Pose Locality Constrained Representation for 3D Human Pose Reconstruction

Xiaochuan Fan, Kang Zheng, Youjie Zhou, and Song Wang

Department of Computer Science & Engineering, University of South Carolina, USA
{fan23,zheng37,zhou42}@email.sc.edu, songwang@cec.sc.edu

Abstract. Reconstructing 3D human poses from a single 2D image is an ill-posed problem without considering the human body model. Explicitly enforcing physiological constraints is known to be non-convex and usually leads to difficulty in finding an optimal solution. An attractive alternative is to learn a prior model of the human body from a set of human pose data. In this paper, we develop a new approach, namely pose locality constrained representation (PLCR), to model the 3D human body and use it to improve 3D human pose reconstruction. In this approach, the human pose space is first hierarchically divided into lower-dimensional pose subspaces by subspace clustering. After that, a block-structural pose dictionary is constructed by concatenating the basis poses from all the pose subspaces. Finally, PLCR utilizes the block-structural pose dictionary to explicitly encourage pose locality in human-body modeling – nonzero coefficients are only assigned to the basis poses from a small number of pose subspaces that are close to each other in the pose-subspace hierarchy. We combine PLCR into the matching-pursuit based 3D human-pose reconstruction algorithm and show that the proposed PLCR-based algorithm outperforms the state-of-the-art algorithm that uses the standard sparse representation and physiological regularity in reconstructing a variety of human poses from both synthetic data and real images.

Keywords: 3D human pose reconstruction, subspace clustering, hierarchical pose tree.

1 Introduction

3D human pose reconstruction plays an important role in many vision applications, such as image retrieval, video surveillance and human-computer interaction. In this paper, we focus on the problem of reconstructing 3D human poses from the 2D locations of human joints that are annotated in a monocular image. Without considering any prior knowledge on human body, this is obviously an ill-posed problem. Previous works have explicitly utilized physiological knowledge of the human body, such as the body-segment length [6,11], the joint-angle limits [1] and the skeletal size [9], to regularize the 3D pose reconstruction. However, due to the large diversity of human poses, it is usually intractable to find an optimal solution under non-convex physiological constraints [13].

D. Fleet et al. (Eds.): ECCV 2014, Part I, LNCS 8689, pp. 174–188, 2014.

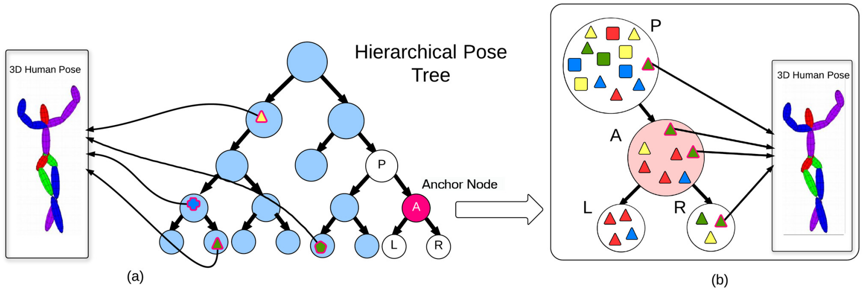


Fig. 1. An illustration of the proposed method. (a) The standard sparse representation allows the non-zero coefficients to be assigned to blocks (nodes) that are distant from each other. (b) The proposed PLCR-based algorithm assigns the non-zero coefficients only to a small set of blocks (nodes) that are close to each other. Basis poses are shown in different-shape elements (e.g., triangles, squares). The selected basis poses for 3D pose reconstruction are linked to the final reconstruction with arrows.

Recently, many efforts have been made in inferring the semantic concepts of the pose or action presented in the 2D image and then using these semantic concepts to help the 3D pose reconstruction. Ramakrishna et al [11] categorize human poses by the human actions, like walking and jumping, and construct sparse representations of human poses. Recently, supervised action classification [23,22,10] was also introduced to automate the action categorization for 3D pose reconstruction. While human actions are semantically well defined, one action may still consist of a variety of human poses and the action-based categorization may not provide sufficiently specific knowledge for 3D pose reconstruction. To address this problem, in this paper we propose a pose locality constrained representation (PLCR) approach for improving the 3D pose reconstruction. In this approach, we construct a hierarchical pose tree, as shown in Figure 1 to model the human poses by subspace clustering [8], where each tree node represents a lower-dimensional pose subspace and nodes with a larger depth in the tree represents more specific pose subspaces. In addition, nodes that are closer to each other in this tree indicate pose subspaces with higher similarity and/or overlap.

In using PLCR for 3D pose reconstruction, we build a block-structural dictionary by concatenating the basis poses from all the nodes of the tree and basis poses from each node constitute a block. With the dictionary, we apply the projected matching pursuit (PMP) algorithm to estimate the most likely 3D human pose. The proposed method explicitly encourages pose locality – nonzero coefficients are only assigned to the basis poses from a small number of blocks (tree nodes) that are close to each other. A comparison between the proposed PLCR representation and the standard sparse representation is shown in Figure 1, where the standard sparse representation may assign nonzero coefficients to distant blocks. Wang et al [19] have shown that locality is more important than sparsity in the image classification. In this paper, we show that, this observation also holds true for the ill-posed problem of 3D human pose reconstruction

– the proposed method can achieve better performance than the state-of-the-art method that uses the standard sparse representation.

