PAUL: Procrustean Autoencoder for Unsupervised Lifting

Chaoyang Wang[1] Simon Lucey[1,2]
[1]Carnegie Mellon University [2]University of Adelaide
{chaoyanw, slucey}@cs.cmu.edu

Abstract

Recent success in casting Non-rigid Structure from Motion (NRSfM) as an unsupervised deep learning problem has raised fundamental questions about what novelty in NRSfM prior could the deep learning offer. In this paper we advocate for a 3D deep auto-encoder framework to be used explicitly as the NRSfM prior. The framework is unique as: (i) it learns the 3D auto-encoder weights solely from 2D projected measurements, and (ii) it is Procrustean in that it jointly resolves the unknown rigid pose for each shape instance. We refer to this architecture as a Procustean Autoencoder for Unsupervised Lifting (PAUL), and demonstrate state-of-the-art performance across a number of benchmarks in comparison to recent innovations such as Deep NRSfM [21] and C3PDO [32].

1. Introduction

Inferring non-rigid 3D structure from multiple unsynchronized 2D imaged observations is an ill-posed problem. Non-Rigid Structure from Motion (NRS*f*M) methods approach the problem by introducing additional priors – of particular note in this regards are low rank [10, 6, 3] and union of subspaces [25, 44] methods.

Recently, NRSfM has seen improvement in performance by recasting the problem as an unsupervised deep learning problem [32, 8, 33]. These 2D-3D *lifting networks* have inherent advantages over classical NRS*f*M as: (i) they are more easily scalable to larger datasets, and (ii) they allow fast feed-forward prediction once trained. These improvement, however, can largely be attributed to the end-to-end reframing of the learning problem rather than any fundamental shift in the prior/constraints being enforced within the NRSfM solution. For example, both Cha *et al*. [8] and Park *et al*. [33] impose a classical low rank constraint on the recovered 3D shape. It is also well understood [10, 21, 44, 25] that such low rank priors have poor performance when applied to more complex 3D shape variations.

The NRSfM field has started to explore new non-rigid shape priors inspired by recent advances in deep learning. Kong & Lucey [21] proposed the use of hierarchical sparsity to have a more expressive shape model while ensuring the inversion problem remains well conditioned. Although achieving significant progress in several benchmarks, the approach is limited by the somewhat adhoc approximations it employs so as to make the entire NRSfM solution realizable as a feed-forward lifting network. Such approximations hamper the interpretability of the method as the final network is a substantial departure from the actually proposed objective. We further argue that this departure from the true objective also comes at the cost of the overall effectiveness of the 2D-3D lifting solution.

In this paper we propose a prior that 3D shapes aligned to a common reference frame are compressible with an undercomplete auto-encoder. This is advantageous over previous linear methods, because the deeper auto-encoder is naturally capable of compressing more complicated non-rigid 3D shapes. What makes learning such an auto-encoder challenging is: (i) it observes only 2D projected measurements of the non-rigid shape; (ii) it must automatically resolve the unknown rigid transformation to align each projected shape instance. We refer to our solution as a *Procustean Autoencoder for Unsupervised Lifting* (PAUL). PAUL is considered unsupervised as it has to handle unknown depth, shape pose, and occlusions. Unlike Deep NRSfM [21], the optimization process of PAUL does not have to be realizable as a feed-forward network – allowing for a solution that stays tightly coupled to the proposed mathematical objective.

We also explore other alternative deep shape priors such as: decoder only and decoder + low-rank. A somewhat similar approach is recently explored by Sidhu *et al*. [34] for dense NRS*f*M. Our empirical results demonstrate the fundamental importance of the auto-encoder architecture for 2D-3D lifting.

Contributions: We make the following contributions:

- We present an optimization objective for joint learning the 2D-3D lifting network and the Procrustean auto-encoder solely from 2D projected measurements.

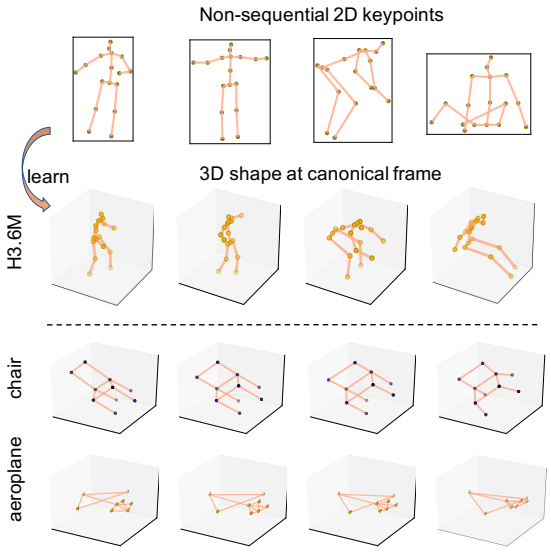


Figure 1: PAUL learns to reconstruct 3D shapes aligned to a canonical frame, using 2D keypoint annotations only. Bottom rows show 3D shapes interpolated from the learned latent space of the Procrustean auto-encoder.

- A naive implementation of PAUL through gradient descent would result in poor local minima, so instead we advocate for a bilevel optimization, whose lower level problem can be efficiently solved by orthographic-N-point (OnP) algorithms.
- Our method achieves state-of-the-art performance across multiple benchmarks, and is empirically shown to be robust against the choice of hyper-parameters such as the dimension of the latent code space.

