Learning to Refine 3D Human Pose Sequences

Jieru Mei[1], Xingyu Chen[2], Chunyu Wang[3], Alan Yuille[1], Xuguang Lan[2], and Wenjun Zeng[3]

[1]Johns Hopkins University
[2]Xian Jiaotong University
[3]Microsoft Research Asia

Abstract

We present a basis approach to refine noisy 3D human pose sequences by jointly projecting them onto a non-linear pose manifold, which is represented by a number of basis dictionaries with each covering a small manifold region. We learn the dictionaries by jointly minimizing the distance between the original poses and their projections on the dictionaries, along with the temporal jittering of the projected poses. During testing, given a sequence of noisy poses which are probably off the manifold, we project them to the manifold using the same strategy as in training for refinement. We apply our approach to the monocular 3D pose estimation and the long term motion prediction tasks. The experimental results on the benchmark dataset shows the estimated 3D poses are notably improved in both tasks. In particular, the smoothness constraint helps generate more robust refinement results even when some poses in the original sequence have large errors.

1. Introduction

A 3D human pose is usually represented by a vector of joint locations in 3D space due to its simplicity. However, the representation is not *compact* because it treats the whole ambient space $\mathcal{R}^{3P}$ without discrimination where P is the number of joints, and ignores the fact that the reasonable 3D human poses, which have valid bending angles and limb lengths, actually lie on a small low-dimensional space embedded in the ambient space.

The above problem may cause serious ambiguities for under-constrained tasks. For example, when we estimate 3D pose from a single image, multiple 3D poses including some *illegitimate* ones, may correspond to the same image after projection. Current works such as [13, 16] rely on deep neural networks to regress legitimate 3D poses. But they may get illegitimate estimations when the input 2D poses have errors as is often the case in practice. This is validated in [13]: when the input 2D poses are from ground truth, the 3D error is only 37.10mm; the error is doubled when the 2D poses are estimated by [17].

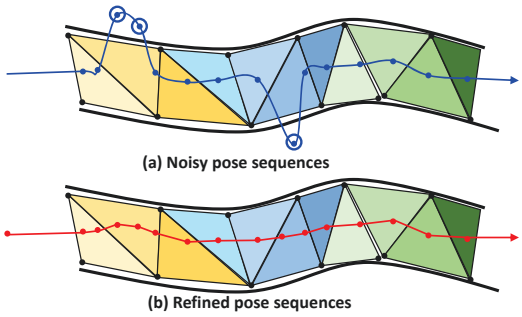

Figure 1. Our 3D pose sequence refinement approach. The mixture of small triangles (basis dictionaries) compactly approximates the pose manifold. The blue points in (a) are the poses in a sequence. The points highlighted by blue circles are off-the-manifold. See (b) for the refined pose sequence.

To resolve those challenges, we propose to learn a *compact* dictionary representation for pose manifold in which *only* reasonable 3D poses can be accurately reconstructed. The approach is motivated by some conjectures about poses. On one hand, human poses are believed to lie on a low-dimensional but non-linear space [20, 29, 10]. This suggests that we need to learn multiple linear dictionaries, instead of a single one, to accurately represent the curved manifold. Second, the manifold is bounded because the joint bending angles and limb lengths of poses are constrained to be within appropriate ranges. Third, the manifold is smooth in the sense that neighboring poses in a sequence should be close on the manifold.

To that end, we present an efficient approach to jointly learn multiple basis dictionaries from public human pose databases such as H36M [9]. Every dictionary consists of a small number of bases and represents poses by their convex combinations. Intuitively, convex combinations of the bases forms a *bounded* convex hull, covering a small region of the

whole pose manifold. For example, a small triangle in Figure 1 corresponds to the convex hull of one dictionary. The union of all convex hulls (dictionaries) serves as piece-wise linear approximations of the curved manifold. Meanwhile, we also encourage the neighboring poses in a sequence are represented by the same or adjacent dictionaries.

However, learning such dictionaries with the desired properties is a challenging problem because it involves two interrelated factors: (1) assign each pose to the corresponding dictionary; (2) optimize the bases in the dictionary using the assigned poses. We may solve the problem in theory by the EM algorithm but it will become extremely slow when scaling to millions of poses. Alternatively, we adopt the normalization trick proposed in [27, 15] which significantly simplifies the problem by simply normalizing all training poses and bases to a unit sphere. The merit of normalization is that it allows us to safely ignore the first assignment step and directly optimize the bases by the fast stochastic gradient descent algorithm [11].

During testing, given a sequence of noisy poses, we refine them by projecting them to the dictionaries. It jointly minimizes the distance between the original and the refined poses, along with the temporal jittering of the latter. It is worth noting that the two targets can be naturally achieved by our basis representation.

We apply our approach to refine the output 3D poses of different tasks. The first is monocular 3D human pose estimation from a 2D pose. We obtain initial 3D pose estimations by an existing state-of-the-art approach [13] which usually has large errors when the input 2D poses are inaccurate. We observe that by projecting the estimated 3D poses to the manifold using our approach, we obtain more legitimate poses with smaller errors. The second task is long term motion prediction given the first few frames. Most of methods fail to generate long sequences because errors will accumulate over time. We apply our approach to refine the predicted pose at each time step before it is used to generate the next pose. This small modification notably improves the robustness of long term motion prediction.

2. Related Work

We first review the existing work on pose refinement. Akhter *et al*. [1] propose a pose-conditioned joint angle prior for 3D poses which is learned on motion capture dataset. It refines illegitimate segments of a pose by truncating the joint angles to be valid values. It locally refines a pose in a segment-by-segment basis but does not consider the global configuration of all joints. Fieraru *et al*. [7] use a network to refine a 2D pose by exploring the dependency between the image and the pose space. But the approach is not validated for 3D poses.

