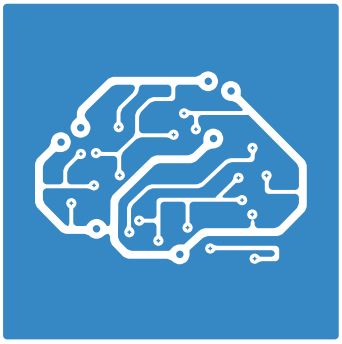


Center for Brains, Minds & Machines

CBMM Memo No. 013 **June 10, 2014**

Robust Estimation of 3D Human Poses from a Single Image

by

Chunyu Wang[1],Yizhou Wang[1],Zhouchen Lin[1],Alan L. Yuille[2],Wen Gao[1]
[1]Peking University, Beijing, China [2]University of California, Los Angeles
{wangchunyu,yizhou.wang, zlin}@pku.edu.cn @yuille@stat.ucla.edu wgao@pku.edu.cn

Abstract: Human pose estimation is a key step to action recognition. We propose a method of estimating 3D human poses from a single image, which works in conjunction with an existing 2D pose/joint detector. 3D pose estimation is challenging because multiple 3D poses may correspond to the same 2D pose after projection due to the lack of depth information. Moreover, current 2D pose estimators are usually inaccurate which may cause errors in the 3D estimation. We address the challenges in three ways: (i) We represent a 3D pose as a linear combination of a sparse set of bases learned from 3D human skeletons. (ii) We enforce limb length constraints to eliminate anthropomorphically implausible skeletons. (iii) We estimate a 3D pose by minimizing the L_1-norm error between the projection of the 3D pose and the corresponding 2D detection. The L_1-norm loss term is robust to inaccurate 2D joint estimations. We use the alternating direction method (ADM) to solve the optimization problem efficiently. Our approach outperforms the state-of-the-arts on three benchmark datasets.

This work was supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF - 1231216.

Robust Estimation of 3D Human Poses from a Single Image

Chunyu Wang[1,2], Yizhou Wang[1,2], Zhouchen Lin[2], Alan L. Yuille[3], and Wen Gao[1]

[1]Nat'l Engineering Lab for Video Technology, Sch'l of EECS, Peking University, Beijing, China
[2]Key Lab. of Machine Perception (MOE), Sch'l of EECS, Peking University, Beijing, China
[3]Department of Statistics, University of California, Los Angeles (UCLA), USA

Abstract

Human pose estimation is a key step to action recognition. We propose a method of estimating 3D human poses from a single image, which works in conjunction with an existing 2D pose/joint detector. 3D pose estimation is challenging because multiple 3D poses may correspond to the same 2D pose after projection due to the lack of depth information. Moreover, current 2D pose estimators are usually inaccurate which may cause errors in the 3D estimation. We address the challenges in three ways: (i) We represent a 3D pose as a linear combination of a sparse set of bases learned from 3D human skeletons. (ii) We enforce limb length constraints to eliminate anthropomorphically implausible skeletons. (iii) We estimate a 3D pose by minimizing the L_1-norm error between the projection of the 3D pose and the corresponding 2D detection. The L_1-norm loss term is robust to inaccurate 2D joint estimations. We use the alternating direction method (ADM) to solve the optimization problem efficiently. Our approach outperforms the state-of-the-arts on three benchmark datasets.

1. Introduction

Action recognition is a key problem in computer vision [19] and has many applications such as human-computer interaction and video surveillance. Since an action is naturally represented by human poses [18], 2D and 3D pose estimation has attracted a lot of attention. A 2D pose is usually represented by a set of joint locations [21] whose estimation remains challenging because of the huge human appearance variation, viewpoint change, etc.

A 3D pose is typically represented by a skeleton model parameterized by joint locations [16] or by rotation angles [8]. The representation is intrinsic as it is invariant to viewpoint changes. However, estimating 3D poses from a single image remains a difficult problem. First, a 3D pose is usually inferred from 2D joint locations. So, the accuracy of 2D joint estimation can greatly affect the 3D estimation performance. Second, multiple 3D poses may correspond to the same 2D pose after projection. This introduces severe ambiguities in 3D pose estimation. Third, the problem is further complicated when camera parameters are unknown.

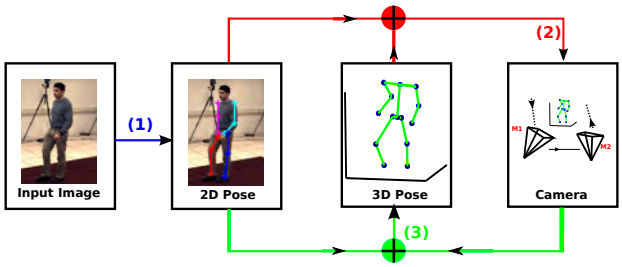


Figure 1. **Method overview**. (1) On a test image, we first estimate the 2D joint locations and initialize a 3D pose. (2) Then camera parameters are estimated from the 2D and 3D poses. (3) Next we update the 3D pose with the current camera parameters and the 2D pose. We repeat steps (2) and (3) until convergence.