2. Related Work

Non-rigid structure from motion. NRS*f*M concerns the problem of recovering 3D shapes from 2D point correspondences from multiple images, *without* the assumption of the 3D shape being rigid. It is ill-posed by nature, and additional priors are necessary to guarantee the uniqueness of the solution. We focus our discussion on the type of priors imposed on shape/trajectory:
(i) *low-rank* was advocated by Bregler *et al*. [6] based on the insight that rigid 3D structure has a fixed rank of three [36]. Dai *et al*. [10] proved that the low-lank assumption is standalone sufficient to solve NRS*f*M. It is also applied temporally [12, 3] to constraint 3D trajectories. Kumar [24] recently revisited Dai's approach [10] and showed that by properly utilizing the assumptions that deformation is smooth over frames, it is able to obtain competitive accuracy on benchmarks. However, since the rank is strictly limited by the minimum of the number of points and frames [10], it becomes infeasible to solve large-scale problems with complex shape variations when the number of points is substantially smaller than the number of frames [21].
(ii) *union-of-subspaces* is inspired by the intuition that complex non-rigid deformations could be clustered into a sequence of simple motions [44]. It was extended to spatial-temporal domain [25] and structure from category [2]. The main limitation of using union-of-subspaces is how to effectively cluster deforming shapes from 2D measurements, and how to compute affinity matrix when the number of frames is huge.
(iii) *sparsity* [22, 20, 43], is a more generic prior compared to union-of-subspaces. However, due to the sheer number of possible subspaces to choose, it is sensitive to noise.
(iv) *Procrustean normal distribution* [27] assumes that the 3D structure follows a normal distribution if aligned to a common reference frame. It allows reconstruction without specifying ranks which are typically required by other methods. It was extended temporally as a Procrustean Markov process [29]. Limited by assuming normal distribution, it is less favorable to model deformation which is not Gaussian.
Unsupervised 2D-3D lifting. NRS*f*M can be recast for unsupervised learning 2D-3D lifting. Cha *et al*. [8] use low-rank loss as a learning objective to constraint the shape output of the 2D-3D lifting network. Park *et al*. [33] further modifies Cha's approach by replacing the camera estimation network with an analytic least square solution which aligns 3D structures to the mean shape of a sequence. Due to the inefficiency of low-rank to model complex shape variations, these methods are restricted to datasets with simpler shape variations, or requires temporal order so as to avoid directly handling global shape variations.

Instead of using classical NRS*f*M priors, recent works explore the use of deeper constraints. Generative Adversarial Networks (GANs) [14] are used to enforce realism of 2D reprojections across novel viewpoints [9, 39, 11, 23]. These methods are only applicable for large datasets due to the requirement of learning GANs. It is also unclear how to directly learn GANs with training set existing missing data.

Novotny *et al*. [32] instead enforces self-consistency on the predicted canonicalization of the randomly perturbed 3D shapes. Kong & Lucey [21] proposed the use of hierarchical sparsity as constraint, and approximate the optimization procedure of hierarchical sparse coding as a feed-forward lifting network. It was recently extended by Wang *et al*. [40] to handle missing data and perspective projection. These approaches use complicated network architecture to enforce constraints as well as estimating camera motion, while our method uses simpler constraint formulation, and realized with efficient solution.

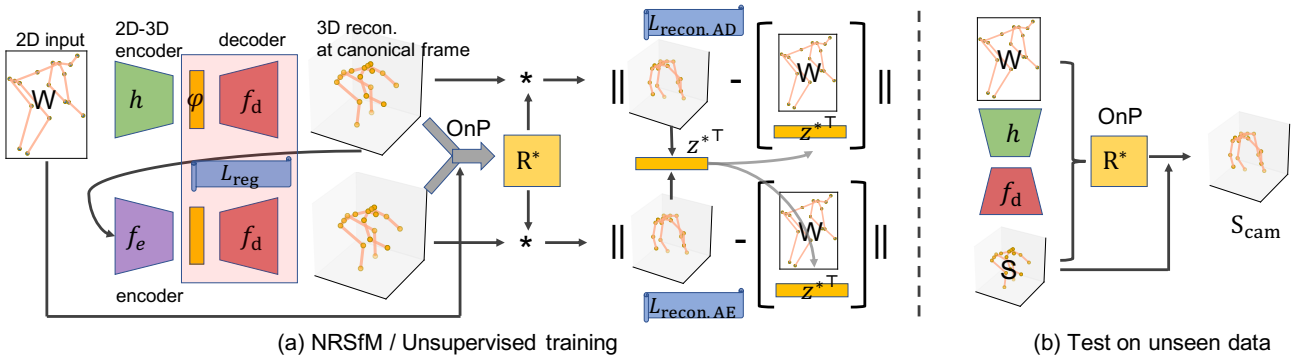


Figure 2: (a) In training, PAUL jointly optimizes the depth value $\mathbf{z}$, camera rotation $\mathbf{R}$ together with the network weights. It is realized through a bilevel optimization strategy, which analytically computes $\mathbf{R}^*$ and $\mathbf{z}^*$ as the solution to an OnP problem. The learning objective is formulated as a combination of reconstruction loss for the decoder-only stream (top row) and the auto-encoder (bottom row) together with regularizer $\mathcal{L}_{\text{reg}}$ applied on the code $\boldsymbol{\varphi}$ and decoder's network weights; (b) in testing, only 2D-3D encoder h and decoder f_{d} are used. Camera rotation is directly estimated by OnP.

3. Preliminary

Problem setup. We are interested in the atemporal setup for unsupervised 2D-3D lifting, which is a general setup that not only works with single deforming objects, but also multiple objects from the same object category. Specifically, given a non-sequential dataset consist of N frames of 2D keypoint locations $\{\mathbf{W}^{(1)}, \dots, \mathbf{W}^{(N)}\}$, where each $\mathbf{W} \in \mathbb{R}^{2\times P}$ represents 2D location for P keypoints, and visibility masks represented as diagonal binary matrices $\{\mathbf{M}^{(1)}, \dots, \mathbf{M}^{(N)}\}$, we want to (i) recover the 3D locations for every keypoints in the dataset, and (ii) train a 2D-3D lifting network capable of making single frame prediction for unseen data.

Weak perspective camera model. We assume weak perspective projections, *i.e.* for a 3D structure $\mathbf{S}$ defined at a canonical frame, its 2D projection is approximated as:

$$\mathbf{W} \approx s\mathbf{R}_{xy}\mathbf{S} + \mathbf{t}_{xy} \tag{1}$$

where $\mathbf{R}_{xy} \in \mathbb{R}^{2\times 3}$, $\mathbf{t}_{xy} \in \mathbb{R}^2$ are the x-y component of a rigid transformation, and $s > 0$ is the scaling factor inversely proportional to the object depth if the true camera model is pin-hole. If all 2D points are visible and centered, $\mathbf{t}_{xy}$ could be omitted by assuming the origin of the canonical frame is at the center of the object. Due to the bilinear form of (1), s is ambiguous and becomes up-to-scale recoverable only when $\mathbf{S}$ is assumed to follow certain prior statistics. A typical treatment to handle scale is to approximate with orthogonal projection by normalizing the scale of $\mathbf{W}$, setting $s = 1$ and leaving $\mathbf{S}$ to be scaled reconstruction.