We also review the techniques which are used in 3D pose estimation to suppress illegitimate poses. The first type of approaches [19, 24, 5, 27, 25, 28] learn lower dimensional representations for 3D poses in order to avoid generating illegitimate poses that are off-the-manifold. Typical dimension reduction methods include Principal Component Analysis (PCA) [20] Sparse Coding (SC) [29] and Sparse Subspace Clustering (SSC) [6]. They represent the pose manifold by unbounded hyper-planes which contradicts that the pose manifold is bounded. As demonstrated in their experiments [29], they still admit illegitimate poses suggesting that the representation is not compact. Our approach belongs to this type of works. But different from PCA, SC and SSC, our representation is bounded, which is more effective in terms of suppressing illegitimate poses.

The second type of approaches [22, 3, 20, 29, 18, 1] enforce limb length constraints on the 3D pose to suppress the estimations which have illegitimate limb lengths. For example, the pioneering works [22, 3] use the limb lengths to compute the relative depth between neighboring joints and manually resolve the sign ambiguity. Later work [20, 29, 18] leverages these constraints in modeling and encourages estimations that have correct limb lengths. The optimization algorithm in [29] is complex and may not reach the global minimum. The authors in [20] solve the problem by using a relaxed easy-to-optimize constraint. However, theses approaches are not adequate because the poses having correct limb lengths are not necessarily legitimate. For example, they may have incorrect joint angles.

3. Dictionary Learning

We first give a straightforward formulation based on our conjectures of the pose manifold. Then we present a reformulation which is easier to optimize. Finally, we show how to introduce the smoothness constraint into dictionary learning which generally outputs more robust bases.

3.1. Straightforward Formulation

Denote a set of N training poses as $\mathcal{Y} = \{\mathbf{y}_1, \cdots, \mathbf{y}_N\}$. The core of our approach is to learn a set of bases $\mathcal{D} = \{\mathbf{d}_1, \cdots, \mathbf{d}_K\}$ and the division of the big dictionary $\mathcal{D}$ into multiple small dictionaries $\{\mathcal{D}^{(1)}, \cdots, \mathcal{D}^{(M)}\}$ where $\mathcal{D}^{(z)} \subseteq \mathcal{D}$. A 3D pose $\mathbf{y}$ is represented by a single dictionary which has the smallest reconstruction error:

$$\begin{aligned} \boldsymbol{\alpha}^*, \mathbf{I}^* = \arg\min_{\boldsymbol{\alpha}, \mathbf{I}} \|\mathbf{y} - \sum_{z=1}^{M} \mathbf{I}^{(z)} \cdot \sum_{l=1}^{|\mathcal{D}^{(z)}|} \alpha_{z,l} \cdot \mathbf{d}_{z,l}\|^2 \\ s.t. \quad \mathbf{I}^{(z)} \in \{0,1\}, \quad \sum_{z=1}^{M} \mathbf{I}^{(z)} = 1 \\ \alpha_{z,l} \geq 0, \quad \sum_{z=1}^{M} \sum_{l=1}^{|D^{(z)}|} \mathbf{I}^{(z)} \alpha_{z,l} = 1 \end{aligned} \tag{1}$$

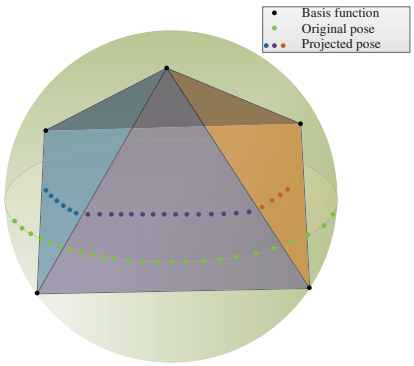

Figure 2. Illustration of the pose manifold and the proposed representation. The green and black points represent the normalized 3D poses and bases, respectively. We learn a convex hull (*i.e.* the pentahedron) inside the sphere and represent the poses by the surfaces of the convex hull. Each pose is represented by its projection on the closest surface.

where $\mathbf{I}^{(z)}$ is a binary indicator which is one only when $\mathcal{D}^{(z)}$ is the assigned dictionary and zero otherwise. The coefficient $\alpha_{z,l}$ corresponds to the l_{th} basis $\mathbf{d}_{z,l}$ in $\mathcal{D}^{(z)}$. The reconstructed pose $\hat{\mathbf{y}}$ is computed as: $\hat{\mathbf{y}} = \sum_{z=1}^{M} \mathbf{I}^{*(z)} \cdot \sum_{l=1}^{|\mathcal{D}^{(z)}|} \alpha^*_{z,l} \mathbf{d}_{z,l}$. The convex constraint on the coefficients ensures the representation is bounded.

The reconstruction error ϵ between the input and reconstructed poses is $\epsilon = \|\mathbf{y} - \hat{\mathbf{y}}\|_2$. We learn the dictionaries $\mathcal{D}$ to minimize the average reconstruction error on the training set. There are actually two sub-problems involved: (1) assign each training pose to one dictionary; (2) learn the dictionary based on the assigned poses. The problem is extremely difficult to optimize in its original form when the number of training data is large. But we show that it can be reformulated into a simpler form which has geometric interpretations and can be optimized efficiently.

3.2. Reformulation

We require the poses $\mathbf{y}$ and the bases $\mathbf{d}$ to have unit l_2-norm. Geometrically, the poses and the basis functions are normalized to lie on a unit hyper-sphere. We show this simple normalization step enables us to safely ignore the sub-problems described in the above section and focus on learning the big dictionary $\mathcal{D}$.

As shown in Figure 2, after normalization, both poses and bases lie on the unit sphere. The convex combination of the basis functions $\mathcal{D}$ forms a convex hull $C_{\mathcal{D}}$ inside the sphere. The *boundary* of the convex hull is defined by a set $\mathcal{F}_\partial$ of boundary surfaces. Each surface $\Delta_z \in \mathcal{F}_\partial$ is specified by a set of basis functions $\{\mathbf{d}_{z,l} : l = 1, ..., |\Delta_z|\}$.