We propose a novel method, which alternately updates the 3D pose and camera parameters. Figure 1 shows the overview of the method. On an input image, we first employ a 2D pose estimator (*e.g.* [21]) to detect the 2D joints. Then we initialize a 3D pose (*e.g.* the mean pose). Using both the poses, we estimate the camera parameters (step 2). Next, we re-estimate the 3D pose with the current camera parameters (step 3). Step 2 and 3 are iterated until convergence.

We represent a 3D human pose by a linear combination of a set of overcomplete bases. Since human poses lie in a low dimensional space [3], in the basis pursuit optimization, we enforce an L_1-norm regularization on the basis coefficients so that only a few of them are activated. Such holistic representation is able to reduce the ambiguities in the 3D pose estimation and is robust to occlusions (*e.g.* missing joints), because it encodes the structural prior of the human skeleton manifold.

We estimate a 3D pose (*i.e.* basis coefficients) by minimizing an L_1-norm penalty between the projection of the 3D joints and the 2D detections. The commonly used L_2-norm tends to distribute errors evenly over all variables. When some joints of the estimated 2D pose are inaccurate, the inferred 3D pose may be biased to a completely wrong configuration. In contrast, L_1-norm is more tolerant to the inaccurate 2D joints. However, even if the L_1-norm error is adopted, the inferred 3D skeleton may still violate the anthropometric quantities such as limb proportions. Hence, we enforce eight limb length constraints on the estimated 3D pose to eliminate the incorrect ones.

We use an efficient alternating direction method (ADM) to solve our optimization problem. Although global optimality is not guaranteed, we obtain reasonably good solutions. Our method outperforms the state-of-the-arts on three benchmark datasets.

The paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the proposed approach. Section 4 shows implementation details and experiment results. Conclusion is in section 5. Section 6 (Appendix) presents the optimization method in detail.

2. Related Work

Existing work on 3D pose estimation can be classified into four categories according to their inputs to the system, *e.g.* the image, image features, camera parameters, etc. The first class [7] [15] takes camera parameters as inputs. For example, Lee et al. [7] represent a 3D pose by a skeleton model and parameterize the body parts by truncated cones. They estimate the rotation angles of body parts by minimizing the silhouette discrepancy between the model projections and the input image by applying Markov Chain Monte Carlo (MCMC). Simo-Serra et al. [15] represent a 3D pose by a set of joint locations. They automatically estimate the 2D pose, model each joint by a Gaussian distribution, and propagate the uncertainty to 3D pose space. They sample a set of 3D skeletons from the space and learn a SVM to determine the most feasible one.

The second class [17] [20] requires manually labeled 2D joint locations in a video as input. Valmadre et al. [17] first apply structure from motion to estimate the camera parameters and the 3D pose of the rigid torso, and then require human input to resolve the depth ambiguities for non-torso joints. Wei et al. [20] propose the "rigid body constraints", *e.g.* the pelvis, left and right hip joints form a rigid structure, and require that the distance between any two joints on the rigid structure remains unchanged across time. They estimate the 3D poses by minimizing the discrepancy between the projection of the 3D poses and the 2D joint detections without violating the "rigid body constraints".

The third class [16] [12] requires manually labeled 2D joints in one image. Taylor [16] assumes that the limb lengths are known and calculates the relative depths of the limbs by considering foreshortening. It requires human input to resolve the depth ambiguities at each joint. Ramakrishna et al. [12] represent a 3D pose by a linear combination of a set of bases. They split the training data into classes, apply PCA to each class, and combine the principal components as bases. They greedily add the most correlated basis into the model and estimate the coefficients by minimizing an L_2-norm error between the projection of 3D pose and the 2D pose. They enforce a constraint on the sum of the limb lengths, which is just a weak constraint. This work [12] achieves the state-of-the-art performance but relies on *manually labeled* 2D joint locations.

The fourth class [11] [3] requires only a single image or image features (*e.g.* silhouettes). For example, Mori et al. [11] match a test image to the stored exemplars using shape context descriptors, and transfer the matched 2D pose to the test image. They lift the 2D pose to 3D using the method proposed in [16]. Elgammal et al.[3] propose to learn view-based silhouettes manifolds and the mapping function from the manifold to 3D poses. These approaches do not explicitly estimate camera parameters, but require a lot of training data from various viewpoints.