2 Related Work

Low-dimensional action priors for pose reconstruction. Many human motion analysis systems used low-dimensional action priors to handle their problems such as human motion optimization [12], human action classification [21], and 3D human body pose tracking [5]. Recently, action priors were also used to assist 3D human pose reconstruction. Yao et al [22] used 2D action recognition as a prior for 3D pose reconstruction, where action specific regression models were trained separately based on low-level appearance features. More recently, Yu et al [23] used action detection on video snippets to derive strong spatiotemporal action priors, which was combined with part-based 2D pose estimation for 3D pose reconstruction. While providing a prior for reconstructing the 3D pose, action labels are still not sufficiently specific since poses from one action class may still show a large diversity.

3D pose reconstruction with physiological regularity. An example of early works on reconstructing 3D poses using physiological regularity is [4] in which physical and motion constraints were applied to prune a binary interpretation tree that records all possible body configurations. Liebowitz and Carlsson [6] assumed known body segment lengths and reconstructed 3D poses from uncalibrated multiple views by using articulated structural constraints. Taylor et al [16] recovered the poses from a single view by assuming known skeletal sizes and resolving the depth ambiguity manually. In [1], the maximum a posterior 3D trajectory was estimated based on a 3D kinematic model including joint angle limits, dynamic smoothing, and 3D key frames. [9] assumed known skeletal size and dealt with a perspective uncalibrated camera. Wei and Chai [20] reconstructed 3D human poses using the bone symmetric constraint from biomechanical data. Valmadre and Lucey [17] extended Wei and Chai [20]'s work by using a deterministic structure-from-motion method. As discussed above, due to the large diversity of human poses, it is usually intractable to find an optimal solution under non-convex physiological constraints [13].

Sparse representation for 3D pose reconstruction. Recently, Ramakrishna et al [11] presented an activity-independent pose-reconstruction method in which the 3D pose is sparsely represented by an overcomplete dictionary learned from a large motion capture dataset. A projected matching pursuit (PMP) algorithm was proposed to infer the underlying 3D poses and the camera settings by minimizing the reprojection error greedily. In this paper, we further introduce pose locality into 3D pose reconstruction – the sparse set of basis poses selected for reconstruction are always from a small number of specific subspaces with high similarity. Through experiments, we will show that the introduction of pose locality can further improve the accuracy of 3D pose reconstruction.

3 Proposed Method

In this section, we first give a formal definition of 3D human pose reconstruction from a 2D projection in Section 3.1. Then, in Section 3.2, we describe an unsupervised pose subspace clustering method for constructing hierarchical pose tree. Based on this tree, we detail the idea of the PLCR and the algorithm that use PLCR for 3D human pose reconstruction in Section 3.3. Finally, we summarize the entire PLCR-based algorithm for 3D pose reconstruction in Section 3.4.

3.1 Problem Description

A 3D human pose can be represented by a set of human joints $\mathbf{J} = \{\mathbf{j}_i\}_{i=1}^{L} \in \mathbb{R}^{3L\times 1}$, where $\mathbf{j}_i$ denotes the 3D coordinates of joint i and L is the number of human joints. In this paper, we are interested in estimating 3D joint locations $\mathbf{J}$ from their 2D projections $\mathbf{p} \in \mathbb{R}^{2L\times 1}$, with unknown camera parameters.

Under the weak perspective camera projection model, the projected 2D coordinates can be represented as

$$\mathbf{p} = (\mathbf{I}_L \otimes \mathbf{M})\,\mathbf{J} + \mathbf{1}_{L\times 1} \otimes \mathbf{T} \tag{1}$$

where $\otimes$ is the Kronecker product, $\mathbf{T} \in \mathbb{R}^{2\times 1}$ is the translation vector, and $\mathbf{M} \in \mathbb{R}^{2\times 3}$ contains both rotation and scaling parameters. Assuming that the camera intrinsic parameters are known, the degree of freedom of the camera parameters is 7. Therefore, in total there are $3L + 7$ unknowns while only $2L$ equations are available. Obviously, this is an under-determined problem, and we need to apply dimensionality reduction to make it determined.

However, due to the large diversity of human poses, a direct application of linear dimensionality reduction on the entire pose space is difficult and usually results in large reconstruction errors. This problem can be solved by restricting the pose reconstruction on a more specific pose subspace. To achieve this goal, two problems need to be addressed: 1) effectively dividing the entire pose space into subspaces, 2) finding the subspace in which the underlying 3D pose belongs to, based only on its 2D projection. For the first problem, we construct a hierarchical pose tree, where each node represents a pose subspace and the node with a larger depth in the tree represents a more specific pose subspace. For the second problem, given a 2D pose projection we find an anchor node in the tree by minimizing the reprojection error. In practice, the underlying 3D pose may not exactly belong to the subspace defined by the anchor node because of the information loss in 2D projection. To address this issue, we additionally include nodes close to the anchor node in the tree and use all their basis poses for 3D pose reconstruction.

3.2 Hierarchical Pose Tree

We construct pose subspaces with different levels of specificity by using subspace clustering. In particular, given a large set of 3D pose training data, we cluster them into different groups in a hierarchical way, such that each group of pose data represents a subspace.