Regularized auto-encoder (RAE) for S. We assume that the 3D shapes, if aligned to a canonical frame, are compressible by an undercomplete auto-encoder with a low-dimensional bottleneck, *i.e.*

$$\mathbf{S} \approx f_{\text{d}} \circ f_{\text{e}}(\mathbf{S}), \tag{2}$$

where f_{e} is the encoder which maps $\mathbf{S}$ to a K-dimensional latent code $\boldsymbol{\varphi} \in \mathbb{R}^K$, f_{d} is the decoder function and $\circ$ denotes function composition. In this work, we choose deterministic RAE [13] instead of variational auto-encoder (VAE) [19] since RAE is easier to train and still leads to an equally smooth and meaningful latent space. The learning objective for RAE is a combination of reconstruction loss and regularizers on the latent codes as well as the decoder's weights,

$$\mathcal{L}_{\text{RAE}}(\mathbf{x}; \boldsymbol{\theta}_{\text{d}}, \boldsymbol{\theta}_{\text{e}}) = \|f_{\text{d}} \circ f_{\text{e}}(\mathbf{x}) - \mathbf{x}\|_F + \mathcal{L}_{\text{reg}}. \tag{3}$$

where $\boldsymbol{\theta}_{\text{d}}, \boldsymbol{\theta}_{\text{e}}$ are network weights for the auto-encode, $\mathbf{x}$ denote data samples, and the regularizer $\mathcal{L}_{\text{reg}}$ is picked to be $\|\boldsymbol{\varphi}\|_2^2$ and weight decay, which was shown to give comparable performance to VAE when generating images and structured objects [13].

2D-3D lifting network. A 2D-3D lifting network is designed to take input from 2D keypoints and visiblity mask, and outputs 3D keypoint locations. We assume the network architecture is decomposed into two parts (i) a 2D-3D encoder h which maps 2D observations to latent code $\boldsymbol{\varphi}$, and (ii) a decoder f_{d} (reused from the auto-encoder) to generate 3D shapes from $\boldsymbol{\varphi}$. Thus this type of 2D-3D lifting network can be expressed as $f_{\text{d}} \circ h(\mathbf{W}, \mathbf{M})$, which is a general form for network architectures used in literature [21, 30, 32].

4. Learning Procrustean auto-encoder from 2D

For clarity, in this section we simplify the problem by assuming all points are visible, which allows removing trans-

lational component in (1). Description of handling occlusions are given in Sec. 4.3. Fig. 2 illustrates the proposed approach.

4.1. Learning objective

Procrustean auto-encoder. Directly compressing 3D shapes $\mathbf{S}_{\text{cam}}$ at camera frame is inefficient due to the inclusion of the degrees of freedom from camera motion. Therefore, we choose to impose compressibility on $\mathbf{S}$ at canonical frame as shown in (2). However, learning such auto-encoder from 2D observations requires overcoming several obstacles: (i) due to the objects being non-rigid, the definition of canonical frame is statistical and implicitly represented by the unknown rigid transformations to align $\mathbf{S}_{\text{cam}}$'s; (ii) choosing canonical frame requires knowing the statistics of $\mathbf{S}_{\text{cam}}$ which we do not have complete information, since only the first two rows of $\mathbf{S}_{\text{cam}}$ are given as $\mathbf{W}$ representing the x-y coordinates, while the 3rd row $\mathbf{z}$ representing depth values are unknown; (iii) reconstructing $\mathbf{S}_{\text{cam}}$ in turn requires the estimation of the rigid transformation as well as the statistical model of the shape. To overcome these, we propose a joint optimization scheme:

$$\min_{\substack{\theta_e,\theta_d,\\ \{\mathbf{z}^{(i)}\},\{\mathbf{R}^{(i)}\in SO(3)\}}} \sum_{i=1}^{N} \mathcal{L}_{\text{RAE}}({\mathbf{R}^{(i)}}^\top \begin{bmatrix}\mathbf{W}^{(i)}\\ {\mathbf{z}^{(i)}}^\top\end{bmatrix}; \boldsymbol{\theta}_e, \boldsymbol{\theta}_d), \quad (4)$$

where θ_e, θ_d are network weights for the auto-encoder, $\mathbf{R}^\top \begin{bmatrix}\mathbf{W}\\ \mathbf{z}^\top\end{bmatrix}$ computes $\mathbf{S}$ at the canonical frame.

However, (4) is still steps away from being applicable to unsupervised 2D-3D lifting, since it misses the 2D-3D lifting network module in the objective function, and is difficult to optimize due to the inclusion of unknown rotation matrices in the input of the auto-encoder. In the following, we address these by reparameterizing the learning objective and propose an efficient optimization scheme.

Reparameterization for learning 2D-3D lifting. First we introduce an auxiliary variable as the latent code $\boldsymbol{\varphi}$, which satisfies

$$f_d(\boldsymbol{\varphi}) = \mathbf{R}^\top \begin{bmatrix}\mathbf{W}^\top & \mathbf{z}\end{bmatrix}^\top. \quad (5)$$

This leads to transforming (4) to a constrained optimization objective with (5) as the constraint and the input to $f_{\text{d}} \circ f_{\text{e}}$ replaced by $f_{\text{d}}(\boldsymbol{\varphi})$,

$$\min_{\substack{\theta_{\text{e}},\theta_{\text{d}},\\ \{\mathbf{R}^{(i)}\in SO(3)\}\\ \{\mathbf{z}^{(i)},\,\boldsymbol{\varphi}^{(i)}\}}} \sum_{i=1}^{N} \underbrace{\| f_{\text{d}} \circ f_{\text{e}} \circ f_{\text{d}}(\boldsymbol{\varphi}^{(i)}) - {\mathbf{R}^{(i)}}^\top \begin{bmatrix}\mathbf{W}^{(i)}\\ {\mathbf{z}^{(i)}}^\top\end{bmatrix} \|_F}_{\mathcal{L}_{\text{recon. AE}}(\boldsymbol{\varphi},\mathbf{R},\mathbf{z};\theta_{\text{e}},\theta_{\text{d}})} + \mathcal{L}_{\text{reg}},$$
$$\text{s.t. } f_{\text{d}}(\boldsymbol{\varphi}) = \mathbf{R}^\top \begin{bmatrix}\mathbf{W}^\top & \mathbf{z}\end{bmatrix}^\top. \quad (6)$$

