Deep NRS*f*M++: Towards 3D Reconstruction 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 solely from 2D landmarks stemming from a large ensemble of images can be viewed as a non-rigid structure from motion (NRS*f*M) problem. To date, however, the application of NRS*f*M to problems in the wild has been problematic. Classical NRS*f*M approaches do not scale to large numbers of images and can only handle certain types of 3D structure (e.g. low-rank). A recent breakthrough [22] in this problem has allowed for the reconstruction of a substantially broader set of 3D structures, dramatically expanding the approach's importance to many problems in computer vision. However, the approach is still limited in that (i) it cannot handle missing/occluded points, and (ii) it is applicable only to weak-perspective camera models. In this paper, we present* Deep NRS*f*M++*, an approach to allow NRS*f*M to be truly applicable in the wild by offering up innovative solutions to the above two issues. Furthermore, we demonstrate state-of-the-art performance across numerous benchmarks, even against recent methods based on deep neural networks.*

1. Introduction

It is well understood [12, 28] how to recover the pose of an object through 2D landmarks if one knows the 3D structure beforehand. In most practical circumstances, however, one does not know that 3D structure. Some [35, 45] have advocated to instead use a dictionary of 3D structures – where the shape instance with the smallest re-projection error is chosen so as to recover the joint pose and shape. This strategy is also problematic as there is little guarantee that the dictionary of 3D shapes reflects the geometry of the object within the image. Increasingly, the vision community is needing solutions to this problem that rely solely on 2D landmarks from a large ensemble of images. This problem can be viewed as non-rigid structure from motion (NRS*f*M).

Current state-of-the-art NRS*f*M techniques, however, come with severe limitations. For one, NRS*f*M optimizers impose prior assumptions (*e.g.* low-rank dictionaries and temporal coherence) that limits their capabilities to deal with large-scale datasets. In addition, most NRS*f*M techniques are restricted to weak perspective camera models and assume fully annotated point correspondences to be available. This becomes a challenging scenario for objects in the wild, where occlusions and strong perspective effects are inevitable.

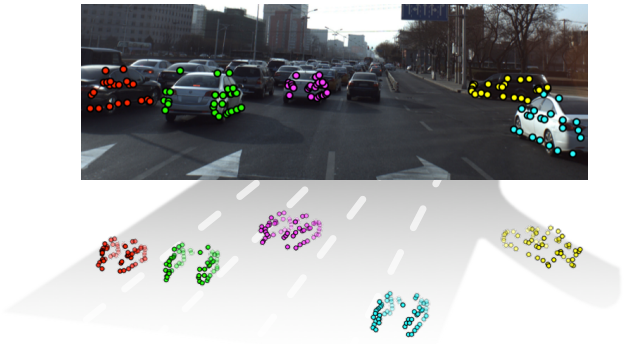

Figure 1: We present **Deep NRS*f*M++**, a general NRS*f*M framework of joint 3D shape and camera recovery from in-the-wild datasets, possibly consisting of occluded 2D landmarks and strong perspective camera projections.

Recently, Kong & Lucey [22] introduced a new NRS*f*M technique, termed Deep NRS*f*M, that learns the dictionaries for joint recovery of 3D shapes and camera poses. Deep NRS*f*M solves the hierarchical block-sparse dictionary learning problem by optimizing the objective end-to-end using deep learning as the machinery, where the structured inductive biases in the network architecture reflect the optimization procedure used for estimating block-sparse codes. With its ability to learn solely from 2D landmarks, Deep NRS*f*M circumvents the limitations of having to learn from the associated ground-truth 3D CAD models. However, its assumptions of fully-annotated observations under weak perspective camera models makes it yet to be practical on datasets collected in the wild, where (a) images can exhibit strong perspective effects and (b) missing landmark annotations due to heavy (self-)occlusions.

In this work, we develop upon the theoretical elegance of Deep NRS*f*M and propose **Deep NRS*f*M++**, a more gen-

eral framework applicable to more relaxed settings. Deep NRS*f*M++ is able to model both weak and strong perspective camera models with the ability to tolerate missing data, which is a fundamental breakthrough of NRS*f*M problems to real-world scenarios where camera perspectives and missing landmarks inherently exhibit within in-the-wild image datasets. We show that missing data and perspective projection can be accounted for by adaptively normalizing both the input 2D landmarks and the shape dictionary; in addition, explicit estimation of the camera's translational component can be circumvented by fully taking advantage of the object-centric nature of the problem. These reformulations lead to a unified framework under both strong and weak perspective camera models, capable of handling missing data.

