Deep Kinematic Pose Regression

Xingyi Zhou[1], Xiao Sun[2], Wei Zhang[1], Shuang Liang[3(✉)], and Yichen Wei[2]

[1] Shanghai Key Laboratory of Intelligent Information Processing, School of Computer Science, Fudan University, Shanghai, China
{zhouxy13,weizh}@fudan.edu.cn
[2] Microsoft Research, Beijing, China
{xias,yichenw}@microsoft.com
[3] Tongji University, Shanghai, China
shuangliang@tongji.edu.cn

Abstract. Learning articulated object pose is inherently difficult because the pose is high dimensional but has many structural constraints. Most existing work do not model such constraints and does not guarantee the geometric validity of their pose estimation, therefore requiring a post-processing to recover the correct geometry if desired, which is cumbersome and sub-optimal. In this work, we propose to directly embed a kinematic object model into the deep neutral network learning for general articulated object pose estimation. The kinematic function is defined on the appropriately parameterized object motion variables. It is differentiable and can be used in the gradient descent based optimization in network training. The prior knowledge on the object geometric model is fully exploited and the structure is guaranteed to be valid. We show convincing experiment results on a toy example and the 3D human pose estimation problem. For the latter we achieve state-of-the-art result on Human3.6M dataset.

Keywords: Kinematic model · Human pose estimation · Deep learning

1 Introduction

Estimating the pose of objects is important for understanding the behavior of the object and relevant high level tasks, e.g., facial point localization for expression recognition, human pose estimation for action recognition. It is a fundamental problem in computer vision and has been heavily studied for decades. Yet, it remains challenging, especially when object pose and appearance is complex, e.g., human pose estimation from single view RGB images.

There is a vast range of definitions for object pose. In the simple case, the pose just refers to the global viewpoint of rigid objects, such as car [41] or head [18]. But more often, the pose refers to a set of semantically important points on the object (rigid or non-rigid). The points could be landmarks that can be easily distinguished from their appearances, e.g., eyes or nose on human face [15], and wings or tail on bird [37]. The points could further be the physical joints that

© Springer International Publishing Switzerland 2016
G. Hua and H. Jégou (Eds.): ECCV 2016 Workshops, Part III, LNCS 9915, pp. 186–201, 2016.
DOI: 10.1007/978-3-319-49409-8_17

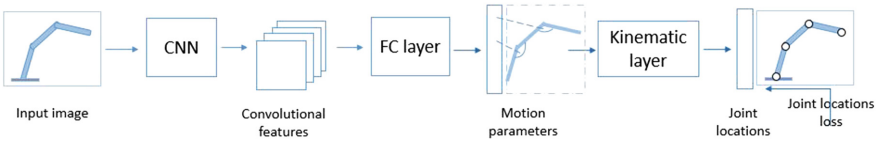


Fig. 1. Illustration of our framework. The input image undergoes a convolutional neutral network and a fully connected layer to output model motion parameters (global potision and rotation angles). The kinematic layer maps the motion parameters to joints. The joints are connected to ground truth joints to compute the joint loss that drives the network training.

defines the geometry of complex articulated objects, such as human hand [20,40] and human body [16,30,39].

Arguably, the articulated object pose estimation is the most challenging. Such object pose is usually very high dimensional and inherently structured. How to effectively represent the pose and perform structure-preserving learning is hard and have been heavily studied. Some approaches represent the object pose in a non-parametric way (as a number of points) and directly learn the pose from data [5,26,27]. The inherent structure is implicitly learnt and modeled from data. Many other approaches use a low dimensional representation by using dimensionality reduction techniques such as PCA [12,20], sparse coding [33, 38,39] or auto-encoder [29]. The structure information is embedded in the low dimensional space. Yet, such embedding is mostly linear and cannot well preserve the complex articulated structural constraints.

In this work, we propose to directly incorporate the articulated object model into the deep neutral network learning, which is the dominant approach for object pose estimation nowadays, for hand [8,20,21,28,31,40] or human body [10,16,17,19,29,30,32,34,39]. Our motivation is simple and intuitive. The kinematic model of such objects is well known as prior knowledge, such as the object bone lengths, bone connections and definition of joint rotations. From such knowledge, it is feasible to define a continuous and differentiable kinematic function with respect to the model motion parameters, which are the rotation angles. The kinematic function can be readily put into a neutral network as a special layer. The standard gradient descent based optimization can be performed in the same way for network training. The learning framework is exemplified in Fig. 1. In this way, the learning fully respects the model geometry and preserves the structural constraints. Such end-to-end learning is better than the previous approaches that rely on a separate post-processing step to recover the object geometry [31,39].

This idea is firstly proposed in the recent work [40] for depth based hand pose estimation and is shown working well. However, estimating 3D structure from depth is a simple problem by nature. It is still unclear how well the idea can be generalized to other objects and RGB images. In this work, we apply the idea to more problems (a toy example and human pose estimation) and for the first time show that the idea works successfully on different articulated pose estimation

problems and inputs, indicating that the idea works in general. Especially, for the challenging 3D human pose estimation from single view RGB images, we present state-of-the-art results on the Human3.6M dataset [12].