A pose is represented by the closest surface of the convex hull. This is similar to formulation (1). The difference is that, for normalized poses and bases, directly minimizing the reconstruction error over the big dictionary $\mathcal{D}$ is equivalent to enumerating all the sub-dictionaries $\mathcal{D}^{(z)}$ and then selecting the closest one. This is because when we project a pose onto the convex hull, the minimum projection error is achieved when projected to the closest surface. This observation enables us to reformulate the problem as follows

$$\min_{\boldsymbol{\alpha}} \|\mathbf{y} - \mathcal{D}\boldsymbol{\alpha}\|^2, \text{ s.t. } \|\boldsymbol{\alpha}\|_1 = 1, \ \boldsymbol{\alpha} \succeq 0 \tag{2}$$

The surfaces of the convex hull naturally divide D into smaller dictionaries — each surface is a small dictionary. See how the polygons in Figure 2 naturally divide the bases into sub-groups. In other words, directly minimizing the reconstruction error on the training set gives the optimal basis dictionary and its divisions:

$$\min_{\mathcal{D},\boldsymbol{\alpha_i}} \sum_{i=1}^{N} ||\mathbf{y}_i - \mathcal{D}\boldsymbol{\alpha}_i||^2, \text{ s.t. } ||\boldsymbol{\alpha}_i||_1 = 1, \ \boldsymbol{\alpha}_i \succeq 0, \forall i \tag{3}$$

3.3. Smoothness Constraint

We encourage the neighboring poses in a sequence to be represented by similar bases. Specifically, as shown in Figure 2, most neighboring poses in a sequence are represented by the same surface, *e.g.*, the purple surface. Few poses which are near the surface boundaries may be represented by different surfaces but these surfaces usually share a number of bases. This strategy encourages to learn a smooth charting of the pose manifold. In addition, we experimentally find that using the smoothness prior during basis learning ends up with consistently better bases especially when the training poses have noise.

Incorporating this prior can be easily achieved by adding a smoothness term to the original objective

$$\begin{aligned} \mathcal{D}^* = \arg\min_{\mathcal{D},\boldsymbol{\alpha}} \sum_{i=1}^{N} \{\|\mathbf{y}_i - \mathcal{D}\boldsymbol{\alpha}_i\|^2 + \lambda \sum_{j \in \mathbb{N}_i} \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j\|^2\} \\ \boldsymbol{\alpha}_i \succeq 0, \quad \|\boldsymbol{\alpha}_i\|_1 = 1, \quad \forall i \end{aligned} \tag{4}$$

where $\mathbb{N}_i$ defines the neighbors of pose $\mathbf{y}_i$ and λ is the balancing parameter. We consider the simplest chain model where a pose depends on only the previous and next poses. More complex graph models can also be used to have stronger regularization but it is beyond the scope of this work.

The problem (4) is not convex with respect to $\mathcal{D}$ and $\boldsymbol{\alpha}$ jointly. But it is convex when we fix one and optimize the other. If we fix $\boldsymbol{\alpha}$, updating $\mathcal{D}$ can be solved by the projected gradient descent algorithm as in [11]. If we fix $\mathcal{D}$, updating $\boldsymbol{\alpha}$ can be solved by the active set algorithm as in [4]. Learning several hundreds of basis functions on the H36M dataset takes only several minutes.

Table 1. Reconstruction errors measured by MPJPE (mm) when we learn different numbers of bases on the H36M dataset w/o the smoothness constraint. 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.
100	24.71	23.64	23.35	28.01	29.70	33.02	36.37	26.94	33.68	44.62	29.57	32.10	32.91	25.30	25.89	29.99
200	18.25	17.63	17.94	21.17	23.19	24.59	28.81	21.46	26.07	34.78	22.24	24.31	25.53	19.04	19.71	22.98
400	13.62	13.55	13.35	16.03	17.68	18.69	21.86	16.70	19.92	26.54	16.91	17.74	19.95	14.10	14.33	17.40
600	11.31	11.31	11.05	13.56	14.56	15.21	17.59	14.17	17.14	22.27	14.23	14.52	17.04	11.67	11.83	14.50
1000	8.93	8.94	8.96	10.88	11.42	11.84	13.51	11.47	13.04	17.59	11.19	10.82	13.70	9.12	9.26	11.38
K **(test)**	Direc.	Discu.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
100	24.72	28.95	29.30	29.33	36.97	46.00	29.69	27.37	41.65	46.96	33.61	29.70	31.02	24.40	28.77	32.56
200	20.74	25.10	24.89	24.93	31.10	40.64	25.73	23.13	36.93	40.92	29.35	26.01	27.30	20.94	23.92	28.11
400	18.13	22.46	22.09	21.65	27.63	35.69	22.34	20.33	33.00	36.67	26.67	22.87	24.37	17.96	21.08	24.86
600	16.73	20.75	20.45	19.84	26.04	33.47	20.71	18.82	31.03	34.70	24.78	21.18	22.87	16.75	19.19	23.15
1000	15.36	18.78	18.41	17.90	23.79	31.03	18.69	17.49	29.40	32.67	23.27	19.42	21.08	15.08	17.16	21.30

3.4. Refine Pose Sequences

After learning the dictionary $\mathcal{D}$, we refine a sequence of poses $(\mathbf{y}_1, \cdots, \mathbf{y}_L)$ by solving

$$\boldsymbol{\alpha}_i^* = \arg\min_{\boldsymbol{\alpha}_i} \sum_{i=1}^{L} \{\|\mathbf{y}_i - \mathcal{D}\boldsymbol{\alpha}_i\|^2 + \lambda \sum_{j \in \mathbb{N}_i} \|\boldsymbol{\alpha}_i - \boldsymbol{\alpha}_j\|^2\}$$
$$\boldsymbol{\alpha}_i \succeq 0, \quad \|\boldsymbol{\alpha}_i\|_1 = 1, \quad \forall i \tag{5}$$

Each pose $\mathbf{y}_i$ is projected to a refined pose $\hat{\mathbf{y}}_i = \mathcal{D}\boldsymbol{\alpha}_i^*$. The first term in the equation forces the refined pose to be on the manifold and to be similar to the input pose. The second term forces the refined poses to have smooth transitions over time. The problem can be efficiently solved by the active set algorithm [4].