Our contributions are summarized as below:

- We offer a novel formulation of the NRS*f*M problem to handle missing landmark data and perspective projection that is compatible to block sparse coding.
- 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 demonstrate state-of-the-art performance of Deep NRS*f*M++ across multiple benchmarks when comparing against classical NRS*f*M and deep learning methods, showing the effectiveness of Deep NRS*f*M++ in handling high percentage of missing data under both weak and strong perspective camera models.

We note that in this paper, we are considering approaches to NRS*f*M that make no assumptions about the temporal relationship between images and more generally applicable to datasets that are disjoint in both space and time.

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. Under orhtogonal projection, for each image, NRS*f*M is framed as factorizing the 2D projection matrix $\mathbf{W} \in \mathbb{R}^{P\times 2}$ as the product of a 3D shape matrix $\mathbf{S} \in \mathbb{R}^{P\times 3}$ and camera projection matrix $\mathbf{P} \in \mathbb{R}^{3\times 2}$:

$$\mathbf{W} = \mathbf{SP}, \quad \mathbf{P}^T\mathbf{P} = \mathbf{I}_2 \tag{1}$$

where each row of $\mathbf{W}$, $\mathbf{S}$ corresponds to the image coordinate (u_i, v_i) and world coordinate (x_i, y_i, z_i) of the i-th point of a shape with a total of P points. This factorization problem is obviously ill-posed. To resolve the ambiguities in solutions, priors are enforced upon the stack of shape matrices under multiple views, and also on the trajectories if temporal information is available. Such priors include the assumptios of shape/trajectory matrices being low rank [9, 6, 3, 14]; lying in a union of subspaces [25, 49, 2]; or being sparsely compressible [23, 21, 48].

These methods are mathematically well interpreted, and usually guarantees the uniqueness of the solution. However, many of them meet limitations in large scale data: Low-rank assumption is infeasible when the data has complex variations and the number of points is much smaller than the number of frames; Union-of-subspaces NRS*f*M has difficulties in how to effectively cluster shape deformations solely from 2D inputs, and how to estimate affinity matrix when the number of frames is large; Sparsity prior enables more power to model complex shape variations with large number of subspaces. But because there are many possible subspaces to choose from, it is sensitive to noise.

Perspective projection. Most NRS*f*M research assumes the weak perspective camera model. But in the real-world data that has objects close to the camera, modeling perspective projection is necessary for accurate reconstruction. Sturm & Triggs [36] formulate S*f*M under perspective camera as a factorization problem. This formulation was later developed to solve NRS*f*M. Xiao & Kanade [46] develop a factorization algorithm with two steps: projective depths are first recovered with sub-space constraints embeded in the 2D measurement, and then solve the factorization by weak perspective NRS*f*M. Wang *et al*. [42] propose to update solutions from a weak perspective to a full perspective projection by refining the projective depths recursively. Hartley & Vidal [17] derive a closed form linear solution. Their algorithm requires an initial estimation of a multifocal tensor, which is reported to be sensitive to noise.

Instead of directly solving the factorization problem in the form of Sturm & Triggs [36], we simplify the problem by fully utilizing the object-centric settings, and reformulate it to be compatible to block sparse coding.

Missing data. Missing point correspondences are inevitable in real-world data due to occlusions. Handling missing data is a non-trivial task not only due to the input data being incomplete, but also the fact that translation can no longer be removed simply by centering the 2D data. Prior works employ the following strategies : (i) introduce a visibility mask to exclude missing 2D points from the evaluation of the objective function [10, 15, 23, 43, 22], (ii) recover the missing 2D points using matrix completion algorithms, and then run the NRS*f*M [9, 27, 16], or (iii) treat missing points as unkonwns, and update them iteratively [31]. In this work, we follow the first strategy, and derives a novel approach under the framework of block sparse coding.

Deep learning methods for NRS*f*M. In recent literature of unsupervised 3D pose learning, equivalent problem is approached through training neural networks to lift 3D from 2D input poses. The lifting network is primarily trained by

minimizing the 2D reprojection error. However 2D reprojection error alone is obviously not enough to constrain the problem, and thus other constraints are needed. GANs and various forms of self consistency loss [30, 7, 40, 11, 24], are used to assist training. These methods are developed mostly from the machine learning point of view, and as a consequence, lack geometric interpretability especially when dealing with both missing data and perspective projection.

