Deep NRS*f*M++: Towards Unsupervised 2D-3D Lifting in the Wild

Chaoyang Wang[1]   Chen-Hsuan Lin[1]   Simon Lucey[1,2]
[1]Carnegie Mellon University   [2]Argo AI, LLC
{chaoyanw, chlin, slucey}@cs.cmu.edu

Abstract

The recovery of 3D shape and pose from 2D landmarks stemming from a large ensemble of images can be viewed as a non-rigid structure from motion (NRSfM) problem. Classical NRSfM approaches, however, are problematic as they rely on heuristic priors on the 3D structure (e.g. low rank) that do not scale well to large datasets. Learning-based methods are showing the potential to reconstruct a much broader set of 3D structures than classical methods – dramatically expanding the importance of NRSfM to atemporal unsupervised 2D to 3D lifting. Hitherto, these learning approaches have not been able to effectively model perspective cameras or handle missing/occluded points – limiting their applicability to in-the-wild datasets. In this paper, we present a generalized strategy for improving learning-based NRSfM methods [32] to tackle the above issues. Our approach, Deep NRS*f*M++*, achieves state-of-the-art performance across numerous large-scale benchmarks, outperforming both classical and learning-based 2D-3D lifting methods.*

1. Introduction

Non-rigid Structure from Motion (NRS*f*M) aims to reconstruct the 3D structure of a deforming object from 2D keypoint correspondences observed from multiple views [5, 14, 37, 8, 35, 31]. While the object deformation has classically been assumed to occur in time, the vision community has increasingly drawn attention to *atemporal* applications – commonly known as unsupervised 2D-3D lifting. Notable examples include (i) Structure from Category (S*f*C) [33, 2, 3, 18], where the deformation exists as the shape variation within a category-specific object dataset, and (ii) unsupervised human pose estimation [10, 34, 60], which aims to recover the 3D human structure from 2D keypoint annotations. More recently, learning-based methods [44, 32, 9] have shown impressive results for solving atemporal NRS*f*M problems. These models are advantageous because (i) they are scalable to large datasets and (ii) they allow fast feed-forward predictions once trained, with-

out the need of rerunning costly optimization procedures for new samples during inference.

In this regard Deep NRS*f*M [32] is of particular interest. Deep NRS*f*M at its heart is a factorization method that learns a series of dictionaries end-to-end for the joint recovery of 3D shapes and camera poses. Unlike classical NRS*f*M approaches, it incorporates hierarchical block sparsity as the shape prior, which was shown to be more expressible than the popular low-rank assumption [14, 36, 8, 4, 17] and more robust than single-level block sparsity [33, 31]. A key concept in Deep NRS*f*M is its imposition of this prior in the *architectural design* of the network. This allows one to solve the hierarchical block-sparse dictionary learning problem by solely optimizing the reprojection objective end-to-end using deep learning as the machinery. This also makes such structured bias in the network architecture to be interpretable as the optimization procedure used for estimating block-sparse codes. Deep NRS*f*M has shown superior performance on S*f*C and temporally unordered motion capture datasets, outperforming comparable NRS*f*M methods [20, 23, 41, 3] by an order of magnitude.

Despite the recent advances on NRS*f*M, however, two common drawbacks to state-of-the-art NRS*f*M methods [37, 36, 20, 23, 21] still remain: (i) the assumption of an *orthogonal camera model* and (ii) the need for cumbersome post-processing (*e.g.* low-rank matrix completion) to handle *missing data* (keypoints). This makes them impractical for most datasets collected in the wild, where images can exhibit strong perspective effects and structured missing keypoint annotations due to self/out-of-view occlusions. Although perspective effects have been studied in physics-based NRS*f*M [48, 38, 57, 11] where priors such as isometric deformation are enforced, they rarely hold for generic atemporal shapes (*e.g.* object categories in S*f*C) under consideration in this paper. The application of a perspective camera model to factorization methods, on the other hand, is nontrivial. Specifically, the bilinear relationship of 2D observations to the factorized 3D shape and camera matrices in (1) breaks down; as a consequence, the naïve block sparsity prior would no longer mathematically hold.

In this paper, we address the above issues of recent

learning-based NRS*f*M methods and provide a solution to incorporate perspective cameras and handle missing data at a theoretical angle. We focus on Deep NRS*f*M [32] and show that the bilinear factorization relationship in (1) can be preserved by centering the common reference frame to the object center. This seemingly simple step, which draws inspiration from Procrustes analysis [22, 6, 40] and previously applied in Perspective-n-Point (PnP) problem [65], circumvents the need for explicitly modeling translation and projective depths in NRS*f*M. The resulting bilinear factorization induces a novel 2D-3D lifting network architecture which is adaptive to camera models as well as the visibility of input 2D points. We refer to our approach as **Deep NRS*f*M++**, which achieves state-of-the-art results across a myriad of benchmarks. In addition, we demonstrate significant improvements of Deep NRS*f*M++ in reconstruction accuracy by correcting an inherent caveat of Deep NRS*f*M, which made unwarranted relaxations of the block sparsity prior that deviates from the true block-sparse coding objective.

Our contributions are summarized as below:

- We derive a bilinear factorization formulation to incorporate perspective projection and handle missing data in learning-based NRS*f*M methods.

- We propose a solution at the architectural level that keeps a closer mathematical proximity to the hierarchical block-sparse coding objective in NRS*f*M.

- We outperform numerous classical and learning-based NRS*f*M methods and achieve state-of-the-art performance across multiple benchmarks, showing its effectiveness in handling large amounts of missing data under both weak and strong perspective camera models.

2. Related Work

Non-rigid structure from motion. NRS*f*M concerns the problem of reconstructing 3D shapes from 2D point correspondences from multiple images, *without* the assumption of the 3D shape being rigid. For a shape of P points under orthogonal projection, *atemporal* NRS*f*M is typically framed as factorizing the 2D measurements $\mathbf{W} \in \mathbb{R}^{P \times 2}$ as the product of a 3D shape matrix $\mathbf{S} \in \mathbb{R}^{P \times 3}$ and camera rotation matrix $\mathbf{P} \in \mathbb{R}^{3 \times 2}$:

$$\mathbf{W} = \mathbf{SP} \quad \text{subject to } \mathbf{P}^\top \mathbf{P} = \mathbf{I}_2 , \qquad (1)$$

where $\mathbf{I}$ is the identity matrix, and the i-th row of $\mathbf{W}$ and $\mathbf{S}$ corresponds respectively to the image coordinates (u_i, v_i) and world coordinates (x_i, y_i, z_i) of the i-th point.

This factorization problem is ill-posed by nature; in order to resolve the ambiguities in solutions, additional priors are necessary to guarantee the uniqueness of the solution.

These priors include the assumption of shape/trajectory matrices being (i) low-rank [14, 8, 4, 17, 36], (ii) being compressible [33, 31, 66], or (iii) lying in a union of subspaces [37, 67, 3]. Although classical NRS*f*M methods incorporating such priors are mathematically well interpreted, they encounter limitations in large-scale datasets (*e.g.* with a large number of frames). The low-rank assumption becomes infeasible when the data exhibits complex shapes variations. Union-of-subspaces NRS*f*M methods have difficulty clustering shape deformations and estimating affinity matrices effectively. The sparsity prior allows more powerful modeling of shape variations with large number of subspaces but also suffers from sensitivity to noise.