4. Experiments on Pose Reconstruction

We first evaluate whether a small number of bases can accurately reconstruct the poses in the dataset. We learn different numbers of bases ranging from 100 to $1,000$ on the training set and report the reconstruction errors on the training and testing sets, respectively.

4.1. Datasets and Metrics

We evaluate on the H36M dataset [9]. The poses of subjects $1, 5, 6, 7$ and 8 are used for training, while 9 and 11 are used for testing. We transform the 3D poses to the local camera coordinate system to remove the global rotations and translations. We normalize an input 3D pose so that it has unit l_2-norm. The normalized pose is reconstructed by the basis dictionaries. We align the reconstructed pose to the input to recover its scale.

The 3D pose estimation accuracy is measured by Mean Per Joint Position Error (MPJPE) which is computed between the ground-truth 3D pose $\mathbf{y} = [p_1^3, \cdots, p_m^3]$ and the reconstructed 3D pose $\hat{\mathbf{y}} = [\hat{p}_1^3, \cdots, \hat{p}_m^3]$. The error (for one pose) is computed as MPJPE $= \frac{1}{m}\sum_{i=1}^{m} \|p_i^3 - \hat{p}_i^3\|_2$. Then we compute the average error over all poses in the dataset.

4.2. Reconstruction w/o Smoothness Constraint

We independently reconstruct each pose in the *training* set without the smoothness constraint by solving equation (2). The top section of Table 1 shows the results. We can see that the reconstruction error is already as small as 17.40mm when we learn only 400 bases for the whole dataset with several million poses. It suggests that 3D poses do lie on a low dimensional space which can be accurately reconstructed by a small number of bases. In addition, increasing the number of bases consistently decreases the reconstruction error. For example, the error becomes 11.38mm when we learn $1,000$ bases.

We also compute the reconstruction error on the testing set (which have not been used for learning the dictionary). Achieving a small error on this subset is critical for the approach to have practical values. The bottom section of Table 1 shows the results. First, the errors on the testing set are generally larger than those on the training set. Second, further increasing the number of bases only marginally decreases the reconstruction error after after exceeding 400. This is reasonable because the pose distributions are different for the training and testing sets, and increasing the number of bases does not help decrease the representation error on the poses which have not been seen during basis learning. But in practice, the error of 21.30mm is sufficiently small for many tasks such as pose estimation.

4.3. Reconstruction with Smoothness Constraint

We also report the reconstruction error when we use the temporal constraint (*i.e.* equation (4) and (5)) to reconstruct pose sequences. Table 2 shows the results. Note that since we are reconstructing the ground truth 3D poses in this experiment, the reconstruction errors are slightly larger than the errors when we reconstruct them without the smoothness constraint. But the increase of error is very subtle which means that adding the temporal constraint will not affect its representation capability. This actually suggests that the pose sequences are indeed smooth and our learned bases support smooth reconstruction of them.

As will be demonstrated in the subsequent experiments,

Table 2. Reconstruction errors measured by MPJPE (mm) when we learn different numbers of bases on the H36M dataset with the smoothness constraint. 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.
100	25.26	24.28	23.76	28.84	30.03	33.22	36.70	27.20	33.74	44.67	29.90	32.39	33.69	26.46	26.79	30.46
200	18.96	18.49	18.48	22.25	23.67	24.88	29.28	21.83	26.18	34.86	22.68	24.71	26.50	20.52	20.90	23.61
400	14.53	14.62	14.07	17.40	18.29	19.12	22.48	17.26	20.08	26.66	17.48	18.27	21.22	15.96	15.80	18.22
600	12.39	12.56	11.90	15.11	15.30	15.74	18.32	14.85	17.36	22.43	14.90	15.16	18.49	13.83	13.55	15.46
1000	10.18	10.39	9.97	12.66	12.31	12.51	14.39	12.32	13.32	17.82	11.97	11.59	15.42	11.59	11.32	12.52
K **(test)**	Direc.	Discu.	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
100	25.64	29.40	29.64	30.05	37.20	46.21	30.19	27.74	41.69	46.96	33.85	30.09	31.86	25.92	29.85	33.08
200	21.86	25.68	25.39	25.87	31.44	40.97	26.38	23.68	36.99	40.94	29.68	26.51	28.32	22.80	25.19	28.78
400	19.43	23.21	22.74	22.84	28.04	36.11	23.15	21.02	33.11	36.71	27.10	23.51	25.65	20.32	22.71	25.71
600	18.14	21.63	21.23	21.15	26.51	33.96	21.60	19.64	31.16	34.76	25.27	21.92	24.29	19.33	20.75	24.09
1000	16.93	19.78	19.28	19.36	24.33	31.62	19.74	18.48	29.57	32.75	23.84	20.25	22.67	17.99	19.04	22.37