Unsupervised Human Pose Subspace Clustering. Considering the code efficiency and availability, in this paper we use the low-rank representation algorithm [8,7] for 3D human pose subspace clustering. Other subspace clustering algorithms, such as the K-subspaces algorithm [18] and the sparse subspace clustering (SSC) algorithm [2,15], can also be used here.

Specifically, given a set of 3D human poses $\mathcal{J} = \{\mathbf{J}_i\}_{i=1}^{N} \in \mathbb{R}^{3L \times N}$, we first construct the lowest-rank representation $\mathbf{Z} \in \mathbb{R}^{N \times N}$ that satisfies $\mathcal{J} = \mathcal{J}\mathbf{Z}$. Let the skinny SVD of $\mathbf{Z}$ be $U\Sigma V^T$. We define the affinity matrix $\mathbf{W}$ as

$$w_{ij} = \left(\left[\widetilde{U}\widetilde{U}^T \right]_{ij} \right)^2 ,$$

where $\widetilde{U}$ is formed by $U\Sigma^{\frac{1}{2}}$ with normalized rows. Each element $w_{ij} \in \mathbf{W}$ measures the likelihood that two poses $\mathbf{J}_i$ and $\mathbf{J}_j$ are located in the same subspace. Finally, we apply the spectral clustering [14] on the affinity matrix $\mathbf{W}$.

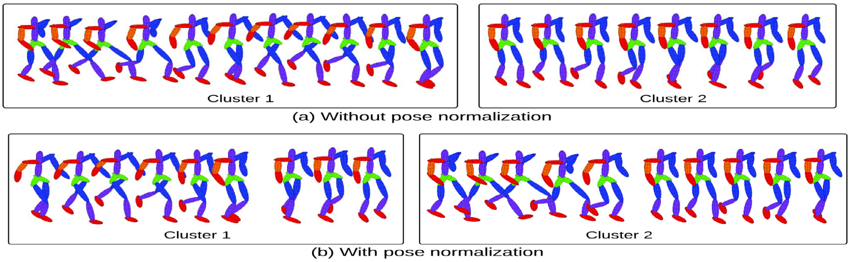
Pose Data Normalization. The goal of subspace clustering is to group similar poses, even performed by different subjects, into a same subspace. However, in practice, we found that the physiological difference between subjects may dominate the pose clustering, e.g., different poses from similar-size subjects may be clustered together. To address this problem, we propose to normalize all the 3D pose data before applying the above subspace clustering. In this paper, we normalize the length of each segment between adjacent joints in the human skeleton.

A segment that connects two joints $\mathbf{j}_a$ and $\mathbf{j}_b$ in the human skeleton can be written as $\mathbf{j}_a - \mathbf{j}_b$. We then convert it to the spherical coordinates as

$$\mathbf{j}_a - \mathbf{j}_b = (\theta_{ab}, \phi_{ab}, r_{ab}) ,$$

where, θ_{ab} is the zenith angle from the z axis, ϕ_{ab} is the azimuth angle from the x axis in the xy plane, and r_{ab} is the radius or the length of the segment. Obviously, r_{ab} is a constant for all the poses performed by a same subject, but different for the poses performed by different subjects. We normalize r_{ab} to the average length of this segment over all the training pose data. For the rigid parts of the human body, such as clavicles and hips, we also normalize the zenith and azimuth angles to be constants, by averaging over all the training pose data. After normalizing in the spherical coordinates, we convert the pose data back to the Cartesian coordinates. In this step, to ensure the segments are connected at the corresponding joints, we take advantage of the tree structure of the human skeleton – starting from the root (e.g., human head), normalized segments are assembled layer by layer to determine the coordinates of each joint. Figure 2 shows the sample subspace clusters with and without the normalization step. We can see that, with the data normalization, similar poses from different subjects can be clustered together.

Hierarchical Pose Tree. To construct subspaces with various levels of specificity, we recursively perform subspace clustering to construct a hierarchical pose



(a) Without pose normalization

(b) With pose normalization

Fig. 2. Sample subspace clustering results (a) without and (b) with the pose data normalization. With the normalization, the data that describe similar poses from different subjects are successfully clustered together, e.g., cluster 1 for the pose of moving right foot ahead and cluster 2 for the pose of moving left foot ahead.

tree – a cluster of pose data may be further clustered into smaller groups, where each group represents a more specific pose subspace. In this paper, we use two parameters, the branch factor K and a subspace complexity k, to control the number of clusters and the height of the resulting pose tree. A branch factor K indicates that each node in the pose tree, except for the leaves, has K children – each subspace is partitioned into K more specific subspaces in the recursive clustering. The subspace complexity k can be estimated using the method proposed in [7] – a subspace will not be further divided if $k < K$ and this subspace becomes a leaf node in the pose tree. This way, nodes with a larger depth in the constructed pose tree represent more specific pose subspaces and the pose similarity between different subspaces can be measured by the shortest-path distance between the corresponding nodes in the pose tree.

3.3 Pose Locality for Reconstruction

In this section, we first build a block-structural pose dictionary based on all the subspaces (nodes) in the constructed pose tree, taking the basis poses at each node as a block. We then describe a new pose locality constrained representation (PLCR) for reconstructing the 3D pose.