2 Related Work

The literature on pose estimation is comprehensive. We review previous work from two perspectives that are mostly related to our work: object pose representation and deep learning based human pose estimation.

2.1 Pose Representation

An object pose consists of a number of related points. The key for pose representation is how to represent the mutual relationship or structural constraints between these points. There are a few different previous approaches.

Pictorial Structure Model. Pictorial structure model [7] is one of the most popular methods in early age. It represents joints as vertexes and joint relations as edges in a non-circular graph. Pose estimation is formulated as inference problems on the graph and solved with certain optimization algorithms. Its extensions [14,23,35] achieve promising results in 2D human estimation, and has been extended to 3D human pose [2]. The main drawback is that the inference algorithm on the graph is usually complex and slow.

Linear Dictionary. A widely-used method is to denote the structural points as a linear combination of templates or basis [15,33,38,39]. [15] represent 3D face landmarks by a linear combination of shape bases [22] and expression bases [4]. It learns the shape, expression coefficients and camera view parameters alternatively. [33] express 3D human pose by an over-complex dictionary with a sparse prior, and solve the sparse coding problem with alternating direction method. [38] assign individual camera view parameters for each pose template. The sparse representation is then relaxed to be a convex problem that can be solved efficiently.

Linear Feature Embedding. Some approaches learn a low dimensional embedding [12,20,29] from the high dimensional pose. [12] applies PCA to the labeled 3D points of human pose. The pose estimation is then performed in the new orthogonal space. The similar idea is applied to 3D hand pose estimation [20]. It uses PCA to project the 3D hand joints to a lower space as a physical constraint prior for hand. [29] extend the linear PCA projector to a multi-layer anto-encoder. The decoder part is fine-tuned jointly with a convolutional neural network in an end-to-end manner. A common drawback in above linear representations is that the complex object pose is usually on a non-linear manifold in the high dimensional space that cannot be easily captured by a linear representation.

Implicit Representation by Retrieval. Many approaches [6,17,36] store massive examples in a database and perform pose estimation as retrieval, therefore avoiding the difficult pose representation problem. [6] uses a nearest neighbors search of local shape descriptors. [17] proposes a max-margin structured

learning framework to jointly embed the image and pose into the same space, and then estimates the pose of a new image by nearest neighbor search in this space. [36] builds an image database with 3D and 2D annotations, and uses a KD-tree to retrieve 3D pose whose 2D projection is similar to the input image. The performance of these approaches highly depends on the quality of the database. The efficiency of nearest neighbor search could be an issue when the database is large.

Explicit Geometric Model. The most aggressive and thorough representation is to use an explicit and generative geometric model, including the motion and shape parameters of the object [3,25]. Estimating the parameters of the model from the input image(s) is performed by heavy optimization algorithms. Such methods are rarely used in a learning based manner. The work in [40] firstly uses a generative kinematic model for hand pose estimation in the deep learning framework. Inspire by this work, we extend the idea to more object pose estimation problems and different inputs, showing its general applicability, especially for the challenging problem of 3D human pose estimation from single view RGB images.

2.2 Deep Learning on Human Pose Estimation

The human pose estimation problem has been significantly advanced using deep learning since the pioneer deep pose work [32]. All current leading methods are based on deep neutral networks. [34] shows that using 2D heat maps as intermediate supervision can dramatically improve the 2D human part detection results. [19] use an hourglass shaped network to capture both bottom-up and top-down cues for accurate pose detection. [10] shows that directly using a deep residual network (152 layers) [9] is sufficient for high performance part detection. To adopt these fully-convolutional based heat map regression method for 3D pose estimation, an additional model fitting step is used [39] as a post processing. Other approaches directly regress the 2D human pose [5,32] or 3D human pose [16,29,30]. These detection or regression based approaches ignore the prior knowledge of the human model and does not guarantee to preserve the object structure. They sometimes output geometrically invalid poses.

To our best knowledge, for the first time we show that integrating a kinematic object model into deep learning achieves state-of-the-art results in 3D human pose estimation from single view RGB images.

3 Deep Kinematic Pose Estimation

3.1 Kinematic Model

An articulated object is modeled as a kinematic model. A kinematic model is composed of several *bones* and *joints*. A bone is a segment of a fixed length, and a joint is the end point of a bone. One bone meets at another at a joint, forming a tree structure. Bones can rotate among a conjunct joint. Without

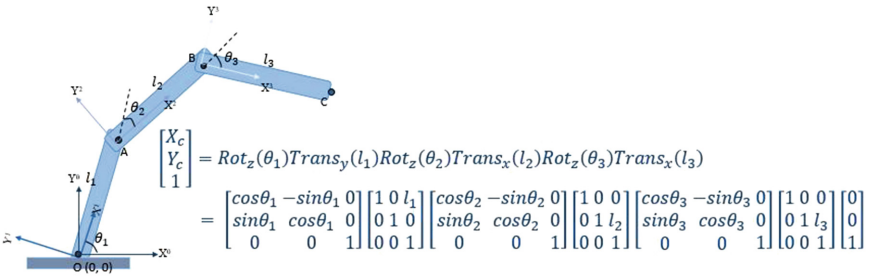


Fig. 2. A sample 2D kinematic model. It has 3 and 4 joints. The joint location is calculated by multiplying a series of transformation matrices.

loss generality, one joint is considered as the root joint (For example, wrist for human hand and pelvis for human body). The root defines the global position and global orientation of the object.