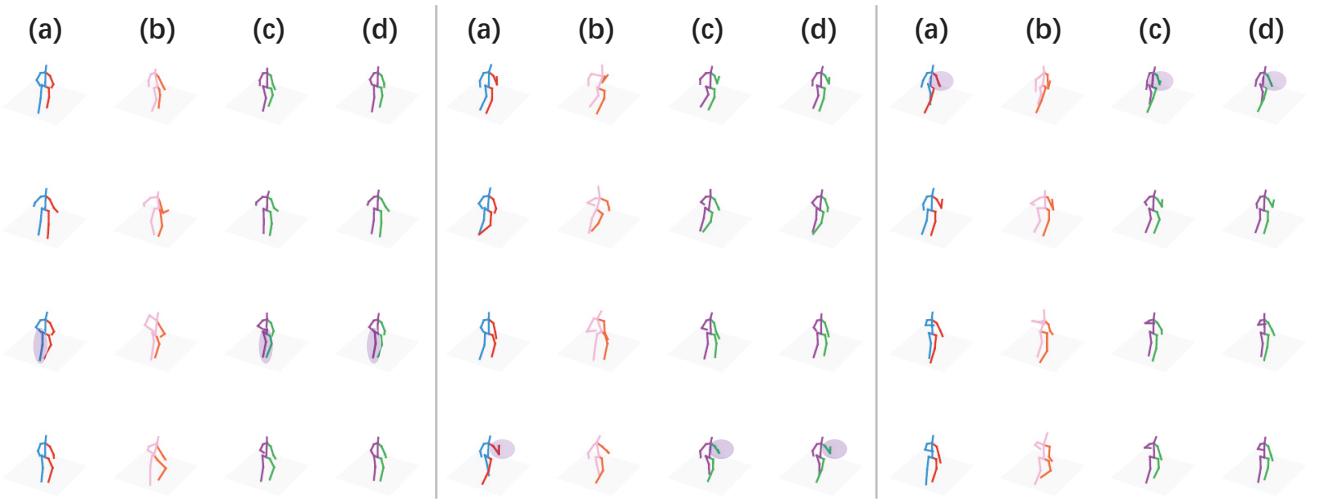

Figure 3. Sample pose refinement results from H36M testing set. The figures of (a), (b), (c) and (d) denote the ground truth poses, corrupted poses, refined poses w/o smoothness constraint and refined poses with smoothness constraint, respectively. First, we can see that when the ground truth poses are severely corrupted by noise, both of our approaches (c) and (d) achieve reasonably good results. Second, the poses refined with smoothness constraint (d) are closer to the ground truth compared to (c).

when the 3D poses are not from ground truth and have errors, for example, when they are estimated from images, using the smoothness constraint will significantly decrease the reconstruction error by regularizing the outlier poses which have large errors using the neighboring poses.

5. Experiments on Pose Refinement

We design two experiments to evaluate our approach when it is used to refine 3D poses. In the first experiment, we create a synthetic dataset by adding noise to the ground truth 3D poses, and then refine them using our approach. We evaluate whether our approach improves the quality of the poses. In the second experiment, we directly work on the 3D poses estimated by [13].

5.1. Implementation Details

We learn the bases from the ground truth poses of the training set in H36M. The number of bases is set to be 400 by cross validation. Changing this number does not significantly impact the refinement performance as long as it is sufficiently large. The balancing parameter λ is set to be 100 by cross validation.

5.2. Metrics

We use two evaluation metrics in this experiment. The first is the MPJPE described in the previous section. The second is the Percentage of Correct Keypoints for 3D poses (PCK3D) [14] which is a 3D extension of the PCK used in 2D pose estimation [23]. If the estimated joint location is within a neighborhood of the ground truth location, it is regarded as being correctly estimated. We compute the percentage of the correctly estimated joints. The neighborhood threshold is set to be 150mm as in the previous work. This metric is more expressive than MPJPE, revealing individual joint mispredictions more strongly.

5.3. Refine Synthetic 3D Poses

We first create a synthetic dataset consisting of several sequences of *illegitimate* 3D poses. This is achieved by

Table 3. The metric is **MPJPE(mm)** on the detected 2D poses. The top section shows the results when the 2D pose estimator is finetuned on the H36M dataset. The bottom section shows the results when it is not finetuned. **Simple** means the results are not refined. **DAE** means denoising auto-encoder. **w/o T** means our approach without temporal constraint. **with T** means our approach with temporal constraint.

finetuned	Direct.	Discuss	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
Simple	43.3	46.4	48.2	49.5	53.3	54.6	44.3	43.8	56.9	62.8	52.3	47.5	52.7	46.5	50.2	50.2
PCA	45.0	47.4	50.4	50.6	56.3	57.4	45.5	44.4	58.9	65.8	53.8	49.4	54.1	46.9	51.2	51.8
DAE	43.5	46.5	49.7	49.8	53.9	57.0	46.1	43.8	58.4	63.4	52.5	48.6	52.7	46.0	50.2	50.8
w/o T	43.0	46.6	49.3	49.9	54.0	59.2	45.7	44.0	60.0	64.7	52.4	48.4	52.3	43.7	48.9	50.8
with T	40.7	44.5	47.3	47.6	52.0	57.6	43.9	41.8	58.3	62.7	50.1	46.3	50.3	41.8	46.9	48.8
not finetuned	Direct.	Discuss	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
Simple	58.4	70.8	62.0	68.7	82.2	72.8	63.0	80.0	81.0	92.9	74.7	69.1	81.9	77.8	79.1	74.3
PCA	58.7	67.5	61.4	67.3	81.0	71.6	62.4	76.1	79.7	91.0	72.5	67.9	75.3	73.3	75.2	72.1
DAE	54.7	64.3	59.1	64.8	78.5	69.5	60.8	74.7	77.4	88.6	71.0	66.0	73.5	72.3	73.8	69.9
w/o T	54.3	63.1	57.3	63.3	76.0	69.7	59.9	72.5	77.8	87.8	68.5	64.4	70.9	67.3	69.3	68.1
with T	52.4	61.2	55.5	61.0	73.6	68.0	58.3	71.1	75.7	85.4	66.3	62.7	69.0	65.4	67.1	66.2

adding Gaussian noise to the ground truth 3D poses in the H36M dataset. For each pose corrupted by noise, we apply the approach proposed in [1] to test whether its bones are valid in terms of joint angles. The corrupted pose will be added to our synthetic dataset if any of its bones is invalid. The synthetic dataset consists of 10k illegitimate poses.

