

- Update G: We update the Lagrangian multiplier G by:

$$G^{l+1} = G^l + \delta^l(Q^{l+1} - P^{l+1}) \tag{19}$$

- Update δ: We update the penaly parameter by:

$$\delta^{l+1} = \min(\delta^l \cdot \rho, \delta^{max}), \tag{20}$$

  where $\rho \geq 1$ and δ^{max} are constant parameters.

Lemma A.1: The solution to

$$\min_{P}\|P - S\|_F^2 \quad \text{s.t.} \quad P \succeq 0, \quad \text{rank}(P) \leq 1 \tag{21}$$

is $P = \max(\xi_1, 0)\nu_1\nu_1^T$, where S is a symmetric matrix and ξ_1 and ν_1 are the largest eigenvalue and eigenvector of S, respectively.

Proof: Since P is a symmetric semi-positive definite matrix and its rank is one, we can write P as: $P = \xi\nu\nu^T$, where $\xi \geq 0$. Let the largest eigenvalue of S be ξ_1, then we have $\nu^T S \nu \leq \xi_1$, $\forall \nu$. Then we have:

$$\begin{aligned} \|P - S\|_F^2 & = \|P\|_F^2 + \|S\|_F^2 - 2\text{tr}(P^T S) \\ & \geq \xi^2 + \textstyle\sum_{i=1}^n \xi_i^2 - 2\xi\xi_1 \\ & = (\xi - \xi_1)^2 + \textstyle\sum_{i=2}^n \xi_i^2 \\ & \geq \textstyle\sum_{i=2}^n \xi_i^2 + \min(\xi_1, 0)^2 \end{aligned} \tag{22}$$

□

The minimum value can be achieved when $\xi = \max(\xi_1, 0)$ and $\nu = \nu_1$.

A.1.4 Update λ_1

We update the Lagrangian multiplier λ_1 by:

$$\lambda_1^{k+1} = \lambda_1^k + \eta^k\left(\gamma^{k+1} - x + M\left(B\alpha^{k+1} + \mu\right)\right) \tag{23}$$

A.1.5 Update λ_2

We update the Lagrangian multiplier λ_2 by:

$$\lambda_2^{k+1} = \lambda_2^k + \eta^k\left(\alpha^{k+1} - \beta^{k+1}\right) \tag{24}$$

A.1.6 Update η

We update the penalty parameter η by:

$$\eta^{k+1} = \min(\eta^k \cdot \rho, \eta^{max}), \tag{25}$$

where $\rho \geq 1$ and η^{max} are the constant parameters.

A.2 Camera Parameter Estimation

Given estimated 2D pose X and 3D pose Y, we estimate camera parameters by solving the following optimization problem:

$$\min_{m_1, m_2}\left\|X - \begin{pmatrix} m_1^T \\ m_2^T \end{pmatrix} Y\right\|_1, \quad \text{s.t.} \quad m_1^T m_2 = 0. \tag{26}$$

We introduce an auxiliary variable R and rewrite Eq. (26) as:

$$\begin{aligned} \min_{R, m_1, m_2} \quad & \|R\|_1 \\ \text{s.t.} \quad & R = X - \begin{pmatrix} m_1^T \\ m_2^T \end{pmatrix} Y, \quad m_1^T m_2 = 0. \end{aligned} \tag{27}$$

We still use ADM to solve problem (27). Its augmented Lagrangian function is:

$$\begin{aligned}&\mathcal{L}_3(R,m_1,m_2,H,\zeta,\tau)\\ =\ & \|R\|_1 + \text{tr}\left(H^T\left[\begin{pmatrix} m_1^T\\ m_2^T\end{pmatrix} Y + R - X\right]\right) + \zeta(m_1^T m_2)\\ &+\tfrac{\tau}{2}\left[\left\|\begin{pmatrix} m_1^T\\ m_2^T\end{pmatrix}Y+R-X\right\|_F^2 + \left(m_1^T m_2\right)^2\right]\end{aligned}$$

where H and ζ are Lagrange multipliers and $\tau > 0$ is the penalty parameter.

A.2.1 Update R

We discard the terms in $\mathcal{L}_3$ which are independent of R and update R by:

$$R^{k+1} = \underset{R}{\text{argmin}}\|R\|_1 + \frac{\tau_k}{2}\left\|R + \begin{pmatrix}\left(m_1^k\right)^T\\ \left(m_2^k\right)^T\end{pmatrix}Y - X + \frac{H^k}{\tau_k}\right\|_F^2$$

which has a closed form solution [57].