Perspective projection. Most factorization-based NRS*f*M research assumes an orthogonal camera model, but this assumption often breaks down for real-world data, where strong perspective effects may exhibit (*e.g.* when objects are sufficiently close to the camera). Modeling perspective projection thus become necessary for more accurate 3D reconstruction. Sturm & Triggs [52] formulate the rigid S*f*M problem under perspective camera as

$$\text{diag}(z_1, \ldots, z_P)[\mathbf{W} \; \mathbf{1}_P] = [\mathbf{S} \; \mathbf{1}_P]\tilde{\mathbf{P}} , \qquad (2)$$

where z_i is the projective depth of the i-th point and the camera matrix $\tilde{\mathbf{P}} \in \mathbb{R}^{4 \times 3}$ additionally models translation. This is similar to (1) except that the 2D measurements $\mathbf{W}$ is now multiplied by the unknown depth. Since the problem is nonlinear, iterative optimization on the reprojection error is adopted to adjust the projective depth while adding constraints on the rank [52, 45, 13, 56, 26].

This formulation was later extended to solve NRS*f*M. Xiao & Kanade [64] developed a two-step factorization algorithm by recovering the projective depths with subspace constraints and then solving the factorization with orthogonal NRS*f*M. Wang *et al.* [61] proposed to update solutions by gradually increasing the perspectiveness of the camera and recursively refining the projective depth. Hartley & Vidal [24] derived a closed-form linear solution requiring an initial estimation of a multi-focal tensor, which was reported sensitive to noise. Instead of directly solving the problem in (2), we simplify the problem by fully utilizing the object-centric constraint, which has been previously used to remove projective depth and translaton in PnP [65]. This results in a bilinear reformulation which is more compatible to factorization-based methods.

Missing data. Missing (annotated) point correspondences are inevitable in real-world data due to self/out-of-view occlusions of objects. Handling missing data is a nontrivial task not only because the input data is incomplete, but also because the true object center becomes unknown, making naïve centering of 2D data an ineffective strategy for compensating translation. Previous works employ a strategy of

either (i) introducing a visibility mask to exclude missing points from the objective function [15, 20, 33, 62, 32], (ii) recovering the missing 2D points with matrix completion algorithms prior running NRSfM [14, 41, 23, 36], or (iii) treating missing points as unknowns and updating them in an iterative fashion [46]. We follow the first strategy and introduce a visibility mask, but with an additional object-centric constraint [40]. Our key difference to Lee *et al*. [40] is that we (i) derive a bilinear form with dictionaries adaptively normalized to the data; (ii) extend to handle perspective projection which is missing in [40].

Unsupervised 2D-3D lifting The problem of unsupervised 2D-3D lifting, which is equivalent to *atemporal* NRSfM, is recently approached by training neural networks to predict the third (depth) coordinate of each input 2D point. These networks are trained by minimizing the 2D reprojection error, but as with classical NRSfM, this is inherently an ill-posed problem that requires priors or constraints in certain forms. One popular prior is the use of Generative Adversarial Networks (GANs) [19] to enforce realism of 2D reprojections across novel viewpoints [10, 58, 16, 34]. Low-rank priors used in classical NRSfM was also explored as a loss function [9]. The recently proposed C3DPO [44] instead enforces self-consistency on the predicted canonicalization of the same 3D shapes. These methods use black-boxed network architectures without geometric interpretability, and thus their robustness to missing data and perspective effects depends solely on the variations within the data, making learning inefficient. In addition, most of these methods [10, 34, 9, 16, 58] are incapable of handling missing data. We mathematically derive a general framework which is applicable for both orthogonal and perspective cameras, robust to large portion of missing data, and interpretable as solving hierarchical block-sparse dictionary coding.

3. Generalized Bilinear Factorization

A wide range of 2D-3D lifting methods assume a linear model for the 3D shapes to be reconstructed, *i.e.* at canonical coordinates, the vectorization of 3D shape $\mathbf{S}$ in (1), denoted as $\mathbf{s} = \text{vec}(\mathbf{S}) \in \mathbb{R}^{3P}$, can be written as $\mathbf{s} = \mathbf{D}\boldsymbol{\varphi}$, where $\mathbf{D} \in \mathbb{R}^{3P \times K}$ is the shape dictionary with K basis and $\boldsymbol{\varphi} \in \mathbb{R}^K$ is the code vector. Equivalently, this linear model could be writen as $\mathbf{S} = \mathbf{D}^\sharp(\boldsymbol{\varphi} \otimes \mathbf{I}_3)$, where $\mathbf{D}^\sharp$ is a $P \times 3K$ reshape of $\mathbf{D}$ and $\otimes$ denotes Kronecker product. Applying the camera extrinsics (*i.e.* rotation $\mathbf{R} \in \mathbb{SO}(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$) gives the 3D reconstruction in the camera coordinates, written as

$$\mathbf{S}_{\text{cam}} = \mathbf{D}^\sharp(\boldsymbol{\varphi} \otimes \mathbf{R}) + \mathbf{1}_P \mathbf{t}^\top , \qquad (3)$$

where $\mathbf{1}$ is the one vector. Under unsupervised settings, $\mathbf{D}$, $\boldsymbol{\varphi}$, $\mathbf{R}$, and $\mathbf{t}$ are all unknowns and are usually solved under simplified assumptions, *i.e.* complete input 2D points un-

der orthogonal camera projection. In addition, the translational component $\mathbf{t}$ could be removed if the input 2D points were pre-centered under the implicit object-centric assumption that the origin of the canonical coordinates is placed at the object center. This leads to a bilinear factorization problem which classical NRSfM methods [32, 14, 36] employ:

$$\mathbf{W} = \mathbf{D}^\sharp \boldsymbol{\Psi}_{\text{xy}} \quad \text{s.t.} \quad \boldsymbol{\Psi} = \boldsymbol{\varphi} \otimes \mathbf{R} \text{ and } \boldsymbol{\varphi} \in \mathcal{C} , \quad (4)$$

where $\boldsymbol{\Psi}_{\text{xy}} \in \mathbb{R}^{3 \times 2}$ denotes the first two columns of $\boldsymbol{\Psi}$, $\boldsymbol{\Psi} \in \mathbb{R}^{3K \times 3}$ is the block code (as it is a Kronecker product), and $\mathcal{C}$ denotes the prior constraints applied on the code $\boldsymbol{\varphi}$, *e.g.* low rank [14, 38] or (hierarchical) sparsity [33, 31, 32].

In practical settings, however, the presence of either *missing data* (from occlusions) or *perspective projection* would break this bilinear form. With missing data in the 2D points $\mathbf{W}$, the translation $\mathbf{t}$ does not vanish if $\mathbf{W}$ is simply pre-centered by the average of visible points. Moreover, directly applying Strum & Triggs [52] under perspective projection would introduce unknown projective depth z_i, resulting in a non-bilinear form of

$$\text{diag}(z_1, \ldots, z_P)\mathbf{W} = \mathbf{D}^\sharp \boldsymbol{\Psi}_{\text{xy}} + \mathbf{1}_P \mathbf{t}_{\text{xy}}^\top , \qquad (5)$$

which prevents direct application of NRSfM algorithms designed for the orthogonal case. Iterative methods which alternates between solving orthogonal NRSfM and optimizing z_p [61] can be applied, but it is cumbersome and prone to poor local minima. On the other hand, similar issue is also encountered in PnP problems, where the projective depths make the problem nonlinear. A simple and effective solution for PnP is to turn the implicit object-centric assumption into an *explicit* constraint, *i.e.*:

$$\mathbf{t} = \frac{1}{P}\mathbf{S}_{\text{cam}}^\top \mathbf{1}_P , \qquad (6)$$