5.3.1 Experimental Results

We test whether the bones of the poses become valid after refinement by [1]. The number of valid bones in a pose is used as a metric to reflect how "legitimate" it is. Figure 4 shows the results. When no refinement is applied to the corrupted poses, only 21.5% poses have more than 13 valid bones. The total number of bones in a pose is 16. But after refining the corrupted poses using our approach, most poses become legitimate and have much more valid bones.

Figure 3 shows some examples of the refined poses. First, the corrupted 3D poses become severely degraded in terms of both limb lengths and joint angles. Second, after being refined by our approach w/o the smoothness constraint, almost all poses become legitimate. Third, adding the smoothness constraint further improves the results. In particular, the refined poses are closer to the ground truth poses compared to our approach w/o smoothness constraint. See the regions highlighted by purple ellipses.

5.4. Refine Estimated 3D Poses

We refine the estimated 3D poses by [13]. The 3D pose estimator is trained on the ground truth 2D and 3D pose pairs. We compare our approach to two baselines. The first is denoising auto-encoder [26] which projects poses to a low-dimensional latent space using a neural network. The second is Principal Component Analysis (PCA) which approximates the pose manifold by a hyperplane.

5.4.1 Estimating 2D Poses by Pose Detector

We estimate 2D poses using the stacked hourglass model [17] which was first trained on the MPII dataset [2] and then finetuned on the H36M dataset. We also conduct experiments when it is not finetuned on H36M. The former model achieves better 2D poses.

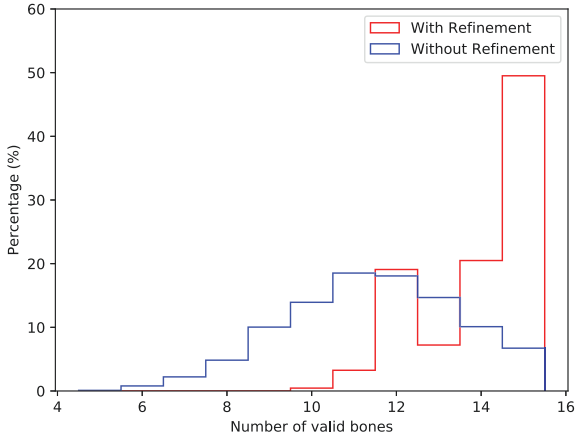

Figure 4. Histogram of the valid bones in the refined poses. The x-axis denotes the number of valid bones. The y-axis denotes the percentage of the poses having such a number of valid bones.

The top section of Table 3 shows the results when the 2D pose estimator is finetuned on the H36M dataset. The average error of the initially estimated 3D poses by [13] (denoted as **Simple**) is about 50.2mm. Applying PCA and denoising auto-encoder on the estimated poses doesn't improve the poses. This is because the input 2D poses are mostly accurate and the estimated 3D poses are mostly legitimate. However, our approach with the smoothness constraint (**with T**) decreases the error by about 1.4mm.

The gain is much larger when the 2D poses are estimated by the model which is not finetuned on the H36M dataset. The bottom section of Table 3 shows the 3D pose estimation results. We can see that the estimation error of the baseline method increases significantly to 74.3mm. Applying PCA and DAE both decreases the error. But our approaches with and without the smoothness constraint outperforms PCA and DAE. In particular, using the smoothness constraint provides larger improvement.

Table 4. The metric is **MPJPE(mm)** on synthetic 2D poses. The 2D poses are the corrupted ground truth by adding Gaussian noise of different variances δ.

Noise=5	Direct.	Discuss	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
Simple	41.1	46.5	46.3	49.1	49.0	54.8	44.9	44.3	52.7	60.6	48.6	47.1	51.0	51.3	52.5	49.3
PCA	43.5	48.0	49.4	50.5	52.6	58.0	46.4	44.8	55.8	64.0	50.9	49.7	52.8	51.9	53.8	51.5
DAE	41.3	46.0	47.1	48.5	48.9	56.3	46.1	43.2	53.0	59.1	48.4	47.7	50.2	49.4	51.2	49.1
w/o T	40.8	46.4	47.0	49.0	50.5	59.4	46.1	43.5	56.1	61.7	49.8	48.4	50.3	48.8	51.1	49.9
with T	34.0	39.6	39.7	41.5	43.1	52.5	39.9	36.8	49.9	53.8	42.9	42.0	43.3	40.2	42.4	42.8
Noise=10	Direct.	Discuss	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
Simple	62.6	69.8	71.0	75.4	73.7	81.2	63.7	68.6	77.3	93.0	72.4	68.5	77.3	84.4	83.7	74.9
PCA	62.2	68.4	70.2	74.2	73.0	80.1	62.7	65.0	76.4	91.2	70.9	67.9	75.4	82.2	82.0	73.5
DAE	57.3	63.1	66.4	68.7	67.3	75.9	59.7	61.3	71.2	84.2	66.0	63.8	70.2	75.9	75.4	68.4
w/o T	57.1	62.7	65.4	68.5	67.3	77.0	59.8	59.4	72.4	83.7	66.1	63.7	68.1	74.0	73.9	67.9
with T	41.6	46.9	49.4	52.2	51.1	61.8	45.2	43.6	57.8	66.7	50.2	49.4	52.2	56.3	55.9	52.0
Noise=15	Direct.	Discuss	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
Simple	84.6	94.3	96.2	101.5	99.9	109.4	83.7	95.2	103.6	125.1	97.6	90.6	104.8	115.7	113.9	101.1
PCA	82.3	90.7	92.4	98.5	95.8	104.9	80.9	88.6	99.5	119.8	93.3	87.8	100.1	111.9	110.1	97.1
DAE	74.4	82.1	86.3	89.1	87.6	97.1	74.7	82.1	92.0	110.1	85.6	81.2	91.6	101.3	99.2	89.0
w/o T	73.0	79.5	83.0	87.1	84.4	95.2	73.9	76.8	90.3	105.9	82.8	79.1	86.1	96.2	94.6	85.9
with T	51.5	57.2	60.0	64.2	61.4	74.4	52.7	54.9	69.2	83.1	60.1	58.8	63.9	71.4	69.3	63.5

Table 5. The metric is **PCK3D** on synthetic 2D poses. The 2D poses are the corrupted ground truth by adding Gaussian noise of different variances δ.