Depending on the type of task, $\boldsymbol{\varphi}$ could be either treated as free variables to optimize if the task is to reconstruct the 'training' set as a NRS*f*M problem, or $\boldsymbol{\varphi}$ could be the network output of the 2D-3D encoder *i.e.* $\boldsymbol{\varphi} = h(\mathbf{W};\theta_{\text{h}})$. We then relax the constrained optimization into an unconstrained one, which allows passing gradients to the weights of the 2D-3D encoder h,

$$\min_{\substack{\theta_{\text{h}},\theta_{\text{e}},\theta_{\text{d}},\\ \{\mathbf{z}^{(i)}\},\{\mathbf{R}^{(i)}\in SO(3)\}}} \sum_{i=1}^{N} \mathcal{L}_{\text{recon. AE}}(h(\mathbf{W}^{(i)}), \mathbf{R}^{(i)}, \mathbf{z}^{(i)}) + \| f_{\text{d}} \circ h(\mathbf{W}^{(i)}) - {\mathbf{R}^{(i)}}^\top \begin{bmatrix}\mathbf{W}^{(i)}\\ {\mathbf{z}^{(i)}}^\top\end{bmatrix} \|_F + \mathcal{L}_{\text{reg}}. \quad (7)$$

This loss function could be understood as the combination of the reconstruction losses for both an auto-encoder and an auto-decoder together with the regularizer from RAE, *i.e.* $\mathcal{L}_{\text{recon. AE}} + \mathcal{L}_{\text{recon. AD}} + \mathcal{L}_{\text{reg}}$.

Relation to learning with auto-decoder. We note that an alternative auto-decoder approach with learning objective $\mathcal{L}_{\text{recon. AD}} + \mathcal{L}_{\text{reg}}$ is applicable, the additional $\mathcal{L}_{\text{recon. AE}}$ in our approach is to enforce the existence of a continuous inverse mapping from 3D shape to latent code. This encourages shapes with small variation to stay close in the latent space, which is helpful to learn a meaningful and smoother latent space. We investigate both approaches in Fig. 5 and compare the learnt latent space visually in Fig. 3.

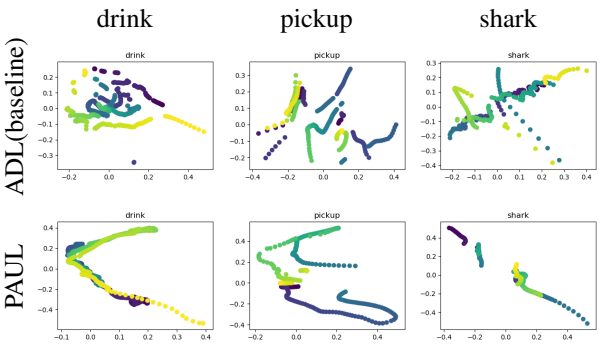
4.2. Efficient bilevel optimization

Directly optimizing (7) with gradient descent is inefficient due to (i) the objective is non-convex and it is prone to poor local minima especially with respect to $\mathbf{R}$. One could use an off-the-shelf NRS*f*M method to provide initialization for $\mathbf{R}$ [34]. However this would make the solution sensitive to the accuracy of the chosen NRS*f*M algorithm. (ii) when using SGD for large datasets, it is problematic to properly update $\mathbf{R}^{(i)}$ and $\mathbf{z}^{(i)}$ if they are left as independent variables. Alternatively, one could introduce additional networks to output $\mathbf{R}$ or $\mathbf{z}$ conditioned on 2D inputs [39, 21]. We find this unnecessary because it introduces extra complexity to solve the problem but is still subject to the inefficiency of gradient descent.

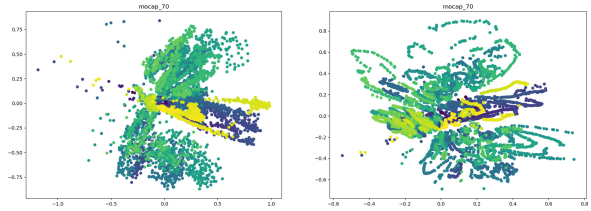
For a more efficient optimization strategy, we propose to first rearrange (7) to an equivalent bilevel objective:

$$\min_{\theta_{\text{h}},\theta_{\text{d}},\theta_{\text{e}}} \sum_{i=1}^{N} \min_{\mathbf{R}^{(i)}\in SO(3),\ \mathbf{z}^{(i)}} \mathcal{L}^{(i)}_{\text{recon. AE}} + \mathcal{L}^{(i)}_{\text{recon. AD}} + \mathcal{L}^{(i)}_{\text{reg}}, \quad (8)$$

The benifit of this rearrangement is that the lower level problem, *i.e.* minimizing the reconstruction losses with respect to $\mathbf{R}$ and $\mathbf{z}$ can be viewed as an extension of the orthographic-N-point (OnP) problem [35], which allows the use of efficient solvers [16, 5, 31]. In addition, if an OnP solver refined by geometric loss is able to converge to local minima, it is not required to be differentiable due to the fact that both lower-level and upper-level problems share



(a) 2D-latent space on **short** sequences with **smooth** camera trajectories.



ADL (baseline) PAUL

(b) 2D-latent space on **long** sequence (CMU-MoCap S70) perturbed with **random** cameras.

Figure 3: Visualization of 2D latent space for ADL & PAUL. Each point represents the 2D latent code recovered for each frame of a sequence. The color of points (from dark blue to bright yellow) indicates the temporal order of points. Ideally, the points should form trajectories in the temporal order. PAUL gives clearer trajectory-like structures in its latent space, while ADL's recovered codes are either more spread-out or form broken trajectories.

the same objective function, thus the gradient is zero at local minima [15]. This would lift the restrictions for the type of solvers we could use for the lower-level problem.

Differentiable fast solver for the lower-level problem. On the other hand, we opt to use an algebraic solution which is computationally more light-weight compared to OnP solvers iteratively minimizing the geometric error. The compromise of using an approximate (*e.g.* algebraic) solution is that, since it does not not necessarily reach local minima, it is required to be implemented as a differentiable operator, which could be easily accomplished via modern autograd packages. The solution we picked is:

1. Find the closed-form least square solution $\tilde{\mathbf{R}}^*$ for minimizing the reprojection error:

$$\min_{\tilde{\mathbf{R}}} \|\tilde{\mathbf{R}}(f_\mathrm{d} \circ f_\mathrm{e} \circ f_\mathrm{d})(\boldsymbol{\varphi}) - \mathbf{W}\|_2^2 + \|\tilde{\mathbf{R}} f_\mathrm{d}(\boldsymbol{\varphi}) - \mathbf{W}\|_2^2. \quad (9)$$

2. Project $\tilde{\mathbf{R}}^*$ to become a rotation matrix $\mathbf{R}^* \in SO(3)$ using SVD.

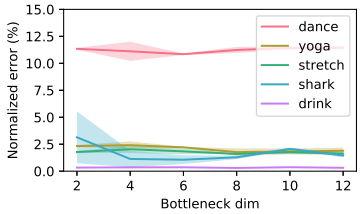


Figure 4: 3D reconstruction error with different bottleneck dimensions. For each configuration, PAUL is run 10 times and visualize with average accuracy (solid lines) together with standard deviation (colored area).