The introduction of this simple equation allows the projective depth and translation to be eliminated for PnP [65]. Similar tricks also apply to NRSfM to remove z_i and $\mathbf{t}$ in (5), and naturally extends to handle missing data. Hence, we derive a generalized bilinear factorization, which considers both orthogonal and perspective projections, as well as missing data:

$$\mathbf{M}\widetilde{\mathbf{W}} = \mathbf{M}\widetilde{\mathbf{D}}\widetilde{\boldsymbol{\Psi}} \quad \text{s.t.} \quad \mathbf{R} \in \mathbb{SO}(3) \text{ and } \boldsymbol{\varphi} \in \mathcal{C} , \quad (7)$$

where $\mathbf{M} = \text{diag}(m_1, \cdots, m_P)$ is the input binary diagonal matrix indicating visibility of points; and $\widetilde{\mathbf{W}}, \widetilde{\mathbf{D}}, \widetilde{\boldsymbol{\Psi}}$ are the transformed forms of $\mathbf{W}, \mathbf{D}, \boldsymbol{\Psi}$, whose formulation under different settings are summarized in Table 1. Mathematical derivations are provided in Sec. 3.1 for perspective camera and Sec. 3.2 for missing data.

14

	$\widetilde{\mathbf{W}}$		$\widetilde{\mathbf{D}}$		$\widetilde{\mathbf{\Psi}}$	
	formulation	shape	formulation	shape	formulation	shape
orthogonal	$\mathbf{W} - \frac{1}{\widetilde{P}}\mathbf{1}_P\mathbf{1}_P^\top\mathbf{M}\mathbf{W}$	$\mathbb{R}^{P\times 2}$	$\mathbf{D}^\sharp + \frac{1}{\widetilde{P}}\mathbf{1}_P\mathbf{1}_P^\top(\mathbf{I}_P - \mathbf{M})\mathbf{D}^\sharp$	$\mathbb{R}^{P\times 3K}$	$\boldsymbol{\varphi}\otimes\mathbf{R}_{\mathrm{xy}}$	$\mathbb{R}^{3K\times 2}$
perspective	(14)	$\mathbb{R}^{2P\times 1}$	(14)	$\mathbb{R}^{2P\times 9K}$	$\mathrm{vec}(\boldsymbol{\varphi}\otimes\mathbf{R})$	$\mathbb{R}^{9K\times 1}$

Table 1: Summary of the formulations of matrices $\widetilde{\mathbf{W}}$, $\widetilde{\mathbf{D}}$, and $\widetilde{\mathbf{\Psi}}$ under orthogonal and perspective cameras.

3.1. Perspective Camera

We first consider the case where all points are visible. Let (x_i', y_i', z_i') be the 3D coordinates of the i-th point in $\mathbf{SR}$. Since $\mathbf{SR} = \mathbf{D}^\sharp\mathbf{\Psi}$, we can also express (x_i', y_i', z_i') as

$$x_i' = \mathbf{d}_i^\top\boldsymbol{\psi}_x, \quad y_i' = \mathbf{d}_i^\top\boldsymbol{\psi}_y, \quad z_i' = \mathbf{d}_i^\top\boldsymbol{\psi}_z, \qquad (8)$$

where $\mathbf{d}_i^T$ is the i-th row of $\mathbf{D}^\sharp$ and $\boldsymbol{\psi}_x, \boldsymbol{\psi}_y$, and $\boldsymbol{\psi}_z$ are the three columns in $\mathbf{\Psi}$ corresponding to the i-th point. The 3D point (x_i', y_i', z_i'), its 3D translation (t_x, t_y, t_z), and its 2D projection (u_i, v_i) on the unit focal plane are related via

$$x_i' + t_x = u_i(z_i' + t_z) \quad \text{and} \quad y_i' + t_y = v_i(z_i' + t_z), \quad (9)$$

which states that the product of the depth and 2D coordinates is equivalent to the x-y coordinates in 3D. From the object-centric constraint in (6), the translation can be expressed as the mean of back-projection of the 2D points as

$$t_x = \frac{1}{P}\sum_{i=1}^P u_i(z_i' + t_z) \quad \text{and} \quad t_y = \frac{1}{P}\sum_{i=1}^P v_i(z_i' + t_z). \quad (10)$$

Substituting (10) and (8) into (9) and rearranging, we obtain the compact bilinear relationship of $\widetilde{\mathbf{W}} = \widetilde{\mathbf{D}}\widetilde{\mathbf{\Psi}}$, written as

$$\underbrace{\begin{pmatrix} \vdots \\ (u_i - \frac{1}{P}\sum_{j=1}^P u_j)t_z \\ (v_i - \frac{1}{P}\sum_{j=1}^P v_j)t_z \\ \vdots \end{pmatrix}}_{\widetilde{\mathbf{W}}:\text{ normalized 2D projection}} = \underbrace{\begin{pmatrix} \vdots & \vdots & \vdots \\ \mathbf{d}_i^\top & 0 & -u_i\mathbf{d}_i^\top + \frac{1}{P}\sum_{j=1}^P u_j\mathbf{d}_j^\top \\ 0 & \mathbf{d}_i^\top & -v_i\mathbf{d}_i^\top + \frac{1}{P}\sum_{j=1}^P v_j\mathbf{d}_j^\top \\ \vdots & \vdots & \vdots \end{pmatrix}}_{\widetilde{\mathbf{D}}:\text{ normalized dictionary}}\underbrace{\begin{pmatrix}\boldsymbol{\psi}_x \\ \boldsymbol{\psi}_y \\ \boldsymbol{\psi}_z\end{pmatrix}}_{\widetilde{\mathbf{\Psi}}}, \quad (11)$$

where $\widetilde{\mathbf{W}}$ is computed from $\mathbf{W}$ via zero-centering by the mean and rescaling by t_z, the depth of the object center to the camera. Here, t_z is a scalar that normalizes the 2D input and controls the scale of the 3D reconstruction, which is similar to the weak perspective case. $\widetilde{\mathbf{\Psi}}$ becomes a vectorization of $\mathbf{\Psi}$ where the columns are simply concatenated.

Shape scale t_z. As shown in (11), t_z serves the purpose of normalizing the 2D inputs and consequently sets the scale of the shape reconstruction. We note that unlike rigid SfM or trajectory reconstruction, where scale constancy between frames (*e.g.* object staticity or temporal smoothness within

the same dataset) is important, the purpose of "estimating" scale is fundamentally inapplicable to this work. Under the atemporal NRSfM settings, where we deal with non-rigid objects only in the shape space, it is sufficient to obtain a scaled 3D reconstruction, as was in the weak perspective case. This in turn allows us to determine an arbitrary scale t_z for each sample as a preprocessing step to facilitate training. We leave details of determining t_z in practice to Sec. 5.