Block-Structural Pose Dictionary. As described in Section 3.2, each node in the constructed pose tree represents a pose subspace, which is described by a cluster of training pose data. At each node i, we can draw all pose data in the corresponding cluster and apply PCA to construct D_i basis poses, denoted as a block $\mathbf{B}_i$. The pose dictionary $\mathcal{B} = \{\mathbf{B}_i\}_{i=1}^{M}$ is constructed by concatenating the basis poses over all the M nodes. The total number of basis poses in the dictionary is $D = \sum_{i=1}^{M} D_i$. Thus, the pose dictionary can also be written as $\mathcal{B} = \{\mathbf{b}_j\}_{j=1}^{D}$, where each $\mathbf{b}_j$ denotes one basis pose.

Given a pose dictionary $\mathcal{B}$, each 3D human pose $\mathbf{J}$ can be represented by a linear combination of basis poses in $\mathcal{B}$, i.e.,

$$\mathbf{J} = \mathbf{m} + \mathcal{B}\boldsymbol{\Omega} = \mathbf{m} + \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_M \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\Omega}_1 \\ \vdots \\ \boldsymbol{\Omega}_M \end{bmatrix} = \mathbf{m} + \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_D \end{bmatrix}^T \begin{bmatrix} \omega_1 \\ \vdots \\ \omega_D \end{bmatrix} \tag{2}$$

where $\mathbf{m} \in \mathbb{R}^{3L\times 1}$ is the mean pose calculated over all the pose data and $\boldsymbol{\Omega} \in \mathbb{R}^{D\times 1}$ are the coefficients. We also denote $E = \|\boldsymbol{\Omega}\|_0$ to be the sparsity of $\boldsymbol{\Omega}$, with which the number of unknowns in Eq. (1) can be reduced to $E + 7$.

Pose Reconstruction with PLCR. For reconstructing the 3D pose from a 2D projection, we need to select E basis poses from the dictionary. Previous method [11] uses sparse representation to sequentially select E basis poses that minimize the reprojection error

$$\mathbf{R}(\mathcal{B}, \boldsymbol{\Omega}, \mathbf{M}, \mathbf{T}) = \mathbf{p} - (\mathbf{I}_L \otimes \mathbf{M}) \left(\mathbf{m} + \sum_{i=1}^{M} \mathbf{B}_i \boldsymbol{\Omega}_i \right) - \mathbf{1}_{L\times 1} \otimes \mathbf{T},$$

where the weak-projective camera parameters $\mathbf{M}$ and $\mathbf{T}$ can be refined iteratively with an initialization.

However, standard sparse representation does not enforce pose locality – the selected basis poses can be drawn from subspaces (nodes) that are far from each other in the pose tree. In this section, our main goal is to ensure that the E selected basis poses are drawn from a small number of subspaces (nodes) that are close to each other. To achieve this goal, we calculate the initial reprojection error $\mathbf{r}_i$ for each block $\mathbf{B}_i$ based on the initial camera parameters, i.e.

$$\mathbf{r}_i = \mathbf{R}\left(\mathbf{B}_i, \boldsymbol{\Omega}_i^*, \mathbf{M}, \mathbf{T}\right),$$

where the coefficients $\boldsymbol{\Omega}_i^*$ can be calculated by

$$\boldsymbol{\Omega}_i^* = \arg\min_{\boldsymbol{\Omega}_i} \|\mathbf{R}\left(\mathbf{B}_i, \boldsymbol{\Omega}_i, \mathbf{M}, \mathbf{T}\right)\|_2. \tag{3}$$

Given the 2D projection $\mathbf{p}$ of the pose and initial camera parameters, we define the *anchor node* $A(\mathbf{p})$ to be the node in the pose tree that leads to the smallest reprojection error, by using the basis poses at this node (subspace), i.e.,

$$A(\mathbf{p}) = \arg\min_i \|\mathbf{r}_i\|_2. \tag{4}$$

To make the search process of anchor node more efficient, we use the following top-down search algorithm.

1. Examine the root of the tree and calculate the reprojection error.
2. Examine all the K child nodes of the root and pick the one with the smallest reprojection error.

3. For the picked node, we further examine its K children and pick the one with the smallest reprojection error and repeat this process until we reach a leaf node.
4. All the picked nodes constitute a path from the root to a leaf and each node along this path has an associated reprojection error. We then pick the node along this path with the smallest reprojection error as the anchor node.

Given the information loss in the 2D projection, the anchor node may not provide a subspace that well describes the underlying 3D pose. We select E basis poses not only from the subspace described by the anchor node, but also from the nodes nearby. Specifically, we use the projected matching pursuit (PMP) to select the basis poses and in each iteration, a new pose basis $\mathbf{b}_{j*}$ is chosen from $\mathcal{B}$ by

$$j^* = \arg\min_j \ \left(|\theta_j| + \lambda d_j \right), \tag{5}$$

where θ_j is the angle between $(\mathbf{I} \otimes \mathbf{M})\,\mathbf{b}_j$ and the reprojection error $\mathbf{r}$ in the current iteration, λ is the locality weight. The locality adaptor

$$d_j = \begin{cases} e^{\frac{d(N(j),A(\mathbf{p}))}{\sigma}}, & \text{if } d\left(N\left(j\right), A\left(\mathbf{p}\right)\right) \leq d_M, \\ +\infty, & \text{otherwise}, \end{cases}$$