3. $\mathbf{z}^* = \frac{1}{2}((f_\mathrm{d} \circ f_\mathrm{e} \circ f_\mathrm{d})^\top(\boldsymbol{\varphi})\mathbf{r}_z^* + f_\mathrm{d}(\boldsymbol{\varphi})^\top \mathbf{r}_z^*)$, which is the closed-form least square solution for minimizing:

$$\min_{\mathbf{z}} \|(f_\mathrm{d} \circ f_\mathrm{e} \circ f_\mathrm{d})^\top(\boldsymbol{\varphi})\mathbf{r}_z^* - \mathbf{z}\|_2^2 + \|f_\mathrm{d}(\boldsymbol{\varphi})^\top \mathbf{r}_z^* - \mathbf{z}\|_2^2, \quad (10)$$

where $\mathbf{r}_z^{*\top}$ denotes the 3rd row of $\mathbf{R}^*$.

End-to-end training. Finally, with the approximate solution $\mathbf{R}^*$, $\mathbf{z}^*$ for the lower-level problem, the learning objective once again becomes a single level one, which is identical to (7) except that $\mathbf{R}$, $\mathbf{z}$ instead of being free variables, they are now replaced by $\mathbf{R}^*$, $\mathbf{z}^*$ which are differentiable functions conditioned on the network weights $\theta_\mathrm{h}, \theta_\mathrm{e}, \theta_\mathrm{d}$. This allows learning these weights end-to-end via gradient descent.

Prediction on unseen data. To make 3D prediction of a single frame from unseen data, we first use the learned 2D-3D encoder h and the decoder f_d to compute $\mathbf{S}$ at the canonical frame, and then run OnP algorithm [35] to align it to the camera frame.

4.3. Handling missing data

If there exists 2D keypoints missing from the observation due to occlusions or out of image, the translational component in (1) is no longer removable simply by centering the visible 2D points. To avoid reintroducing $\mathbf{t}$ which would complicate derivations, we choose to follow the object centric trick to absorb translation through adaptively normalizing $\mathbf{S}$ according to the visibility mask $\mathbf{M}$ [40]. The normalized $\tilde{\mathbf{S}}$ is computed as:

$$\tilde{\mathbf{S}} = \mathbf{S} + \mathbf{S}(\mathbf{I}_P - \mathbf{M})\mathbf{1}_P\mathbf{1}_P^\top. \quad (11)$$

with this, the projection equation remains bilinear, *i.e.* $\tilde{\mathbf{W}} = \mathbf{R}_{\mathrm{xy}}\tilde{\mathbf{S}}$, where $\tilde{\mathbf{W}}$ denotes the centered $\mathbf{W}$ by the average of visible 2D points. This allows to adapt PAUL to handle missing data with minimal changes. The detailed description is provided in the supp. material.

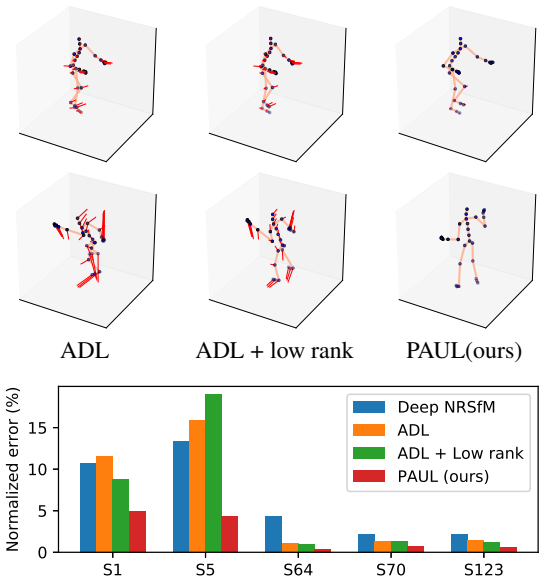


Figure 5: Comparison with auto-decoder baseline (*i.e.* ADL), and low rank constraint (ADL + low-rank) on CMU motion capture dataset. PAUL gives significantly more accurate reconstruction compared to ADL and low-rank. Red line visualizes the difference between reconstructed and groundtruth points.

5. Experiments

5.1. Implementation details

Network architecture. Throughout our experiment, we use the same auto-encoder architecture across datasets except the bottleneck dimension. The number of neural units in each layer is decreased exponentially, *i.e.* $\{256, 128, 64, 32, 16\}$. Ideally, if validation set with 3D groundtruth is provided, we could select optimal architecture based on cross validation. However, due to the unsupervised setting, we rather set the hyperparameters heuristically. We pick a smaller bottleneck dimension, *i.e.* 4 for smaller datasets (*e.g.* synthetic NRSfM benchmarks) or datasets with mostly rigid objects (*e.g.* Pascal3D+), and pick a larger dimension, *i.e.* 8 for articulated objects such as human skeleton (H3.6M, CMU motion capture dataset) and meshes (UP3D). The robustness of our method against variations in hyperparameter settings is investigated in Sec. 5.3.

For the 2D-3D encoder, we experiment with both fully connected residual network [30] and convolutional network [21]. The only modification we make to those architecture is the dimension of their output so as to match the picked bottleneck dimension.

Training details. We keep the same weightings for $\mathcal{L}_{\text{reg}}$ across all experiments, *i.e.*

$$\mathcal{L}_{\text{reg}} = 0.01\|\boldsymbol{\varphi}\|_2^2 + 10^{-4}\|\boldsymbol{\theta}_{\text{d}}\|_2^2. \tag{12}$$

We use the Adam optimizer [18] for training. The optimization parameters are tuned according to specific datasets so as to guarantee convergence.