For a kinematic model of J joints, it has $J-1$ bones. Let $\{l_i\}_{i=1}^{J-1}$ be the collection of bone lengths, they are fixed for a specific subject and provided as prior knowledge. For different subjects, we assume they only differ in a global scale, i.e. $\forall i, l_i' = s \times l_i$. The scale is also provided as prior knowledge, e.g. through a calibration process.

Let the rotation angle of the i-th joint be θ_i, the motion parameter Θ includes the global position $\mathbf{p}$, global orientation $\mathbf{o}$, and all the rotation angles, $\Theta = \{\mathbf{p}, \mathbf{o}\} \cup \{\theta_i\}_{i=1}^{J}$. The forward kinematic function is a mapping from motion parameter space to joint location space.

$$\mathcal{F} : \{\Theta\} \rightarrow \mathcal{Y} \tag{1}$$

where $\mathcal{Y}$ is the coordinate for all joints, $\mathcal{Y} \in \mathcal{R}^{3\times J}$ for 3D object and $\mathcal{Y} \in \mathcal{R}^{2\times J}$ for 2D object.

The kinematic function is defined on a kinematic tree. An example is shown in Fig. 2. Each joint is associated with a local coordinate transformation defined in the motion parameter, including a rotation from its rotation angles and a translation from its out-coming bones. The final coordinate of a joint is obtained by multiplying a series of transformation matrices along the path from the root joint to itself. Generally, the global position of joint u is

$$p_u = (\prod_{v \in Pa(u)} Rot(\theta_v) \times Trans(l_v))\mathbf{O}^\top \tag{2}$$

where $Pa(u)$ is the set of its parents nodes at the kinematic tree, and $\mathbf{O}$ is the origin in homogenous coordinate, i.e., $\mathbf{O} = [0, 0, 1]^\top$ for 2D and $\mathbf{O} = [0, 0, 0, 1]^\top$ for 3D. For 3D kinematic model, each rotation is assigned with one of the $\{X, Y, Z\}$ axis, and at each joint there can be multiple rotations. The direction of translation is defined in the canonical local coordinate frame where the motion parameters are all zeros.

In [40], individual bounds for each angle can be set as additional prior knowledge for the objects. It is feasible for human hand since all the joints have at most 2 rotation angles and their physical meaning is clear. However, in the case of human body, angle constraint are not individual, it is conditioned on pose [1] and hard to formulate. We leave it as future work to explore more efficient and expressive constraints.

As shown in Fig. 2, the forward kinematic function is continuous with respect to the motion parameter. It is thus differentiable. As each parameter occurs in one matrix, this allows easy implementation of back-propagation. We simply replace the corresponding rotational matrix by its derivation matrix and keep other items unchanged. The kinematic model can be easily put in a neural network as a layer for gradient descent-based optimization.

3.2 Deep Learning with a Kinematic Layer

We discuss our proposed approach and the other two baseline methods to learn the pose of an articulated object. They are illustrated in Fig. 3. All three methods share the same basic convolutional neutral network and only differs in their ending parts, which is parameter-free. Therefore, we can make fair comparison between the three methods.

Now we elaborate on them. The first method is a baseline. It directly estimates the joint locations by a convolutional neural network, using Euclidean Loss on the joints. It is called **direct joint**. It has been used for human pose estimation [16,32] and hand pose estimation [20]. This approach does not consider the geometry constraints of the object. The output is less structured and could be invalid, geometrically.

Instead, we propose to use a kinematic layer at the top of the network. The network predicts the motion parameters of the object, while the learning is still guided by the joint location loss. We call this approach **kinematic joint**. The joint location loss with respect to model parameter Θ is Euclidean Loss

$$L(\Theta) = \frac{1}{2}||\mathcal{F}(\Theta) - Y||^2 \tag{3}$$

where $Y \in \mathcal{Y}$ is the ground truth joint location in the input image. Since this layer has no free parameters to learn and appears in the end of the network, we can think of the layer as coupled with the Euclidean loss Layer, serving as a geometrically more accurate loss layer.

Compared to direct joint approach, our proposed method fully incorporates prior geometric knowledge of the object, such as the bone lengths and spatial relations between the joints. The joint location is obtained by a generative process and guaranteed to be valid. The motion parameter space is more compact than the unconstrained joint space, that is, the degrees of freedom of motion parameters are smaller than that of joints, for example, in Sect. 4.2, the DOF is 27 for motion parameters but 51 for joints. Overall, our method can be considered as a better regularization on the output space.

