Learning Basis Representation to Refine 3D Human Pose Estimations

Chunyu Wang,* **Haibo Qiu,*** **Alan L. Yuille,†** **Wenjun Zeng***
*Microsoft Research Asia, Beijing, China
†The Johns Hopkins University, Baltimore, MD 21218, USA

Abstract

Estimating 3D human poses from 2D joint positions is an ill-posed problem, and is further complicated by the fact that the estimated 2D joints usually have errors to which most of the 3D pose estimators are sensitive. In this work, we present an approach to refine inaccurate 3D pose estimations. The core idea of the approach is to learn a number of bases to obtain tight approximations of the low-dimensional pose manifold where a 3D pose is represented by a convex combination of the bases. The representation requires that globally the refined poses are close to the pose manifold thus avoiding generating illegitimate poses. Second, the designed bases also have the property to guarantee that the distances among the body joints of a pose are within reasonable ranges. Experiments on benchmark datasets show that our approach obtains more legitimate poses over the baselines. In particular, the limb lengths are closer to the ground truth.

Estimating 3D human poses from monocular images (Nie, Wei, and Zhu 2017; Tekin et al. 2017; Martinez et al. 2017; Sun et al. 2017; Pavlakos et al. 2017) is useful for many applications such as action recognition and human computer interaction. Compared to 2D poses, 3D poses are viewpoint invariant thus are more intrinsic representations of human motion. On the other hand, compared to raw RGB images, 3D poses have lower dimensions and suffer less from the risk of over-fitting to small datasets.

The task however is very challenging due to several reasons. First, multiple 3D poses may correspond to the same 2D pose and several candidates are even illegitimate. For example, the poses may have invalid anthropometric measurements, *e.g.* limb lengths or bending angles. Second, the input 2D poses are usually automatically estimated from images and may have large errors in some challenging cases (*e.g.* severe occlusions). The errors in 2D poses usually degrade the estimation accuracy of 3D poses severely. Figure 1 (b) shows a sample (inaccurate) 3D pose estimated by a strong baseline (Moreno-Noguer 2017).

In this work, we present an approach to refine inaccurate 3D poses by learning manifold priors from 3D pose datasets. There are two motivations behind our approach. First, we know that although 3D poses lie in a high dimensional ambient space (*e.g.* $3\times m$, m is the number of joints), legitimate

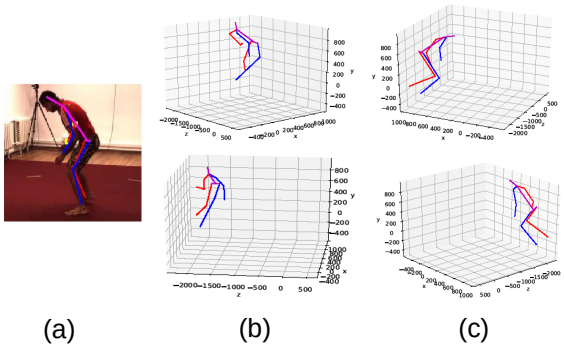

Figure 1: (a) shows the 2D pose estimated by (Cao et al. 2017). The joints highlighted by the yellow dots are those that were not successfully detected. (b) shows the estimated 3D pose (two views) by the baseline method (Moreno-Noguer 2017). (c) shows our refined 3D poses.

3D poses only lie in a small bounded region of that space. See the cyan region in Figure 2 for illustration. The core of our approach is to learn a bounded and low-dimensional representation for that region utilizing a dictionary of bases. See the red circles in Figure 2. A pose is represented by a **convex** combination of the **neighboring** bases. The neighborhood requirement enables a local and tight approximation of the pose manifold. For refinement purpose, a 3D pose will be projected to generate a valid 3D pose on the manifold. This basis representation captures the global configuration of all the body joints.

The second motivation is that, due to the skeletal structure of a pose, the distances among the body joints of a scale-normalized pose are bounded by the minimum and maximum distances. In particular, if a pair of joints form a rigid limb, then their distance is almost constant for all the normalized poses. This is a local prior for poses which we can use to locally refine illegitimate poses. We embed this information into bases and guarantee that locally a pose has legitimate distances among all joints.

We evaluate our 3D pose refinement approach on two benchmark datasets: H36M (Ionescu et al. 2014) and MPI-INF-3DHP (Mehta et al. 2017). We use two strong baselines (Moreno-Noguer 2017; Martinez et al. 2017) to obtain ini-

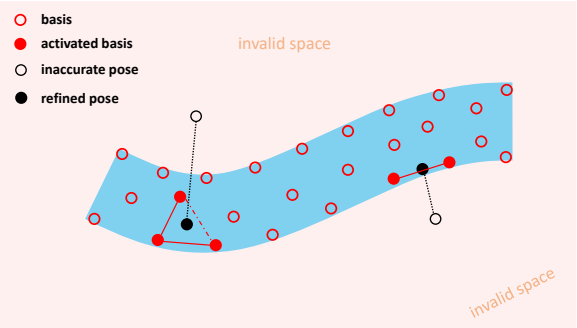

Figure 2: Illustration of the pose refinement approach. The pink area represents the whole ambient space of the 3D poses. However, only a small (cyan) region of the space contains legitimate poses. We learn a number of bases (*i.e.* the red circles) in the valid region. A 3D pose is represented by a convex combination of the neighboring bases which guarantees that the refined pose is close to the manifold.

tial 3D pose estimations which are fed to our algorithm for refinement. We observe that our approach consistently obtains more accurate poses after refinement on both datasets. The improvement is larger when the initially estimated poses have larger errors. In addition, the limb lengths of the refined poses are closer to the ground truth.