Our method requires only a single image to infer 3D human poses. It is similar to [12] but there are five distinctive differences. (i) We obtain 2D joint locations by running a detector [21] rather than by manual labeling. (ii) We use L_1-norm penalty instead of the L_2-norm one as it is more robust to inaccurate 2D joint locations. (iii) They [12] enforce a weak anthropomorphic constraint (*i.e.* sum of limb length) for the sake of computational simplicity, which is insufficient to eliminate incorrect poses; while we enforce eight limb length constraints, which is much more effective. (iv) We enforce an L_1-norm constraint on the basis coefficients rather than greedily adding bases into the model to encourage sparsity. They need to re-estimate the coefficients every time a new basis is introduced, which is inefficient. (v) We use an efficient alternating direction method to solve our optimization problem.

3. Our Approach

We represent 2D and 3D poses by n joint locations $x \in \mathbb{R}^{2n}$ and $y \in \mathbb{R}^{3n}$, respectively. By assuming a weak perspective camera model, the 2D projection x of a 3D pose y in an image are related as: $x = My$, where $M = I_n \otimes M_0$, in which I is the identity matrix, $\otimes$ is the Kronecker product, and $M_0 = \begin{pmatrix} m_1^T \\ m_2^T \end{pmatrix} \in \mathbb{R}^{2\times 3}$ is the camera projection matrix. Given the estimated x, we alternately estimate the camera parameter M_0 and the 3D pose y. We describe the details for 3D pose estimation in section 3.1 and for camera parameter estimation in section 3.2.

3.1. Robust 3D Pose Estimation

We represent a 3D pose y as a linear combination of a set of bases $B = \{b_1, \cdots, b_k\}$, *i.e.* $y = \sum_{i=1}^{k} \alpha_i \cdot b_i + \mu$, where α are the basis coefficients and μ is the mean pose. Given a 2D pose x and camera parameter M_0, we estimate the coefficients α by minimizing an L_1-norm error between the projection of the estimated 3D pose and the 2D pose: $\|M(B\alpha + \mu) - x\|_1$. We also enforce L_1-norm regularization on the basis coefficients α and eight limb length constraints on the inferred 3D pose.

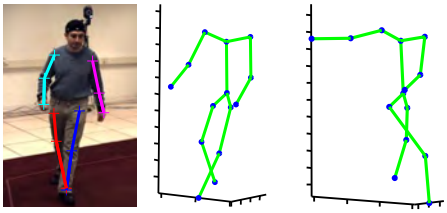


Figure 2. Comparison of 3D pose estimation by minimizing L_1-norm vs L_2-norm penalty. **(a)** estimated 2D joint locations where the right foot location is inaccurate. **(b-c)** are the estimated 3D poses using the L_1-norm and L_2-norm, respectively. The L_2-norm penalty biases the estimation to a wrong pose.

3.1.1 L_1-norm Objective Function

L_2-norm is the most widely used error metric in the literature. However, it is sensitive to inaccuracies in 2D pose estimation, which are usually caused by failures in feature detections and other factors, because it tends to distribute errors uniformly. In this work, we propose to minimize an L_1-norm error, *i.e.* $\|x - M(B\alpha + \mu)\|_1$. As a widely used robust regularizer in statistics, the L_1 penalty is robust to inaccurate 2D joint outliers. For example, in Figure 2 the 2D location of the right foot is inaccurate. The estimated 3D pose using L_2-norm error is drastically biased to a wrong configuration. The camera parameter estimation is also incorrect. However, using L_1-norm returns a reasonable 3D pose. Extensive experiments on benchmark datasets justify that using the L_1-norm can improve the performance, especially when 2D pose estimation is inaccurate.

3.1.2 Sparsity Constraint on the Basis Coefficients

Although human poses are highly variant, they lie in a low dimensional space [13]. Hence, we enforce sparsity on the basis coefficients α so that each 3D pose is represented by only a few bases. The sparsity can be induced by minimizing the L_1-norm of α. This is an important structural prior to remove incorrect or anthropomorphically implausible 3D poses. In addition, the sparsity constraint can also prevent overfitting to (inaccurate) 2D pose estimations. If there is no sparsity constraint, given a large number of bases we can always decrease the projection error to zero for an arbitrary 2D pose; however, there is no guarantee that the resulted 3D pose is correct. In experiments, we observe that the sparsity constraint is quite important. In summary, the resulted objective function is:

$$\min_{\alpha} \quad \|x - M(B\alpha + \mu)\|_1 + \theta \|\alpha\|_1 \tag{1}$$

where $\theta > 0$ is a parameter that balances the loss term and the regularization term.