Noise=5	Direct.	Discuss	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
Simple	98.85	97.98	98.55	97.57	98.17	96.76	98.68	98.70	96.71	94.84	97.77	97.80	97.22	96.79	96.57	97.53
PCA	98.78	97.90	98.46	97.53	97.76	96.43	98.61	98.66	96.52	93.96	97.63	97.67	96.98	96.70	96.45	97.34
DAE	98.91	98.16	98.68	97.73	98.36	96.55	98.74	98.79	96.98	95.34	97.86	97.83	97.25	96.88	96.69	97.65
w/o T	98.87	98.10	98.68	97.54	98.03	95.57	98.56	98.70	96.39	94.64	97.60	97.56	97.27	97.02	96.78	97.42
with T	99.35	98.75	99.42	98.54	98.91	96.94	99.42	99.24	97.41	96.53	98.60	98.44	98.21	98.07	98.12	98.40
Noise=10	Direct.	Discuss	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
Simple	95.14	92.96	92.86	91.41	92.13	89.61	95.47	93.65	90.49	84.56	92.33	93.27	90.77	87.92	88.15	91.38
PCA	95.12	93.09	93.00	91.41	91.96	89.57	95.46	94.19	90.42	84.60	92.36	93.14	90.91	88.00	88.09	91.42
DAE	96.15	94.61	94.25	93.01	94.05	91.20	96.20	95.18	92.44	87.56	93.93	94.29	92.52	90.09	90.40	93.06
w/o T	96.14	94.75	94.53	92.94	93.94	90.38	96.02	95.64	91.80	87.42	93.68	94.13	93.05	90.53	90.63	93.04
with T	98.51	97.94	97.98	96.65	97.78	95.04	99.04	98.49	96.19	93.20	97.66	97.44	96.68	94.68	94.95	96.82
Noise=15	Direct.	Discuss	Eat	Greet	Phone	Photo	Pose	Purch.	Sit	SitD	Smoke	Wait	WalkD	Walk	WalkT	Avg.
Simple	88.48	84.08	83.30	81.50	81.59	77.99	89.03	83.96	80.40	71.37	82.72	85.39	79.86	74.79	75.65	81.34
PCA	88.77	85.06	84.42	82.21	82.76	79.43	89.37	85.87	81.40	73.33	83.84	85.93	81.33	76.05	76.70	82.43
DAE	91.48	88.48	87.18	86.17	86.65	83.16	91.81	88.60	84.97	77.68	87.25	88.61	85.05	80.88	81.86	85.99
w/o T	91.88	89.50	88.39	86.84	87.72	83.54	91.89	90.39	85.11	78.77	87.90	89.18	87.06	82.78	83.41	86.96
with T	97.29	96.12	96.22	94.17	95.87	91.14	97.83	96.51	93.21	87.73	95.59	95.64	94.24	90.95	91.33	94.26

5.4.2 Adding Noise to Ground truth 2D Poses

Since the images in the H36M dataset are relatively simple, achieving accurate 2D pose estimations on this dataset does not mean we can achieve similar performance in practical scenarios. So in this section, we systematically evaluate the 3D pose estimator and our approach when the 2D poses are corrupted by different levels of noise.

We estimate the 2D pose scale s by finding the maximal length of each pose along x, y-axis. Then we sample zero-mean Gaussian noise with standard variance $\sigma = \delta\% \times s$, where δ are set to be 5, 10 and 15, respectively. Table 4, 5 shows the 3D estimation errors when the noise are added to the ground truth 2D poses. First, when the smallest amount of noise ($\delta = 5$) are injected into 2D poses, most 3D estimations are rather accurate. For example, the error of the baseline **Simple** is 49.3mm. Directly projecting the estimations to the learned manifold doesn't offer improvement because most 3D estimations are already legitimate. However, we see a big improvement by using the temporal smoothness constraint. The error decreases from 49.3mm to 42.8mm.

Second, when we add large noise (δ=10) to 2D poses, the 3D estimation error of the baseline approach increases significantly to 74.9mm. In this case, projecting the estimations to the manifold significantly decreases the error to 67.9mm. This means many 3D estimations are illegitimate when the 2D poses have large errors.

Third, when we increase the noise from 5 to 15, the error of the baseline increases significantly from 49.3mm to 101.1mm which shows that most approaches heavily depend on the accuracy of the 2D poses. However, the error for our approach increases from 42.8mm to 63.5mm which is much smaller than the baseline.

6. Experiments on Motion Prediction

In this section, we evaluate our approach in the context of human motion prediction. The task aims at generating

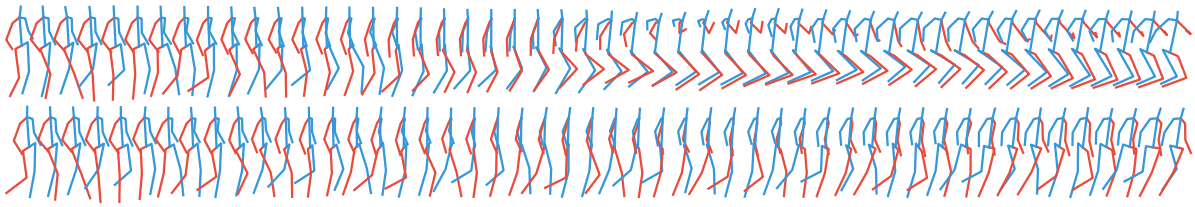

Figure 5. Typical motion prediction results of the baseline (top) and our method (bottom).