In the contrary, we mathematically derive a general framework that is applicable for both weak perspective and perspective projection, robust to missing data, and interpretable as solving hierarchical block sparse dictionary coding.

3. Background: Deep NRS*f*M

At its core, Deep NRS*f*M [22] solves a hierarchical block sparse dictionary learning problem under the assumption that the 3D shape $\mathbf{S} \in \mathbb{R}^{P\times 3}$ are compressible via multi-layer sparse coding, where P is the number of landmarks. This is formally written as

$$\begin{aligned} \mathbf{s} = \mathbf{D}_1\boldsymbol{\varphi}_0 \text{ and } \ \boldsymbol{\varphi}_{l-1} &= \mathbf{D}_l\boldsymbol{\varphi}_l \\ \text{s.t. } \|\boldsymbol{\varphi}_l\|_1 \leq \lambda_l, \ \ \boldsymbol{\varphi}_l &\geq \mathbf{0}, \ l = \{1 \dots L\} \ , \end{aligned} \tag{2}$$

where $\mathbf{s} \in \mathbb{R}^{3P}$ is the vectorization of $\mathbf{S}$, and $\mathbf{D}_1 \in \mathbb{R}^{3P\times K_1}$, $\mathbf{D}_l \in \mathbb{R}^{K_{l-1}\times K_l} \ \forall l$ are the hierarchical dictionaries. Constraints on multi-layer sparsity not only preserves sufficient freedom on shape variation, but also results in more constrained code recovery.

Hierarchical block sparse coding. Multiplying both sides of equations in (2) by the camera projection matrix $\mathbf{P} \in \mathbb{R}^{3\times 2}$ leads to the following hierarchical block sparse coding objective:

$$\mathbf{W} = \mathbf{D}_1^\sharp\boldsymbol{\Psi}_1, \ \ \boldsymbol{\Psi}_1 = (\mathbf{D}_2 \otimes \mathbf{I}_3)\boldsymbol{\Psi}_2, \ \ \dots \tag{3}$$

where $\otimes$ denotes Kronecker product and $\mathbf{D}_1^\sharp \in \mathbb{R}^{P\times 3K_1}$ is a reshape of $\mathbf{D}_1$. The 3D shape matrix $\mathbf{S}$ can be recovered by

$$\begin{aligned} \mathbf{S} &= \mathbf{D}_1^\sharp(\boldsymbol{\varphi}_1 \otimes \mathbf{I}_3) \\ \boldsymbol{\Psi}_l &= \boldsymbol{\varphi}_l \otimes \mathbf{P} \ , \end{aligned} \tag{4}$$

which is a concatenation of K_l 3×2 code blocks. Since the camera matrix $\mathbf{P}$ is orthonormal, we have

$$\frac{1}{\sqrt{2}}\|\boldsymbol{\Psi}_l\|_F^{(3\times 2)} = \|\boldsymbol{\varphi}_l\|_1 \leq \lambda_l \ \ \forall l \ \ , \tag{5}$$

where $\|.\|_F^{(3\times 2)}$ denotes the sum of the Frobenius norm of each 3×2 block. Therefore, we equivalently enforce $\|\boldsymbol{\Psi}\|_F^{(3\times 2)} \leq \lambda_l$. This hierarchical block sparse coding formulation can be generalized to handle occlusions and also perspective projections, as we show in later sections.

Bilevel optimization. Kong & Lucey [22] proposed to solve the hierarchical block sparse dictionary learning problem through a *bilevel* optimization procedure. With the dictionaries fixed, the lower-level optimization solves for the camera matrices and shapes by minimizing the block sparse coding objective. Kong & Lucey [22] advocated a neural network formulation inspired by the Iterative Shrinkage and Thresholding Algorithm (ISTA) [5, 34, 18], which can be interpreted as an approximation of a single ISTA iteration of inferring the sparse codes using the network weights as the dictionaries [32]. The neural network f approximates the solution of the lower-level optimization problem, *i.e.* $f(\mathbf{W}; \mathbf{D}_1, \cdots \mathbf{D}_L) \longrightarrow \mathbf{S}^*, \mathbf{P}^*$. The block sparsity constraint, however, was further relaxed to be non-negative with the ReLU operation, which we empirically find to degrade performance. At the higher level, the network weights are updated to minimize the 2D reprojection error, *i.e.* the difference between the reprojection of the 3D reconstruction and the 2D measurement.