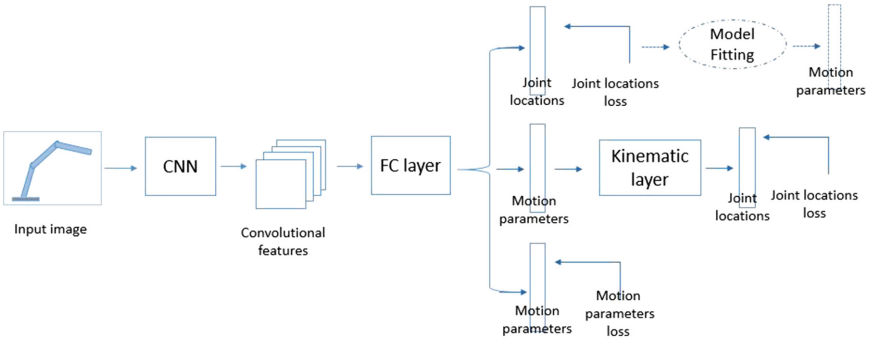


Fig. 3. Three methods for object pose estimation. Top (**Direct Joint**): the network directly outputs all the joints. Such estimated joints could be invalid geometrically. Optionally, they can be optimized via a model-fitting step to recover a correct model, referred to as **ModelFit** in the text. Middle (**Kinematic Joint**): our proposed approach. The network outputs motion parameters to the kinematic layer. The layer outputs joints. Bottom (**Direct Parameter**): the network directly outputs motion parameters.

Unlike dictionary-based representations [33,38] that require a heuristic sparse regularization, our approach has a clear geometrical interpretation and its optimization is feasible in deep neutral network training. Besides, it produces joint rotation angles that could be useful in certain applications.

The third method is a less obvious baseline. It directly estimates the motion parameters, using Euclidean loss on those parameters. It is called **direct parameter**. Intuitively, this approach cannot work well because the roles of different parameters are quite different and it is hard to balance the learning weights between those parameters. For example, the global rotation angles on the root joint affects all joints. It has much more impacts than those parameters on distal joints but it is hard to quantify this observation. Moreover, for complex articulated objects the joint locations to joint angles mapping is not one-to-one but ambiguous, e.g., when the entire arm is straight, roll angle on the shoulder joint can be arbitrary and it does not affect the location of elbow and wrist. It is hard to resolve such ambiguity in the network training. By contrast, the joint location loss in our kinematic approach is widely distributed over all object parts. It is well behaved and less ambiguous.

We note that it is possible to enforce the geometric constraints by fitting a kinematic model to some estimated joints as a post-processing [31,39]. For example, [31] recovers a 3D kinematic hand model using a PSO-based optimization, by fitting the model into the 2D hand joint heat maps. [39] obtains 3D human joints represented by a sparse dictionary using an EM optimization algorithm. In our case, we provide an additional **ModelFit** baseline that recovers a kinematic model from the output of direct joint baseline by minimizing the loss in Eq. 3.

4 Experiment

The work in [40] applies the kinematic pose regression approach for depth based 3D hand pose estimation and has shown good results. To verify the generality of the idea, we apply this approach for two more different problems. The first is a toy example for simple 2D articulated object on synthesized binary image. The second is 3D human pose estimation from single RGB images, which is very challenging.

4.1 A Toy Problem

In the toy problem, the object is 2D. The image is synthesized and binary. As shown in Fig. 4 top, the input image is generated from a 3 dimensional motion parameter $\Theta = \{x, y, \theta\}$, where x, y is the image coordinate (normalized between $0-1$) of the root joint, and θ indicates the angle between the each bone and the vertical line.

We use a 5 layer convolutional neutral network. The network structure and hyper-parameters are the same as [40]. The input image resolution is 128×128. The bone length is fixed as 45 pixels. We randomly synthesize $16k$ samples for training and $1k$ samples for testing. Each model is trained for 50 epoches.

As described in Fig. 3, we perform our **direct joint**, **kinematic joint** and **direct parameter** on this task. The joint location for **direct parameter** is computed by the kinematic layer as a post process in testing. It turns out all the 3 methods achieve low joint errors in this simple case. The mean joint errors for **direct joint**, **kinematic Joint**, **direct parameter** are 5.1 pixels, 4.9 pixels, and 4.8 pixels, respectively. **direct joint** is the worst, probably because the task is easy for all the setting and these two require to learn more parameters. When we evaluate the average length of the two bones for **direct joint** regression, we find it has a standard deviation of 5.3 pixels (11.8 % of the bone length 45 pixels), indicating that the geometry constraint is badly violated.

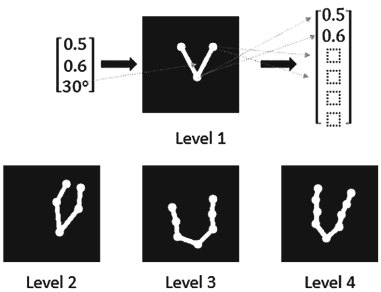


Fig. 4. Illustration of the toy problem. The input images are synthesized and binary. **Top**: Motion parameter and joint representation of a simple object with 3 motion parameters. **Bottom**: Example input images for 3 objects with different complexity levels. They have 6, 8, and 10 motion parameters, respectively.

