Procrustean Regression Networks: Learning 3D Structure of Non-rigid Objects from 2D Annotations

Sungheon Park[1], Minsik Lee[2], and Nojun Kwak[3(✉)]

[1] Samsung Advanced Institute of Technology (SAIT), Yongin-si, Korea
sungheonpark@snu.ac.kr
[2] Hanyang University, Seoul, Korea
mleepaper@hanyang.ac.kr
[3] Seoul National University, Seoul, Korea
nojunk@snu.ac.kr

Abstract. We propose a novel framework for training neural networks which is capable of learning 3D information of non-rigid objects when only 2D annotations are available as ground truths. Recently, there have been some approaches that incorporate the problem setting of non-rigid structure-from-motion (NRSfM) into deep learning to learn 3D structure reconstruction. The most important difficulty of NRSfM is to estimate both the rotation and deformation at the same time, and previous works handle this by regressing both of them. In this paper, we resolve this difficulty by proposing a loss function wherein the suitable rotation is automatically determined. Trained with the cost function consisting of the reprojection error and the low-rank term of aligned shapes, the network learns the 3D structures of such objects as human skeletons and faces during the training, whereas the testing is done in a single-frame basis. The proposed method can handle inputs with missing entries and experimental results validate that the proposed framework shows superior reconstruction performance to the state-of-the-art method on the Human 3.6M, 300-VW, and SURREAL datasets, even though the underlying network structure is very simple.

1 Introduction

Inferring 3D poses from several 2D observations is inherently an underconstrained problem. Especially, for non-rigid objects such as human faces or bodies, it is harder to retrieve the 3D shapes than for rigid objects due to their shape deformations.

S. Park, M. Lee—Authors contributed equally.
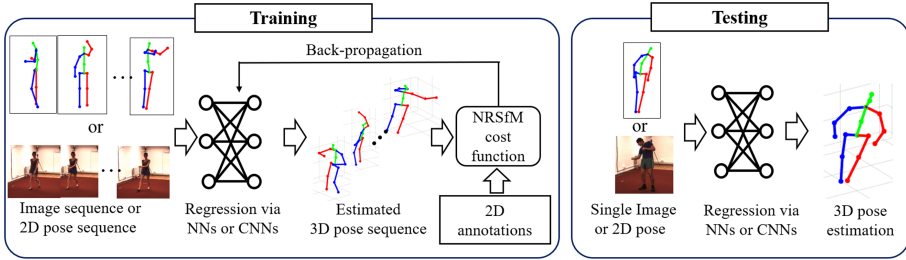
Fig. 1. Illustration of PRN. During the training, sequences of images or 2D poses are fed to the network, and their 3D shapes are estimated as the network outputs. The network is trained using the cost function which is based on an NRSfM algorithm. Testing is done by a simple feed-forward operation in a single-frame basis.

There are two distinct ways to retrieve 3D shapes of non-rigid objects from 2D observations. The first approach is to use a 3D reconstruction algorithm. Non-rigid structure from motion (NRSfM) algorithms [2,4,9,12,21] are designed to reconstruct 3D shapes of non-rigid objects from a sequence of 2D observations. Since NRSfM algorithms are not based on any learned models, the algorithms should be applied to each individual sequence, which makes the algorithm time-consuming when there are numerous number of sequences. The second approach is to learn the mappings from 2D to 3D with 3D ground truth training data. Prior knowledge can be obtained by dictionary learning [43,44], but neural networks or convolutional neural networks (CNNs) are the most-used methods to learn the 2D-to-3D or image-to-3D mappings [24,30], recently. However, 3D ground truth data are essential to learn those mappings, which requires large amounts of costs and efforts compared to the 2D data acquisition.

There is another possibility: With the framework which combines those two different frameworks, *i.e.*, NRSfM and neural networks, it is possible to overcome the limitations and to take advantages of both. There have been a couple of works that implement NRSfM using deep neural networks [6,19], but these methods mostly focus on the structure-from-category (SfC) problem, in which the 3D shapes of different rigid subjects in a category are reconstructed, and the deformation between subjects are not very diverse. Experiments on the CMU MoCap data in [19] show that, for data with diverse deformations, its generalization performance is not very good. Recently, Novotny *et al.* [27] proposed a neural network that reconstructs 3D shapes from monocular images by canonicalizing 3D shapes so that the 3D rigid motion is registered. This method has shown successful reconstruction results for data with more diverse deformations, which has been used in traditional NRSfM research. Wang *et al.* [38] also proposed knowledge distillation method that incorporate NRSfM algorithms as a teacher, which showed promising results on learning 3D human poses from 2D points.

The main difficulty of NRSfM is that one has to estimate both the rigid motion and the non-rigid shape deformation, which has been discussed

extensively in the field of NRSfM throughout the past two decades. Especially, motion and deformation can get mixed up and some parts of rigid motions can be mistaken to be deformations. This has been first pointed out in [21], in which conditions derived from the generalized Procrustes analysis (GPA) has been adopted to resolve the problem. Meanwhile, all recent neural-network-based NRSfM approaches attempt to regress both the rigid motion and non-rigid deformation at the same time. Among these, only Novotny *et al.* [27] deals with the motion-deformation-separation problem in NRSfM, which is addressed as "shape transversality." Their solution is to register motions of different frames using an auxiliary neural network.

In this paper, we propose an alternative to this problem: First, we prove that a set of Procrustes-aligned shapes is transversal. Based on this fact, rather than explicitly estimating rigid motions, we propose a novel loss, in which suitable motions are determined automatically based on Procrustes alignment. This is achieved by modifying the cost function recently proposed in Procrustean regression (PR) [29], an NRSfM scheme that shares similar motivations with our work, which is used to train neural networks via back-propagation. Thanks to this new loss function, the network can concentrate only on the 3D shape estimation, and accordingly, the underlying structure of the proposed neural network is quite simple. The proposed framework, *Procrustean Regression Network* (PRN), learns to infer 3D structures of deformable objects using only 2D ground truths as training data.

Figure 1 illustrates the flow of the proposed framework. PRN accepts a set of image sequences or 2D point sequences as inputs at the training phase. The cost function of PRN is formulated to minimize the reprojection error and the nuclear norm of aligned shapes. The whole training procedure is done in an end-to-end manner, and the reconstruction result for an individual image is generated at the test phase via a simple forward propagation without requiring any post processing step for 3D reconstruction. Unlike the conventional NRSfM algorithms, PRN robustly estimates 3D structure of unseen test data with feed-forward operations in the test phase, taking the advantage of neural networks. The experimental results verify that PRN effectively reconstructs the 3D shapes of non-rigid objects such as human faces and bodies.