Related Work

We first review the existing work which treats pose refinement as a post-processing step similar to what we do in this work. Then we discuss the common techniques that are used directly in the process of estimating human poses which are targeted to obtain more legitimate poses.

Akhter and Black (Akhter and Black 2015) proposed a pose-conditioned joint angle prior for 3D poses. The prior is learned on a large scale motion capture dataset and can determine whether each segment of a pose is legitimate in terms of joint angles. If a segment is illegitimate, it is capable of refining the segment by truncating the joint angle to be a legitimate value. This is a local refinement approach which refines a pose in a segment by segment basis but does not consider the global configurations of all joints. Fieraru *et al.* (Fieraru et al. 2018) proposed a network which takes inputs of an image and an estimated pose, and outputs a refined 2D pose by exploring the dependency between the image and the pose space. However, the effectiveness of the approach is not validated for 3D poses.

We also review the techniques that are used directly in the process of estimating 3D poses. The first class (Pitelis, Russell, and Agapito 2013; Urtasun, Fleet, and Fua 2005; Elgammal and Lee 2009; Wang et al. 2016; Urtasun et al. 2005; Fan et al. 2014) proposes to learn a low-dimensional representation for 3D poses in order to suppress the generation of illegitimate poses that are off-the-manifold. Typical representation learning methods include Principal Component Analysis (PCA) (Ramakrishna, Kanade, and Sheikh 2012) Sparse Coding (SC) (Wang et al. 2014) and Sparse Subspace Clustering (SSC) (Elhamifar and Vidal 2009). Generally speaking, these methods all propose to represent the pose manifold by a set of unbounded hyper-planes. However, they do not take advantage of the fact that the pose manifold is actually bounded. As demonstrated in their experiments (Wang et al. 2014), the approach still admits illegitimate poses if no additional constraints are used. We will compare with these representations in the experiments section.

The second class (Taylor 2000; Barrón and Kakadiaris 2001; Ramakrishna, Kanade, and Sheikh 2012; Wang et al. 2014; Akhter and Black 2015) proposes to enforce limb length constraints on the 3D poses. The pioneer work (Taylor 2000; Barrón and Kakadiaris 2001) use the limb lengths to compute the relative depth between neighboring joints. Later work (Ramakrishna, Kanade, and Sheikh 2012; Wang et al. 2014) leverages these constraints in modeling. The optimization algorithm in (Wang et al. 2014) is complex and may not reach global minimum. The authors in (Ramakrishna, Kanade, and Sheikh 2012) solve the problem by using a relaxed constraint.

Our approach learns a piece-wise low-dimensional representation for the pose manifold. But different from PCA, SC and SSC, our representation is bounded which is more effective in terms of suppressing illegitimate poses. In addition, the limb length constraints are implicitly embedded in the representation thus effectively enhances the prior and simplifies the optimization process.

Notations

We represent a 3D pose y by m joint locations $y = [p_1^3, \cdots, p_m^3]$ where $p_i^3 \in \mathbb{R}^3$. We normalize a 3D pose by the length of its left-lower-arm and denote it as: $\hat{y} = [\hat{p}_1^3, \cdots, \hat{p}_m^3]$. We can recover the unnormalized refined pose based on the scale of the input pose.

We embed a normalized pose $\hat{y}$ into a distance matrix $d \in \mathbb{R}^{m \times m}$ by computing the distances between every pair of joints $d(i,j) = \|\hat{p}_i^3 - \hat{p}_j^3\|_2$. This distance representation is rotation and translation invariant which effectively reduces the pose space and facilitates the learning of the bases. In addition, this representation also simplifies the enforcement of the joint distance constraints as discussed in the next section. In the rest of the paper, we will directly work with distance matrices unless stated otherwise. In the end we can recover the 3D pose from the distance matrix by using Multidimensional Scaling (Biswas et al. 2006). The recovered poses will be aligned to the (probably inaccurate) input pose to recover the rotation, translation and scale by performing Procrustes (Gower and Dijksterhuis 2004).

The Basis Representation

We learn a set of K bases $\mathbb{B} = \{b_1, \cdots, b_K\}$ to represent a distance matrix d by their linear combinations $d = \sum_{k=1}^{K} b_k \alpha_k$. Recall that each entry $d(i,j)$ represents the distance between the joints i and j. Because of the articulated structure of human poses, the value of $d(i,j)$ is bounded by the minimum and the maximum values, *i.e.* $d(i,j) \in [\delta_{\min}^{i,j}, \delta_{\max}^{i,j}]$. We require every basis b to respect the distance ranges: $b(i,j) \in [\delta_{\min}^{i,j}, \delta_{\max}^{i,j}]$ (we will describe how to learn

such bases in subsequent sections) Then a convex combination of the bases will generate a distance matrix whose entries are guaranteed to be within the corresponding distance ranges. So we propose to represent a distance matrix d by convex combinations of the bases $\mathbb{B}$:

$$\alpha^* = \arg\min_{\alpha} \|d - \sum_{k=1}^{K} b_k \alpha_k\|^2, \quad \alpha_k \geq 0, \quad \sum_{k=1}^{K} \alpha_k = 1 \tag{1}$$

We also require the minimum and maximum distances among the joints are captured by at least one basis to ensure that all (valid) distance matrices can be accurately represented. Mathematically it means: $\max_k b_k(i,j) = \delta^{i,j}_{\text{max}}$ and $\min_k b_k(i,j) = \delta^{i,j}_{\text{min}}$ where $k \in \{1, \cdots, K\}$ is the index of the bases in $\mathbb{B}$. Geometrically, it is equivalent to saying that some bases should be positioned at the boundaries of the manifold. See the red circles on the manifold boundaries in Figure 2 for illustration.