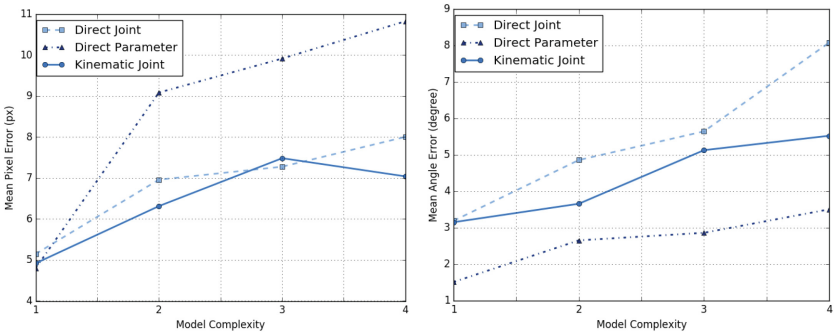


Fig. 5. Experimental results on mean joint locations error(**Left**) and mean angle error(**Right**) with respect to model complexity. It shows when as kinematic model becoming complex, our approach is stable in both metric.

Since it is hard to claim any other significant difference between the 3 method in such a simple case, we gradually increase the model complexity. Global orientation and more joint angles are added to the kinematic model. For each level of complexity, we add one more bone with one rotational angle on each distal bone. Example input image are illustrated in Fig. 4 bottom.

The joint location errors and angle errors with respect to the model complexity are shown in Fig. 5. Note that for **direct joint** regression, the angles are directly computed from the triangle. The results show that the task become more difficult for all methods. **Direct parameter** gets high joint location errors, probably because a low motion parameter error does not necessarily implies a low joint error. It is intuitive that it always get best performance on joint angle, since it is the desired learning target. **Direct joint** regression also has large error on its recovered joint angles, and the average length of each bone becomes more unstable. It shows that geometry structure is not easy to learn. Using a generative **kinematic joint** layer keeps a decent accuracy on both metric among all model complexity. This is important for complex objects in real applications, such as human body.

4.2 3D Human Pose Regression

We test our method on the problem of full 3D human pose estimation from single view RGB images. Following [16], the 3D coordinate of joints is represented by its offset to a root joint. We use Human 3.6M dataset [12]. Following the standard protocol in [12,16,38], we define $J = 17$ joints on the human body. The dataset contains millions of frames of RGB images. They are captured over 7 subjects performing 15 actions from 4 different camera views. Each frame is accurately annotated by a MoCap system. We treat the 4 cameras of the same subject separately. The training and testing data partition follows previous works [12,16,39]. All frames from 5 subjects(S1, S5, S6, S7, S8) are used for training. The remaining 2 subjects(S9, S11) are for testing.

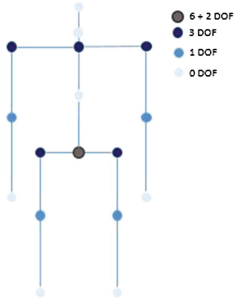


Fig. 6. Illustration of Human Model. It contains 17 joints and 27 motion parameters. See text for the detail kinematic structure.

Our kinematic human model is illustrated in Fig. 6. It defines 17 joints with 27 motion parameters. The pelvis is set as the root joint. Upside it is the neck, which can roll and yaw among the root. Torso is defined as the mid point of neck and pelvis. It has no motion parameter. Pelvis and neck orientation determine the positions of shoulders and hips by a fixed bone transform. Each shoulder/hip has full 3 rotational angles, and elbow/knee has 1 rotational angle. Neck also has 3 rotational angles for nose and head orientation. Note that there can be additional rotation angles on the model, for example shoulders can rotate among neck within a subtle degree and elbows can roll itself. Our rule of thumb is to simulate real human structure and keep the model simple.

Table 1. Results of Human3.6M Dataset. The numbers are mean Euclidean distance (mm) between the ground-truth 3D joints and the estimations of different methods.

	Directions	Discussion	Eating	Greeting	Phoning	Photo	Posing	Purchases
LinKDE [12]	132.71	183.55	132.37	164.39	162.12	205.94	150.61	171.31
Li et al. [16]	-	148.79	104.01	127.17	-	189.08	-	-
Li et al. [17]	-	136.88	96.94	124.74	-	168.68	-	-
Tekin et al. [29]	-	129.06	91.43	121.68	-	162.17	-	-
Tekin et al. [30]	132.71	158.52	87.95	126.83	118.37	185.02	114.69	107.61
Zhou et al. [39]	**87.36**	109.31	**87.05**	103.16	116.18	143.32	106.88	99.78
Ours(Direct)	106.38	104.68	104.28	107.80	115.44	**114.05**	103.80	109.03
Ours(ModelFit)	109.75	110.47	113.98	112.17	123.66	122.82	121.27	117.98
Ours(Kinematic)	91.83	**102.41**	96.95	**98.75**	**113.35**	125.22	**90.04**	**93.84**
	Sitting	SittingDown	Smoking	Waiting	WalkDog	Walking	WalkPair	Average
LinKDE [12]	151.57	243.03	162.14	170.69	177.13	96.60	127.88	162.14
Li et al. [16]	-	-	-	-	146.59	77.60	-	-
Li et al. [17]	-	-	-	-	132.17	69.97	-	-
Tekin et al. [29]	-	-	-	-	130.53	**65.75**	-	-
Tekin et al. [30]	136.15	205.65	118.21	146.66	128.11	65.86	**77.21**	125.28
Zhou et al. [39]	**124.52**	199.23	107.42	118.09	114.23	79.39	97.70	113.01
Ours(Direct)	125.87	**149.15**	112.64	105.37	**113.69**	98.19	110.17	112.03
Ours(ModelFit)	137.29	157.44	136.85	110.57	128.16	102.25	114.61	121.28
Ours(Kinematic)	132.16	158.97	**106.91**	**94.41**	126.04	79.02	98.96	**107.26**