3.2. Handling Missing Data

Perspective camera. When all the 2D keypoint locations might not be fully available, it becomes insufficient to compute $\mathbf{t}$ from (10). To resolve this, we propose to replace the occluded keypoints with the ones directly from $\mathbf{S}_{\mathrm{cam}}$, *i.e.*

$$t_x = \frac{1}{P}\sum_{i=1}^P \underbrace{m_i u_i(z_i' + t_z)}_{\text{visible points}} + \underbrace{(1 - m_i)(x_i' + t_x)}_{\text{occluded points}}, \quad (12)$$

where m_i indicates the visibility. Rearranging (12), we have

$$t_x = \frac{\sum_{i=1}^P m_i u_i(z_i' + t_z) + (1 - m_i)x_i'}{\sum_{i=1}^P m_i} \quad (13)$$

and similarly for t_y. Substituting the new expressions of the translational components t_x and t_y into (9), we have

$$\mathbf{M}\underbrace{\begin{pmatrix} \vdots \\ (u_i - \frac{\sum_{j=1}^P m_j u_j}{\sum_{j=1}^P m_j})t_z \\ (v_i - \frac{\sum_{j=1}^P m_j v_j}{\sum_{j=1}^P m_j})t_z \\ \vdots \end{pmatrix}}_{\widetilde{\mathbf{W}}:\text{ normalized 2D projection}} = \mathbf{M}\underbrace{\begin{pmatrix} \vdots & \vdots & \vdots \\ \widetilde{\mathbf{d}}_i^\top & 0 & -u_i\mathbf{d}_i^\top + \frac{\sum_{j=1}^P m_j u_j\mathbf{d}_j^\top}{\sum_{j=1}^P m_j} \\ 0 & \widetilde{\mathbf{d}}_i^\top & -v_i\mathbf{d}_i^\top + \frac{\sum_{j=1}^P m_j v_j\mathbf{d}_j^\top}{\sum_{j=1}^P m_j} \\ \vdots & \vdots & \vdots \end{pmatrix}}_{\widetilde{\mathbf{D}}:\text{ normalized dictionary}}\begin{pmatrix}\boldsymbol{\psi}_x \\ \boldsymbol{\psi}_y \\ \boldsymbol{\psi}_z\end{pmatrix}, \quad (14)$$

where $\widetilde{\mathbf{d}}_i = \mathbf{d}_i + \frac{\sum_{j=1}^P(1 - m_j)\mathbf{d}_j}{\sum_{j=1}^P m_j}$.

Orthogonal camera. Handling missing data under orthogonal cameras can be derived using as identical strategy as

$$\mathbf{M}\underbrace{\left(\mathbf{W} - \frac{1}{\widetilde{P}}\mathbf{1}_P\mathbf{1}_P^\top\mathbf{M}\mathbf{W}\right)}_{\widetilde{\mathbf{W}}:\text{ normalized 2D projection}} = \mathbf{M}\underbrace{\left(\mathbf{D}^\sharp + \frac{1}{\widetilde{P}}\mathbf{1}_P\mathbf{1}_P^\top(\mathbf{I}_P - \mathbf{M})\mathbf{D}^\sharp\right)}_{\widetilde{\mathbf{D}}:\text{ normalized dictionary}}\mathbf{\Psi}, \quad (15)$$

where $\widetilde{P}$ is the number of visible points. We leave the derivations to the supplementary material for conciseness. This solution aligns with the common practice employed

15

for data normalization [32, 44, 40]. The difference is that we offset the shape dictionary as well to align the statistics to the shifted 2D inputs.

4. Deep NRSfM++

To solve the generalized bilinear factorization (7) given N tuples of $(\mathbf{W}^{(n)}, \mathbf{M}^{(n)})$ as the dataset (with n indexing the samples), two problems remain to address: (i) how to define heuristics $\mathcal{C}$, which is crucial to have an accurate solution, and (ii) how to formulate the optimization strategy. We choose to follow Deep NRSfM [32], which instead of using simple handcrafted priors (*e.g.* low rank or sparsity), it imposes hierarchical sparsity constraint $\mathcal{C}_\Theta$ with learnable parameters Θ (see Sec. 4.1). The learning strategy of Deep NRSfM is then interpreted as solving a bilevel optimization problem:

$$\min_{\mathbf{D}, \Theta} \sum_{n=1}^{N} \min_{\substack{\varphi^{(n)} \in \mathcal{C}_\Theta \\ \mathbf{R}^{(n)} \in \mathrm{SO}(3)}} \|\mathbf{M}^{(n)}\tilde{\mathbf{W}}^{(n)} - \mathbf{M}^{(n)}\tilde{\mathbf{D}}\tilde{\mathbf{\Psi}}^{(n)}\|_F, \quad (16)$$

where the lower level problems are to solve single frame 2D-3D lifting with given $\mathbf{D}, \Theta$; and the upper level problem is to find optimal $\mathbf{D}, \Theta$ for the whole dataset.

Descent method is employed to solve this bilevel problem. We first approximate the solver of the lower level problem as a feedforward network, *i.e.* $f(\mathbf{M}, \mathbf{W}^{(n)}; \mathbf{D}, \Theta) \mapsto (\mathbf{R}^{*(n)}_{\mathbf{D},\Theta}, \varphi^{*(n)}_{\mathbf{D},\Theta})$. The architecture of the network (as illustrated in Fig. 1) is induced from unrolling one iteration of Iterative Shrinkage and Thresholding Algorithm (ISTA) [7, 50, 25] (see Sec 4.2). Then with $\mathbf{\Psi}^{*(n)}_{\mathbf{D},\Theta} = \varphi^{*(n)}_{\mathbf{D},\Theta} \otimes \mathbf{R}^{*(n)}_{\mathbf{D},\Theta}$, the original bilevel problem is reduced to a single level unconstrained problem, *i.e.*

$$\min_{\mathbf{D}, \Theta} \sum_{n=1}^{N} \|\mathbf{M}^{(n)}\tilde{\mathbf{W}}^{(n)} - \mathbf{M}^{(n)}\tilde{\mathbf{D}}\tilde{\mathbf{\Psi}}^{*}_{\mathbf{D},\Theta}\|_F, \quad (17)$$

which allows the use of solvers such as gradient descent. Finally, with $\mathbf{D}, \Theta$ learned, $f(\mathbf{M}, \mathbf{W}; \mathbf{D}, \Theta)$ is the 2D-3D lifting network applicable to unseen data.

Note: Due to the introduction of multiple levels of dictionaries and codes in Sec. 4.1, we will abuse the notation of $\mathbf{D}, \varphi, \mathbf{\Psi}$ by adding subscript 1, *i.e.* $\mathbf{D}_1, \varphi_1, \mathbf{\Psi}_1$ indicating that they are from the first level of the hierarchy.

4.1. Hierarchical Block-Sparse Coding (HBSC)

Assuming the canonical 3D shapes are compressible via multi-layer sparse coding, the shape code φ_1 is constrained by $\mathcal{C}_\Theta$ as:

$$\varphi_{l-1} = \mathbf{D}_l\varphi_l, \quad \|\varphi_l\|_1 \leq \lambda_l, \quad \varphi_l \geq \mathbf{0}, \quad \forall l \in \{1, \cdots, L\}, \quad (18)$$

where $\mathbf{D}_l \in \mathbb{R}^{K_{l-1} \times K_l}$ are the hierarchical dictionaries, l is the index of hierarchy level, and λ_l is the scalar specifying the amount of sparsity in each level. Thus the learnable parameters for $\mathcal{C}_\Theta$ is $\Theta = \{\cdots, \mathbf{D}_l, \lambda_l, \cdots\}$. Constraints on

multi-layer sparsity not only preserves sufficient freedom on shape variation, but it also results in more constrained code recovery.