Topology Preservation by Local Bases

Convex combinations of the bases will construct a big convex hull which still admits invalid poses especially when the shape of the manifold is not convex. To alleviate the problem, we propose to learn a tighter approximation of the pose manifold by a mixture of small convex hulls as opposed to a single big convex hull.

Inspired by (Wang et al. 2010), we propose to only activate the bases in the neighborhood of a pose to represent it. For example, in Figure 2, the two black circles are represented by the bases which are close to them, respectively. This strategy will effectively construct a set of small convex hulls, rather than a big convex hull, which provide tighter approximations of the (non-convex) manifold.

The neighborhood requirement is formulated as follows:

$$\arg\min_{\alpha} \|d - \mathbb{B}\alpha\|^2 + \lambda \|\alpha \odot s\|^2, \quad \alpha \succeq 0, \quad |\alpha|_1 = 1, \tag{2}$$

where $s = [s_1, \cdots, s_K]^T$ is a column vector of dimension K encoding the distances between the distance matrix d and the K bases: $s_k = \exp(\frac{\text{dist}(d, b_k)}{\sigma})$. Operator $\odot$ denotes element-wise multiplication. σ is a preset parameter that is used to adjust the weight decay speed. We can see that this formulation encourages to represent a pose by the neighboring bases. To enable a compact notation, we assume each basis b_k and distance matrix d (originally has shape of $m \times m$) have been reshaped to a column vector. So the shape of the basis dictionary $\mathbb{B}$ is $m^2 \times K$.

Basis Learning

Now we discuss how to learn the bases having the two properties discussed in the beginning of this section: (1) respect the distance ranges, and meanwhile (2) capture the minimum and maximum distances.

We denote the set of training distance matrices as $\mathbb{D} = \{d_1, \cdots, d_N\}$ which are computed from the normalized training poses. We assume the training set $\mathbb{D}$ is representative so as to have the following two properties: $\max_l d_l(i) = \delta^i_{\text{max}}$ and $\min_l d_l(i) = \delta^i_{\text{min}}$ where $l \in \{1, \cdots, N\}$. Recall that d_l is a column vector and $d_l(i)$ is the i_{th} dimension of d_l. It means the extremal values of each dimension are present in the dataset $\mathbb{D}$.

We first formulate the basis learning problem and then present two lemmas proving why the formulation has the two properties described above. We construct each basis by a convex combination of the training distance matrices. So learning the bases is equivalent to learning the coefficients of the combinations:

$$\begin{aligned} &\arg\min_{\beta,\alpha} \sum_{i=1}^{N} \|d_i - \sum_{k=1}^{K} (\sum_{l=1}^{N} d_l \cdot \beta_{l,k}) \cdot \alpha_{i,k}\|^2 \\ &s.t. \beta_{l,k} \geq 0, \alpha_{i,k} \geq 0, \sum_{l=1}^{N} \beta_{l,k} = 1, \sum_{k=1}^{K} \alpha_{i,k} = 1 \end{aligned} \tag{3}$$

We omit the neighborhood requirement here for simplicity. But adding the requirement is trivial. Now we provide a sketch proof why the bases learned by the above formulation are guaranteed to have the two properties.

Lemma 1. *A basis constructed by a convex combination of the training matrices $b = \sum_{l=1}^{N} d_l \beta_l, \beta_l \geq 0, \sum_{l=1}^{N} \beta_l = 1$ is guaranteed to respect the distance limits. In other words, for each entry j of the basis b we have $\delta^j_{min} \leq b(j) \leq \delta^j_{max}$*

Proof. Since $\min_l d_l(j) \leq \sum_{l=1}^{N} d_l(j)\beta_l \leq \max_l d_l(j)$ when $\beta_l \geq 0, \sum_{l=1}^{N} \beta_l = 1$, so we have $\delta^j_{\text{min}} = \min_l d_l(j) \leq b(j) \leq \max_l d_l(j) = \delta^j_{\text{max}}$. □

Lemma 2. *The bases $\{b_k = \sum_{l=1}^{N} d_l \cdot \beta_{l,k} | k = 1, \cdots, K\}$ learned by minimizing equation (3) are guaranteed to satisfy: $\max_k b_k(j) = \delta^j_{max}$ and $\min_k b_k(j) = \delta^j_{min}$.*

Proof. First, we can prove that when $\max_k b_k(j) = \delta^j_{\text{max}}$ and $\min_k b_k(j) = \delta^j_{\text{min}}$, then the value of the objective function in equation (3) decreases to 0 because all the possible values between the minimum and the maximum values can be accurately represented. Then we can also prove that when $\max_k b_k(j) \leq \delta^j_{\text{max}}$ or $\min_k b_k(j) \geq \delta^j_{\text{min}}$, then there are distance matrices whose j_{th} entries cannot be accurately reconstructed. So the value of the objective function is larger than 0 which is not optimal. So by minimizing equation (3), we will have $\max_k b_k(j) = \delta^j_{\text{max}}$ and $\min_k b_k(j) = \delta^j_{\text{min}}$ which achieves the optimal objective value of zero. □

Optimization

The formulation in equation (3) is non-convex. But it is convex when optimizing each set of variables αs and βs individually with the other set fixed. We use the alternating method proposed in (Chen, Mairal, and Harchaoui 2014) to solve the problem. Although there is no guarantee for global optimum, in our experiments, we observe that the extreme distances can be well captured by the bases. See Figure 3 for illustrations. We can see that, for most entries, the extreme values are approximately captured by the bases.