The theoretical connection between solving a hierarchical block sparse coding and a feed-forward network is the major breakthrough in Deep NRS*f*M. However, this connection was formulated under the assumption of having perfect point correspondences with weak perspective camera model. We address these issues and propose Deep NRS*f*M++, a more general framework capable of handling missing points and perspective camera projections.

4. Deep NRS*f*M++

Deep NRS*f*M++ is a general framework for NRS*f*M that can handle an arbitrary number of missing landmarks under both weak and strong perspective camera models. Similar to Deep NRS*f*M, it formulates the problem as learning hierarchical block sparse dictionaries. The major difference is that the 2D input $\mathbf{W}$ and the first dictionary $\mathbf{D}_1$ are adaptive according to the visibility of the input points as well as the selected camera model. The block sparse coding objective can be written in a generic form as:

$$\begin{aligned} \mathbf{M}\tilde{\mathbf{W}} = \mathbf{M}\tilde{\mathbf{D}}_1\tilde{\boldsymbol{\Psi}}_1, &\qquad \|\boldsymbol{\Psi}_1\|_F^{(3\times M)} \leq \lambda_1 \\ \boldsymbol{\Psi}_1 = (\mathbf{D}_2 \otimes \mathbf{I}_3)\boldsymbol{\Psi}_2, &\qquad \|\boldsymbol{\Psi}_2\|_F^{(3\times M)} \leq \lambda_2 \\ \vdots & \\ \boldsymbol{\Psi}_{N-1} = (\mathbf{D}_N \otimes \mathbf{I}_3)\boldsymbol{\Psi}_N, &\qquad \|\boldsymbol{\Psi}_N\|_F^{(3\times M)} \leq \lambda_N \end{aligned} \tag{6}$$

where $\mathbf{M}$ is a diagonal binary matrix indicating visibility and M is the number of columns of the camera matrix $\mathbf{P}$, where $M = 2$ for weak perspective and $M = 3$ for perspective camera models. $\tilde{\mathbf{W}}$, $\tilde{\mathbf{D}}_1$, $\tilde{\boldsymbol{\Psi}}_1$ denote generic forms and are functions of $\mathbf{W}$, $\mathbf{D}_1$, and $\boldsymbol{\Psi}_1$ respectively; for the special case of full landmark visibility under weak perspective cameras, these relationships fall back to (3) in Deep NRS*f*M

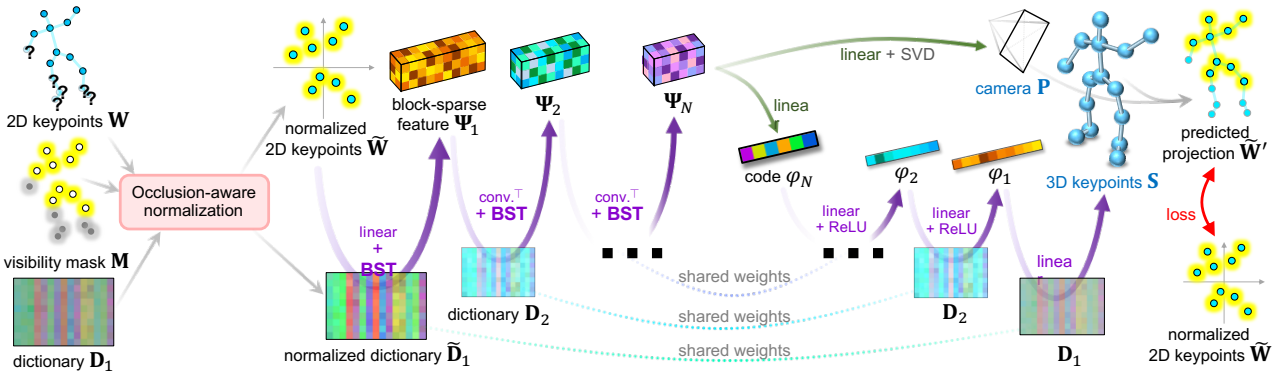

Figure 2: Deep NRS*f*M++: 2D keypoint input $\mathbf{W}$ and shape dictionary $\mathbf{D}_1$ are normalized according to the visibility mask $\mathbf{M}$ and camera type (summarized in Table 1). The normalized 2D input $\tilde{\mathbf{W}}$ is then fed into an encoder-decoder network derived from hierarchical block sparse coding (see Sec. 4.1). The network outputs the camera matrix $\mathbf{P}$ and the 3D shape $\mathbf{S}$, from which we can have a 2D reconstruction $\tilde{\mathbf{W}}'$. The training objective is to minimize the difference between $\tilde{\mathbf{W}}'$ and $\tilde{\mathbf{W}}$.