A.2.2 Update m_1

We discard the terms in $\mathcal{L}_3$ which are independent of m_1 and update m_1 by:

$$\begin{aligned}m_1^{k+1} = \underset{m_1}{\text{argmin}}&\left\|\begin{pmatrix} m_1^T\\ (m_2^k)^T\end{pmatrix}Y + R^{k+1} - X + \tfrac{H^k}{\tau_k}\right\|_F^2\\ &+\left(m_1^T m_2^k + \tfrac{\zeta^k}{\tau_k}\right)^2\end{aligned}$$

This is a least square problem and has a closed form solution.

A.2.3 Update m_2

We discard the terms in $\mathcal{L}_3$ which are independent of m_2 and update m_2 by:

$$\begin{aligned}m_2^{k+1} = \underset{m_2}{\text{argmin}}&\left\|\begin{pmatrix} \left(m_1^{k+1}\right)^T\\ m_2^T\end{pmatrix}Y + R^{k+1} - X + \tfrac{H^k}{\tau_k}\right\|_F^2\\ &+\left(\left(m_1^{k+1}\right)^T m_2 + \tfrac{\zeta^k}{\tau_k}\right)^2\end{aligned}$$

This is a least square problem and has a closed form solution.

A.2.4 Update H

We update Lagrange multiplier H by:

$$H^{k+1} = H^k + \tau^k\left(\begin{pmatrix}\left(m_1^{k+1}\right)^T\\ \left(m_2^{k+1}\right)^T\end{pmatrix}Y + R^{k+1} - X\right) \quad (28)$$

A.2.5 Update ζ

We update the Lagrange multiplier ζ by:

$$\zeta^{k+1} = \zeta^k + \tau^k\cdot\left(m_1^{k+1}\right)^T m_2^{k+1} \quad (29)$$

Acknowledgment

Y. Wang was supported by National Basic Research Program of China (973 Program) (grant no. 2015CB351800), National Natural Science Foundation (NSF) of China (grant nos. 61625201 and 61527804). Z. Lin was supported by 973 Program (grant no. 2015CB352502), NSF of China (grant nos. 61625301 and 61731018), Qualcomm, and Microsoft Research Asia. A. Yuille was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior/ Interior Business Center (DOI/IBC) contract number D17PC00345 .

References

[1] L. W. Campbell and A. F. Bobick, “Recognition of human body motion using phase space constraints,” in *ICCV*, 1995, pp. 624–630.

[2] Y. Yacoob and M. J. Black, “Parameterized modeling and recognition of activities,” in *ICCV*, 1998, pp. 120–127.

[3] C. Wang, Y. Wang, and A. L. Yuille, “An approach to pose-based action recognition,” in *CVPR*, 2013, pp. 915–922.

[4] J. Wang, Z. Liu, Y. Wu, and J. Yuan, “Mining actionlet ensemble for action recognition with depth cameras,” in *CVPR*, 2012, pp. 1290–1297.

[5] I. Laptev, “On space-time interest points,” *IJCV*, vol. 64, no. 2-3, pp. 107–123, 2005.

[6] Y. Yang and D. Ramanan, “Articulated pose estimation with flexible mixtures-of-parts,” in *CVPR*, 2011, pp. 1385–1392.

[7] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, “Progressive search space reduction for human pose estimation,” in *CVPR*, 2008, pp. 1–8.

[8] S. Ioffe and D. Forsyth, “Human tracking with mixtures of trees,” in *ICCV*, vol. 1, 2001, pp. 690–695.

[9] C. Ionescu, J. Carreira, and C. Sminchisescu, “Iterated second-order label sensitive pooling for 3d human pose estimation,” in *CVPR*, 2014, pp. 1661–1668.

[10] D. Ramanan, D. A. Forsyth, and A. Zisserman, “Strike a pose: Tracking people by finding stylized poses,” in *CVPR*, vol. 1, 2005, pp. 271–278.

[11] B. Sapp, D. Weiss, and B. Taskar, “Parsing human motion with stretchable models,” in *CVPR*, 2011, pp. 1281–1288.

[12] M. Dantone, J. Gall, C. Leistner, and L. Van Gool, “Human pose estimation using body parts dependent joint regressors,” in *CVPR*, 2013, pp. 3041–3048.

[13] V. Ramakrishna, T. Kanade, and Y. Sheikh, “Reconstructing 3d human pose from 2d image landmarks,” in *ECCV*, 2012, pp. 573–586.

[14] C. Wang, Y. Wang, Z. Lin, A. L. Yuille, and W. Gao, “Robust estimation of 3d human poses from single images,” in *CVPR*, 2014.