2   Related Works

The underlying assumption of NRSfM methods is that the 3D shape or the 3D trajectory of a point is interpreted as a weighted sum of several bases [2,4]. 3D shapes are obtained by factorizing a shape matrix or a trajectory matrix so that the matrix has a pre-defined rank. Improvements have been made by several works which use probabilistic principal components analysis [34], metric constraints [28], course-to-fine reconstruction algorithm [3], complementary-space modeling [12], block sparse dictionary learning [18], or force-based models [1]. The major disadvantage of early NRSfM methods is that the number of basis should be determined explicitly while the optimal number of bases is usually

unknown and is different from sequence to sequence. NRSfM methods using low-rank optimization have been proposed to overcome this problem [9,11].

It was proven that shape alignment also helps to increase the performance of NRSfM [7,20–22]. Procrustean normal distribution (PND) [21] is a powerful framework to separate rigid shape variations from the non-rigid ones. The expectation-maximization-based optimization algorithm applied to PND, EM-PND, showed superior performance to other NRSfM algorithms. Based on this idea, Procrustean Regression (PR) [29] has been proposed to optimize an NRSfM cost function via a simple gradient descent method. In [29], the cost function consists of a data term and a regularization term where low-rankness is imposed not directly on the reconstructed 3D shapes but on the aligned shapes with respect to the reference shape. Any type of differentiable function can be applied for both terms, which has allowed its applicability to perspective NRSfM.

On the other hand, along with recent rise of deep learning, there have been efforts to solve 3D reconstruction problems using CNNs. Object reconstruction from a single image with CNNs is an active field of research. The densely reconstructed shapes are often represented as 3D voxels or depth maps. While some works use ground truth 3D shapes [8,33,39], other works enable the networks to learn 3D reconstruction from multiple 2D observations [10,36,41,42]. The networks used in aforementioned works include a transformation layer that estimates the viewpoint of observations and/or a reprojection layer to minimize the error between input images and projected images. However, they mostly restrict the class of objects to ones that are rigid and have small amounts of deformations within each class, such as chairs and tables.

The 3D interpreter network [40] took a similar approach to NRSfM methods in that it formulates 3D shapes as the weighted sum of base shapes, but it used synthetic 3D models for network training. Warpnet [16] successfully reconstructs 3D shapes of non-rigid objects without supervision, but the results are only provided for birds datasets which have smaller deformations than human skeletons. Tulsiani *et al*. [35] provided a learning algorithm that automatically localize and reconstruct deformable 3D objects, and Kanazawa *et al*. [17] also infer 3D shapes as well as texture information from a single image. Although those methods output dense 3D meshes, the reconstruction is conducted on rigid objects or birds which do not contain large deformations. Our method provides a way to learn 3D structure of non-rigid objects that contain relatively large deformations and pose variations such as human skeletons or faces.

Training a neural network using the loss function based on NRSfM algorithms has been rarely studied. Kong and Lucey [19] proposed to interpret NRSfM as multi-layer sparse coding, and Cha *et al*. [6] proposed to estimate multiple basis shapes and rotations from 2D observations based on a deep neural network. However, they mostly focused on solving SfC problems which have rather small deformations, and the generalization performance of Kong and Lucey [19] is not very good for unseen data with large deformations. Recently, Novotny *et al*. [27] proposed a network structure which factors object deformation and viewpoint changes. Even though many existing ideas in NRSfM are nicely implemented in

[27], this in turn makes the network structure quite complicated. Unlike [27], the 3D shapes are aligned to the mean of aligned shapes in each minibatch in PRN, which enables the use of a simple network structure. Moreover, PRN does not need to set the number of basis shapes explicitly, because it is adjusted automatically in the low-rank loss.

3   Method

We briefly review PR [29] in Sect. 3.1, which is a regression problem based on Procrustes-aligned shapes and is the basis of PRN. Here, we also introduce the concept of "shape transversality" proposed by Novotny *et al.* [27] and prove that a set of Procrustes-aligned shapes is transversal, which means that Procrustes alignment can determine unique motions and eliminate the rigid motion components from reconstructed shapes. The cost function of PRN and its derivatives are explained in Sect. 3.2. The data term and the regularization term for PRN are proposed in Sect. 3.3. Lastly, network structures and training strategy is described in Sect. 3.4.

3.1   Procrustean Regression

NRSfM aims to recover 3D positions of the deformable objects from 2D correspondences. Concretely, given 2D observations of n_p points $\mathbf{U}_i(1 \leq i \leq n_f)$ in n_f frames, NRSfM reconstructs 3D shapes of each frame $\mathbf{X}_i$. PR [29] formulated NRSfM as a regression problem. The cost function of PR consists of data term that corresponds to the reprojection error and the regularization term that minimizes the rank of the aligned 3D shapes, which has the following form:

$$\mathcal{J} = \sum_{i=1}^{n_f} f(\mathbf{X}_i) + \lambda g(\widetilde{\mathbf{X}}, \overline{\mathbf{X}}). \tag{1}$$

Here, $\mathbf{X}_i$ is a $3 \times n_p$ matrix of the reconstructed 3D shapes on the ith frame, and $\overline{\mathbf{X}}$ is a reference shape for Procrustes alignment. $\widetilde{\mathbf{X}}$ is a $3n_p \times n_f$ matrix which is defined as $\widetilde{\mathbf{X}} \triangleq [\mathrm{vec}(\widetilde{\mathbf{X}}_1) \, \mathrm{vec}(\widetilde{\mathbf{X}}_2) \, \cdots \, \mathrm{vec}(\widetilde{\mathbf{X}}_{n_f})]$, where $\mathrm{vec}(\cdot)$ is a vectorization operator. $\widetilde{\mathbf{X}}_i$ is an aligned shape of the ith frame. The aligned shapes are retrieved via Procrustes analysis without scale alignment. In other words, the aligning rotation matrix for each frame is calculated as

$$\mathbf{R}_i = \underset{\mathbf{R}}{\mathrm{argmin}} \, \|\mathbf{R}\mathbf{X}_i\mathbf{T} - \overline{\mathbf{X}}\| \quad \text{s.t.} \quad \mathbf{R}^T\mathbf{R} = \mathbf{I}. \tag{2}$$

Here, $\mathbf{T} \triangleq \mathbf{I}_{n_p} - \frac{1}{n_p}\mathbf{1}_{n_p}\mathbf{1}_{n_p}^T$ is the translation matrix that makes the shape centered at origin. $\mathbf{I}_n$ is an $n \times n$ identity matrix, and $\mathbf{1}_n$ is an all-one vector of size n. The aligned shape of the ith frame becomes $\tilde{\mathbf{X}}_i = \mathbf{R}_i\mathbf{X}_i\mathbf{T}$.