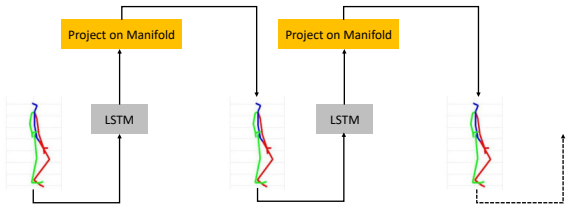

Figure 6. Human motion prediction pipeline. We use LSTM to model the historical motions following many related works [12, 21, 8]. Since the predicted 3D pose may be inaccurate, especially when the input (estimated previously) has errors, we project the 3D pose to the nearest pose on the manifold. In this way, we can solve the problem of error accumulation and significantly improve the robustness of the predictions.

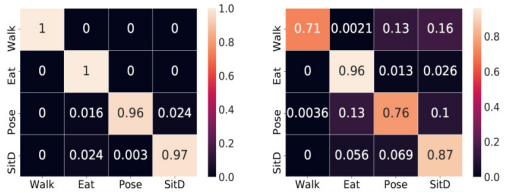

Figure 7. Confusion matrixes of the predicted pose sequences of our approach (left) and baseline (right).

future frames of human poses based on an observed short sequence of poses. Recent work has focused on using deep recurrent neural networks (RNNs) to model human motion, with the goal of learning time-dependent representations. But there is a drawback with this model. Since the predicted pose at time t will be used to predict pose at $t+1$, the prediction error of the current frame degrades the following predictions. We find in experiments that the long term predictions either gradually converge to the natural poses or end up with predicting very inaccurate poses. The problem is more severe for long term predictions because the errors accumulate over time.

Our solution is to refine the predicted pose before it is fed to the network for the following predictions. In particular, we project the pose to the manifold and represent it by the nearest data on the manifold. Although the approach is simple, it enables long term motion predictions up to several minutes without converging to the natural poses or very inaccurate poses. Figure 6 shows the pipeline for motion prediction. The network structure in LSTM consists of two fully connected layers with each having 1024 neurons.

Figure 5 shows two example sequences predicted by the baseline and our method, respectively. We can see that the baseline converges to a certain pose and fails to further generate meaningful sequences. In contrast, our method, by refining the pose at each time step, generates consistently better pose predictions.

6.1. Quantitative Evaluation

We also adopt an approach to quantitatively evaluate the motion prediction method. For long term prediction, it is not appropriate to evaluate the predicted poses frame by frame with the ground truth poses because there are many possibilities to accomplish the target action. In other words, a good (predicted) sequence of poses are not necessarily to be similar to the ground truth sequence in a frame-to-frame matching scheme. So we follow the previous work and evaluate the predicted sequences by action recognition.

We select four actions (*i.e.* walking, eating, posing and sitting down) from the H36M dataset which have large inter-class distances. Then we train LSTM based action classifiers based on the training set. Then we feed the predicted pose sequences to the classifier. Ideally, if the predictions are good, the classifiers should be able to correctly classify them. Figure 7 shows the classification results of our method and the baseline, respectively. The classification accuracy of our approach is about 98.25%. In contrast, if we don't refine the pose sequences, the classification accuracy decreases to 82.50%. The main reason for the degraded accuracy is because the prediction error will accumulate overtime if we don't correct them in time.

7. Conclusion

We present an approach to refine pose sequences by jointly considering (1) to constrain the poses to lie on a manifold, (2) to constrain the pose sequences to be smooth. This is achieved by learning basis dictionaries to approximate the pose manifold. We evaluate the proposed approach by two important tasks: (1) 3D pose estimation from a monocular video, and (2) long-term motion prediction. Consistently better pose sequences are obtained by our approach which demonstrates its practical values.

Acknowledgement: This work was supported by NSF No.1763705 and NSFC No.91748208.

References

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

[2] M. Andriluka, L. Pishchulin, P. V. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *CVPR*, pages 3686–3693, 2014.

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

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

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

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

[7] M. Fieraru, A. Khoreva, L. Pishchulin, and B. Schiele. Learning to refine human pose estimation. In *CVPR Workshops*, pages 205–214, 2018.

[8] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *ICCV*, pages 4346–4354, 2015.

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

[10] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. SMPL: a skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):248:1–248:16, 2015.

[11] J. Mairal, F. R. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *ICML*, pages 689–696, 2009.

[12] J. Martinez, M. J. Black, and J. Romero. On human motion prediction using recurrent neural networks. In *CVPR*, pages 4674–4683, 2017.

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

[14] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt. Monocular 3d human pose estimation in the wild using improved CNN supervision. In *3DV*, pages 506–516, 2017.

[15] J. Mei, C. Wang, and W. Zeng. Online dictionary learning for approximate archetypal analysis. In *ECCV*, pages 501–516, 2018.

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

[17] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499, 2016.

[18] G. Pavlakos, X. Zhou, and K. Daniilidis. Ordinal depth supervision for 3d human pose estimation. In *CVPR*, pages 7307–7316, 2018.

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

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

[21] Y. Tang, L. Ma, W. Liu, and W. Zheng. Long-term human motion prediction by modeling motion context and enhancing motion dynamics. In *IJCAI*, pages 935–941, 2018.

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

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

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

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

[26] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, pages 1096–1103, 2008.

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

[28] C. Wang, H. Qiu, A. L. Yuille, and W. Zeng. Learning basis representation to refine 3d human pose estimations. In *AAAI*, pages 8925–8932, 2019.

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