controls the pose locality – only the nodes (subspaces) that are close to the anchor node in the pose tree are included for basis-pose selection. $N\left(j\right)$ denotes the node (subspace) that $\mathbf{b}_j$ belongs to and $d\left(N\left(j\right), A\left(\mathbf{p}\right)\right)$ is the distance or the length of the shortest path between two nodes $N\left(j\right)$ and $A\left(\mathbf{p}\right)$ in the pose tree. d_M is a pre-set threshold and the nodes with a distance to the anchor that is larger than this threshold will not be included for basis-pose selection. Following [19], σ controls the weight decay speed for the locality adaptor. Using this technique to select basis poses, we can iteratively refine the reconstruction of the 3D pose and camera parameters using the PMP algorithm [11]. Note that, the proposed algorithm only selects one more basis pose $\mathbf{b}_j^*$ in each iteration and this is different from the group sparsity technique [3], where all the basis poses at the node $N(j^*)$ are selected in an iteration.

3.4 Algorithm Summary

The complete PLCR-based algorithm for 3D pose reconstruction is summarized in Algorithm 1. As described above, we first construct a hierarchical pose tree, build a block-structural pose dictionary and search for an anchor node. We then iteratively pick the new basis poses that not only reduce the reprojection error, but also satisfy the pose locality constraint. In each iteration, we re-estimate the camera parameters based on the updated pose representations. Specifically, we use the PMP algorithm in [11] for camera parameter estimation. This iterative process is terminated when the 2D reprojection error is lower than a tolerance value, or a pre-set sparsity E has been reached. Using this iterative algorithm, a 3D human pose can be reconstructed using a linear combination of a small number of basis poses.

Algorithm 1. PLCR-based 3D pose reconstruction

```
Input: $\mathbf{p}$: 2D projection of a human pose
       $\mathcal{J}$: a set of 3D human poses
       $E$: pre-set sparsity
       $\tau$: tolerance value for the 2D reprojection error
Construct a hierarchical pose tree using method proposed in Sec. 3.2 and
  build the pose dictionary $\mathcal{B}$.
Estimate initial camera parameters $\langle \mathbf{M} = \mathbf{M}_1, \mathbf{T} = \mathbf{T}_1 \rangle$[11].
Search for an anchor node $A(\mathbf{p})$ using the method proposed in Sec. 3.3
  and initialize $\mathcal{S} = \emptyset$.
FOR $l$ from 1 to $E$
  $j^* = \arg\min_j \left(|\theta_j| + \lambda d_j\right)$
  $\mathcal{S} = \mathcal{S} \cup \mathbf{b}_{j^*}$
  Update the coefficients $\mathbf{\Omega}$ and camera parameters $\langle \mathbf{M}, \mathbf{T} \rangle$ according to
  the updated $\mathcal{S}$.
  Calculate the reprojection error $\mathbf{r} = \mathbf{R}(\mathcal{S}, \mathbf{\Omega}, \mathbf{M}, \mathbf{T})$.
  IF $\|\mathbf{r}\|_2 > \tau$
    BREAK
Calculate the 3D pose $\mathbf{J}$ by Eq.(2) and return.
Output: 3D pose $\mathbf{J}$ and camera parameters $\mathbf{M}$ and $\mathbf{T}$
```

4 Experiments

We use the CMU Motion Capture dataset for quantitative evaluations. This dataset contains more than three million 3D human poses collected on 144 subjects performing 30 different actions, and it has been widely used for evaluating 3D human pose reconstruction [11,17,20]. We also qualitatively evaluate the proposed method on real images collected from the Internet. As in previous works [11,17,20], we randomly selected a subset of $29,336$ 3D human poses from 5 different action categories: 'walking', 'jumping', 'running', 'boxing' and 'climbing' for quantitative performance evaluation. Details on the selected data is shown in Table 1. We can see that, for each action category except for 'climbing', the collected data are preformed by a number of different subjects. We use the 18-joint pose representation for our experiments [20].

Table 1. Detailed information on the $29,336$ 3D poses that are used for quantitative performance evaluation

	Walking	Jumping	Running	Boxing	Climbing
# of Pose	5752	5808	5352	8072	4352
# of Subjects	8	3	8	3	1

To study the generalizability of the proposed method, we use the "leave-one-subject-out" strategy for performance evaluation – the test data and the training data are from different subjects. Furthermore, we exclude the data from the 'climbing' action from training and only use them for testing to examine the generalizability of the proposed 3D pose reconstruction method across different action categories. As shown in Table 1, we in total conducted 23 rounds of experiments. Out of them, we have 22 rounds of experiments that use the pose data from the first four categories, excluding the data from 'climbing' category. In each of these 22 rounds of experiments, we leave out pose data from one subject in one action category for testing, while using the remaining data for training. We then conduct one additional round of experiment which uses pose data from 'climbing' category for testing and all the pose data from the other four categories for training.

When using a pose for testing, we first project it (i.e., the 18 joints) into 2D using a randomly generated camera parameters – both the camera location and orientation conform to a Gaussian distribution. We then reconstruct the 3D pose from this 2D projection (assuming camera parameters are unknown) and measure the reconstruction error at each joint as Euclidean distance between the ground-truth location of this joint and the reconstructed location of this joint. We then take the **maximum** reconstruction error over all the 18 joints and normalize it over the distance between the chest and waist as the *(normalized) reconstruction error* of this pose. Another performance metric used in this paper is the pose *reconstruction rate*, defined as the percentage of the tested poses that result in a low (normalized) reconstruction error, defined by a given threshold, which we use 0.3 for all the experiments.