In [29], (1) is optimized for variables $\mathbf{X}_i$ and $\overline{\mathbf{X}}$ and it is shown that their gradients for (1) can be analytically derived. Hence, any gradient-based optimization

method can be applied for large choices of f and g. What the above formulation implies is that we can impose a regularization loss based on the alignment of reconstructed shapes, and therefore, we can enforce certain properties only to non-rigid deformations in which rigid motions are excluded.

To back up the above claim, we introduce the transversal property introduced in [27]:

Definition 1. The set $\mathcal{X}_0 \subset \mathbb{R}^{3 \times n_p}$ has the transversal property if, for any pair $\mathbf{X}, \mathbf{X}' \in \mathcal{X}_0$ related by a rotation $\mathbf{X}' = \mathbf{R}\mathbf{X}$, then $\mathbf{X} = \mathbf{X}'$.

The above definition basically defines a set of shapes that do not contain any non-trivial rigid transforms of its elements, and its elements can be interpreted as having canonical rigid poses. In other words, if two shapes in the set are distinctive, then they should not be identical up to a rigid transform. Here, we prove that the set of Procrustes-aligned shapes is indeed a transversal set. First, we need an assumption: Each shape should have a unique Procrustes alignment w.r.t. the reference shape. This condition might not be satisfied in some cases, *e.g.*, degenerate shapes such as co-linear shapes.

Lemma 1. *A set $\mathcal{X}_P$ of Procrustes-aligned shapes w.r.t. a reference shape $\overline{\mathbf{X}}$ is transversal if the shapes are not degenerate.*

Proof. Suppose that there are $\mathbf{X}, \mathbf{X}' \in \mathcal{X}_P$ that satisfy $\mathbf{X}' = \mathbf{R}\mathbf{X}$. Based on the assumption, $\min_{\mathbf{R}'} \|\mathbf{R}'\mathbf{X}'\mathbf{T} - \overline{\mathbf{X}}\|^2$ will have a unique minimum at $\mathbf{R}' = \mathbf{I}$. Hence, $\min_{\mathbf{R}'} \|\mathbf{R}'\mathbf{R}\mathbf{X}\mathbf{T} - \overline{\mathbf{X}}\|^2$ will also have a unique minimum at the same point, which indicates that $\min_{\mathbf{R}''} \|\mathbf{R}''\mathbf{X}\mathbf{T} - \overline{\mathbf{X}}\|^2$ will have one at $\mathbf{R}'' = \mathbf{R}$. Based on the assumption, $\mathbf{R}''$ has to be $\mathbf{I}$, and hence $\mathbf{R} = \mathbf{I}$.   □

In [27], an arbitrary registration function f is introduced to ensure the transversality of a given set, which is implemented as an auxiliary neural network that has to be trained together with the main network component. We can interpret the Procrustes alignment in this work as a replacement of f that does not need training and has analytic gradients. Accordingly, the underlying network structure of PRN can become much simpler at the cost of a more complicated loss function.

3.2   PR Loss for Neural Networks

One may directly use the gradients of (1) to train neural networks by designing a neural network that estimates both the 3D shapes $\mathbf{X}_i$ and the reference shape $\overline{\mathbf{X}}$. However, the reference shape here incurs some problems when we are to handle it in a neural network. If the class of objects that we are interested in does not contain large deformations, then imposing this reference shape as a global parameter can be an option. On the contrary, if there can be a large deformation, then optimizing the cost function with minibatches of similar shapes or sequences of shapes can be vital for the success of training. In this case, a separate network module to estimate a good 3D reference shape is inevitable. However, designing a network module that estimates mean shapes may make the network structure

more complex and training procedure harder. To keep it concise, we excluded the reference shape from (1) and defined the reference shape as the mean of the aligned output 3D shapes. The mean shape $\overline{\mathbf{X}}$ in (1) is simply replaced with $\sum_{j=1}^{n_f} \mathbf{R}_j \mathbf{X}_j \mathbf{T}$. Now, $\mathbf{X}_i$ is the only variable in the cost function, and the derivative of the cost function with respect to the estimated 3D shapes, $\frac{\partial \mathcal{J}}{\partial \mathbf{X}_i}$, is derived analytically.

The cost function of PRN can be written as follows:

$$\mathcal{J} = \sum_{i=1}^{n_f} f(\mathbf{X}_i) + \lambda g(\widetilde{\mathbf{X}}). \tag{3}$$

The alignment constraint is also changed to

$$\mathbf{R} = \underset{\mathbf{R}}{\operatorname{argmin}} \sum_{i=1}^{n_f} \|\mathbf{R}_i \mathbf{X}_i \mathbf{T} - \frac{1}{n_f} \sum_{j=1}^{n_f} \mathbf{R}_j \mathbf{X}_j \mathbf{T}\| \qquad \text{s.t.} \quad \mathbf{R}_i^T \mathbf{R}_i = \mathbf{I}. \tag{4}$$

where $\mathbf{R}$ is the concatenation of all rotation matrices, *i.e.*, $\mathbf{R} = [\mathbf{R}_1, \mathbf{R}_2, \cdots, \mathbf{R}_{n_f}]$. Let us define $\mathbf{X}$ and $\widetilde{\mathbf{X}}$ as $\mathbf{X} \triangleq [\operatorname{vec}(\mathbf{X}_1), \operatorname{vec}(\mathbf{X}_2), \cdots, \operatorname{vec}(\mathbf{X}_{n_f})]$ and $\widetilde{\mathbf{X}} \triangleq [\operatorname{vec}(\widetilde{\mathbf{X}}_1), \operatorname{vec}(\widetilde{\mathbf{X}}_2), \cdots, \operatorname{vec}(\widetilde{\mathbf{X}}_{n_f})]$ respectively. The gradient of $\mathcal{J}$ with respect to $\mathbf{X}$ while satisfying the constraint (4) is

$$\frac{\partial \mathcal{J}}{\partial \mathbf{X}} = \frac{\partial f}{\partial \mathbf{X}} + \lambda \left\langle \frac{\partial g}{\partial \widetilde{\mathbf{X}}}, \frac{\partial \widetilde{\mathbf{X}}}{\partial \mathbf{X}} \right\rangle, \tag{5}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. $\frac{\partial f}{\partial \mathbf{X}}$ and $\frac{\partial g}{\partial \mathbf{X}}$ are derived once f and g are determined. The derivation process of $\frac{\partial \widetilde{\mathbf{X}}}{\partial \mathbf{X}}$ is analogous to [29]. We explained detailed process in the supplementary material and provide only the results here, which has the form of

$$\frac{\partial \widetilde{\mathbf{X}}}{\partial \mathbf{X}} = (\mathbf{A}\mathbf{B}^{-1}\mathbf{C} + \mathbf{I}_{3n_p n_f})\mathbf{D}. \tag{6}$$