with $\tilde{\mathbf{W}} = \mathbf{W}$, $\tilde{\mathbf{D}}_1 = \mathbf{D}_1^\sharp$, $\tilde{\mathbf{\Psi}}_1 = \mathbf{\Psi}_1$. Table 1 summarizes the formulation for $\tilde{\mathbf{W}}$, $\tilde{\mathbf{D}}_1$, $\tilde{\mathbf{\Psi}}_1$ for different settings. We provide derivations of these mathematical relationships in Sec. 4.2 for weak perspective cameras and Sec. 4.3 for perspective cameras.

4.1. ISTA with Block Soft Thresholding

Consider the single-layer case for block sparse coding:

$$\min_{\mathbf{\Psi}} \|\mathbf{X} - \mathbf{D}\mathbf{\Psi}\|_F^2 + \lambda \|\mathbf{\Psi}\|_F^{(3\times M)} . \tag{7}$$

One iteration of ISTA is computed as

$$\mathbf{\Psi}^{(t+1)} = \text{prox}_{\lambda\|\cdot\|_F^{(3\times M)}}(\mathbf{\Psi}^{(t)} - \alpha \mathbf{D}^\top(\mathbf{D}\mathbf{\Psi}^{(t)} - \mathbf{X})) \tag{8}$$

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

$$\text{BST}^{(3\times M)}(\mathbf{\Psi}; \lambda) = \left[\dots \quad (1 - \tfrac{\lambda}{\|\Pi_i\|_F})^+ \Pi_i^\top \quad \dots\right]^\top . \tag{9}$$

Suppose $\mathbf{\Psi}$ is initialized to $\mathbf{0}$, and the step size $\alpha = 1$, then the first iteration of ISTA is written as

$$\mathbf{\Psi} = \text{BST}^{(3\times M)}(\mathbf{D}^\top \mathbf{X}; \lambda) . \tag{10}$$

We interpret BST as solving for the block sparse code and incorporate $\text{BST}^{(3\times M)}(\cdot)$ as the nonlinearity in our encoder part of the network, similar to a single-layer feed-forward ReLU network being interpreted as basis pursuit [32]. This formulation is closer to the true objective of NRS*f*M compared to Deep NRS*f*M, which uses ReLU as the nonlinearity to instead *relax* the constraint to L_1 sparsity with non-negative constraint. This is mainly due to the fact that the relationship $\mathbf{\Psi} = \boldsymbol{\varphi} \otimes \mathbf{P}$ is not applicable to the non-negative constraint. We show with empirical evidence of the superiority of $\text{BST}^{(3\times M)}$ over ReLU in Table 2.

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

$$\begin{aligned}
\mathbf{\Psi}_1 =& \text{BST}^{(3\times M)}([\tilde{\mathbf{D}}_1^\top \tilde{\mathbf{W}}]_{3K_1\times M}; \mathbf{b}_1), \\
\mathbf{\Psi}_2 =& \text{BST}^{(3\times M)}((\mathbf{D}_2 \otimes \mathbf{I}_3)^\top \mathbf{\Psi}_1; \mathbf{b}_2), \\
&\vdots \\
\mathbf{\Psi}_N =& \text{BST}^{(3\times M)}((\mathbf{D}_N \otimes \mathbf{I}_3)^\top \mathbf{\Psi}_{N-1}; \mathbf{b}_N),
\end{aligned} \tag{11}$$

where $\mathbf{b}_l \in \mathbb{R}^{K_l}$ are the thresholds for each of the K_l block. Following similar practice in Deep NRS*f*M, we factorize $\mathbf{\Psi}_N$ into the code vector $\boldsymbol{\varphi}_N$ and camera matrix $\mathbf{P}$ with approximate solutions, where the camera matrix $\mathbf{P}$ is further constrained to be orthonormal under weak perspective camera and $\mathbb{SO}(3)$ under perspective camera using SVD [22].