3.1.3 Anthropomorphic Constraints

We require that the eight limb lengths of a 3D pose comply with certain proportions [6]. The eight limbs are left/right-upper/lower-arm/leg. 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 the identity matrix. The product of E_j and y is 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 length of the right lower leg to one and compute the squared lengths of other limbs (say L_i) according to the limb proportions used in [6]. The proportions are kept the same for all people. Now we have constraints $\|C_i(B\alpha + \mu)\|_2^2 = L_i$. Given the camera parameters we can formulate the 3D pose estimation problem as follows:

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

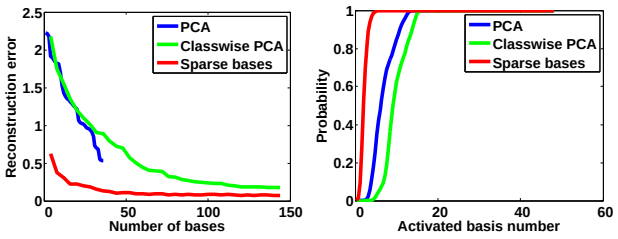
3.2. Robust Camera Parameter Estimation

Given a 3D pose, we estimate the camera parameters by minimizing the L_1-norm projection error. We reshape the 2D and 3D poses, x and y, as $X \in \mathbb{R}^{2\times n}$ and $Y \in \mathbb{R}^{3\times n}$, respectively. Then ideally $X = M_0 Y$ should hold, where $M_0 = \begin{pmatrix} m_1^T \\ m_2^T \end{pmatrix}$ is the projection matrix of a weak projective camera, *i.e.* $m_1^T m_2 = 0$. Due to errors, we estimate the camera parameters m_1 and m_2 by solving the following problem:

$$\min_{m_1, m_2} \left\| X - \begin{pmatrix} m_1^T \\ m_2^T \end{pmatrix} Y \right\|_1, \quad \text{s.t.} \quad m_1^T m_2 = 0. \tag{3}$$

3.3. Optimization

We alternately update the 3D pose and the camera parameters. We first initialize the 3D pose X by the mean pose of the training data, and estimate camera parameters m_1 and m_2 by solving problem (3). With the updated camera parameters, we then re-estimate the 3D pose by solving problem (2). We repeat the above process until convergence or the maximum number of iterations is reached. We use the alternating direction method to solve the two optimization problems efficiently. Please see Appendix for details.



(a) (b)

Figure 3. Comparison of the three basis learning methods on the CMU dataset. **(a)** 3D pose reconstruction errors using different number of bases. **(b)** Cumulative distribution of the number of activated bases in represent the 3D poses. The y-axis is the percentage of the cases whose activated basis number is less than or equal to the corresponding x-axis value on the curves.

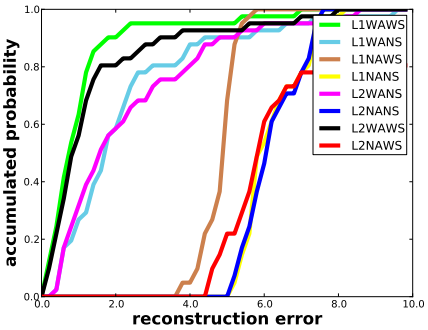


Figure 4. **Controlled experiment:** Cumulative distribution of 3D pose estimation errors on the CMU dataset. The y-axis is the percentage of the cases whose estimation error is less than or equal to the corresponding x-axis value on the curves.

4. The Experimental Results

We conduct two types of experiments to evaluate our approach. The first type is controlled. We assume that the 2D joint locations are known and evaluate the influence: (i) of the **three factors** (*i.e.* the sparsity term, the anthropomorphic constraints and the L_1-norm penalty), (ii) of the inaccurate 2D pose estimations and (iii) of the human-camera angles, on the 3D pose estimation performance. The second type is real. We estimate the 2D pose in an image by a detector [21] and then estimate the 3D skeletons. We compare our method with the state-of-the-art ones [12] [15] [2]. Our approach can also refine the 2D pose estimation by projecting the estimated 3D pose to 2D image.

We use 12 body joints, *i.e.* the left and right shoulders, elbows, hands, hips, knees and feet, being consistent with the 2D pose detector [21]. 200 bases are used for all experiments and about 6 of them are activated for representing a 3D pose. In optimization, we terminate the algorithm if the number of iterations exceeds 20.

4.1. The Datasets

We evaluate our approach on three datasets: the CMU motion dataset [1], the HumanEva dataset [14] and the UvA 3D pose dataset [5]. For the CMU dataset, we learn the bases on actions of "climb", "swing", "sit" and "jump", and test on different actions of "walk", "run", "golf" and "punch" to justify the generalization ability of our method. For the HumanEva dataset, we use the walking and jogging actions of three subjects for evaluation as in [15]. For the UvA dataset, we use the first four sequences for training and the remaining eight for testing.

4.2. Basis Learning

Our approach pursues a set of sparse bases by enforcing an L_1-norm regularization on the basis coefficients (as in [10]). But we also compare with other two basis learning methods. The first method applies principle component analysis (PCA) on the training set and uses the first k principal components as the bases. The second splits the training set into different classes by action labels, then applies PCA on each class, and finally collect the principal components as bases (which we call classwise PCA) as in [12].