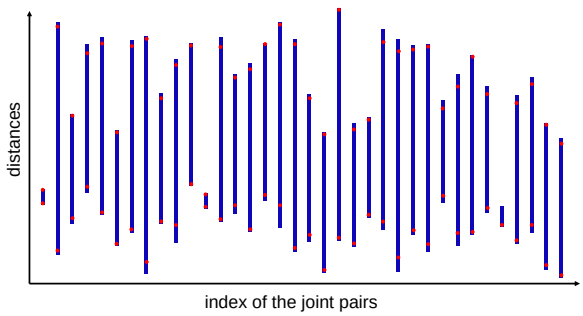

Figure 3: Visualization of the learned bases. The x-axis is the index of the matrix entries. The blue points represent the distances of the data in the training set. The red points are the minimum and maximum values of the bases. For most entries, the bases do capture the extremal values. Because the problem is not convex, it is possible that some entries have sub-optimal values. We only visualize several entries for simplicity. (Best view in color)

Refining Inaccurate Human Poses

After learning the bases, given an input pose y, we first normalize it and compute the corresponding distance matrix d. Then we refine the distance matrix by projecting it to the valid regions spanned by the bases:

$$\alpha^* = \arg\min_{\alpha} \|d - \mathbb{B}\alpha\|^2 + \lambda\|\alpha \odot s\|^2, \quad \alpha \succeq 0, |\alpha|_1 = 1, \tag{4}$$

The refined distance matrix is denoted as $d^* = \mathbb{B}\alpha^*$. Then we recover the 3D pose from d^* by multidimensional scaling (Biswas et al. 2006). The recovered 3D pose is up to a translation, rotation, scaling and reflexion transformation to the input unnormalized pose y. We perform Procrustes to align the recovered pose to y. It is worth noting that the ground truth is not used in the process making the subsequent comparisons completely fair.

Experiments for Property Verification

We first systematically evaluate the properties of the representation. Generally, if a 3D pose is valid (*e.g.* a ground truth 3D pose), the bases should reconstruct it with sufficient accuracy. In contrast, if a pose is invalid (*e.g.* having incorrect limb lengths), it should be projected to generate a valid pose which is close to the input pose.

Datasets

We first evaluate on the H36M dataset (Ionescu et al. 2014). There are 15 daily actions recorded by seven subjects. The dataset provides synchronized images, 2D poses and 3D poses. Following the most common evaluation protocol (Zhou et al. 2017; Pavlakos et al. 2017), we use five subjects (*i.e.* S1, S5, S6, S7, S8) for training and two subjects (S9, S11) for testing. We also evaluate on the MPI-INF-3DHP (Mehta et al. 2017) dataset which covers outdoor images. Following the previous work, we directly use the model trained on the H36M dataset to validate the generalization capability of the approach.

Evaluation Metrics We use two metrics in our experiments. The first one is mean per joint position error (MPJPE) between the groundtruth 3D pose $y = [p_1^3, \cdots, p_m^3]$ and the estimated 3D pose $\bar{y} = [\bar{p_1^3}, \cdots, \bar{p_m^3}]$ which is computed as $\frac{1}{m}\sum_{i=1}^{m} \|p_i^3 - \bar{p_i^3}\|_2$. Then we compute the average error over all poses of each action in the dataset.

The second metric is the Percentage of Correct Keypoints for 3D poses (PCK3D) (Mehta et al. 2017) which is a 3D extension of the PCK used in 2D pose estimation (Tompson et al. 2014). If the estimated joint position is within a neighborhood of the ground truth joint, then it is regarded as being correctly estimated. Then we compute the percentage of the correctly estimated joints. The neighborhood threshold is set to be $150mm$, corresponding to roughly half of the head size, as in the previous work. This metric is more expressive and robust than MPJPE, revealing individual joint mis-predictions more strongly.

Experimental Results

Reconstruct the Training Set of H36M We first learn independent basis dictionaries for each of the 15 actions in the training set by equation 3. Then we reconstruct each distance matrix d in the training set by equation (4). Then we recover the 3D pose from the reconstructed distance matrix using multidimensional scaling (Biswas et al. 2006). Finally, we align the recovered 3D pose to the original pose by Procrustes and compute the MPJPE.

The top section of Table 1 shows the results. When we learn a small number of 200 bases for each of the 15 actions, the reconstruction error is about 7.89mm which is already reasonably small. When we increase the number of bases, the reconstruction error consistently decreases. For example, the error is as small as 1.96mm when we learn $1,000$ bases. The results indicate that the valid 3D poses do lie in a small (probably low dimensional) space which can be accurately reconstructed by a small number of bases.

Reconstruct the Testing Set of H36M We also validate whether the bases can accurately reconstruct the poses in the testing set. This generalization capability is critical for the approach to be used in a real environment. See the bottom section of Table 1 for the results. First, the errors are larger than those on the training set. This is reasonable because the test sets are captured by different subjects who may perform the same action in very different styles. The error is about 28mm when 400 bases are used. Using more bases (*e.g.* 1000) cannot further decrease the error due to the large differences between the two sets.