$\mathbf{A}$ is a $3n_p n_f \times 3n_f$ block diagonal matrix expressed as

$$\mathbf{A} = \operatorname{blkdiag}((\mathbf{X}_1'^T \otimes \mathbf{I}_3)\mathbf{L}, (\mathbf{X}_2'^T \otimes \mathbf{I}_3)\mathbf{L}, \cdots, (\mathbf{X}_{n_f}'^T \otimes \mathbf{I}_3)\mathbf{L}), \tag{7}$$

where $\operatorname{blkdiag}(\cdot)$ is the block-diagonal operator, $\otimes$ denotes the Kronecker product. $\mathbf{X}_i'^T = \hat{\mathbf{R}}_i \mathbf{X}_i \mathbf{T}$, where $\hat{\mathbf{R}}_i$ is the current rotation matrix before the gradient evaluation, and $\mathbf{L}$ is a 9×3 matrix that implies the orthogonality constraint of a rotation matrix [29], whose values are

$$\mathbf{L} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T. \tag{8}$$

$\mathbf{B}$ is a $3n_f \times 3n_f$ matrix whose block elements are

$$\mathbf{b}_{ij} = \begin{cases} \mathbf{L}^T(\sum_{k \neq i} \mathbf{X}_k'^T \mathbf{X}_i'^T \otimes \mathbf{I}_3)\mathbf{L} & i = j \\ \mathbf{L}^T(\mathbf{I}_3 \otimes \mathbf{X}_i' \mathbf{X}_j'^T)\mathbf{E}\mathbf{L} & i \neq j \end{cases} \tag{9}$$

where $\mathbf{b}_{ij}$ means the (i,j)-th 3×3 submatrix of $\mathbf{B}$, i and j are integers ranging from 1 to n_f, and $\mathbf{E}$ is a permutation matrix that satisfies $\mathbf{E}\text{vec}(\mathbf{H}) = \text{vec}(\mathbf{H}^T)$. $\mathbf{C}$ is a $3n_f \times 3n_f n_p$ matrix whose block elements are

$$\mathbf{c}_{ij} = \begin{cases} -\mathbf{L}^T(\sum_{k \neq i} \mathbf{X}'_k \otimes \mathbf{I}_3) & i = j \\ -\mathbf{L}^T(\mathbf{I}_3 \otimes \mathbf{X}'_i)\mathbf{E} & i \neq j \end{cases} \tag{10}$$

where $\mathbf{c}_{ij}$ means the (i,j)-th 3×3 submatrix of $\mathbf{C}$. Finally, $\mathbf{D}$ is a $3n_f n_p \times 3n_f n_p$ block-diagonal matrix expressed as

$$\mathbf{D} = \text{blkdiag}(\mathbf{T} \otimes \hat{\mathbf{R}}_1, \mathbf{T} \otimes \hat{\mathbf{R}}_2, \cdots, \mathbf{T} \otimes \hat{\mathbf{R}}_{n_f}). \tag{11}$$

Even though the size of $\partial \widetilde{\mathbf{X}}/\partial \mathbf{X}$ is quite large, *i.e.*, $3n_f n_p \times 3n_f n_p$, we don't actually have to construct it explicitly since the only thing we need is the ability to backpropagate. Memory space and computations can be largely saved based on clever utilization of batch matrix multiplications and reshapes. In the next section, we will discuss about the design of the functions f and g and their derivatives.

3.3   Design of f and g

In PRN, the network produces the 3D position of each joint of a human body. The network output is fed into the cost function, and the gradients are calculated to update the network. For the data term f, we use the reprojection error between the estimated 3D shapes and the ground truth 2D points. We only consider the orthographic projection in this paper, but the framework can be easily extended to the perspective projection. The function f corresponding to the data term has the following form.

$$f(\mathbf{X}) = \sum_{i=1}^{n_f} \frac{1}{2} \|(\mathbf{U}_i - \mathbf{P}_o \mathbf{X}_i) \odot \mathbf{W}_i\|_F^2. \tag{12}$$

Here, $\mathbf{P}_o = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ is an 2×3 orthographic projection matrix, and $\mathbf{U}_i$ is a $2 \times n_p$ 2D observation matrix (ground truth). $\mathbf{W}_i$ is a $2 \times n_p$ weight matrix whose ith column represents the confidence of the position of ith point. $\mathbf{W}_i$ has values between 0 and 1, where 0 means the keypoint is not observable due to occlusion. Scores from 2D keypoint detectors can be used as values of $\mathbf{W}_i$. Lastly, $\|\cdot\|_F$ and $\odot$ denotes the Frobenius norm and element-wise multiplication respectively. The gradient of (12) is

$$\frac{\partial f}{\partial \mathbf{X}} = \sum_{i=1}^{n_f} \mathbf{P}_o^T((\mathbf{P}_o \mathbf{X}_i - \mathbf{U}_i) \odot \mathbf{W}_i \odot \mathbf{W}_i). \tag{13}$$