We learn the bases on the training data (of four action classes) of the CMU dataset, and reconstruct each test 3D pose by solving an L_1-norm regularized least square problem. The reconstruction errors are shown in Figure 3 (a). The sparse bases consistently achieve the lowest errors among the three methods. Note that the maximum number of bases for PCA and classwise PCA is 36 (i.e. the dimension of a 3D pose) and 144, respectively. In addition, fewer bases are activated using the L_1-norm induced bases (see Figure 3 (b)). This justifies the bases' representative power.

4.3. Controlled Experiments

We assume the ground-truth 2D pose x is known and recover the 3D pose y from x. The residual error between the estimated 3D pose $\hat{y}$ and the ground truth y, *i.e.* $||y - \hat{y}||_2$, is used as the evaluation criterion as in [12].

4.3.1 Influence of the Three Factors

We design seven baselines to evaluate the influence of the three factors, *i.e.* the sparsity term, the anthropomorphic constraints and the L_1-norm penalty. The first baseline is symbolized as L2NAWS, *i.e.* the approach uses an L_2-norm objective function, No Anthropomorphic constraints and With the Sparsity constraint. The remaining baselines are L2NANS, L2WANS, L2WAWS, L1NANS, L1NAWS and L1WANS, which can be similarly understood by their names. We solve the optimization problem in L2WANS and L2WAWS by the alternating direction method. The optimization problems in other baselines can be solved trivially.

Figure 4 shows the results on the CMU dataset. First, the baselines without the sparsity term perform worse than those with the sparsity term. Second, enforcing limb length

Table 1. **Real experiment on the HumanEva dataset:** comparison with the state-of-the-art methods [15] [2]. We present results for both walking and jogging actions of all three subjects and camera C1. The numbers in each cell are the root mean square error (RMS) and standard deviation, respectively. We use the unit of millimeter as in [15] and [2]. The length of the right lower leg is about 380 mm. See Section 4.4.1.

Walking	S1	S2	S3
Ours	71.9 (19.0)	75.7 (15.9)	85.3 (10.3)
[15]	99.6 (42.6)	108.3 (42.3)	127.4 (24.0)
[2]	89.3	108.7	113.5
Jogging	S1	S2	S3
Ours	62.6 (10.2)	77.7 (12.1)	54.4 (9.0)
[15]	109.2 (41.5)	93.1 (41.1)	115.8 (40.6)

constraints improves the performance (*e.g.* L2WAWS outperforms L2NAWS). Third, L_1-norm outperforms L_2-norm (*e.g.* L1NAWS is better than L2NAWS). Finally, our approach performs best among the baselines.

4.3.2 Influence of Inaccurate 2D Poses

We evaluate the robustness of our approach against inaccurate 2D pose estimations. We synthesize noisy 2D poses by generating ten levels of random Gaussian noises and adding them to the original 2D poses. The magnitude of the tenth (largest) level noise is one, which is equal to the normalized length of the right lower leg. We estimate the 3D poses from those corrupted 2D joints.

Figure 5 shows the results. First, L1NANS outperforms L2NANS, which demonstrates that L_1-norm is more robust to 2D pose errors. Second, L2NANS and L2WANS get larger errors than L2NAWS and L2WAWS, respectively, which shows the importance of sparsity in handling inaccurate 2D poses. Our approach achieves a better performance than all baselines and Ramakrishna et al's method [12].

4.3.3 Influence of Human-Camera Angles

We explore the influence of human-camera angles on 3D pose estimation. 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. We estimate the 3D poses from their 2D projections. Figure 6 shows that the estimation errors using [12] increase drastically as human moves from profile (90 degrees) towards frontal pose (0 degree). This may be due to the fact that frontal view has more severe foreshortenings than the profile view, hence introduces more ambiguities into 3D pose estimation. Our approach is more robust against viewpoint changes.

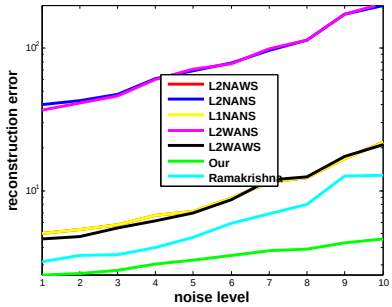


Figure 5. **Controlled experiment on the CMU dataset:** 3D pose estimation errors when different levels of noises are added to 2D poses. See Section 4.3.2.

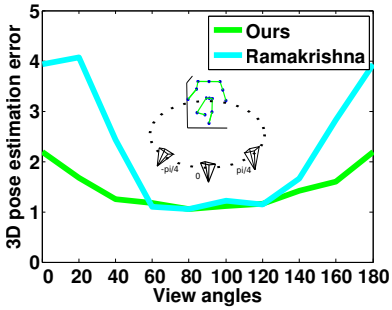


Figure 6. **Controlled experiment on the CMU dataset:** 3D pose estimation error when the human-camera angle varies from 0 to 180. See Section 4.3.3.