Multi-layer sparse coding induces a hierarchical block sparsity constraint on the block codes $\mathbf{\Psi}_l$ (equal to $\varphi_l \otimes \mathbf{R}_{xy}$ if orthogonal projection and $\varphi_l \otimes \mathbf{R}$ if perspective), which leads to a relaxation of the lower level problem in (16):

$$\min_{\{\mathbf{\Psi}_1, \cdots \mathbf{\Psi}_L\}} \|\mathbf{M}\tilde{\mathbf{W}} - \mathbf{M}\tilde{\mathbf{D}}_1\tilde{\mathbf{\Psi}}_1\|_F^2 + \sum_{l=1}^{L} \lambda_l \|\mathbf{\Psi}_l\|_F^{(3 \times a)} \tag{19}$$
$$+ \sum_{l=2}^{L} \|\mathbf{\Psi}_{l-1} - (\mathbf{D}_l \otimes \mathbf{I}_3)\mathbf{\Psi}_l\|_F^2$$

where $\|.\|_F^{(3 \times a)}$ denotes the sum of the Frobenius norm of each $3 \times a$ block. Here, $a = 2$ for orthogonal projections and $a = 3$ for perspective projections.

Remark. Deep NRSfM [32] further relaxes the block sparsity in (19) to L_1 sparsity with a nonnegative constraint to allow for the use of ReLU activations in the network architecture. Such nonnegative constraint is inapplicable to our generalized formulation of $\mathbf{\Psi}$, and we empirically find the use of ReLU activation to degrade performance (Tab. 2).

4.2. HBSC-induced Network Architecture

The 2D-3D lifting network f serves as an approximate solver of the HBSC problem (19). The architecture is induced from unrolling one iteration of the Iterative Shrinkage and Thresholding Algorithm (ISTA) [7, 50, 25], one of the classic methods for sparse coding. Unrolling iterative solver as network architecture has been widely practised to insert inductive bias for better generalization [49, 59, 54, 42, 27, 12, 43]. However, the motivation here is different – it is used to insert learnable priors to constrain an unsupervised learning problem. We provide derivations in the following.

Block soft thresholding. We review the block-sparse coding problem and consider the single-layer case. To reconstruct an input signal $\mathbf{X}$, we solves:

$$\min_{\mathbf{\Psi}} \|\mathbf{X} - \mathbf{D}\mathbf{\Psi}\|_F^2 + \lambda\|\mathbf{\Psi}\|_F^{(3 \times a)} . \tag{20}$$

One iteration of ISTA is computed as

$$\mathbf{\Psi}^{(t+1)} = \mathrm{prox}_{\lambda\|\cdot\|_F^{(3 \times a)}}(\mathbf{\Psi}^{(t)} - \alpha\mathbf{D}^\top(\mathbf{D}\mathbf{\Psi}^{(t)} - \mathbf{X})) , \tag{21}$$

where $\mathrm{prox}_{\lambda\|\cdot\|_F^{(3 \times a)}}$ is the proximal operator for L_1 block sparsity of block size $3 \times a$. Let $\mathbf{\Psi} = [\Pi_1, \Pi_2, \dots \Pi_K]^\top$, where $\Pi_i \in \mathbb{R}^{3 \times a} \, \forall i$. Thus $\mathrm{prox}_{\lambda\|\cdot\|_F^{(3 \times a)}}$ is equivalent to applying *block soft thresholding* (BST) to all Π_i, defined as

$$\mathrm{BST}^{(3 \times a)}(\mathbf{\Psi}; \lambda) = \left[\dots \quad (1 - \frac{\lambda}{\|\Pi_i\|_F})^+ \Pi_i^\top \quad \dots\right]^\top . \tag{22}$$

Assuming the block code $\mathbf{\Psi}$ is initialized to $\mathbf{0}$ with the step size $\alpha = 1$, the first iteration of ISTA can be written as

$$\mathbf{\Psi} = \mathrm{BST}^{(3 \times a)}(\mathbf{D}^\top\mathbf{X}; \lambda) . \tag{23}$$
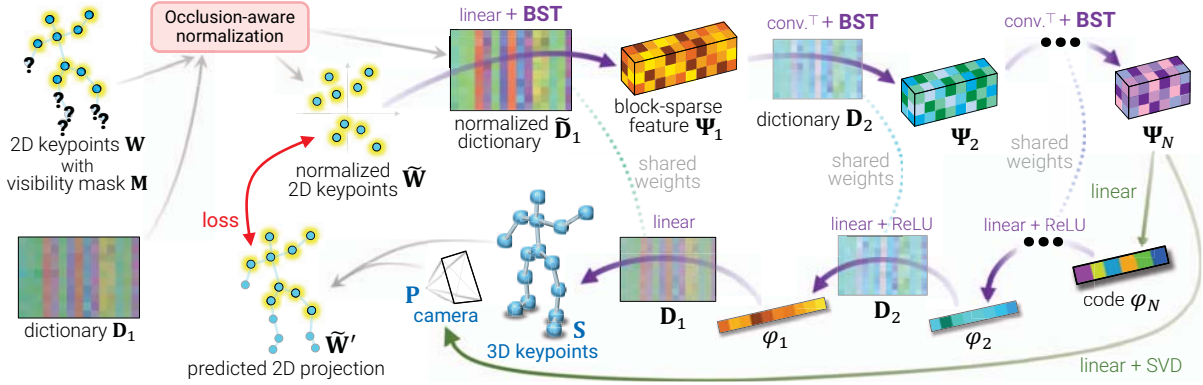
16

Figure 1: **Deep NRSfM++** is a general framework for learning hierarchical block-sparse dictionaries that operates under perspective camera models with the ability to handle missing data. The 2D keypoint input $\mathbf{W}$ and shape dictionary $\mathbf{D}_1$ are normalized according to the visibility mask $\mathbf{M}$ and camera model (Table 1). The encoder-decoder network, derived from hierarchical block-sparse coding, takes the normalized 2D input $\widetilde{\mathbf{W}}$ and jointly predicts the camera matrix $\mathbf{P}$ and the 3D shape $\mathbf{S}$, further reprojected to 2D as $\widetilde{\mathbf{W}}'$. The training objective is to minimize the difference between $\widetilde{\mathbf{W}}'$ and $\widetilde{\mathbf{W}}$.

We interpret BST as solving for the block-sparse code and incorporate $\mathrm{BST}^{(3 \times a)}(\cdot)$ as the nonlinearity in our encoder part of the network, similar to a single-layer ReLU network interpreted as basis pursuit [47]. Our formulation is closer to the true block-sparse coding objective than Deep NRSfM, which uses ReLU as the nonlinearity to *relax* the constraint to L_1 sparsity with nonnegative constraint.

Encoder-decoder network. By unrolling one iteration of block ISTA for each layer, our encoder takes $\widetilde{\mathbf{W}}$ as input and produces the block code for the last layer $\mathbf{\Psi}_L$ as output:

$$
\begin{aligned}
\mathbf{\Psi}_1 &= \mathrm{BST}^{(3 \times a)}\left([\widetilde{\mathbf{D}}_1^\top \widetilde{\mathbf{W}}]_{3K_1 \times a}; \boldsymbol{\lambda}_1\right), \\
\mathbf{\Psi}_2 &= \mathrm{BST}^{(3 \times a)}\left((\mathbf{D}_2 \otimes \mathbf{I}_3)^\top \mathbf{\Psi}_1; \boldsymbol{\lambda}_2\right), \\
&\;\;\vdots \\
\mathbf{\Psi}_L &= \mathrm{BST}^{(3 \times a)}\left((\mathbf{D}_L \otimes \mathbf{I}_3)^\top \mathbf{\Psi}_{L-1}; \boldsymbol{\lambda}_L\right),
\end{aligned}
\tag{24}
$$

where $\boldsymbol{\lambda}_l$ is the learnable threshold for each K_l block and $[\cdot]_{3K_1 \times a}$ is a $3K_1 \times a$ reshape. $(\mathbf{D}_l \otimes \mathbf{I}_3)^T \mathbf{\Psi}_{l-1}$ are implemented by convolution transpose. $\mathbf{\Psi}_L$ are then factorized into $\boldsymbol{\varphi}_L$, $\mathbf{R}$ (constraining to $\mathbb{SO}(3)$ using SVD [32]). The 3D shape $\mathbf{S}$ is recovered from $\boldsymbol{\varphi}_L$ via the decoder as:

$$
\begin{aligned}
\boldsymbol{\varphi}_{L-1} &= \mathrm{ReLU}(\mathbf{D}_L \boldsymbol{\varphi}_L + \mathbf{b}_L), \\
&\;\;\vdots \\
\boldsymbol{\varphi}_1 &= \mathrm{ReLU}(\mathbf{D}_2 \boldsymbol{\varphi}_2 + \mathbf{b}_2), \\
\mathbf{S} &= \mathbf{D}_1 \boldsymbol{\varphi}_1.
\end{aligned}
\tag{25}
$$