To solve the problem, we learn a single big dictionary for all actions together. Compared to the action-wise dictionaries, it is more probable for the big dictionary to have better generalization power because it is learned on a larger set of poses. We can see from Table 1 that the reconstruction errors on the test set decrease significantly. It is surprising that the reconstruction error is as low as 25mm when we learn only 200 bases for all actions. This means many poses are

Table 1: H36M dataset: Reconstruction errors measured by MPJPE (mm) when we learn different numbers of bases for each of the 15 actions. The rows with the superscript * means we learn a single dictionary for all actions. The top and bottom sections of the table show the results on the training and testing sets, respectively.

K **(train)**	Direc.	Discu.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
200	6.90	7.81	5.75	8.51	7.32	9.72	5.77	7.64	13.44	8.88	6.34	9.37	6.81	7.74	6.29	7.89
400	4.37	4.78	3.37	4.55	4.32	4.94	3.02	4.01	7.79	4.80	3.39	5.15	4.31	4.29	3.73	4.45
1000	2.15	2.49	1.52	1.67	1.99	1.84	1.32	1.61	2.91	2.22	1.49	2.08	2.27	1.95	1.94	1.96
200∗	14.96	15.98	14.02	19.05	17.88	29.49	14.89	19.82	23.42	17.97	21.65	23.30	18.10	18.55	18.09	19.15
400∗	12.17	11.98	9.41	13.90	12.75	23.97	10.29	12.06	17.54	13.26	16.14	17.42	12.20	13.27	11.94	13.89
1000∗	6.60	7.14	5.66	8.70	7.15	12.56	6.33	6.82	10.21	7.44	7.91	8.40	6.56	7.88	7.48	7.79
K **(test)**	Direc.	Discu.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
200	27.75	33.58	25.10	27.68	34.25	24.58	18.55	35.58	30.54	36.74	44.72	26.29	17.56	32.82	21.67	29.16
400	24.18	30.92	24.16	25.19	32.84	24.28	20.34	35.04	30.59	36.09	41.83	24.38	14.94	28.33	20.39	27.57
1000	25.19	30.05	23.86	30.09	34.89	26.83	18.93	34.08	32.78	31.27	43.41	36.05	16.05	27.13	23.59	28.95
200∗	22.41	23.11	20.22	26.93	27.92	26.37	17.78	27.28	34.02	25.85	31.53	27.85	19.32	22.79	21.69	25.00
400∗	19.63	19.77	17.57	24.02	22.40	21.12	16.29	24.85	31.01	22.92	28.75	24.92	16.48	19.85	18.15	21.85
1000∗	16.89	16.49	14.81	19.15	19.20	19.41	13.90	22.55	28.88	19.84	25.08	20.04	13.77	16.32	15.56	18.79

actually shared between actions thus it is more reasonable to learn bases for all actions together. In particular, the error decreases to 18.79mm when we learn 1000 bases. This is accurate enough for many applications.

Reconstruct the Invalid Poses We also investigate the behaviors of the approach when the input 3D poses are invalid. We add large noises to the ground truth 3D poses and apply the approach (Akhter and Black 2015) to testify whether the bones of the corrupted pose is valid. If there is any invalid bone, we regard the corrupted pose as invalid. We obtain $30K$ invalid poses for our experiment.

We use the PCK3D metric because it is more expressive in terms of revealing individual joint errors. Table 2 shows the results. As expected, when only a small number of joints are corrupted, the PCK3Ds of the corrupted and the refined poses are very similar. The differences become larger when more joints are corrupted. For example, when 14 out of the 17 joints are corrupted, the average PCK3D of the corrupted poses is about 67%. However, after refinement, the PCK3D increases to 84% which demonstrates that the proposed approach effectively refines the inaccurate poses. Figure 4 shows some typical examples.

Experiments for Human Pose Estimation

We use (Martinez et al. 2017) and (Moreno-Noguer 2017) as our 3D pose estimation baselines which obtain the state-of-the-art performance. We obtain the 2D poses, which are the inputs to the 3D pose estimators, in two ways: (1) they are estimated from images by (Newell, Yang, and Deng 2016). Because the images in the two datasets are simple (*e.g.* simple background and clothing), the estimated 2D poses are rather accurate. However, this is different from real scenarios; (2) we learn an error distribution for the 2D poses in the more complex datasets MPII (Andriluka et al. 2014a) and COCO (Lin et al. 2014). Then we sample errors and add them to the ground truth 2D poses of the H36M dataset to simulate the situation when 2D poses are not perfect.

Implementation Details

For the first baseline (Martinez et al. 2017), which we denote as **Simple Baseline**, we use the code provided by the authors to train the 3D pose estimator. All the details are kept the same as in the original paper for fair comparison. The second baseline (Moreno-Noguer 2017) is denoted as **Matrix Baseline**. We implemented the approach using Pytorch and achieved comparable results as the paper. We learn 1000 bases for all the training poses.

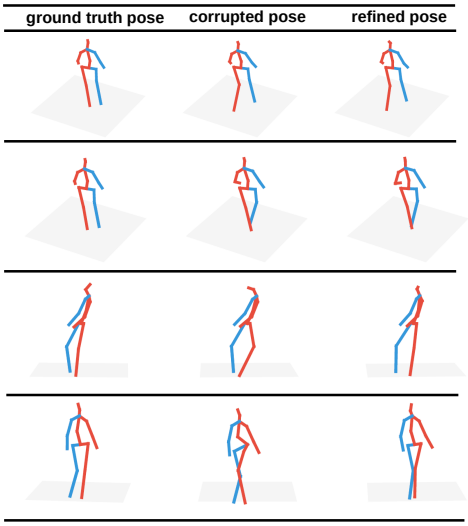

Figure 4: Pose refinement examples. When the corrupted pose is mostly valid, then the refined pose is very similar to it. See the first two rows. When the corrupted pose becomes invalid, then our approach will project it to generate a valid pose. In the third row, the bending angle of the red leg is invalid, but after refinement, it becomes valid.