The recovered code $\boldsymbol{\varphi}_N$ is pass into a decoder network to reconstruct 3D shape $\mathbf{S}$ via

$$\begin{aligned}
\boldsymbol{\varphi}_{N-1} &= \text{ReLU}(\mathbf{D}_N \boldsymbol{\varphi}_N + \mathbf{b}'_N) , \\
&\vdots \\
\boldsymbol{\varphi}_1 &= \text{ReLU}(\mathbf{D}_2 \boldsymbol{\varphi}_2 + \mathbf{b}'_2) , \\
\mathbf{S} &= \mathbf{D}_1 \boldsymbol{\varphi}_1 .
\end{aligned} \tag{12}$$

Training objective. Reflecting the block sparse coding objective in (6), our training objective is to enforce the estimated $\tilde{\mathbf{\Psi}}_1^* = \boldsymbol{\varphi}_1^* \otimes \mathbf{P}^*$ to a closer reconstruction of the normalized 2D input $\tilde{\mathbf{W}}$:

$$\mathcal{L} = \|\tilde{\mathbf{D}}_1 \tilde{\mathbf{\Psi}}_1^* - \tilde{\mathbf{W}}\|_F . \tag{13}$$

	$\tilde{\mathbf{W}}$	$\tilde{\mathbf{D}}_1$	$\tilde{\mathbf{\Psi}}$
weak perspective	$\mathbf{W} - \mathbf{1}_P \otimes \frac{\mathbf{1}_P^\top \mathbf{M}\mathbf{W}}{\tilde{P}}$	$\mathbf{D}_1^\sharp + \mathbf{1}_P \otimes \frac{\mathbf{1}_P^\top (\mathbf{I}_P - \mathbf{M})\mathbf{D}_1^\sharp}{\tilde{P}}$	$\mathbf{\Psi}_1$, $3K_1 \times 2$
perspective	(28)	(29)	vec($\mathbf{\Psi}_1$), $9K_1 \times 1$

Table 1: Summary of $\tilde{\mathbf{W}}$, $\tilde{\mathbf{D}}_1$, $\tilde{\mathbf{\Psi}}$ under weak perspective and perspective camera.

4.2. Weak Perspective Camera

Due to an inherent scale ambiguity between camera and 3D shape in weak perspective camera models, we do not explicitly solve for the camera scale, but rather normalize the input 2D points with a corresponding 2D bounding box. We consider the scale after normalization to be one and reconstruct a scaled 3D shape. We assume the 3D shape $\mathbf{S}$ is zero-centered and lies in an object-centric coordinate system. $\mathbf{S}$ can thus be transformed to the camera coordinates via:

$$\mathbf{S}^{\text{cam}} = \mathbf{S}\mathbf{R} + \mathbf{1}_P \otimes \mathbf{t}^\top \,, \tag{14}$$

where $\mathbf{R}$ is the rotation matrix and $\mathbf{t}$ is the translation vector. Under orthorgraphic projection, the 2D projection $\mathbf{W}$ is

$$\mathbf{W} = \mathbf{S}_{xy}^{\text{cam}} = \mathbf{S}\mathbf{R}_{xy} + \mathbf{1}_P \otimes \mathbf{t}_{xy}^\top \,, \tag{15}$$

where $\mathbf{S}_{xy}^{\text{cam}}$ and $\mathbf{R}_{xy}$ are the first two columns of $\mathbf{S}^{\text{cam}}$ and $\mathbf{R}$ respectively $\mathbf{t}_{xy}$ is the 2D translation in x-y coordinates. Assuming that the object-centric coordinates are centered at the mean of the keypoint locations, we have

$$\mathbf{t}_{xy} = \bar{\mathbf{w}} = \frac{1}{P}\sum_{i=1}^{P} \mathbf{w}_i \,. \tag{16}$$

Handling missing data. To take possible occlusions into account, we can rewrite (15) as a generalized form using the visibility diagonal binary matrix $\mathbf{M}$ as

$$\mathbf{M}\mathbf{W} = \mathbf{M}(\mathbf{S}\mathbf{R}_{xy} + \mathbf{1}_P \otimes \mathbf{t}_{xy}^\top) \,. \tag{17}$$