The parameters that need to be tuned for our algorithms are: the branch factor K, the sparsity E, and the locality-adaptor related ones (λ, σ and d_M). In our experiments, we set $K = 2$, $E = 10$, $\sigma = 1$ and $d_M = 2$. We vary the parameter λ in the experiment to examine its effect to the performance. The choice of the parameters E and d_M will be further discussed at the end of this section.

4.1 Quantitative Results

Table 2 shows the reconstruction error (*rec_error*) and the reconstruction rates (*rec_rate*), averaged over all the subjects for each action category, with varying parameter λ. For comparison, we also report in Table 2 the performance of a state-of-the-art algorithm developed by Ramakrishna et al [11] on the same training and testing data. This comparison method [11] uses standard sparse representation and physiological regularity for 3D pose reconstruction. Note that, to examine the effectiveness of the proposed pose-locality constraints, we do not include any physiological regularity in the proposed method. We can see that, the proposed PLCR-based 3D pose reconstruction method outperforms the Ramakrishna's algorithm for all the action categories.

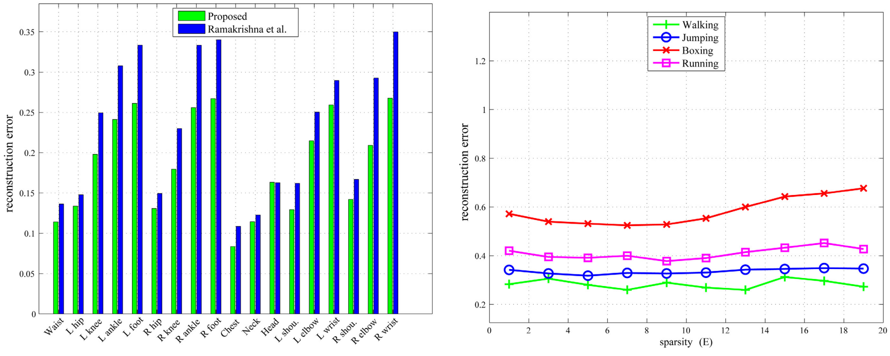
The 2D joint locations annotated in monocular images are often noisy. To examine the performance of 3D pose reconstruction under the 2D annotation noise, we add Gaussian white noise with different standard deviation *std* to the projected 2D joint locations, and then perform the 3D reconstruction, and the average performance over all the action categories is reported in Table 3, where

Table 2. The 3D reconstruction errors and reconstruction rates for different action categories

Action Category	Performance Metrics	Proposed method				Ramakrishna et al [11]
		$\lambda = 0$	$\lambda = 0.1$	$\lambda = 0.2$	$\lambda = 0.3$	
Walking	*rec_error*	0.360	0.300	**0.260**	0.272	0.446
	rec_rate	53.4%	71.2%	**73.9%**	70.4%	29.6%
Running	*rec_error*	0.417	0.390	**0.385**	0.432	0.453
	rec_rate	29.8%	35.1%	**38.2%**	34.0%	23.0%
Jumping	*rec_error*	0.343	0.322	**0.316**	0.321	0.374
	rec_rate	34.12%	39.5%	**41.6%**	40.2%	31.6%
Boxing	*rec_error*	0.579	**0.530**	0.535	0.534	0.584
	rec_rate	13.3%	**17.0%**	16.4%	16.8%	10.7%
Climbing	*rec_error*	0.560	0.528	**0.522**	0.526	0.533
	rec_rate	21.7%	27.9%	27.0%	**28.1%**	20.1%

Table 3. The average 3D reconstruction errors and reconstruction accuracy rates when different levels of noise are added to the 2D projections

std		0.0	0.1	0.2	0.3	0.4
Proposed method	*rec_error*	0.414	0.449	0.485	0.561	0.630
	rec_rate	32.6%	28.7%	24.4%	18.1%	13.1%
Ramakrishna et al [11]	*rec_error*	0.466	0.497	0.558	0.634	0.704
	rec_rate	23.9%	20.5%	13.8%	9.3%	4.8%



(a) (b)

Fig. 3. (a) Reconstruction errors at each of the 18 joints – a comparison between the proposed method and the Ramakrishna et al's algorithm. (b) Average reconstruction errors for four actions by varying the value of E.

the performance under $std = 0.0$ is the one without adding any noise. The values of the std are normalized by the width of the bounding box around the 2D projected pose. We can see that, the stronger the added noise, the larger the 3D reconstruction error and the lower the reconstruction rate. However, with the same level of noise, the proposed method still outperforms the comparison method.

Figure 3(a) shows the reconstruction error at each of the 18 joints, averaged over all rounds of experiments and all action categories. We can see that the proposed method achieves lower reconstruction error at all 18 joints than the Ramakrishna et al's algorithm. We can also see that, the reconstruction errors at feet, wrists, and ankles are larger than those at other joints, because of the larger movements of the hands and feet. Similar phenomena has been reported in [23].