For the regularization term, we imposed a low-rank constraint to the aligned shapes. Log-determinant or the nuclear norm are two widely used functions and we choose the nuclear norm, *i.e.*,
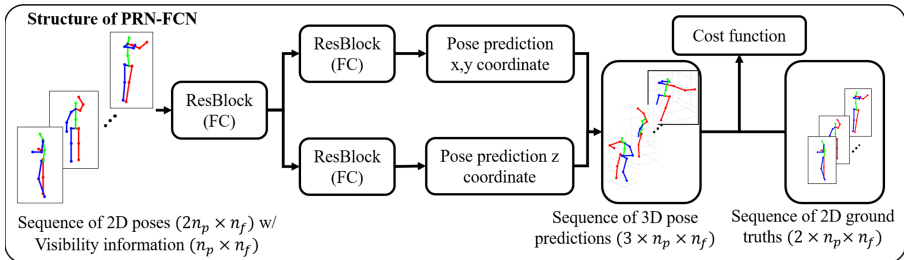
$$g(\widetilde{\mathbf{X}}) = \|\widetilde{\mathbf{X}}\|_*, \tag{14}$$

where $\|\cdot\|_*$ stands for the nuclear norm of a matrix. The subgradient of a nuclear norm can be calculated as

$$\frac{\partial g}{\partial \widetilde{\mathbf{X}}} = \mathbf{U}\mathrm{sign}(\mathbf{\Sigma})\mathbf{V}^T, \tag{15}$$

where $\mathbf{U\Sigma V}^T$ is the singular value decomposition of $\widetilde{\mathbf{X}}$ and $\mathrm{sign}(\cdot)$ is the sign function. Note that the sign function is to deal with zero singular values. $\partial g/\partial \widetilde{\mathbf{X}}_i$ is easily obtained by reordering $\partial g/\partial \widetilde{\mathbf{X}}$.

Fig. 2. Structure of FCNs used in this paper. Structure of CNNs are the same except that ResNet-50 is used as a backbone network instead of a fully connected ResBlock.

3.4 Network Structure

By susbstituting (6), (13), and (15) into (5), the gradient of the cost function of PRN with respect to the 3D shape $\mathbf{X}_i$ can be calculated. Then, the gradient for the entire parameters in the network can also be calculated by back-propagation. We experimented two different structures of PRN in Sect. 4: fully connected networks (FCNs) and convolutional neural networks (CNNs). For the FCN structure, inputs are the 2D point sequences. Each minibatch has a size of $2n_p \times n_f$, and the network produces the 3D positions of the input sequences. We use two stacks of residual modules [13] as the network structure. The prediction parts of x, y coordinates and z coordinates in the network are separated as illustrated in Fig. 2, which achieved better performance in our empirical experience.

For the CNNs, sequences of RGB images are fed into the networks. ResNet-50 [13] is used as a backbone network. The features of the final convolutional layers consisting of 2,048 feature maps of 7×7 size are connected to a network with the same structure as in the previous FCN to produce the final 3D output. We initialize the weights in the convolutional layers to those of the ImageNet [31] pre-trained network. More detailed hyperparameter settings are described in the supplementary material.

4   Experiments

The proposed framework is applied to reconstruct 3D human poses, 3D human faces, and dense 3D human meshes, all of which are the representative types of non-rigid objects. Additional qualitative results and experiments of PRN including comparison with the other methods on the datasets can be found in the supplementary materials.

4.1   Datasets

Human 3.6M [15] contains large-scale action sequences with the ground truth of 3D human poses. We downsampled the frame rate of all sequences to 10 frames per second (fps). Following the previous works on the dataset, we used five subjects (S1, S5, S6, S7, S8) for training, and two subjects (S9, S11) are used as the test set. Both 2D points and RGB images are used for experiments. For the experiments with 2D points, we used ground truth projections provided in the dataset as well as the detection results of a stacked hourglass network [26].

300VW [32] has 114 video clips of faces with 68 landmarks annotations. We used the subset of 64 sequences from the dataset. The dataset is splitted into train and test sets, each of which consists of 32 sequences. 63,205 training images and 60,216 test images are used for the experiment. Since 300-VW dataset only provides 2D annotations and no 3D ground truth data exists, we used the data provided in [5] as 3D ground truths.

SURREAL [37] dataset is used to validate our framework on dense 3D human shapes. It contains 3D human meshes which are created by fitting SMPL body model [23] on CMU Mocap sequences. Each mesh is comprised of 6,890 vertices. We selected 25 sequences from the dataset and split into training and test sets which consist of 5,000 and 2,401 samples respectively. The meshes are randomly rotated around y-axis, and orthographic projection is applied to generate 2D points.

4.2   Implementation Details

The parameter λ is set to $\lambda = 0.05$ for all experiments. The datasets used for our experiment consist of video sequences from fixed monocular cameras. However, most NRSfM algorithms including PRN requires moderate rotation variations in a sequence. To this end, for the Human 3.6M dataset where the sequences are taken by 4 different cameras, we alternately sample the frames or 2D poses from different cameras for consecutive frames. We set the time interval of the samples from different cameras to 0.5 s. Meanwhile, 300-VW dataset does not have multi-view sequences, and each sequence does not have enough rotations. Hence, we randomly sample the inputs in a minibatch from different sequences. For SURREAL datasets, we used 2D poses from consecutive frames.

On the other hand, the rotation alignment is applied to the samples within the same mini-batch. Therefore, if we select the samples in a mini-batch from a

single sequence, the samples within the mini-batch does not have enough variations which affects training speed and performance. To alleviate this problem, we divided a mini-batch into 4 groups and calculated the gradients of the cost function for each group during the training of Human 3.6 M and SURREAL datasets. In addition, since only a small number of different sequences are used in each mini-batch during the training and frames in the same mini-batch are highly correlated as a result, batch normalization [14] may make the training unstable. Hence, we train the networks using batch normalization with moving average for the 70% of the training, and the rest of the iterations are trained with fixed average values as in the test phase.

4.3   Results

The performance of PRN on Human3.6M is evaluated in terms of mean per joint position error (MPJPE) which is the widely used metric in the literature. Meanwhile, we used normalized error as the error metric of 300-VW and SURREAL datasets since the dataset does not provide absolute scales of 3D points. MPJPE and normalized error(NE) are defined as

$$\text{MPJPE}(\hat{\mathbf{X}}_i, \mathbf{X}_i^*) = \frac{1}{n_p} \sum_{j=1}^{n_p} \|\hat{\mathbf{X}}_{ij} - \mathbf{X}_{ij}^*\|, \quad \text{NE}(\hat{\mathbf{X}}_i, \mathbf{X}_i^*) = \frac{\|\hat{\mathbf{X}}_i - \mathbf{X}_i^*\|_F}{\|\mathbf{X}_i^*\|_F}, \quad (16)$$

where $\hat{\mathbf{X}}_i$ and $\mathbf{X}_i^*$ denote the reconstructed 3D shape and the ground truth 3D shape on the ith frame, respectively, and $\hat{\mathbf{X}}_{ij}$ and $\mathbf{X}_{ij}^*$ are the jth keypoint of $\hat{\mathbf{X}}_i$ and $\mathbf{X}_i^*$, respectively. Since orthographic projection has reflection ambiguity, we measure the error also for the reflected shapes and choose the shape that has a smaller error.