Evaluation metric. We follow two commonly used evaluation protocols:

(i) *MPJPE* evaluates the mean per-joint position error. To account for the inherent ambiguity from weak perspective cameras, we flip the depth values of the reconstruction if it leads to lower error. To account for the ambiguity in the object distance, we either subtract the average depth values or subtract the depth value of a root keypoint. The latter is used only for H3.6M dataset due to the evaluation convention in literature.

(ii) *Normalized error* (**NE**) evalutes the relative error by: $\|\mathbf{S}_{\text{pred}} - \mathbf{S}_{\text{GT}}\|_F / \|\mathbf{S}_{\text{GT}}\|_F$.

5.2. Baselines

Auto-decoder lifting (ADL). As discussed in Sec. 4.1, an alternative approach for unsupervised lifting is only minimizing the loss $\mathcal{L}_{\text{recon. AD}} + \mathcal{L}_{\text{reg}}$, without the term $\mathcal{L}_{\text{recon. AE}}$ for the auto-encoder. Hence for this baseline, we are only training with respect to the decoder, thus regarded as an auto-decoder approach.

ADL + low rank. In addition, we experiment with adding the low rank constraint as another baseline. Similar to Cha *et al*. [8] and Park *et al*. [33], we evaluate the nuclear norm of the output of the shape decoder as the approximate low rank loss, *i.e.* $\|\mathbf{S}\|_*$. We empirically pick the weighting for the low rank loss as 0.01.

5.3. NRS*f*M experiments

In the first set of experiments, we evaluate the proposed method for the NRS*f*M task, where we report how well the compared methods are able to reconstruct a dataset. The goal is to evaluate the robustness of the proposed Procrustean auto-encoder shape prior across different shape variations, without being convoluted by the inductive bias from a 2D-3D lifting network, which is not the interest of this work. To achieve this, on short sequences, instead of conditioning $\boldsymbol{\varphi}$ with a 2D-3D encoder, we treat $\boldsymbol{\varphi}$ as free variable to optimize directly; and on long sequences, we use the same 2D-3D encoder as in Deep NRS*f*M [21] to have a fair comparison.

NRS*f*M datasets. We report performance on two types of datasets: (i) short sequences with simple object motions, *e.g. drink, pickup, yoga, stretch, dance, shark* which are standard benchmarks used in NRS*f*M literature [4, 37].

(ii) long sequences with large articulated motions, *i.e.* CMU motion capture dataset [1]. We use the processed data from Kong & Lucey [21] which is intentionally made more challenging by inserting large random camera motions.

Robustness against bottleneck dimension. As shown in Fig. 4, we run the methods with varying bottleneck dimen-

	short sequences						long sequences (random cam. motion)				
	drink	pickup	yoga	stretch	dance	shark	S1	S5	S64	S70	S123
#frames	1102	357	307	370	264	240	45025	13773	11621	10788	10788
CNS [28]	3.04	9.18	11.15	7.97	**7.59**	8.32	37.62	40.02	29.00	26.26	26.46
PND [27]	**0.37**	3.72	1.40	1.56	14.54	1.35	-	-	-	-	-
BMM [10]	2.66	17.31	11.50	10.34	18.64	23.11	16.45	14.07	18.13	18.91	19.32
BMM-v2 [24]	1.19	1.98	**1.29**	**1.44**	10.60	5.51	-	-	-	-	-
Deep NRSfM [21]	17.38	**0.53**	12.54	21.63	20.95	21.83	10.74	13.40	4.38	2.17	2.23
PAUL	0.47	2.03	1.71	1.62	10.22	**0.37**	**4.97**	**4.38**	**0.39**	**0.77**	**0.59**

Table 1: Comparison with state-of-the-art NRS*f*M methods on both short sequences and long sequences, report with normalized error. Long sequences are sampled from CMU motion capture dataset [1] with large random camera motion. Atemporal methods are highlighted by orange, methods using temporal information are marked by green. Due to the code for PND and BMM-v2 is unavailable, they are excluded from evaluation on CMU motion capture sequences.

	aero.	car	tv.	sofa	motor.	dining.	chair	bus	bottle	boat	bicycle	train	Mean	8 cls.
C3DPO	6.56	8.21	15.03	7.30	7.48	**3.77**	3.46	20.41	7.48	**7.58**	3.47	33.70	10.4	7.58
Deep NRS*f*M++	7.51	9.22	17.43	9.37	6.18	12.90	3.97	18.02	2.08	9.18	4.03	**23.67**	10.3	8.90
PAUL	**3.99**	**7.13**	**9.88**	**3.99**	**3.74**	5.70	**2.19**	**14.11**	**1.03**	8.08	**1.74**	38.78	**8.4**	**5.32**

Table 2: Per-category normalized error (%) on Pascal3D+ dataset. Follow the protocol of Agudo *et al*. [2], we further report the average error of 8 object categories which are annotated with ≥ 8 keypoints.

	UP3D 79KP	Pascal3D+
avg occlusion %	61.89	37.68
EM-S*f*M [37]	0.107	131.0
GbNRS*f*M [12]	0.093	184.6
Deep NRS*f*M [21]	0.076	51.3
C3DPO [32]	0.067	36.6
Deep NRS*f*M++ [40]	0.062	34.8
PAUL	**0.058**	**30.9**

Table 3: Comparison on datasets with high percentage of missing data. Test accuracy is reported with MPJPE.

	GT pts.	SH pts. [32]
Pose-GAN [23]	130.9	173.2
C3DPO [32]	95.6	153.0
PRN [33]	86.4	124.5
PAUL	88.3	132.5

Table 4: MPJPE on H3.6M validation set. orange indicates atemporal method and green indicates methods use temporal information.

	NE (%)
C3DPO [32]	35.09
PRN [33]	13.77
PAUL (ours)	**12.36**
PAUL (train set)	4.30

Table 5: Test accuracy on SURREAL synthetic sequences. Training error is also reported for PAUL (bottom row).

sion from 2 to 12 on different datasets. To account for the stochastic behavior due to network initialization and gradient descent on small datasets, we run the methods 10 times and visualize with average accuracy (solid lines) together with standard deviation (colored area). PAUL gives stable results once the bottleneck dimension is sufficiently large. This indicates that PAUL is practical for unseen datasets by using an overestimated bottleneck dimension.

Comparison with ADL and low rank. As shown in Fig. 5, on sequences from CMU motion capture dataset, ADL achieves lower error in most sequences when comparing against Deep NRS*f*M, indicating it is indeed a strong baseline. Augmenting ADL with low rank constraint is able to further decrease error for several sequences, but the improvement is not consistent across the whole dataset. In comparison, PAUL gives significant error reduction for all the evaluated sequences, which demonstrates the effectiveness of the proposed Procrustean anto-encoder prior.