Remark. Our key technical differences to Deep NRSfM are: (i) the first layer of the network is adaptive according to the camera model and keypoint visibility; (ii) replacement

of ReLU with BST; (iii) block size of $\mathbf{\Psi}_l$ becomes 3×3 under the perspective camera model.

5. Experiments

Architectural details. The most important hyperparameter of Deep NRSfM++ is the dictionary size K_L at the last level, which depends on the shape variation that exhibits within the dataset. The rest are set arbitrarily and has less affect on performance. Setting K_L to 8 gives reasonable result across all evaluated tasks. For optimal performance by validating on hold-out validation set, we use 8-10 for articulated objects and 2-4 for rigid ones. The detailed description of the architecture and the analysis of the robustness of K_L are included in the Supp.

Shape scale t_z. To set the scale t_z in practice, we make use of available 2D information such as bounding boxes; in applications where relative scales between samples in a dataset is available (*e.g.* human skeletons), we can utilize a strategy to estimate and recorrect the scale t_z in an iterative fashion. We use the detected 2D object bounding box to provide an initial estimate of t_z and subsequently update the scale estimation (using the Frobenius norm of $\mathbf{S}$ or the average bone length of a skeleton model, if available). Once we have updated the scale estimation t_z, we rerun Deep NRSfM++ and update the reconstruction. This **scale correction** procedure allows the 3D reconstruction and scale estimation to improve each other.

Evaluation metrics. We employ the following metrics to evaluate the accuracy of 3D reconstruction. **MPJPE**: before calculating the mean per-joint position, we normalize the scale of the prediction to match against ground truth (GT). To account for the ambiguity from weak perspec-

17

Subject	CNS	NLO	SPS	Deep NRSfM	ReLU	BST
1	0.613	1.22	1.28	0.175	0.265	**0.112**
5	0.657	1.160	1.122	**0.220**	0.393	0.230
18	0.541	0.917	0.953	0.081	0.117	**0.076**
23	0.603	0.998	0.880	0.053	0.093	**0.048**
64	0.543	1.218	1.119	0.082	0.179	**0.020**
70	0.472	0.836	1.009	0.039	0.030	**0.019**
106	0.636	1.016	0.957	**0.113**	0.364	0.116
123	0.479	1.009	0.828	0.040	0.040	**0.020**

Table 2: 3D reconstruction error on CMU Motion Capture compared with NRSfM methods: CNS [41], NLO [15], SPS [31] and Deep NRSfM [32]. "ReLU" and "BST" are Deep NRSfM reimplemented with different nonlinearities.

tive cameras, we flip the depth value of the prediction if it leads to lower error. **PA-MPJPE**: rigid align the prediction to GT before evaluating MPJPE. **STRESS**: borrowed from Novotny *et al.* [44] is a metric invariant to camera pose and scale. **Normalized 3D error**: 3D error normalized by the scale of GT, used in prior NRSfM works [5, 14, 32, 20].

BST vs ReLU. To study the effect of replacing ReLU with BST as the nonlinearity, we compare our approach against Deep NRSfM [32] on orthogonal projection data with perfect point correspondences on the CMU motion capture dataset [1]. Normalized 3D error is reported per human subject in Table 2. Our approach using BST achieves better accuracy compared to Deep NRSfM, providing empirical benefits of its closer proximity to solving the true block sparse coding objective.

Orthogonal projection with missing data. We evaluate Deep NRSfM++ on two benchmarks with high amount of missing data (Table 3): (1) Synthetic UP-3D, a large synthetic dataset with dense human keypoints based on the UP-3D dataset [39]. The data was generated by orthographic projections of the SMPL body shape with the visibility computed from a ray tracer. We follow the same settings as C3DPO [44] and evaluate the 3D reconstruction of 79 representative vertices of the SMPL model on the test set. (2) PASCAL3D+ [63] consists of images of 12 rigid object categories with sparse keypoints annotations. To ensure consistency between 2D keypoint and 3D ground truth, we follow C3DPO and use the orthographic projections of the aligned CAD models with the visibility taken from the original 2D annotations. A single model is trained to account for all 12 object categories. We also include results where the 2D keypoints are detected by HRNet [53].

Our method achieves over 32% error reduction over Deep NRSfM (Table 3(b)) while comparing favorably against other NRSfM methods and deep learning methods such as C3DPO.

Perspective projection. We evaluate our approach on two datasets with strong perspective effects: (1) Human 3.6M [28], a large-scale human pose dataset annotated by motion capture systems. We closely follow the commonly used evaluation protocol: we use 5 subjects (1, 5, 6, 7, 8) for training and 2 subjects (9, 11) for testing. (2) Apollo-Car3D [51] has 5277 images featuring cars, where each car instance comes with annotated 3D pose. 2D keypoint annotations are also provided without 3D ground truth. To evaluate our method, we render 2D keypoints by projecting 34 car models according to the 3D pose labels. Visibility of each keypoints are marked according to the original 2D keypoint annotations. To showcase strong perspective effects, we select cars within 30 meters with no less than 25 visible points (out of 58 in total), which gives us 2848 samples for training and 1842 for testing.

We evaluate different variants of our approach. We find that modeling perspective projection (Deep NRSfM++ persp.) leads to significant improvement over the orthogonal model (Deep NRSfM++ ortho.) and applying scale correction further improves accuracy. Deep NRSfM++ shows robustness under different level of noise and occlusion (see Table 5) and achieves the best result compared to other unsupervised learning method. We outperform the leading GAN-based method [10] by a significant margin when using the same training set (50.9 v.s. 58); in addition, Chen *et al.* [10] reaches our level of performance (50.9 vs 51) only when external training sources and temporal constraints are used. We provide qualitative results in Fig. 2, which shows the benefit of Deep NRSfM++'s ability to model perspective camera models while also outperforming C3DPO.

Deforming object in short sequences. Our method is applicable to NRSfM problems with limited number of frames. We experiment on **NRSfM Challenge Dataset** [29], which consists of 5 deforming objects captured with six different camera paths. While the leading methods all utilize trajectory information, our atemporal approach still gives reasonable reconstruction as shown in Fig. 6. The detailed comparison is in the supp. material.

6. Conclusion

We propose an atemporal approach applicable to SfC and unsupervised pose estimation. It provides an unified framework to model both orthogonal and perspective camera as well as missing data. The **limitations** of this work is: (i) the hierarchical sparsity prior is less robust to articulated objects when only limited number of frames is available (see Supp.). (ii) the connection between the proposed learning framework and the true objective of the NRSfM problem is still an approximation. Further theoritical investigation is in need to understand the benifit of this approximation or any possible alternatives, but is out of the scope of this paper.