To verify the effectiveness of PRN in the fully-connected network architecture (PRN-FCN), we first applied PRN to the task of 3D reconstruction given an input of 2D points. First, we trained PRN-FCN on the Human 3.6M dataset using either ground truth 2D generated by orthographic projection or perspective projection (GT-ortho, GT-persp) or keypoints detected using Stacked hourglass networks (SH). The detailed results for different actions are illustrated in Table 1. For comparison, we also show the results of C3DPO from [27] under the same training setting. As a baseline, we also provide the performance of FCN trained only on the reprojection error (PRN w/o reg). We also trained the neural nets using the 3D shapes reconstructed from existing NRSfM methods, CSF2 [12] and SPM [9] to compare our framework with NRSfM methods. We applied the NRSfM methods to each sequence with the same strides and camera settings as done in training PRN. The trained networks also have the same structure as the one used for PRN.

Here, we can confirm that the regularization term helps estimating depth information more accurately and drops the error significantly. Moreover, PRN-FCN significantly outperforms the NRSfM methods and is also superior to the

Table 1. MPJPE with 2D inputs on Human 3.6M dataset with different 2D inputs: (GT-ortho) Orthographic projection of 3D GT points. (GT-persp) Perspective projection of 3D GT. (SH) 2D keypoint detection results of a stacked hourglass network either from [27] (SH [27]) or from the network fine-tuned on Human 3.6M (SH-FT). PRN-FCN-W used weighted reprojection error based on the keypoint detection score.

Method (GT-ortho)	Direct	Discuss	Eating	Greet	Phone	Pose	Purch	Sitting	SitingD	Smoke	Photo	Wait	Walk	WalkD	WalkT	Avg
PRN w/o reg	138.1	139.7	146.5	145.2	140	127.6	149.4	170.4	188.4	138.3	150.9	133.1	125.6	143.9	139.3	144.8
CSF2 [12] + NN	87.2	90.1	96.1	95.9	102.9	92.1	99.3	129.8	136.7	99.5	120.1	95.2	90.8	102.4	89.2	101.6
SPM [9] + NN	65.3	68.7	82.0	70.1	95.3	65.1	71.9	117.0	136.0	84.3	88.9	71.2	59.5	73.3	68.3	82.3
C3DPO [27]	56.1	55.6	62.2	66.4	63.2	62.0	62.9	76.3	85.8	59.9	88.7	63.3	71.1	70.7	72.3	67.8
PRN-FCN	65.3	58.2	60.5	73.8	60.7	71.5	64.6	79.8	90.2	60.3	81.2	67.1	54.4	61.2	65.6	66.7
Method (GT-persp)	Direct	Discuss	Eating	Greet	Phone	Pose	Purch	Sitting	SitingD	Smoke	Photo	Wait	Walk	WalkD	WalkT	Avg
C3DPO [27]	96.8	85.7	85.8	107.1	86.0	96.8	93.9	94.9	96.7	86.0	124.3	90.7	95.2	93.4	101.3	95.6
PRN-FCN	93.1	83.3	76.2	98.6	78.8	91.7	81.4	87.4	91.6	78.2	104.3	89.6	83.0	80.5	95.3	86.4
Method (SH)	Direct	Discuss	Eating	Greet	Phone	Pose	Purch	Sitting	SitingD	Smoke	Photo	Wait	Walk	WalkD	WalkT	Avg
C3DPO [27]	131.1	137.4	125.2	146.4	143.2	141.4	137.3	141.4	163.8	136.2	161	143.4	145.9	153.2	168.6	145.0
PRN-FCN (SH [27])	127.2	115.1	109.2	130.0	126.9	122.3	116.4	128.4	149.3	117.3	140.7	124.0	123.9	115.3	140.4	124.5
PRN-FCN (SH FT')	100.2	89.4	83.8	105.5	93.0	97.2	89.2	114.0	141.2	89.1	114.8	97.3	91.0	88.3	107.2	99.1
PRN-FCN-W (SH FT')	100.3	88.8	82.8	105.2	91.4	96.7	88.1	102.1	113.2	87.4	115.1	96.5	91.7	87.6	106.4	95.9
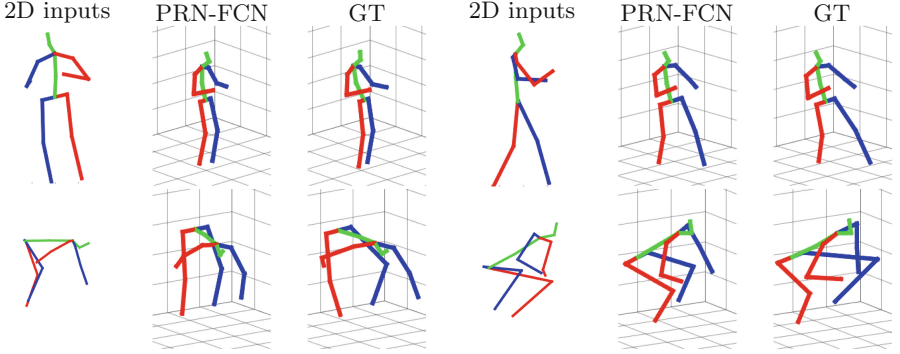
Fig. 3. Qualitative results of PRN-FCN on Human 3.6M dataset. PRN successfully reconstructs 3D shapes from 2D points under various rotations and poses. Left arms and legs are shown in blue, and right arms and legs are shown in red. (Color figure online)

recently proposed work [27] for both ground truth inputs and inputs from keypoint detectors, which proves the effectiveness of the alignment and the low-rank assumption for similar shapes. While PRN-FCN is silghtly better than [27] under orthographic projections, it largely outperforms [27] when trained using 2D points with perspective projections, which indicates that PRN is also robust to the noisy data. The results from the neural networks trained with NRSfM tend to have large variations depending on the types of sequences. This is mainly because the label data comes from NRSfM methods does not show prominent reconstruction results, and this erroneous signal limits the performance of the network in difficult sequences. On the other hand, PRN-FCN robustly reconstruct 3D shapes across all sequences. More interestingly, when the scores of keypoint detectors are used as a weight(PRN-FCN-W), PRN showed improved performance. This result implies that PRN is also robust to inputs with structured missing points since occluded keypoints have lower scores. Although we did not provide the confidence information as input signals, lower weight in the cost function makes the keypoints with lower confidence rely more on the regularization term. As a consequence, PRN-FCN-W performs especially better on the sequences that have complex pose variations such as *Sitting* or *SittingDown*.