[15] C. J. Taylor, “Reconstruction of articulated objects from point correspondences in a single uncalibrated image,” in *CVPR*, vol. 1, 2000, pp. 677–684.

[16] H.-J. Lee and Z. Chen, “Determination of 3d human body postures from a single view,” *CVGIP*, vol. 30, no. 2, pp. 148–168, 1985.

[17] P. J. Huber, *Robust statistics*. Springer, 2011.

[18] P. Yadollahpour, D. Batra, and G. Shakhnarovich, “Diverse m-best solutions in mrfs,” in *Workshop on Discrete Optimization in Machine Learning, NIPS*, 2011.

[19] G. Pons-Moll and B. Rosenhahn, “Model-based pose estimation,” in *Visual analysis of humans*. Springer, 2011, pp. 139–170.

[20] M. W. Lee and I. Cohen, “Proposal maps driven mcmc for estimating human body pose in static images,” in *CVPR*, vol. 2, 2004, pp. II–334.

[21] J. M. Rehg, D. D. Morris, and T. Kanade, “Ambiguities in visual tracking of articulated objects using two-and three-dimensional models,” *IJRR*, vol. 22, no. 6, pp. 393–418, 2003.

[22] C. Sminchisescu and B. Triggs, “Kinematic jump processes for monocular 3d human tracking,” in *CVPR*, vol. 1. IEEE, 2003, pp. I–I.

[23] ——, “Building roadmaps of minima and transitions in visual models,” *IJCV*, vol. 61, no. 1, pp. 81–101, 2005.

[24] E. Simo-Serra, A. Ramisa, G. Alenyà, C. Torras, and F. Moreno-Noguer, “Single Image 3D Human Pose Estimation from Noisy Observations,” in *CVPR*, 2012.

[25] E. Simo-Serra, A. Quattoni, C. Torras, and F. Moreno-Noguer, “A Joint Model for 2D and 3D Pose Estimation from a Single Image,” in *CVPR*, 2013.

[26] J. Valmadre and S. Lucey, “Deterministic 3d human pose estimation using rigid structure,” in *ECCV*, 2010, pp. 467–480.

[27] X. K. Wei and J. Chai, "Modeling 3d human poses from uncalibrated monocular images," in *ICCV*, 2009, pp. 1873–1880.

[28] C. Barron and I. A. Kakadiaris, "Estimating anthropometry and pose from a single image," in *CVPR*, vol. 1, 2000, pp. 669–676.

[29] G. Pons-Moll, D. J. Fleet, and B. Rosenhahn, "Posebits for monocular human pose estimation," in *CVPR*, 2014, pp. 2337–2344.

[30] I. Akhter and M. J. Black, "Pose-conditioned joint angle limits for 3d human pose reconstruction," in *CVPR*, 2015, pp. 1446–1455.

[31] G. Mori and J. Malik, "Recovering 3d human body configurations using shape contexts," *PAMI*, vol. 28, no. 7, pp. 1052–1062, 2006.

[32] A. Elgammal and C.-S. Lee, "Inferring 3d body pose from silhouettes using activity manifold learning," in *CVPR*, vol. 2, 2004, pp. II–681.

[33] G. Shakhnarovich, P. Viola, and T. Darrell, "Fast pose estimation with parameter-sensitive hashing," in *ICCV*, 2003, pp. 750–757.

[34] A. Agarwal and B. Triggs, "Recovering 3d human pose from monocular images," *PAMI*, vol. 28, no. 1, pp. 44–58, 2006.

[35] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. J. Black, "Keep it smpl: Automatic estimation of 3d human pose and shape from a single image," in *ECCV*. Springer, 2016, pp. 561–578.

[36] A.-I. Popa, M. Zanfir, and C. Sminchisescu, "Deep multitask architecture for integrated 2d and 3d human sensing," *arXiv preprint arXiv:1701.08985*, 2017.

[37] H. Yasin, U. Iqbal, B. Kruger, A. Weber, and J. Gall, "A dual-source approach for 3d pose estimation from a single image," in *CVPR*, 2016, pp. 4948–4956.

[38] L. Bo and C. Sminchisescu, "Twin gaussian processes for structured prediction," *IJCV*, vol. 87, no. 1-2, pp. 28–52, 2010.

[39] S. Li and A. B. Chan, "3d human pose estimation from monocular images with deep convolutional neural network," in *ACCV*. Springer, 2014, pp. 332–347.

[40] T. Meltzer, C. Yanover, and Y. Weiss, "Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation," in *ICCV*, vol. 1, 2005, pp. 428–435.

[41] C. Sminchisescu and A. Jepson, "Variational mixture smoothing for non-linear dynamical systems," in *CVPR*, vol. 2. IEEE, 2004, pp. II–II.