We found that the ground truth 3D joints in the dataset has strictly the same length for each bone across all the frames on the same subject. Also, the lengths of the same bone across the 7 subjects are very close. Therefore, in our human model, the bone lengths are simply set as the average bone lengths of the 7 subjects. In addition, every subject is assigned a global scale. The scale is computed from the sum bone lengths divided by the average sum bone length. It is a fixed constant for each subject during training. During testing, we assume the subject scale is unknown and simply set it as 1. In practical scenarios, the subject scale can be estimated by a calibrating pre processing and then fixed.

Following [16,29], we assume the bounding box for the subject in known. The input images are resized to 224×224. Note that it is important not to change the aspect ratio for the kinematic based method, we use border padding to keep the real aspect ratio. The training target is also normalized by the bounding box size. Since our method is not action-dependent, we train our model using all the data from the 15 actions. By contrast, previous methods [12,17,39] use data for each action individually, as their local feature, retrieval database or pose dictionary may prefer more concrete templates.

We use the 50-layer Residual Network [9] that is pre-trained on ImageNet [24] as our initial model. It is then fine-tuned on our task. Totally available training data for the 5 subjects is about 1.5 million images. They are highly similar and redundant. We randomly sample 800 k frames for training. No data augmentation is used. We train our network for 70 epoches, with base learning rate 0.003 (dropped to 0.0003 after 50 epochs), batch size 52 (on 2 GPUs), weight decay 0.0002 and momentum 0.9. Batch-normalization [11] is used. Our implementation is based on Caffe [13].

The experimental results are shown in Table 1. The results for comparison methods [12,16,17,29,29,30,39] are from their published papers. Thanks to the powerful Residual Network [9], our **direct joint** regression base line is already

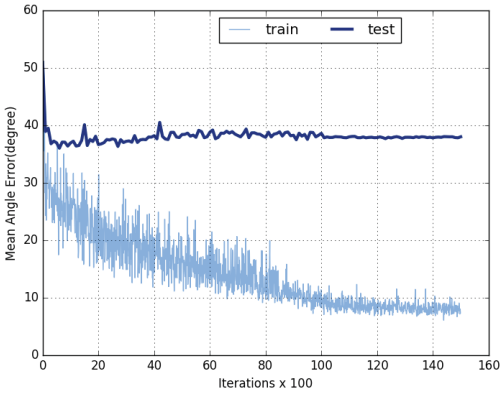
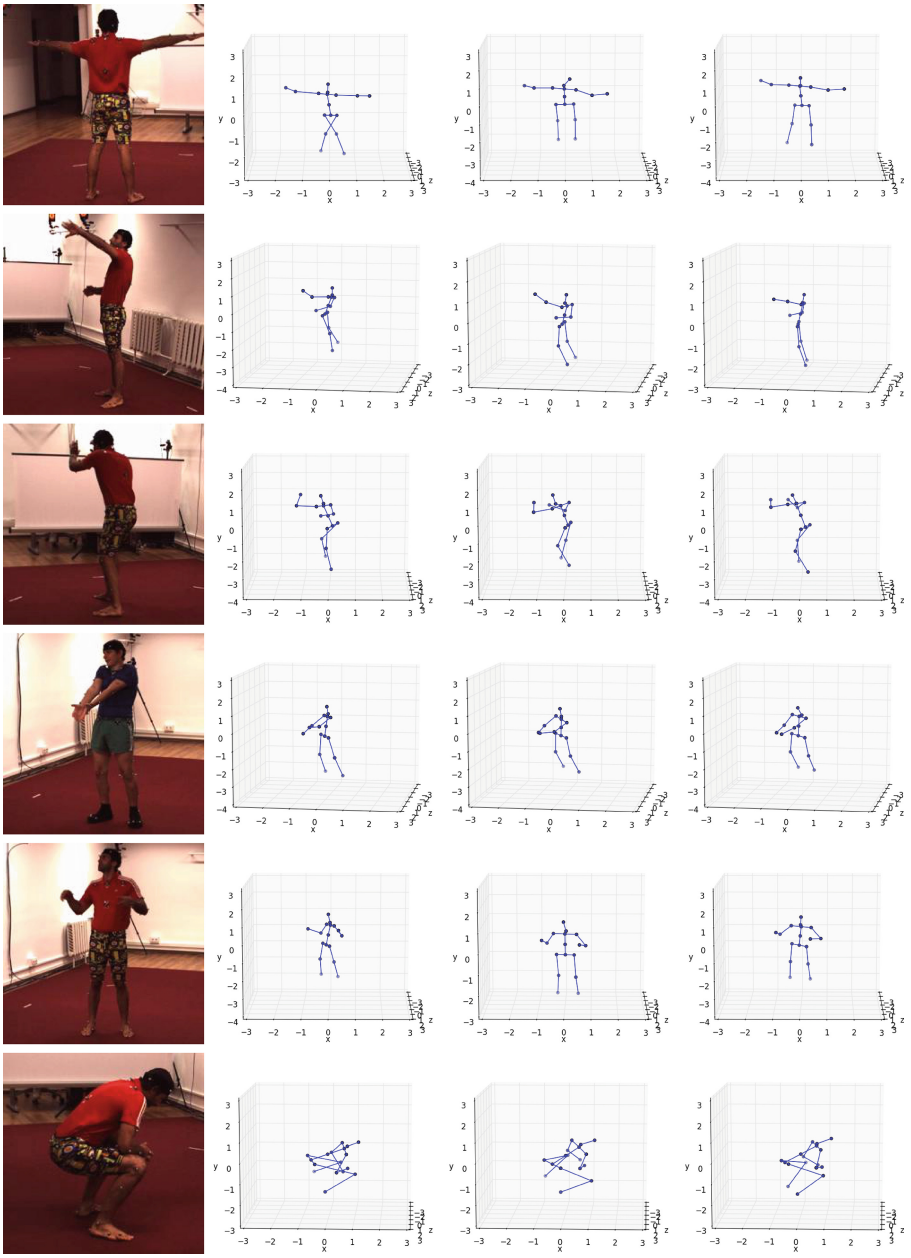