4.2 Qualitative Evaluation

3D pose reconstruction results on four pose samples drawn from CMU Motion Capture dataset are shown in Figure 4. For each sample, we show (from left to right) the ground-truth 3D pose, its 2D projection, the 3D reconstruction using the proposed method, and the 3D reconstruction using the Ramakrishna's algorithm [11], respectively. For all these four cases, we can see that the proposed method generates more accurate and physiologically correct reconstructions, which are particularly clear at the locations indicated by the thick blue arrows on the results from the Ramakrishna et al's algorithm [11].

We also evaluate the proposed method on several images downloaded from the Internet, by manually annotating the 2D locations of the 18 joints on each image. The pose reconstruction results are shown in Figure 5. The reconstructed 3D human poses are shown from two different view angles. We can see that, the proposed method produces reasonable human pose reconstruction results on these real images.

4.3 The Selection of Parameters d_M and E

The parameter d_M defines a range around the anchor node that is allowed to be used for drawing the basis poses for 3D pose reconstruction. Intuitively, this parameter should be the distance between the real node (subspace) a pose belongs to and the anchor node searched by the proposed method. In our case, a pose belongs to one node (subspace) at each level of the tree and all these real nodes from all levels constitute a path from the root to a leaf. We can examine the shortest distance between the anchor node and this path, called *node-path distance,* to select an appropriate value for d_M. Table 4 shows the distribution of this node-path distance for all the collected pose data. We can see that most poses (86%) show such a distance to be no more than 2 (edges). Therefore, in our experiments, we set $d_M = 2$.

The parameter E indicates the sparsity, i.e., the number of basis poses used for 3D pose reconstruction. Figure 3(b) shows the average reconstruction error curves by varying the value of E, one curve for each action category. We can see

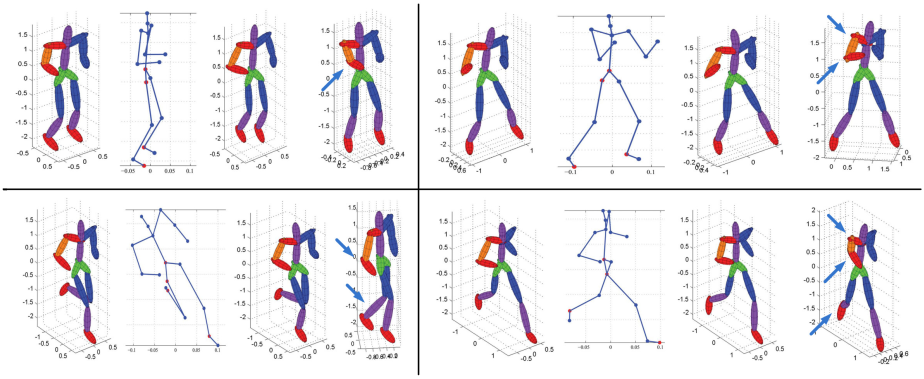


Fig. 4. Qualitative comparison between the proposed method and the Ramakrishna et al's algorithm [11] on the CMU Motion Capture dataset. For each pose, from left to right are the ground-truth 3D pose, its 2D projection, the 3D reconstruction using the proposed method, and the 3D reconstruction using the Ramakrishna et al's algorithm [11], respectively. Indicated by the blue arrows (on the 3D reconstruction produced by the Ramakrishna et al's algorithm) are the locations where the proposed method produces much better 3D reconstruction than the Ramakrishna et al's algorithm.

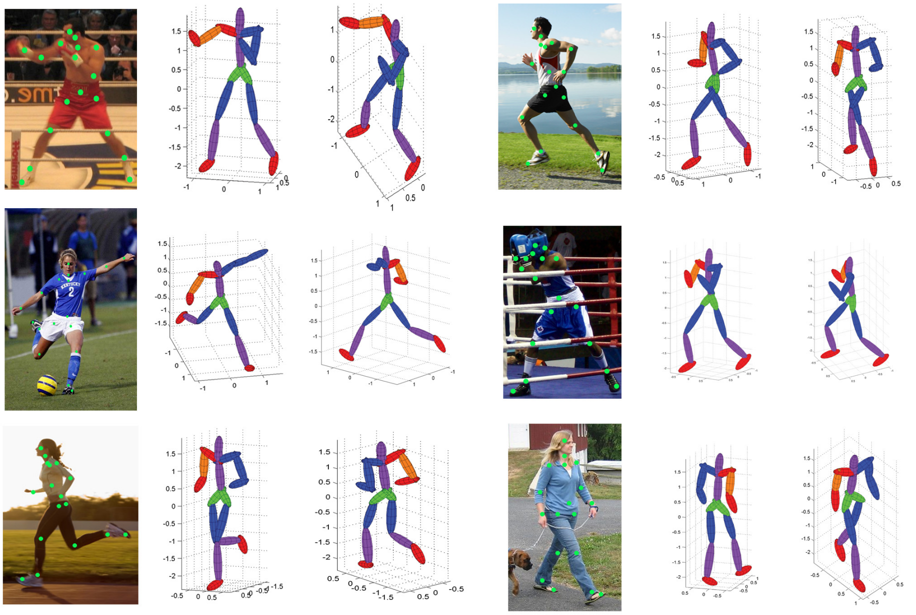


Fig. 5. 3D pose reconstruction from six images collected from Internet

Table 4. Distribution of the distance between the anchor node and the true path in the pose tree

Node-path distance	0	1	2	3	4	5	6	7
% Poses	60.3	14.0	11.8	7.8	3.4	1.0	1.2	0.0

that, varying the value of E from 1 to 19 does not lead to substantial performance difference for the 3D pose reconstruction. In our experiments, we simply select $E = 10$.