4.4. Real Experiments

Given an image, we first detect the 2D joint locations by a detector [21], from which we estimate the corresponding 3D pose using the proposed approach.

4.4.1 Comparisons to the State-of-the-arts

We compare our 3D pose estimator against a state-of-the-art method [12] on the UvA dataset. Figure 7 shows the estimation errors on each joint. Our approach achieves smaller estimation errors on all joints, especially for the left and the right hands. This proves that our approach is robust to inaccurate 2D joint locations. We also compare our approach with the state-of-the-arts [15] [2] on the HumanEva dataset. Table 1 shows the root mean square errors adopted in [15]. Our approach outperforms both [15] and [2].

4.4.2 Evaluation on Camera Parameter Estimation

Our camera parameter estimation usually converges within nine iterations. Figure 8 shows the 3D pose estimation results using the estimated cameras and groundtruth cameras, respectively. We can see that the difference is subtle for 70% of cases. We discover that the initialization of the 3D pose can influence the estimation accuracy. So we cluster the training poses into 30 clusters and initialize the 3D pose

Table 2. **Real experiment on the UvA dataset:** Comparison of 2D pose estimation results. We report: (1) the Probability of Correct Pose (PCP) for the eight body parts (*i.e.* left upper arm (LUA), left lower arm (LLA), right upper arm (RUA), right lower arm (RLA), left upper leg (LUL), left lower leg (LLL), right upper leg (RUL) and right lower leg (RLL)), (2) PCP for 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. [21]	0.751	**0.416**	0.771	**0.286**	0.857	0.825	0.910	0.894	0.714	109
Ramakrishna et al. [12]	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**

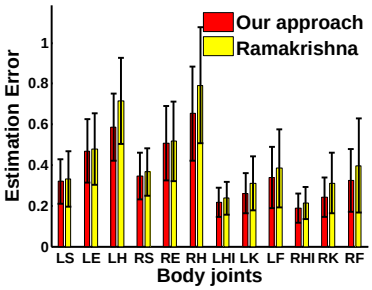


Figure 7. **Real experiment on the UvA dataset:** comparison with a state-of-the-art [12]. Both average estimation errors and standard deviations are shown for each joint (*i.e.* left shoulder, left elbow, left hand, right shoulder, right elbow, right hand, left hip, left knee, left foot, right hip, right knee and right foot). See Section 4.4.1.

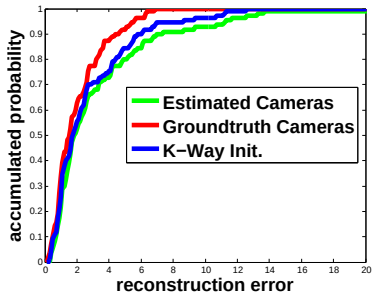


Figure 8. **Real experiment on the CMU dataset:** cumulative distribution of 3D pose estimation errors when camera parameters are (1) assigned by groundtruth, estimated by initializing the 3D pose with (2) mean pose, or (3) 30 cluster centers. The y-axis is the percentage of the cases whose estimation error is less than or equal to the corresponding x-axis value on the curves.

with each of the cluster centers for parallel optimization. We keep the one with the smallest error. We see that the performance can be further improved.

4.4.3 Evaluation on 2D Pose Estimation

We project the estimated 3D pose to 2D and compare with the original 2D estimation [21]. We report the results using two criteria. The first is the probability of correct pose (PCP) [21] — 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. The second criterion is the Euclidean distance between the estimated 2D pose and the groundtruth in pixels as in [15]. Table 2 shows 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 the best using the second criterion.

5. Conclusion

We address the problem of estimating 3D human poses from a single image. The approach is used in conjunction with an existing 2D pose detector. It is robust to inaccurate 2D pose estimations by using a sparse basis representation, anthropometric constraints and an L_1-norm projection error metric. We use an efficient alternating direction method to solve the optimization problem. Our approach outperforms the state-of-the art ones on three benchmark datasets.

Acknowledgements: We'd like to thank for the support from the following research grants NSFC-61272027, NSFC-61231010, NSFC-61121002, NSFC-61210005 and USA ARO Proposal 62250-CS. And, this material is based upon work supported by the Center for Minds, Brains and Machines (CBMM), funded by NSF STC award CCF-1231216. Z. Lin is supported by NSF China (grant nos. 61272341, 61231002, 61121002) and MSRA.

References

[1] CMU human motion capture database. Availabel online at http://mocap.cs.cmu.edu/search.html. 2003.

[2] B. Daubney and X. Xie. Tracking 3D human pose with large root node uncertainty. In *CVPR*, pages 1321–1328, 2011.