Fig. 7. Training curve of direct motion parameter regression. Although the training loss keeps dropping, the testing loss remains high.

Table 2. Qualitative results for direct joint regression and kinematic on Human3.6M dataset. They show some typical characters for these methods. The results are ploted at 3D space from the same viewpoint.

Input Image	Direct Joint	Kinematic Joint	Ground-truth



the state-of-the-art. Since we used additional training data from ImageNet, comparing our results to previous works is unfair, and the superior performance of our approach is not the contribution of this work. We include the previous works' results in Table 1 just as references.

Kinematic joint achieves the best average accuracy among all methods, demonstrating that embedding a kinematic layer in the network is effective. Qualitative results are shown in Table 2, including some typical failure cases for **direct joint** include flipping the left and right leg when the person is back to the camera (Row 1) and abnormal bone length (Row 2,3).

Despite **direct joint** regression achieve a decent accuracy for 3D joint location, we can further apply a kinematic model fitting step, as described in the previous sections. The model fitting is based on gradient-descent for each frame. The results is shown in Table 1 as **ours(Fit)**, it turns out to be worse than **direct joint**, indicating such post-preprocessing is sub-optimal if the initial poses do not have valid structural information.

We also tried **direct parameter** regression on this dataset. The training target for motion parameter is obtained in the same way as described above, by gradient descent. However, as shown in Fig. 7, the testing error keeps high. Indicating direct parameter regression does not work on this task. There could be two reasons: many joints have full 3 rotational angles, this can easily cause ambiguous angle target, for example, if the elbow or knee is straight, the roll angle for shoulder or hip can be arbitrary. Secondly, learning 3D rotational angles is more obscure than learning 3D joint offsets. It is even hard for human to annotate the 3D rotational angles from an RGB image. Thus it may require more data or more time to train.

5 Conclusions

We show that geometric model of articulated objects can be effectively used within the convolutional neural network. The learning is end-to-end and we get rid of the inconvenient post-processing as in previous approaches. The experimental results on 3D human pose estimation shows that our approach is effective for complex problems. In the future work, we plan to investigate more sophisticated constraints such as those on motion parameters. We hope this work can inspire more works on combining geometry with deep learning.

Acknowledgments. We would like to thank anonymous reviewers who gave us useful comments. This work was supported by Natural Science Foundation of China (No. 61473091), National Science Foundation of China (No. 61305091), and The Fundamental Research Funds for the Central Universities (No. 2100219054).

References

1. Akhter, I., Black, M.J.: Pose-conditioned joint angle limits for 3D human pose reconstruction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1446–1455 (2015)