4.4 Distribution of Anchor-Node Depth

The depth of the searched anchor nodes in the pose tree reflects the specificity of the subspace used for 3D pose reconstruction – the deeper the anchor node, the more specific the corresponding subspace and the stronger the regularization for the ill-posed 3D reconstruction. Table 5 shows the distribution of anchor-node depth for all the tested pose data. We can see that for more than 80% of the poses, the searched anchor nodes have a depth larger than or equal to 2.

Table 5. Distribution of the anchor-node depth in the pose tree

Depth	0	1	2	3	4	5	6	7
% Poses	4.0	12.1	21.7	12.9	19.3	19.7	7.2	3.1

5 Conclusions

In this paper, we developed a new pose locality constrained representation (PLCR) of 3D human poses and used it to improve the 3D pose reconstruction from a single 2D image. We first used subspace clustering to construct a hierarchical pose tree, where each node represents a pose subspace and the nodes with larger depth in the tree represent more specific pose subspaces. To reconstruct a 3D pose, an anchor node is searched from the pose tree based on the input 2D projection. We then use the projected matching pursuit algorithm to search for a sparse set of basis poses from the anchor node (subspace) and its nearby nodes, which enforces the pose locality. We tested on $29,336$ pose data randomly selected from five action categories of the CMU Motion Capture dataset and found that the proposed PLCR-based algorithm outperforms a state-of-the-art algorithm using only sparse representation without considering pose locality. Reasonable qualitative results were also shown on real images collected from the Internet.

Acknowledgement. This work was supported in part by AFOSR FA9550-11-1-0327 and NSF IIS-1017199.

References

1. Di Franco, D.E., Cham, T.J., Rehg, J.M.: Reconstruction of 3D figure motion from 2D correspondences. In: CVPR (2001)
2. Elhamifar, E., Vidal, R.: Sparse subspace clustering. In: CVPR (2009)
3. Jenatton, R., Mairal, J., Obozinski, G., Bach, F.: Proximal methods for hierarchical sparse coding. The Journal of Machine Learning Research 12, 2297–2334 (2011)
4. Lee, H.J., Chen, Z.: Determination of 3D human body postures from a single view. Computer Vision, Graphics, and Image Processing 30(2), 148–168 (1985)
5. Li, R., Tian, T.P., Sclaroff, S., Yang, M.H.: 3D human motion tracking with a coordinated mixture of factor analyzers. IJCV 87(1-2), 170–190 (2010)
6. Liebowitz, D., Carlsson, S.: Uncalibrated motion capture exploiting articulated structure constraints. IJCV 51(3), 171–187 (2003)
7. Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., Ma, Y.: Robust recovery of subspace structures by low-rank representation. TPAMI 35(1), 171–184 (2013)
8. Liu, G., Lin, Z., Yu, Y.: Robust subspace segmentation by low-rank representation. In: ICML (2010)
9. Parameswaran, V., Chellappa, R.: View independent human body pose estimation from a single perspective image. In: CVPR (2004)
10. Raja, K., Laptev, I., Pérez, P., Oisel, L.: Joint pose estimation and action recognition in image graphs. In: ICIP (2011)
11. Ramakrishna, V., Kanade, T., Sheikh, Y.: Reconstructing 3D human pose from 2D image landmarks. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part IV. LNCS, vol. 7575, pp. 573–586. Springer, Heidelberg (2012)
12. Safonova, A., Hodgins, J.K., Pollard, N.S.: Synthesizing physically realistic human motion in low-dimensional, behavior-specific spaces. TOG 23(3), 514–521 (2004)
13. Salzmann, M., Urtasun, R.: Implicitly constrained gaussian process regression for monocular non-rigid pose estimation. In: NIPS (2010)
14. Shi, J., Malik, J.: Normalized cuts and image segmentation. TPAMI, 888–905 (2000)
15. Soltanolkotabi, M., Elhamifar, E., Candes, E.: Robust subspace clustering. arXiv preprint arXiv:1301.2603 (2013)
16. Taylor, C.J.: Reconstruction of articulated objects from point correspondences in a single uncalibrated image. In: CVPR (2000)
17. Valmadre, J., Lucey, S.: Deterministic 3D human pose estimation using rigid structure. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 467–480. Springer, Heidelberg (2010)
18. Vidal, R.: Subspace clustering. Signal Processing Magazine 28(2), 52–68 (2011)
19. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR, pp. 3360–3367 (2010)
20. Wei, X.K., Chai, J.: Modeling 3D human poses from uncalibrated monocular images. In: ICCV (2009)
21. Yang, A.Y., Iyengar, S., Sastry, S., Bajcsy, R., Kuryloski, P., Jafari, R.: Distributed segmentation and classification of human actions using a wearable motion sensor network. In: CVPRW (2008)
22. Yao, A., Gall, J., Van Gool, L.: Coupled action recognition and pose estimation from multiple views. IJCV (2012)
23. Yu, T.H., Kim, T.K., Cipolla, R.: Unconstrained monocular 3D human pose estimation by action detection and cross-modality regression forest. In: CVPR (2013)