18

Method	MPJPE	STRESS
EM-SfM [55]	0.107	0.061
GbNRSfM [17]	0.093	0.062
Deep NRSfM [32]	0.076	0.063
C3DPO-base [44]	0.160	0.105
C3DPO [44]	0.067	0.040
Deep NRSfM++	**0.062**	**0.037**

(a) Test error on Synthetic UP-3D.

Method	MPJPE	STRESS
EM-SfM [55]	131.0	116.8
GbNRSfM [17]	184.6	111.3
Deep NRSfM [32]	51.3	44.5
C3DPO-base [44]	53.5	46.8
C3DPO [44]	36.6	31.1
Deep NRSfM++	**34.8**	**27.9**

(b) Test error on PASCAL3D+, where the input keypoints are from the ground truth.

Method	MPJPE	STRESS
Deep NRSfM [32]	65.3	47.7
CMR [30]	74.4	53.7
C3DPO [44]	57.5	41.4
Deep NRSfM++	**53.0**	**36.1**

(c) Test error on PASCAL3D+, where the input keypoints are off-the-shelf keypoint detection results from HRNet [53].

Table 3: Quantitative results on data generated by orthogonal projection with realistic occlusions.

Method	KSTA [20]	CNS [41]	Pose-GAN [34]	C3DPO [44]	Chen *et al.* [10]	Deep NRSfM++ ortho.	persp.	persp. + scale corr.
MPJPE	-	120.1	130.9	95.6	-	104.2	60.5	**56.6**
PA-MPJPE	123.6	79.6	-	-	58	72.9	51.8	**50.9**

(a) Test error on Human 3.6M compared against unsupervised methods. We report our results under both weak perspective and perspective camera models and demonstrate the effectiveness of applying scale corrections (2 iterations).

		w/o missing pts.		w/ missing pts.	
	method	train	test	train	test
	CNS [41]	1.30	-	-	-
	KSTA [20]	1.58	-	1.62	-
Deep NRSfM++	ortho.	0.596	0.591	0.679	0.681
	persp.	0.152	0.145	0.182	0.185
	+ scale correction	0.131	0.124	0.165	0.168

(b) Test error on ApolloCar3D, where we report the MPJPE in meters as the evaluation metric.

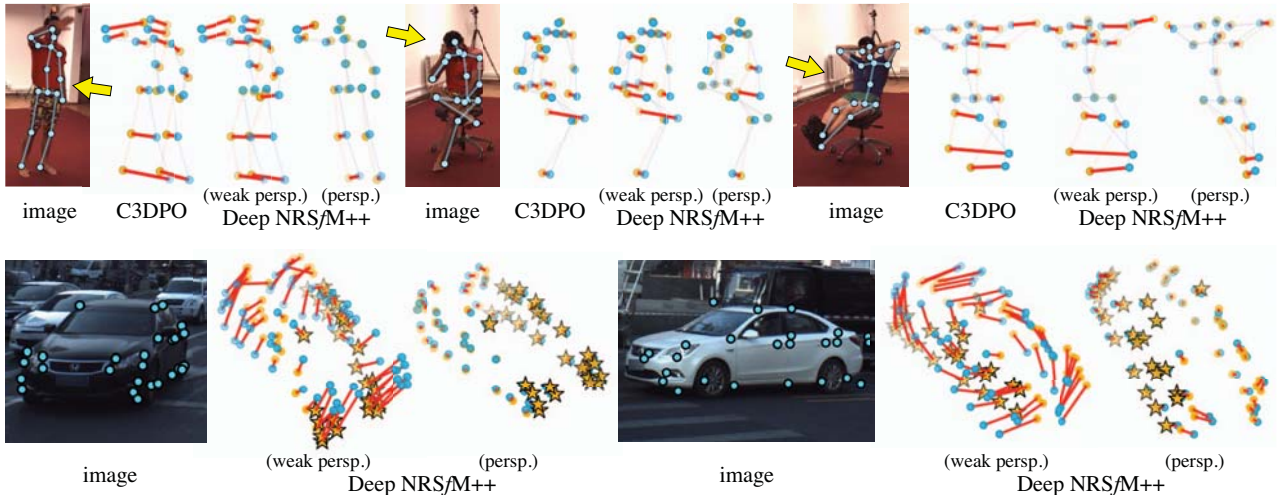Table 4: Quantitative results on real-world data with perspective projection.

Figure 2: Qualitatative comparison. Blue points: GT, yellow points: prediction (stars indicate visible annotations), red lines: prediction error. The two rows are visual results from Human 3.6M and ApolloCar3D datasets respectively.

Noise (σ)	Missing pts. (%) 0	30	60
0	0.124	0.142	0.192
3	0.129	0.144	0.205
5	0.136	0.150	0.202
10	0.125	0.166	0.181
15	0.191	0.188	0.304

Table 5: Robustness with input under different occlusion rates and noises on ApolloCar3D.
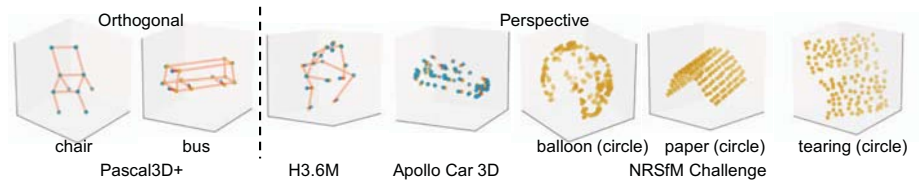

Table 6: Qualitative results across different datasets. Blue points: ground truth; yellow points and orange lines: reconstruction; red lines: difference between ground truth and reconstruction. Best viewed in color and zoomed in.

References

[1] Cmu motion capture dataset. available at http://mocap.cs.cmu.edu/. 7

[2] A. Agudo and F. Moreno-Noguer. Recovering pose and

3d deformable shape from multi-instance image ensembles. In *Asian Conference on Computer Vision*, pages 291–307. Springer, 2016. 1

[3] A. Agudo, M. Pijoan, and F. 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. 1, 2

[4] I. Akhter, Y. Sheikh, and S. 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

[5] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Nonrigid structure from motion in trajectory space. In *Advances in neural information processing systems*, pages 41–48, 2009. 1, 7

[6] A. Bartoli, D. Pizarro, and M. Loog. Stratified generalized procrustes analysis. *International journal of computer vision*, 101(2):227–253, 2013. 2

[7] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. Citeseer. 5

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

[9] G. Cha, M. Lee, and S. Oh. Unsupervised 3d reconstruction networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3849–3858, 2019. 1, 3

[10] C.-H. Chen, A. Tyagi, A. Agrawal, D. Drover, R. MV, S. Stojanov, and J. M. Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 3, 7, 8

[11] A. Chhatkuli, D. Pizarro, T. Collins, and A. Bartoli. Inextensible non-rigid structure-from-motion by second-order cone programming. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2428–2441, 2017. 1

[12] N. Chodosh, C. Wang, and S. Lucey. Deep convolutional compressed sensing for lidar depth completion. In *Asian Conference on Computer Vision*, pages 499–513. Springer, 2018. 5

[13] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(11):1098–1104, 1996. 2

[14] Y. Dai, H. Li, and M. 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, 3, 7

[15] A. Del Bue, F. Smeraldi, and L. Agapito. Non-rigid structure from motion using ranklet-based tracking and non-linear optimization. *Image and Vision Computing*, 25(3):297–310, 2007. 3, 7

[16] D. Drover, R. MV, C.-H. Chen, A. Agrawal, A. Tyagi, and C. Phuoc Huynh. Can 3d pose be learned from 2d projections alone? In *The European Conference on Computer Vision (ECCV) Workshops*, September 2018. 3