Using Estimated 2D Poses

Table 3 shows the 3D pose estimation accuracy when the input 2D poses are automatically estimated. We observe

Table 2: H36M dataset (testing set): Reconstruction errors measured by PCK3D when different numbers of joints are corrupted in a 3D pose. The rows with the superscript* are the results of the refined poses.

Number of Invalid Joints	Direc.	Discu.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
1	97.10	97.84	97.62	97.30	97.53	97.17	96.83	97.50	97.21	97.40	97.21	97.36	97.11	97.16	97.78	97.34
1*	99.55	99.36	99.26	99.10	99.47	98.80	99.70	99.56	98.94	99.52	99.70	99.07	99.40	99.16	99.56	99.34
3	91.95	92.14	93.30	92.32	92.75	91.72	91.70	91.84	92.23	92.16	93.42	93.13	92.92	92.12	92.79	92.43
3*	96.83	96.92	98.73	96.41	97.74	97.28	97.59	98.38	98.04	97.40	98.11	96.76	96.81	98.21	98.34	97.57
5	88.51	87.82	89.13	86.93	88.03	87.80	87.18	85.66	89.06	88.08	87.34	86.38	87.84	88.55	85.79	87.61
5*	94.12	95.77	97.77	91.91	96.01	91.72	96.98	96.69	97.13	96.29	94.82	92.86	93.92	96.95	93.23	95.08
14	64.80	67.05	67.91	64.79	68.38	65.14	65.61	68.38	68.48	67.30	69.29	66.82	69.49	68.07	69.15	67.38
14*	82.17	85.39	90.02	80.80	86.66	81.05	87.33	90.15	90.50	87.02	89.13	80.37	79.36	86.66	76.80	84.89

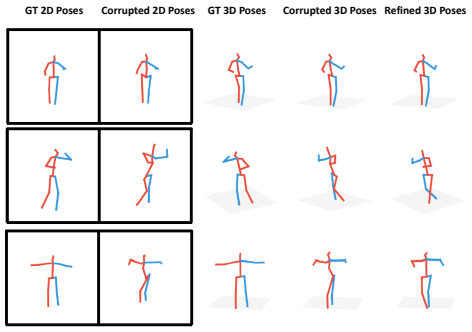

Figure 5: 3D pose estimation and refinement samples. When the added 2D noises are small, the estimated 3D poses are mostly accurate. In these cases, the refined 3D poses are also accurate. When the 2D noises are large, the estimated 3D poses begin to become illegitimate in terms of either limb lengths or bending angles. In these cases, the refined 3D poses are significantly better than the initial estimations.

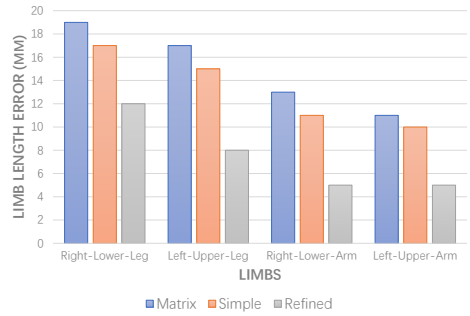

Figure 6: Limb length errors of the estimated 3D poses of the three approaches on the H36M dataset.

marginal but consistent improvement over both baselines. In particular, for the **Matrix** baseline whose initial accuracy is lower, our approach obtains larger improvement. We also observe that the limb lengths of the refined poses are also close to the ground truth. Figure 6 shows the limb length errors of the estimated/refined 3D poses.

Using Synthesized 2D Poses

We first apply the 2D pose estimator (Newell, Yang, and Deng 2016) to the images from the MPII and the COCO datasets. Then we compute an error vector for each estimation. We divide the error vectors into three groups according to the largest absolute value in each vector. The first group contains the error vectors whose largest absolute values are smaller than 10 pixels. The second group contains those which are between 10 and 15 pixels. The third group contains those which are between 15 and 20 pixels. Then we fit a Mixture-of-Gaussian model for each group, respectively. The number of mixtures is set to be five.

In the testing stage, we first sample error vectors from the distributions and then add them to the ground truth 2D poses in the H36M dataset. We feed the corrupted 2D poses into the baseline methods to obtain the 3D estimations. We report errors for each of the three groups, respectively. Table 4 shows the results. When the 2D poses have minimum level of errors, the baselines get reasonably good results, *e.g.* 94.25% for the **Simple** baseline. This result shows that the baselines are robust to small errors in 2D. However, when we consistently increase the errors, the baseline methods begin to generate severely degraded estimations. For example, for the third group, the PCK3D of the **Simple** baseline decreases to 71.73%. In this case, applying our refinement approach increases the accuracy to 81.03%. Figure 5 shows several estimation samples.

We also compare with other basis representations including PCA and sparse coding. The number of bases for each method is set by cross validation. Table 4 shows the results. First, PCA makes negligible difference for the three groups of experiments. In some cases, it even degrades the accuracy. This is reasonable because the poses of the 15 actions cannot accurately represented by the orthogonal bases learned by PCA. Actually, there is a trade-off. We can achieve more accurate representations if we use more bases. But it will also harm the power of refinement. The bases learned by sparse coding (Mairal et al. 2009) can refine the inaccurate poses to some extent. But our approach obtains larger improvement. We guess the main reason is because our representation achieves a tighter approximation of the manifold and generates 3D poses with reasonable limb lengths.