[42] D. Park and D. Ramanan, "N-best maximal decoders for part models," in *ICCV*. IEEE, 2011, pp. 2627–2634.

[43] D. Batra, P. Yadollahpour, A. Guzman-Rivera, and G. Shakhnarovich, "Diverse m-best solutions in markov random fields," in *ECCV*, 2012, pp. 1–16.

[44] V. Kazemi and J. Sullivan, "Using richer models for articulated pose estimation of footballers." in *BMVC*, 2012, pp. 1–10.

[45] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *ECCV*. Springer, 2016, pp. 483–499.

[46] A. Safonova, J. K. Hodgins, and N. S. Pollard, "Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces," *TOG*, vol. 23, no. 3, pp. 514–521, 2004.

[47] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *NIPS*, 2011, pp. 612–620.

[48] V. Ferrari, M. Marín-Jiménez, and A. Zisserman, "2d human pose estimation in tv shows," in *Statistical and Geometrical Approaches to Visual Motion Analysis*, 2009, pp. 128–147.

[49] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *ICML*. ACM, 2009, pp. 689–696.

[50] B. Daubney and X. Xie, "Tracking 3D human pose with large root node uncertainty," in *CVPR*, 2011, pp. 1321–1328.

[51] L. Sigal and M. J. Black, "Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion," *Brown Univertsity TR*, vol. 120, 2006.

[52] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments," *TPAMI*, vol. 36, no. 7, pp. 1325–1339, 2014.

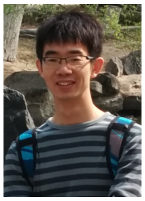
[53] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.0 beta," http://cvxr.com/cvx, Sep. 2013.

[54] S. Li, W. Zhang, and A. B. Chan, "Maximum-margin structured learning with deep networks for 3d human pose estimation," in *ICCV*, 2015, pp. 2848–2856.

[55] B. Tekin, X. Sun, X. Wang, V. Lepetit, and P. Fua, "Predicting peoples 3d poses from short sequences," *arXiv preprint arXiv: 1504.08200*, 2015.

[56] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "Sparseness meets deepness: 3d human pose estimation from monocular video," in *CVPR*, 2016, pp. 4966–4975.

[57] R. Liu, Z. Lin, and Z. Su, "Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning." ACML, 2013.



Chunyu Wang is an associate researcher in Microsoft Research Asia. He received his Ph.D in computer science from Peking University in 2016. His research interests are in computer vision, artificial intelligence and machine learning.

Yizhou Wang is a Professor of the Computer Science Department at Peking University, China. He received his Ph.D. in computer science from University of California at Los Angeles (UCLA) in 2005. He was a Research Staff of the Palo Alto Research Center (Xerox-PARC) from 2005 to 2008. His research interests include computer vision, statistical modeling and learning.

Zhouchen Lin (M'00-SM'08-F'18) received the Ph.D. degree in applied mathematics from Peking University in 2000. He is currently a Professor with the Key Laboratory of Machine Perception, School of Electronics Engineering and Computer Science, Peking University. His research interests include computer vision, image processing, machine learning, pattern recognition, and numerical optimization. He is an area chair of ACCV 2009/2018, CVPR 2014/2016, ICCV 2015, and NIPS 2015/2018, and senior program committee of AAAI 2016/2017/2018 and IJCAI 2016/2018. He is an Associate Editor of the IEEE Transactions on Pattern Analysis And Machine Intelligence and the International Journal of Computer Vision. He is a fellow of IAPR and IEEE.

Alan L. Yuille received the B.A. degree in mathematics and the Ph.D. degree in theoretical physics studying under Stephen Hawking from the University of Cambridge, in 1976 and 1980, respectively. He joined the Artificial Intelligence Laboratory, MIT, from 1982 to 1986, and followed this with a faculty position with the Division of Applied Sciences, Harvard, from 1986 to 1995. From 1995 to 2002, he was a Senior Scientist with the Smith-Kettlewell Eye Research Institute, San Francisco. In 2002, he accepted a position as a Full Professor with the Department of Statistics, University of California, Los Angeles. He has been a Bloomberg Distinguished Professor of Cognitive Science and Computer Science at Johns Hopkins University since 2017. He has over two hundred peer-reviewed publications in vision, neural networks, and physics, and has co-authored two books: Data Fusion for Sensory Information Processing Systems (with J. J. Clark) and Two- and Three-Dimensional Patterns of the Face (with P. W. Hallinan, G. G. Gordon, P. J. Giblin, and D. B. Mumford). He received several academic prizes.