[17] K. Fragkiadaki, M. Salas, P. Arbelaez, and J. Malik. Grouping-based low-rank trajectory completion and 3d reconstruction. In *Advances in Neural Information Processing Systems*, pages 55–63, 2014. 1, 2, 8

[18] Y. Gao and A. L. Yuille. Symmetric non-rigid structure from motion for category-specific object structure estimation. In *European Conference on Computer Vision*, pages 408–424. Springer, 2016. 1

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 3

[20] P. F. Gotardo and A. M. Martinez. Kernel non-rigid structure from motion. In *2011 International Conference on Computer Vision*, pages 802–809. IEEE, 2011. 1, 3, 7, 8

[21] P. F. Gotardo and A. M. Martinez. Non-rigid structure from motion with complementary rank-3 spaces. In *CVPR 2011*, pages 3065–3072. IEEE, 2011. 1

[22] J. C. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975. 2

[23] O. C. Hamsici, P. F. Gotardo, and A. M. Martinez. Learning spatially-smooth mappings in non-rigid structure from motion. In *European Conference on Computer Vision*, pages 260–273. Springer, 2012. 1, 3

[24] R. Hartley and R. Vidal. Perspective nonrigid shape and motion recovery. In *European Conference on Computer Vision*, pages 276–289. Springer, 2008. 2

[25] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005. 5

[26] A. Heyden, R. Berthilsson, and G. Sparr. An iterative factorization method for projective structure and motion from image sequences. *Image and Vision Computing*, 17(13):981–991, 1999. 2

[27] P. Hu and D. Ramanan. Bottom-up and top-down reasoning with hierarchical rectified gaussians. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 5

[28] C. Ionescu, D. Papava, V. Olaru, and C. 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. 7

[29] S. H. N. Jensen, A. Del Bue, M. E. B. Doest, and H. Aanæs. A benchmark and evaluation of non-rigid structure from motion. *arXiv preprint arXiv:1801.08388*, 2018. 7

[30] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018. 8

[31] C. Kong and S. Lucey. Prior-less compressible structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4123–4131, 2016. 1, 2, 3, 7

[32] C. Kong and S. Lucey. Deep non-rigid structure from motion. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 3, 5, 6, 7, 8

[33] C. Kong, R. Zhu, H. Kiani, and S. Lucey. Structure from category: A generic and prior-less approach. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 296–304. IEEE, 2016. 1, 2, 3

[34] Y. Kudo, K. Ogaki, Y. Matsui, and Y. Odagiri. Unsupervised adversarial learning of 3d human pose from 2d joint locations. *arXiv preprint arXiv:1803.08244*, 2018. 1, 3, 8

[35] S. Kumar. Jumping manifolds: Geometry aware dense non-rigid structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5346–5355, 2019. 1

[36] S. Kumar. Non-rigid structure from motion: Prior-free factorization method revisited. In *Winter Conference on Applications of Computer Vision (WACV 2020)*, 2020. 1, 2, 3

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

[38] S. Kumar, Y. Dai, and H. Li. Monocular dense 3d reconstruction of a complex dynamic scene from two perspective frames. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4649–4657, 2017. 1, 3

[39] C. Lassner, J. Romero, M. Kiefel, F. Bogo, M. J. Black, and P. 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. 7

[40] M. Lee, J. Cho, C.-H. Choi, and S. 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, 3, 5

[41] M. Lee, J. Cho, and S. Oh. Consensus of non-rigid reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4670–4678, 2016. 1, 3, 7, 8

[42] Z. Liu, X. Li, P. Luo, C. C. Loy, and X. Tang. Deep learning markov random field for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1814–1828, 2017. 5

[43] C. Murdock, M. Chang, and S. Lucey. Deep component analysis via alternating direction neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 820–836, 2018. 5

[44] D. Novotny, N. Ravi, B. Graham, N. Neverova, and A. Vedaldi. C3dpo: Canonical 3d pose networks for non-rigid structure from motion. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 3, 5, 7, 8

[45] J. Oliensis and R. Hartley. Iterative extensions of the sturm/triggs algorithm: Convergence and nonconvergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2217–2233, 2007. 2

[46] M. Paladini, A. Del Bue, J. Xavier, L. Agapito, M. Stošić, and M. Dodig. Optimal metric projections for deformable and articulated structure-from-motion. *International journal of computer vision*, 96(2):252–276, 2012. 3

[47] V. Papyan, Y. Romano, and M. Elad. Convolutional neural networks analyzed via convolutional sparse coding. *The Journal of Machine Learning Research*, 18(1):2887–2938, 2017. 6

[48] S. Parashar, A. Bartoli, and D. Pizarro. Self-calibrating isometric non-rigid structure-from-motion. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 252–267, 2018. 1

[49] J. Rick Chang, C.-L. Li, B. Poczos, B. Vijaya Kumar, and A. C. Sankaranarayanan. One network to solve them all–solving linear inverse problems using deep projection models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5888–5897, 2017. 5

[50] C. J. Rozell, D. H. Johnson, R. G. Baraniuk, and B. A. Olshausen. Sparse coding via thresholding and local competition in neural circuits. *Neural computation*, 20(10):2526–2563, 2008. 5

[51] X. Song, P. Wang, D. Zhou, R. Zhu, C. Guan, Y. Dai, H. Su, H. Li, and R. Yang. Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5452–5462, 2019. 7

[52] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European conference on computer vision*, pages 709–720. Springer, 1996. 2, 3

[53] K. Sun, B. Xiao, D. Liu, and J. Wang. Deep high-resolution representation learning for human pose estimation. *arXiv preprint arXiv:1902.09212*, 2019. 7, 8

[54] C. Tang and P. Tan. Ba-net: Dense bundle adjustment network. *arXiv preprint arXiv:1806.04807*, 2018. 5

[55] L. Torresani, A. Hertzmann, and C. 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. 8

[56] T. Ueshiba and F. Tomita. A factorization method for projective and euclidean reconstruction from multiple perspective views via iterative depth estimation. In *European conference on computer vision*, pages 296–310. Springer, 1998. 2

[57] S. Vicente and L. Agapito. Soft inextensibility constraints for template-free non-rigid reconstruction. In *European conference on computer vision*, pages 426–440. Springer, 2012. 1

[58] B. Wandt and B. 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. 3

[59] C. Wang, H. K. Galoogahi, C.-H. Lin, and S. Lucey. Deeplk for efficient adaptive object tracking. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 627–634. IEEE, 2018. 5

[60] C. Wang, C. Kong, and S. Lucey. Distill knowledge from nrsfm for weakly supervised 3d pose learning. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1

[61] G. Wang, H.-T. Tsui, and Z. Hu. Structure and motion of nonrigid object under perspective projection. *Pattern recognition letters*, 28(4):507–515, 2007. 2, 3

[62] G. Wang, H.-T. Tsui, and Q. M. J. Wu. Rotation constrained power factorization for structure from motion of nonrigid objects. *Pattern Recognition Letters*, 29:72–80, 2008. 3

[63] Y. Xiang, R. Mottaghi, and S. 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. 7

[64] J. Xiao and T. Kanade. Uncalibrated perspective reconstruction of deformable structures. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1075–1082. IEEE, 2005. 2

[65] Y. Zheng, Y. Kuang, S. Sugimoto, K. Astrom, and M. 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. 2, 3

[66] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. 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

[67] Y. Zhu, D. Huang, F. De La Torre, and S. 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. 2