Generalization Capability

We first evaluate our approach on the wild images from the MPII dataset (Andriluka et al. 2014b) without re-training the

Table 3: H36M dataset: 3D pose estimation results of the two baselines and our refinement approach. The metric is PCK3D.

Methods	Direc.	Discu.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
Simple	98.96	99.12	98.24	98.72	99.20	96.54	96.50	96.63	98.33	98.10	98.42	98.62	97.33	97.61	98.31	98.03
Refine	100.00	98.53	98.05	97.30	100.00	98.90	98.18	99.10	100.00	99.15	100.00	97.03	97.65	99.14	98.30	98.75
Matrix	97.87	93.41	97.67	95.94	99.53	97.50	96.81	96.21	95.97	97.55	98.31	94.68	96.40	97.55	96.71	96.81
Refine	97.92	94.73	97.93	97.59	100.00	100.00	96.84	97.42	96.21	98.09	98.41	95.41	96.81	98.03	96.87	97.60

Table 4: H36M dataset: 3D Pose estimation results of the two baselines and our approach. The 2D poses are the corrupted ground truth by adding noises sampled from Gaussian distributions of different variances. The metric is PCK3D.

Noise Level	Direc.	Discu.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
Noise=10																
Simple	95.80	90.31	95.33	93.59	96.76	93.03	94.50	95.49	93.40	91.48	94.84	94.71	92.88	95.59	96.11	94.25
PCA	94.46	89.89	94.13	92.54	94.64	92.47	93.08	95.38	93.04	90.95	94.47	94.17	91.80	95.43	95.43	93.46
Sparse	94.77	90.39	95.09	92.90	96.51	92.86	94.68	94.45	93.49	91.06	93.01	95.75	91.62	95.81	96.14	93.97
Refine	94.96	90.66	95.85	94.33	95.59	93.90	93.36	95.49	93.11	93.31	94.84	92.55	92.41	96.32	95.93	94.17
Noise=15																
Simple	84.98	86.97	84.79	86.38	87.18	86.47	84.22	77.42	81.26	85.74	90.76	85.59	84.51	82.70	79.58	84.57
PCA	82.37	86.75	82.27	88.53	82.51	84.20	80.86	81.65	78.35	89.97	93.64	88.18	83.53	77.98	80.49	84.09
Sparse	86.00	89.28	85.93	86.73	88.29	86.99	87.08	78.36	82.74	89.59	91.27	86.84	89.67	86.10	83.69	86.57
Refine	88.39	88.87	88.03	90.30	89.08	90.00	87.37	82.92	83.88	90.44	91.29	92.94	86.86	85.64	83.22	87.94
Noise=20																
Simple	63.84	72.45	76.47	75.00	70.12	71.35	74.65	65.92	74.07	71.49	67.35	73.92	72.90	76.08	70.30	71.73
PCA	64.04	74.21	76.52	76.96	71.84	72.68	75.72	68.84	74.20	75.34	68.75	75.81	73.56	77.42	75.04	73.39
Sparse	71.27	79.58	77.74	76.11	70.77	78.99	76.45	68.61	85.42	75.36	70.91	76.06	75.73	76.80	82.96	76.18
Refine	77.68	82.77	81.93	83.38	80.65	79.51	83.98	75.61	79.74	82.90	80.88	84.51	82.46	80.20	79.34	81.03

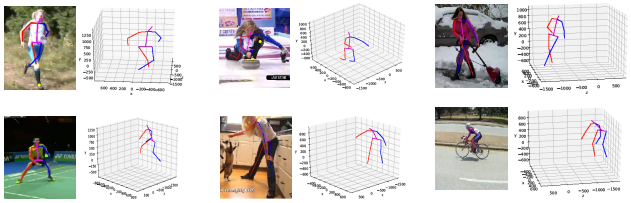

Figure 7: Estimated 3D poses on the MPII dataset. We do not train our model on this dataset. If a 2D joint was not detected, it is highlighted by a yellow marker. The results validate that our method has good generalization power.

baselines and bases. Since we do not have the ground truth 3D poses for this dataset, we show several estimation examples in Figure 7. The visualized results are obtained by refining the poses estimated by the **Simple** baseline. Note that some 2D joints were not estimated successfully as highlighted in the figure. We can see that the refined poses are mostly reasonable which suggests the bases generalize well to unseen poses in other datasets.

We also test on the MPI-INF-3DHP dataset. Table 5 shows the results. Our method outperforms the baselines consistently which demonstrates the effectiveness and the generalization power of the refinement approach.

Conclusion

We present a basis representation for refining illegitimate 3D human poses. The representation captures the global configurations of all joints and suppresses the generation of illegitimate 3D poses that are off-the-manifold. Locally, the representation guarantees that the refined 3D poses have reasonable limb lengths. The effect is significant when 3D pose estimations have large errors, *e.g.* when the input 2D poses are not very accurate, which is common in real deployment. The learned bases have reasonably good generalization capabilities as validated on other datasets.

Table 5: Comparison with the state-of-the-arts of the 3D pose estimation results on the MPI-INF-3DHP dataset measured by PCK3D.