2. Belagiannis, V., Amin, S., Andriluka, M., Schiele, B., Navab, N., Ilic, S.: 3D pictorial structures for multiple human pose estimation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014
3. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M.J.: http://arxiv.org/abs/1607.08128
4. Cao, C., Weng, Y., Zhou, S., Tong, Y., Zhou, K.: Facewarehouse: a 3D facial expression database for visual computing. IEEE Trans. Vis. Comput. Graph. **20**(3), 413–425 (2014)
5. Carreira, J., Agrawal, P., Fragkiadaki, K., Malik, J.: Human pose estimation with iterative error feedback. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
6. Choi, C., Sinha, A., Hee Choi, J., Jang, S., Ramani, K.: A collaborative filtering approach to real-time hand pose estimation. In: The IEEE International Conference on Computer Vision (ICCV), December 2015
7. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. Int. J. Comput. Vis. **61**(1), 55–79 (2005)
8. Ge, L., Liang, H., Yuan, J., Thalmann, D.: Robust 3D hand pose estimation in single depth images: from single-view CNN to multi-view CNNs. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
10. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B.: http://arxiv.org/abs/1605.03170
11. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
12. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6M: large scale datasets and predictive methods for 3D human sensing in natural environments. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1325–1339 (2014)
13. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093 (2014)
14. Johnson, S., Everingham, M.: Learning effective human pose estimation from inaccurate annotation. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1465–1472. IEEE (2011)
15. Jourabloo, A., Liu, X.: Large-pose face alignment via CNN-based dense 3D model fitting. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
16. Li, S., Chan, A.B.: 3D human pose estimation from monocular images with deep convolutional neural network. In: Cremers, D., Reid, I., Saito, H., Yang, M.-H. (eds.) ACCV 2014. LNCS, vol. 9004, pp. 332–347. Springer, Heidelberg (2015). doi:10.1007/978-3-319-16808-1_23
17. Li, S., Zhang, W., Chan, A.B.: Maximum-margin structured learning with deep networks for 3D human pose estimation. In: The IEEE International Conference on Computer Vision (ICCV), December 2015
18. Meyer, G.P., Gupta, S., Frosio, I., Reddy, D., Kautz, J.: Robust model-based 3D head pose estimation. In: The IEEE International Conference on Computer Vision (ICCV), December 2015
19. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. CoRR abs/1603.06937 (2016). http://arxiv.org/abs/1603.06937

20. Oberweger, M., Wohlhart, P., Lepetit, V.: Hands deep in deep learning for hand pose estimation. arXiv preprint arXiv:1502.06807 (2015)
21. Oberweger, M., Wohlhart, P., Lepetit, V.: Training a feedback loop for hand pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3316–3324 (2015)
22. Paysan, P., Knothe, R., Amberg, B., Romdhani, S., Vetter, T.: A 3D face model for pose and illumination invariant face recognition. In: Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2009, pp. 296–301. IEEE (2009)
23. Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Poselet conditioned pictorial structures. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2013
24. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.: Imagenet large scale visual recognition challenge. CoRR abs/1409.0575 (2014). http://arxiv.org/abs/1409.0575
25. Sharp, T., Keskin, C., Robertson, D., Taylor, J., Shotton, J., Kim, D., Rhemann, C., Leichter, I., Vinnikov, A., Wei, Y., Freedman, D., Kohli, P., Krupka, E., Fitzgibbon, A., Izadi, S.: Accurate, robust, and flexible realtime hand tracking. In: CHI (2015)
26. Shotton, J., Girshick, R., Fitzgibbon, A., Sharp, T., Cook, M., Finocchio, M., Moore, R., Kohli, P., Criminisi, A., Kipman, A., et al.: Efficient human pose estimation from single depth images. IEEE Trans. Pattern Anal. Mach. Intell. **35**(12), 2821–2840 (2013)
27. Sun, X., Wei, Y., Liang, S., Tang, X., Sun, J.: Cascaded hand pose regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 824–832 (2015)
28. Supancic III., J.S., Rogez, G., Yang, Y., Shotton, J., Ramanan, D.: Depth-based hand pose estimation: methods, data, and challenges. arXiv preprint arXiv:1504.06378 (2015)
29. Tekin, B., Katircioglu, I., Salzmann, M., Lepetit, V., Fua, P.: Structured prediction of 3D human pose with deep neural networks. arXiv preprint arXiv:1605.05180 (2016)
30. Tekin, B., Rozantsev, A., Lepetit, V., Fua, P.: Direct prediction of 3D body poses from motion compensated sequences. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
31. Tompson, J., Stein, M., Lecun, Y., Perlin, K.: Real-time continuous pose recovery of human hands using convolutional networks. ACM Trans. Graph. **33**, 169 (2014)
32. Toshev, A., Szegedy, C.: Deeppose: human pose estimation via deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1653–1660 (2014)
33. Wang, C., Wang, Y., Lin, Z., Yuille, A.L., Gao, W.: Robust estimation of 3D human poses from a single image. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014
34. Wei, S.E., Ramakrishna, V., Kanade, T., Sheikh, Y.: Convolutional pose machines. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
35. Yang, Y., Ramanan, D.: Articulated pose estimation with flexible mixtures-of-parts. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1385–1392. IEEE (2011)

36. Yasin, H., Iqbal, U., Kruger, B., Weber, A., Gall, J.: A dual-source approach for 3D pose estimation from a single image. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
37. Yu, X., Zhou, F., Chandraker, M.: Deep deformation network for object landmark localization. arXiv preprint arXiv:1605.01014 (2016)
38. Zhou, X., Leonardos, S., Hu, X., Daniilidis, K.: 3D shape estimation from 2D landmarks: a convex relaxation approach. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015
39. Zhou, X., Zhu, M., Leonardos, S., Derpanis, K.G., Daniilidis, K.: Sparseness meets deepness: 3D human pose estimation from monocular video. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
40. Zhou, X., Wan, Q., Zhang, W., Xue, X., Wei, Y.: Model-based deep hand pose estimation. In: IJCAI (2016)
41. Zhu, M., Zhou, X., Daniilidis, K.: Single image pop-up from discriminatively learned parts. In: The IEEE International Conference on Computer Vision (ICCV), December 2015