[3] A. Elgammal and C.-S. Lee. Inferring 3d body pose from silhouettes using activity manifold learning. In *CVPR*, volume 2, pages II–681. IEEE, 2004.

[4] M. Grant, S. Boyd, and Y. Ye. Cvx: Matlab software for disciplined convex programming, 2008.

[5] M. Hofmann and D. M. Gavrila. Multi-view 3d human pose estimation combining single-frame recovery, temporal integration and model adaptation. In *CVPR*, pages 2214–2221. IEEE, 2009.

[6] H.-J. Lee and Z. Chen. Determination of 3d human body postures from a single view. *CVGIP*, 30(2):148–168, 1985.

[7] M. W. Lee and I. Cohen. Proposal maps driven mcmc for estimating human body pose in static images. In *CVPR*, volume 2, pages II–334. IEEE, 2004.
[8] M. W. Lee and R. Nevatia. Human pose tracking in monocular sequence using multilevel structured models. *PAMI*, 31(1):27–38, 2009.
[9] 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.
[10] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *ICML*, pages 689–696. ACM, 2009.
[11] G. Mori and J. Malik. Recovering 3d human body configurations using shape contexts. *PAMI*, 28(7):1052–1062, 2006.
[12] V. Ramakrishna, T. Kanade, and Y. Sheikh. Reconstructing 3d human pose from 2d image landmarks. In *ECCV*, pages 573–586. Springer, 2012.
[13] A. Safonova, J. K. Hodgins, and N. S. Pollard. Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. *TOG*, 23(3):514–521, 2004.
[14] L. Sigal and M. J. Black. Humaneva: Synchronized video and motion capture dataset for evaluation of articulated human motion. *Brown Univertsity TR*, 120, 2006.
[15] 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.
[16] C. J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. In *CVPR*, volume 1, pages 677–684. IEEE, 2000.
[17] J. Valmadre and S. Lucey. Deterministic 3d human pose estimation using rigid structure. In *ECCV*, pages 467–480. Springer, 2010.
[18] C. Wang, Y. Wang, and A. L. Yuille. An approach to pose-based action recognition. In *CVPR*, pages 915–922, 2013.
[19] L. Wang, Y. Wang, and W. Gao. Mining layered grammar rules for action recognition. *International journal of computer vision*, 93(2):162–182, 2011.
[20] X. K. Wei and J. Chai. Modeling 3d human poses from uncalibrated monocular images. In *ICCV*, pages 1873–1880. IEEE, 2009.
[21] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392. IEEE, 2011.

6. Appendix: Optimization by ADM

Due to space limit, we only sketch the major steps of ADM for our optimization problems. In the following, k and l are the number of iterations.

6.1. 3D Pose Estimation

We introduce two auxiliary variables β and γ and rewrite Eq. (2) as:

$$\begin{aligned} \min_{\alpha,\beta,\gamma} \quad & \|\gamma\|_1 + \theta \|\beta\|_1 \\ \text{s.t.} \quad & \gamma = x - M(B\alpha + \mu), \quad \alpha = \beta, \\ & \|C_i(B\alpha + \mu)\|_2^2 = L_i, i = 1, \cdots, m. \end{aligned} \tag{4}$$

The augmented Lagrangian function of Eq. (4) is:

$$\begin{aligned} &\mathcal{L}_1(\alpha, \beta, \gamma, \lambda_1, \lambda_2, \eta) = ||\gamma||_1 + \theta||\beta||_1 + \\ &\lambda_1^T[\gamma - x + M(B\alpha + \mu)] + \lambda_2^T(\alpha - \beta) + \\ &\frac{\eta}{2}\left[||\gamma - x + M(B\alpha + \mu)||^2 + ||\alpha - \beta||^2\right] \end{aligned}$$

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 alternately.

6.1.1 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 [9].

6.1.2 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 [9].

6.1.3 Update α

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

$$\begin{aligned} \alpha^{k+1} = \arg\min_{\alpha} \quad & z^T W z \\ \text{s.t.} \quad & z^T \Omega_i z = 0, \quad i = 1, \cdots, m \end{aligned} \tag{5}$$

where $z = [\alpha^T \quad 1]^T$,

$$W = \begin{pmatrix} B^T M^T M B + I & 0 \\ 2\left[\left(\gamma^{k+1} - x + M\mu + \frac{\lambda_1^k}{\eta_k}\right)^T M B - \beta^{k+1} + \frac{\lambda_2^k}{\eta_k}\right] & 0 \end{pmatrix}$$

and $\Omega_i = \begin{pmatrix} B^T C_i^T C_i B & B^T C_i^T C_i \mu \\ \mu^T C_i^T C_i B & \mu^T C_i^T C_i \mu - L_i \end{pmatrix}$.