Qualitative results for PRN and comparison with the ground truth shapes are illustrated in Fig. 3. It is shown that PRN accurately reconstructs 3D shapes of human bodies from various challenging 2D poses.

Next, we apply PRN to the CNNs to learn the 3D shapes directly from RGB images. MPJPE on the Human 3.6M test set are provided in Table 2. For comparison, we also trained the networks using only the reprojection error and excluding the regularization term in the cost function of PRN (PRN w/o reg). Moreover, we also trained the networks using the 3D shapes reconstructed from existing NRSfM methods, CSF2 [12] and PR [29] since SPM [9] diverged for many sequences in this dataset. Estimating 3D poses from RGB images directly

Table 2. MPJPE with RGB image inputs on the Human 3.6M dataset.

Table 3. Normalized error with 2D inputs on the 300-VW dataset.

Table 4. Normalized error with 2D inputs on the SURREAL dataset.

Method	MPJPE
PRN w/o reg	164.5
CSF2 [12] + CNN	130.6
SPM [9] + CNN	114.4
PRN-CNN	**108.9**
GT 3D	98.8

Method	NE
PRN w/o reg	0.5201
CSF2 [12] + NN	0.2751
PR [29] + NN	0.2730
C3DPO [27]	0.1715
PRN-FCN	**0.1512**
GT 3D	0.0441

Method	NE
PRN w/o reg	0.3565
C3DPO [27]	0.3509
PRN-FCN	**0.1377**

is more challenging than using 2D points as inputs because 2D information as well as depth information should also be learned, and images also contain photometric variations or self-occlusions. PRN largely outperforms the model without regularization term and shows better results than the CNNs trained using NRSfM reconstruction results. It can be observed that the CNN trained with ground truth 3D still has large errors. The performance may be improved if recently-proposed networks for 3D human pose estimation [25,30] is applied here. However, a large network structure reduces the batch size, which can ruin the entire training process of PRN. Therefore, we instead used the largest network we can afford with maintaining the batch size to at least 16. Even though this limits the performance gain due to network structure, we can still compare the results from other similar-sized networks to verify that the proposed training strategy is effective. Qualitative results of PRN-CNN are provided in the supplementary materials.

Next, for the task of 3D face reconstruction, we used the 300-VW dataset [32] which has a video sequence of human faces. We used the reconstruction results from [5] as 3D ground truths. The reconstruction performance is evaluated in terms of normalized error, and the results are illustrated in Table 3. PRN-FCN is also superior to the other methods, including C3DPO [27], in 300-VW datasets. Qualitative results are shown in the two leftmost columns of Fig. 4. Both PRN and C3DPO output plausible results, but C3DPO tends to have larger depth ranges than ground truth depths, which led to increase the normalized errors.

Lastly, we validated the effectiveness of PRN on dense 3D models. Human meshes in SURREAL datasets consist of 6890 3D points for each shape. Since calculating the cost function on dense 3D data imposes heavy computational burden, we subdivided the 3D points into a few groups and compute the cost function for a small set of points. The groups are randomly organized in every iteration. Normalized errors on the SURREAL dataset is shown in Table 4. As it can be seen in Table 4 and the two rightmost columns of Fig. 4, PRN-FCN effectively reconstruct 3D human mesh models from 2D inputs while C3DPO [27] fails to recover depth information.
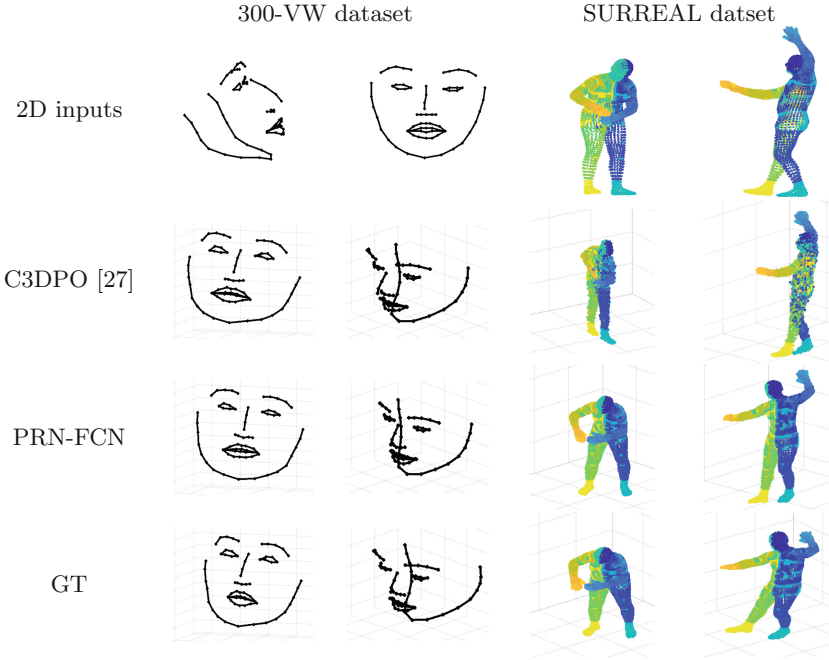
Fig. 4. Qualitative results of PRN-FCN on 300-VW datasets (two leftmost columns) and SURREAL datasets (two rightmost columns).

5   Conclusion

In this paper, a novel framework for training neural networks to estimate 3D shapes of non-rigid objects based on only 2D annotations is proposed. 3D shapes of an image can be rapidly estimated using the trained networks unlike existing NRSfM algorithms. The performance of PRN can be improved by adopting different network architectures. For example, CNNs based on heatmap representations may provide accurate 2D poses and improve reconstruction performance. Moreover, the flexibility for designing the data term and the regularization term in PRN makes it easier to extend the framework to handle perspective projection. Nonetheless, the proposed PRN with simple network structures outperforms the existing state-of-the-art. Although solving NRSfM with deep learning still has some challenges, we believe that the proposed framework establishes the connection between NRSfM algorithms and deep learning which will be useful for future research.