	GS	NO GS	Outdoor	ALL PCK	AUC
Matrix	69.1	61.3	70.7	67.2	29.3
Refined	**73.4**	**65.2**	**74.6**	**72.4**	**34.6**
Simple	71.9	63.7	72.7	69.4	30.5
Refined	**74.2**	**67.8**	**76.2**	**74.5**	**36.6**

Acknowledgement: We would like to thank for the support from the grant ONR N00014-18-1-2119.

References

Akhter, I., and Black, M. J. 2015. Pose-conditioned joint angle limits for 3d human pose reconstruction. In *CVPR*, 1446–1455.

Andriluka, M.; Pishchulin, L.; Gehler, P.; and Schiele, B. 2014a. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, 3686–3693.

Andriluka, M.; Pishchulin, L.; Gehler, P.; and Schiele, B.

2014b. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, 3686–3693.

Barrón, C., and Kakadiaris, I. A. 2001. Estimating anthropometry and pose from a single uncalibrated image. *CVIU* 81(3):269–284.

Biswas, P.; Liang, T.-C.; Toh, K.-C.; Ye, Y.; and Wang, T.-C. 2006. Semidefinite programming approaches for sensor network localization with noisy distance measurements. *TASE* 3(4):360–371.

Cao, Z.; Simon, T.; Wei, S.-E.; and Sheikh, Y. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *CVPR*, 1302–1310.

Chen, Y.; Mairal, J.; and Harchaoui, Z. 2014. Fast and robust archetypal analysis for representation learning. In *CVPR*, 1478–1485.

Elgammal, A., and Lee, C.-S. 2009. Tracking people on a torus. *PAMI* 31(3):520–538.

Elhamifar, E., and Vidal, R. 2009. Sparse subspace clustering. In *CVPR*, 2790–2797.

Fan, X.; Zheng, K.; Zhou, Y.; and Wang, S. 2014. Pose locality constrained representation for 3d human pose reconstruction. In *ECCV*, 174–188. Springer.

Fieraru, M.; Khoreva, A.; Pishchulin, L.; and Schiele, B. 2018. Learning to refine human pose estimation. *arXiv preprint arXiv:1804.07909*.

Gower, J. C., and Dijksterhuis, G. B. 2004. Procrustes problems, volume 30 of oxford statistical science series.

Ionescu, C.; Papava, D.; Olaru, V.; and Sminchisescu, C. 2014. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI* 36(7):1325–1339.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755. Springer.

Mairal, J.; Bach, F.; Ponce, J.; and Sapiro, G. 2009. Online dictionary learning for sparse coding. In *ICML*, 689–696.

Martinez, J.; Hossain, R.; Romero, J.; and Little, J. J. 2017. A simple yet effective baseline for 3d human pose estimation. In *ICCV*, 2659–2668.

Mehta, D.; Rhodin, H.; Casas, D.; Fua, P.; Sotnychenko, O.; Xu, W.; and Theobalt, C. 2017. Monocular 3d human pose estimation in the wild using improved cnn supervision. In *3D Vision (3DV), 2017 International Conference on*, 506–516. IEEE.

Moreno-Noguer, F. 2017. 3d human pose estimation from a single image via distance matrix regression. In *CVPR*, 1561–1570.

Newell, A.; Yang, K.; and Deng, J. 2016. Stacked hourglass networks for human pose estimation. In *ECCV*, 483–499. Springer.

Nie, B. X.; Wei, P.; and Zhu, S.-C. 2017. Monocular 3d human pose estimation by predicting depth on joints. In *ICCV*, 3467–3475.

Pavlakos, G.; Zhou, X.; Derpanis, K. G.; and Daniilidis, K. 2017. Coarse-to-fine volumetric prediction for single-image 3d human pose. In *CVPR*, 1263–1272.

Pitelis, N.; Russell, C.; and Agapito, L. 2013. Learning a manifold as an atlas. In *CVPR*, 1642–1649.

Ramakrishna, V.; Kanade, T.; and Sheikh, Y. 2012. Reconstructing 3d human pose from 2d image landmarks. In *ECCV*. 573–586.

Sun, X.; Shang, J.; Liang, S.; and Wei, Y. 2017. Compositional human pose regression. In *ICCV*, 2621–2630.

Taylor, C. J. 2000. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. In *CVPR*, 677–684.

Tekin, B.; Marquez Neila, P.; Salzmann, M.; and Fua, P. 2017. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In *ICCV*, 3961–3970.

Tompson, J. J.; Jain, A.; LeCun, Y.; and Bregler, C. 2014. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 1799–1807.

Urtasun, R.; Fleet, D. J.; Hertzmann, A.; and Fua, P. 2005. Priors for people tracking from small training sets. In *ICCV*, 403–410.

Urtasun, R.; Fleet, D. J.; and Fua, P. 2005. Monocular 3d tracking of the golf swing. In *CVPR*, 932–938.

Wang, J.; Yang, J.; Yu, K.; Lv, F.; Huang, T.; and Gong, Y. 2010. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, 3360–3367. IEEE.

Wang, C.; Wang, Y.; Lin, Z.; Yuille, A. L.; and Gao, W. 2014. Robust estimation of 3d human poses from single images. In *CVPR*, 2361–2368.

Wang, C.; Flynn, J.; Wang, Y.; and Yuille, A. L. 2016. Recognizing actions in 3d using action-snippets and activated simplices. In *AAAI*, 3604–3610.

Zhou, X.; Huang, Q.; Sun, X.; Xue, X.; and Wei, Y. 2017. Weakly-supervised transfer for 3d human pose estimation in the wild. *arXiv:1704.02447*.