Comparison with state-of-the-art NRS*f*M methods. Table 1 collects results from some of the state-of-the-art NRS*f*M methods on the synthetic bechmarks, *e.g*. BMM-v2 [24], CNS [28] and PND [27]. All the well-performing methods utilize temporal information while PAUL does not, but still achieves competitive accuracy on short sequences. On long sequences from CMU motion capture dataset, the accuracy of temporal-based methods *e.g*. CNS deteriorates significantly due to the data perturbed by large random camera motion. Atemporal methods on the other hand gives stable results and PAUL outcompetes all the compared methods by a wide margin.

5.4. 2D-3D lifting on unseen data

We compare against recent unsupervised 2D-3D lifting methods on the processed datasets by Novotny *et al*. [32]:
Datasets. (i) *Synthetic UP-3D* is a large synthetic dataset with dense human keypoints collected from the UP-3D

dataset [26]. The 2D keypoints are generated by orthographic projection of the SMPL body shape with the visibility computed from a ray traccer. Similar to C3DPO, we report result for 79 representative vertices of the SMPL on the test set;

(ii) *Pascal3D+* [41] consists of images from 12 object categories with sparse keypoint annotations. The 3D keypoint groundtruth are created by selecting and aligning CAD models. To ensure consistency between 2D keypoint and 3D groundtruth, the orthographic projections of the aligned 3D CAD models are used as 2D keypoint annotations, and the visibility mask are taken from the original 2D annotations. For a fair comparison against C3DPO, we use the same fully-connected residual network as the 2D-3D encoder, and train a single model to account for all 12 object categories.

(iii) *Human 3.6 Million dataset* (H3.6M) [17] is a large-scale human pose dataset annotated by motion capture systems. Following the commonly used evaluation protocol, the first 5 human subjects (1, 5, 6, 7, 8) are used for training and 2 subjects (9, 11) for testing. The 2D keypoint annotations of H3.6M preserves perspective effect, thus is a realistic dataset for evaluating the practical usage of 2D-3D lifting.

Robustness against occlusion. Both synthetic UP3D and Pascal3D+ dataset simulate realistic occlusions with high occlusion percentage. We focus our comparison against C3DPO and Deep NRS*f*M++ [40] which is a recent update of Deep NRS*f*M for better handling missing data and perspective projections. As shown in Table 3, PAUL significantly outperforms both of them. To account for the distortion caused by the object scale, we switch the evaluation metric from MPJPE to normalized error in Table 2 and report per-class error. PAUL leads with even bigger margin.

Robustness against labeling noise. To work with in-the-wild data, 2D-3D lifting methods are required to be robust against annotation noise, which could be simulated by using 2D keypoints detected by a pretrained keypoint detector. In addition, 2D annotation with perspective effect could also be regarded as noise since it is not modeled by the assumed weak perspective camera model. We evaluate both scenarios on H3.6M dataset (see Table 4). PAUL outperforms the compared atemporal methods (*i.e*. C3DPO and Deep NRS*f*M++) and is competitive to recently proposed PRN [33] which requires training data to be sequential.

5.5. Dense reconstruction

We follow the comparison in Park *et al*. [33] on the synthetic SURREAL dataset [38], which consists of 5k frames with 6890 points for training, and 2,401 frames for testing. Unlike PRN [33] which subsamples a subset of points when evaluating the low rank shape prior due to the intense computational cost of evaluating nuclear norm, our auto-encoder shape prior is computationally cheaper when dealing with dense inputs, thus we made no modification when applying PAUL to SURREAL. As shown in Table 5, PAUL achieves lower test error compared to PRN, even though we use no temporal information in training. It is worth to point out that the current bottleneck in achieving better test accuracy is at the generalization ability of the 2D-3D encoder network, not at the proposed unsupervised training framework. As shown in the last row of Table 5, the reconstruction error on the training set is already much lower than the test error (*i.e*. 4.30% vs 12.36%).

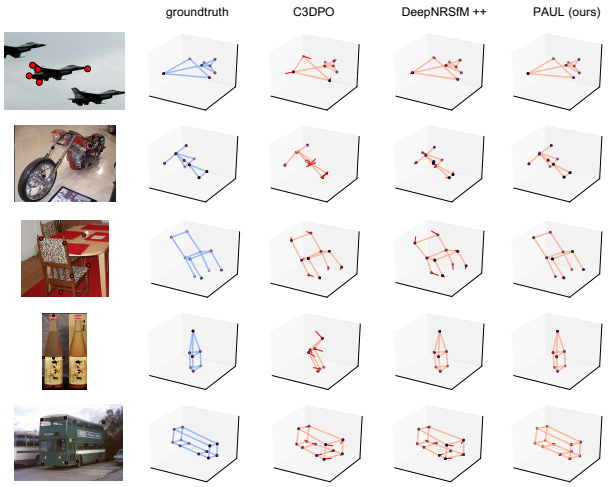


Figure 6: Qualitative comparison on Pascal3D+ dataset. Red lines visualize the difference between groundtruth points and predicted points. PAUL shows more accurate prediction in the compared samples.

6. Conclusion

We propose learning a Procrustean auto-encoder for unsupervised 2D-3D lifting capable of learning from non-sequential 2D observations with large shape variations. We demonstrate that having an auto-encoder performs favorably compared to an alternative auto-decoder approach. The proposed method achieves state-of-the-art accuracy across NRS*f*M and 2D-3D lifting tasks. For future work, theoretical analysis of the characterization of the solution (*e.g*. uniqueness) may help inspire further development. Interpreting the approach as learning manifold may also help provide guidance such as setting hyperparameters [7]. Finally, it is straightforward to extend the method to model perspective projection using similar extensions outlined in [40, 42].

Acknowledgement This work was partially supported by the National Science Foundation under Grant No.1925281.