Let $Q = zz^T$. Then the objective function becomes $z^T W z = \text{tr}(WQ)$ and Eq. (5) is transformed to:

$$\begin{aligned} \min_{Q} \quad & \text{tr}(WQ) \\ \text{s.t.} \quad & \text{tr}(\Omega_i Q) = 0, \quad i = 1, \cdots, m, \\ & Q \succeq 0, \quad \text{rank}(Q) \leq 1. \end{aligned} \tag{6}$$

We still solve problem (6) by the alternating direction method [9]. We introduce an auxiliary variable P and

rewrite the problem as:

$$\begin{array}{ll}\min\limits_{Q,P} & \text{tr}(WQ) \\ \text{s.t.} & \text{tr}(\Omega_i Q)=0, \quad i=1,\cdots,m, \\ & P=Q, \quad \text{rank}(P)\leq 1, \quad P\succeq 0.\end{array} \tag{7}$$

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:

$$Q^{l+1}=\begin{array}{l}\text{argmin} \\ \text{tr}(\Omega_i Q)=0, \\ i=1,\cdots,m\end{array} \mathcal{L}_2(Q,P^l,G^l,\delta_l). \tag{8}$$

  This is convex and solved using CVX [4], a package for specifying and solving convex programs.

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

$$P^{l+1}=\begin{array}{l}\text{argmin} \\ P\succeq 0, \\ \text{rank}(P)\leq 1\end{array} \left\|P-\tilde{Q}\right\|_F^2 \tag{9}$$

  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 (9) has a closed form solution by the following lemma.

Lemma 6.1 *The solution to*

$$\min_P ||P-S||_F^2 \quad \textit{s.t.} \quad P\succeq 0, \quad \textit{rank}(P)\leq 1 \tag{10}$$

is $P=\max(\zeta_1,0)\nu_1\nu_1^T$, where S is a symmetric matrix and ζ_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: $P=\zeta\nu\nu^T$, where $\zeta\geq 0$. Let the largest eigenvalue of S be ζ_1, then we have $\nu^T S\nu\leq\zeta_1, \forall\nu$. Then we have:

$$\begin{array}{ll}||P-S||_F^2 & =||P||_F^2+||S||_F^2-2\text{tr}(P^TS) \\ & \geq \zeta^2+\sum_{i=1}^n\zeta_i^2-2\zeta\zeta_1 \\ & =(\zeta-\zeta_1)^2+\sum_{i=2}^n\zeta_i^2 \\ & \geq \sum_{i=2}^n\zeta_i^2+\min(\zeta_1,0)^2\end{array} \tag{11}$$

The minimum value can be achieved when $\zeta=\max(\zeta_1,0)$ and $\nu=\nu_1$.

6.2. Camera Parameter Estimation

We introduce an auxiliary variable R and rewrite Eq. (3):

$$\begin{array}{ll}\min\limits_{R,m_1,m_2} & \|R\|_1 \\ \text{s.t.} & R=X-\begin{pmatrix} m_1^T \\ m_2^T\end{pmatrix}Y, \quad m_1^Tm_2=0.\end{array} \tag{12}$$

We still use ADM to solve problem (12). Its augmented Lagrangian function is:

$$\begin{array}{ll} & \mathcal{L}_3(R,m_1,m_2,H,\zeta,\tau) \\ = & \|R\|_1+\text{tr}\left(H^T\left[\begin{pmatrix} m_1^T \\ m_2^T\end{pmatrix}Y+R-X\right]\right)+\zeta(m_1^Tm_2) \\ & +\frac{\tau}{2}\left[\left\|\begin{pmatrix} m_1^T \\ m_2^T\end{pmatrix}Y+R-X\right\|_F^2+\left(m_1^Tm_2\right)^2\right]\end{array}$$

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

6.2.1 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 [9].

6.2.2 Update m_1

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

$$\begin{array}{l}m_1^{k+1}=\underset{m_1}{\text{argmin}}\left\|\begin{pmatrix} m_1^T \\ \left(m_2^k\right)^T\end{pmatrix}Y+R^{k+1}-X+\frac{H^k}{\tau_k}\right\|_F^2 \\ \qquad +\left(m_1^Tm_2^k+\frac{\zeta^k}{\tau_k}\right)^2\end{array}$$

This has a closed form solution.

6.2.3 Update m_2

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

$$\begin{array}{l}m_2^{k+1}=\underset{m_2}{\text{argmin}}\left\|\begin{pmatrix} \left(m_1^{k+1}\right)^T \\ m_2^T\end{pmatrix}Y+R^{k+1}-X+\frac{H^k}{\tau_k}\right\|_F^2 \\ \qquad +\left(\left(m_1^{k+1}\right)^Tm_2+\frac{\zeta^k}{\tau_k}\right)^2\end{array}$$

This has a closed form solution.