References

1. Agudo, A., Moreno-Noguer, F.: Force-based representation for non-rigid shape and elastic model estimation. IEEE Trans. Pattern Anal. Mach. Intell. **40**(9), 2137–2150 (2018)
2. Akhter, I., Sheikh, Y., Khan, S., Kanade, T.: Trajectory space: a dual representation for nonrigid structure from motion. IEEE Trans. Pattern Anal. Mach. Intell. **33**(7), 1442–1456 (2011)
3. Bartoli, A., Gay-Bellile, V., Castellani, U., Peyras, J., Olsen, S., Sayd, P.: Coarse-to-fine low-rank structure-from-motion. In: IEEE Conference on Computer Vision and Pattern Recognition 2008, CVPR 2008, pp. 1–8. IEEE (2008)
4. Bregler, C., Hertzmann, A., Biermann, H.: Recovering non-rigid 3D shape from image streams. In: IEEE Conference on Computer Vision and Pattern Recognition 2000, Proceedings, vol. 2, pp. 690–696. IEEE (2000)
5. Bulat, A., Tzimiropoulos, G.: How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks). In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1021–1030 (2017)
6. Cha, G., Lee, M., Oh, S.: Unsupervised 3D reconstruction networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3849–3858 (2019)
7. Cho, J., Lee, M., Oh, S.: Complex non-rigid 3D shape recovery using a procrustean normal distribution mixture model. Int. J. Comput. Vis. **117**(3), 226–246 (2016)
8. Choy, C.B., Xu, D., Gwak, J.Y., Chen, K., Savarese, S.: 3D-R2N2: a unified approach for single and multi-view 3D object reconstruction. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 628–644. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_38
9. Dai, Y., Li, H., He, M.: A simple prior-free method for non-rigid structure-from-motion factorization. Int. J. Comput. Vis. **107**(2), 101–122 (2014)
10. Gadelha, M., Maji, S., Wang, R.: 3D shape induction from 2D views of multiple objects. arXiv preprint arXiv:1612.05872 (2016)
11. Garg, R., Roussos, A., Agapito, L.: Dense variational reconstruction of non-rigid surfaces from monocular video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1272–1279 (2013)
12. Gotardo, P.F., Martinez, A.M.: Non-rigid structure from motion with complementary rank-3 spaces. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3065–3072. IEEE (2011)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
14. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37, pp. 448–456. JMLR.org (2015)
15. 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)
16. Kanazawa, A., Jacobs, D.W., Chandraker, M.: WarpNet: weakly supervised matching for single-view reconstruction. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016

17. Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11219, pp. 386–402. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01267-0_23

18. Kong, C., Lucey, S.: Prior-less compressible structure from motion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4123–4131 (2016)

19. Kong, C., Lucey, S.: Deep non-rigid structure from motion. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1558–1567 (2019)

20. Lee, M., Cho, J., Oh, S.: Procrustean normal distribution for non-rigid structure from motion. IEEE Trans. Pattern Anal. Mach. Intell. **39**(7), 1388–1400 (2017). https://doi.org/10.1109/TPAMI.2016.2596720

21. Lee, M., Cho, J., Choi, C.H., Oh, S.: Procrustean normal distribution for non-rigid structure from motion. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1280–1287. IEEE (2013)

22. Lee, M., Choi, C.H., Oh, S.: A procrustean Markov process for non-rigid structure recovery. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1550–1557 (2014)

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

24. Martinez, J., Hossain, R., Romero, J., Little, J.J.: A simple yet effective baseline for 3D human pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2640–2649 (2017)

25. Mehta, D., et al.: VNect: real-time 3D human pose estimation with a single RGB camera. ACM Trans. Graph. (TOG) **36**(4), 44 (2017)

26. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 483–499. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_29

27. Novotny, D., Ravi, N., Graham, B., Neverova, N., Vedaldi, A.: C3DPO: canonical 3D pose networks for non-rigid structure from motion. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 7688–7697 (2019)

28. Paladini, M., Del Bue, A., Stosic, M., Dodig, M., Xavier, J., Agapito, L.: Factorization for non-rigid and articulated structure using metric projections. In: IEEE Conference on Computer Vision and Pattern Recognition 2009, CVPR 2009, pp. 2898–2905. IEEE (2009)

29. Park, S., Lee, M., Kwak, N.: Procrustean regression: a flexible alignment-based framework for nonrigid structure estimation. IEEE Trans. Image Process. **27**(1), 249–264 (2018)

30. Pavlakos, G., Zhou, X., Derpanis, K.G., Daniilidis, K.: Coarse-to-fine volumetric prediction for single-image 3D human pose. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7025–7034 (2017)

31. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y

32. Shen, J., Zafeiriou, S., Chrysos, G.G., Kossaifi, J., Tzimiropoulos, G., Pantic, M.: The first facial landmark tracking in-the-wild challenge: benchmark and results. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 50–58 (2015)

33. Tatarchenko, M., Dosovitskiy, A., Brox, T.: Multi-view 3D models from single images with a convolutional network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 322–337. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_20

34. Torresani, L., Hertzmann, A., Bregler, C.: Nonrigid structure-from-motion: estimating shape and motion with hierarchical priors. IEEE Trans. Pattern Anal. Mach. Intell. **30**(5), 878–892 (2008)

35. Tulsiani, S., Kar, A., Carreira, J., Malik, J.: Learning category-specific deformable 3D models for object reconstruction. IEEE Trans. Pattern Anal. Mach. Intell. **39**(4), 719–731 (2017)

36. Tulsiani, S., Zhou, T., Efros, A.A., Malik, J.: Multi-view supervision for single-view reconstruction via differentiable ray consistency. In: CVPR, vol. 1, p. 3 (2017)

37. Varol, G., et al.: Learning from synthetic humans. In: CVPR (2017)

38. Wang, C., Kong, C., Lucey, S.: Distill knowledge from NRSfM for weakly supervised 3D pose learning. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 743–752 (2019)

39. Wu, J., Wang, Y., Xue, T., Sun, X., Freeman, B., Tenenbaum, J.: MarrNet: 3D shape reconstruction via 2.5 D sketches. In: Advances in Neural Information Processing Systems, pp. 540–550 (2017)

40. Wu, J., et al.: Single image 3D interpreter network. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 365–382. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_22

41. Yan, X., Yang, J., Yumer, E., Guo, Y., Lee, H.: Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In: Advances in Neural Information Processing Systems, pp. 1696–1704 (2016)

42. Zhang, D., Han, J., Yang, Y., Huang, D.: Learning category-specific 3D shape models from weakly labeled 2D images. In: Proceedings of the CVPR, pp. 4573–4581 (2017)

43. Zhou, X., Leonardos, S., Hu, X., Daniilidis, K.: 3D shape estimation from 2D landmarks: a convex relaxation approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4447–4455 (2015)

44. 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