References

[1] CMU Motion Capture Dataset. available at http://mocap.cs.cmu.edu/. 6, 7

[2] Antonio Agudo, Melcior Pijoan, and Francesc Moreno-Noguer. Image collection pop-up: 3d reconstruction and clustering of rigid and non-rigid categories. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2, 7

[3] Ijaz Akhter, Yaser Sheikh, and Sohaib Khan. In defense of orthonormality constraints for nonrigid structure from motion. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1534–1541. IEEE, 2009. 1, 2

[4] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Nonrigid structure from motion in trajectory space. In *Advances in neural information processing systems*, pages 41–48, 2009. 6

[5] Adam W Bojanczyk and Adam Lutoborski. The procrustes problem for orthogonal stiefel matrices. *SIAM Journal on Scientific Computing*, 21(4):1291–1304, 1999. 4

[6] Christoph Bregler. Recovering non-rigid 3d shape from image streams. Citeseer. 1, 2

[7] Francesco Camastra and Antonino Staiano. Intrinsic dimension estimation: Advances and open problems. *Information Sciences*, 328:26–41, 2016. 8

[8] Geonho Cha, Minsik Lee, and Songhwai Oh. Unsupervised 3d reconstruction networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3849–3858, 2019. 1, 2, 6

[9] Ching-Hang Chen, Ambrish Tyagi, Amit Agrawal, Dylan Drover, Rohith MV, Stefan Stojanov, and James M. Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2

[10] Yuchao Dai, Hongdong Li, and Mingyi He. A simple prior-free method for non-rigid structure-from-motion factorization. *International Journal of Computer Vision*, 107(2):101–122, 2014. 1, 2, 7

[11] Dylan Drover, Rohith MV, Ching-Hang Chen, Amit Agrawal, Ambrish Tyagi, and Cong Phuoc Huynh. Can 3d pose be learned from 2d projections alone? In *The European Conference on Computer Vision (ECCV) Workshops*, September 2018. 2

[12] Katerina Fragkiadaki, Marta Salas, Pablo Arbelaez, and Jitendra Malik. Grouping-based low-rank trajectory completion and 3d reconstruction. In *Advances in Neural Information Processing Systems*, pages 55–63, 2014. 2, 7

[13] P. Ghosh, M. S. M. Sajjadi, A. Vergari, M. J. Black, and B. Schölkopf. From variational to deterministic autoencoders. In *8th International Conference on Learning Representations (ICLR)*, Apr. 2020. 3

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[15] Stephen Gould, Richard Hartley, and Dylan Campbell. Deep declarative networks: A new hope. *arXiv preprint arXiv:1909.04866*, 2019. 5

[16] John C Gower, Garmt B Dijksterhuis, et al. *Procrustes problems*, volume 30. Oxford University Press on Demand, 2004. 4

[17] Catalin Ionescu, Dragos Papava, Vlad Olaru, and Cristian Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013. 8

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *stat*, 1050:1, 2014. 3

[20] Chen Kong and Simon Lucey. Prior-less compressible structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4123–4131, 2016. 2

[21] Chen Kong and Simon Lucey. Deep non-rigid structure from motion. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 3, 4, 6, 7

[22] Chen Kong, Rui Zhu, Hamed Kiani, and Simon Lucey. Structure from category: A generic and prior-less approach. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 296–304. IEEE, 2016. 2

[23] Yasunori Kudo, Keisuke Ogaki, Yusuke Matsui, and Yuri Odagiri. Unsupervised adversarial learning of 3d human pose from 2d joint locations. *arXiv preprint arXiv:1803.08244*, 2018. 2, 7

[24] Suryansh Kumar. Non-rigid structure from motion: Prior-free factorization method revisited. In *Winter Conference on Applications of Computer Vision (WACV 2020)*, 2020. 2, 7

[25] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Multi-body non-rigid structure-from-motion. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 148–156. IEEE, 2016. 1, 2

[26] Christoph Lassner, Javier Romero, Martin Kiefel, Federica Bogo, Michael J. Black, and Peter V. Gehler. Unite the people: Closing the loop between 3d and 2d human representations. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 8

[27] Minsik Lee, Jungchan Cho, Chong-Ho Choi, and Songhwai Oh. Procrustean normal distribution for non-rigid structure from motion. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 1280–1287, 2013. 2, 7

[28] Minsik Lee, Jungchan Cho, and Songhwai Oh. Consensus of non-rigid reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4670–4678, 2016. 7

[29] Minsik Lee, Chong-Ho Choi, and Songhwai Oh. A procrustean markov process for non-rigid structure recovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2

[30] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little. A simple yet effective baseline for 3d human pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2640–2649, 2017. 3, 6

[31] Ab Mooijaart and Jacques JF Commandeur. A general solution of the weighted orthonormal procrustes problem. *Psychometrika*, 55(4):657–663, 1990. 4

[32] David Novotny, Nikhila Ravi, Benjamin Graham, Natalia Neverova, and Andrea Vedaldi. C3dpo: Canonical 3d pose networks for non-rigid structure from motion. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 3, 7

[33] Sungheon Park, Minsik Lee, and Nojun Kwak. Procrustean regression networks: Learning 3d structure of non-rigid objects from 2d annotations. In *European Conference on Computer Vision*, pages 1–18. Springer, 2020. 1, 2, 6, 7, 8

[34] Vikramjit Sidhu, Edgar Tretschk, Vladislav Golyanik, Antonio Agudo, and Christian Theobalt. Neural dense non-rigid structure from motion with latent space constraints. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 4

[35] Carsten Steger. Algorithms for the orthographic-n-point problem. *Journal of Mathematical Imaging and Vision*, 60(2):246–266, 2018. 4, 5

[36] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International journal of computer vision*, 9(2):137–154, 1992. 2

[37] Lorenzo Torresani, Aaron Hertzmann, and Chris Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *IEEE transactions on pattern analysis and machine intelligence*, 30(5):878–892, 2008. 6, 7

[38] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017. 8

[39] Bastian Wandt and Bodo Rosenhahn. Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7782–7791, 2019. 2, 4

[40] Chaoyang Wang, Chen-Hsuan Lin, and Simon Lucey. Deep nrsfm++: Towards 3d reconstruction in the wild. *arXiv preprint arXiv:2001.10090*, 2020. 2, 5, 7, 8

[41] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 75–82. IEEE, 2014. 8

[42] Yinqiang Zheng, Yubin Kuang, Shigeki Sugimoto, Kalle Astrom, and Masatoshi Okutomi. Revisiting the pnp problem: A fast, general and optimal solution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2344–2351, 2013. 8

[43] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Konstantinos G Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4966–4975, 2016. 2

[44] Yingying Zhu, Dong Huang, Fernando De La Torre, and Simon Lucey. Complex non-rigid motion 3d reconstruction by union of subspaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1542–1549, 2014. 1, 2