Since not all 2D keypoint locations are guaranteed available, $\mathbf{t}_{xy}$ cannot be computed via (16) anymore. To resolve this, we propose to replace the occluded keypoints with the projection from $\mathbf{S}^{\text{cam}}$, with $\mathbf{t}_{xy}$ rewritten as

$$\mathbf{t}_{xy} = \frac{1}{P}\sum_{i=1}^{P} \underbrace{m_i \mathbf{w}_i}_{\text{visible points}} + \underbrace{(1-m_i)(\mathbf{R}_{xy}^\top \mathbf{s}_i + \mathbf{t}_{xy})}_{\text{occluded points}}, \tag{18}$$

where m_i indicates the visibility of the i-th keypoint and $\mathbf{s}_i$ denotes the i-th 3D point in $\mathbf{S}$. Rearranging (18) yields

$$\begin{aligned} \mathbf{t}_{xy} &= \frac{1}{\tilde{P}}\sum_{i=1}^{P} m_i \mathbf{w}_i + (1-m_i)\mathbf{R}_{xy}^\top \mathbf{s}_i \\ &= \frac{1}{\tilde{P}}[\mathbf{M}\mathbf{W} + (\mathbf{I}_P - \mathbf{M})\mathbf{S}\mathbf{R}_{xy}]^\top \mathbf{1}_P \,, \end{aligned} \tag{19}$$

where $\tilde{P}$ denotes the number of visible points. Substituting (19) into (17) and rearranging, we have

$$\mathbf{M}(\mathbf{W} - \mathbf{1}_P \otimes \frac{\mathbf{1}_P^\top \mathbf{M}\mathbf{W}}{\tilde{P}}) = \mathbf{M}(\mathbf{S}\mathbf{R}_{xy} + \mathbf{1}_P \otimes \frac{\mathbf{1}_P^\top (\mathbf{I}_P - \mathbf{M})\mathbf{S}\mathbf{R}_{xy}}{\tilde{P}}) \,. \tag{20}$$

Since $\mathbf{S}\mathbf{R}_{xy} = \mathbf{D}_1^\sharp(\boldsymbol{\varphi}_1 \otimes \mathbf{R}_{xy}) = \mathbf{D}_1^\sharp \mathbf{\Psi}_1$ from (4), we have

$$\mathbf{M}\ \underbrace{(\mathbf{W} - \mathbf{1}_P \otimes \frac{\mathbf{1}_P^\top \mathbf{M}\mathbf{W}}{\tilde{P}})}_{\tilde{W}\text{: normalized 2D projection}} = \mathbf{M}\ \underbrace{(\mathbf{D}_1^\sharp + \mathbf{1}_P \otimes \frac{\mathbf{1}_P^\top (\mathbf{I}_P - \mathbf{M})\mathbf{D}_1^\sharp}{\tilde{P}})}_{\tilde{\mathbf{D}}_1\text{: normalized dictionary}} \mathbf{\Psi}_1 \tag{21}$$

In other words, $\tilde{\mathbf{W}}$ is formed by shifting $\mathbf{W}$ with the average of visible keypoints locations. This aligns with the common practice employed for data normalization [22, 30]. For the special case where all points are visible, the expressions in (21) degenerates back to (3).

4.3. Perspective Camera

We first consider the case where all points are visible. Let (x_i', y_i', z_i') be the point coordinates in $\mathbf{S}\mathbf{R}$. Due to the fact that $\mathbf{S}\mathbf{R} = \mathbf{D}_1^\sharp \mathbf{\Psi}_1$, (x_i', y_i', z_i') can also be expressed as

$$x_i' = \mathbf{d}_i^T \boldsymbol{\psi}_x, \quad y_i' = \mathbf{d}_i^T \boldsymbol{\psi}_y, \quad z_i' = \mathbf{d}_i^T \boldsymbol{\psi}_z \,, \tag{22}$$

where $\mathbf{d}_i$ refers to each row of $\mathbf{D}_1^\sharp$, and $\boldsymbol{\psi}_x$, $\boldsymbol{\psi}_y$, $\boldsymbol{\psi}_z$ are the three columns of $\mathbf{\Psi}_1$. Assuming that camera intrinsics $\mathbf{K} = \mathbf{I}_3$, we have the linear relationships

$$x_i' + t_x = u_i(z_i' + t_z) \ \text{ and } \ y_i' + t_y = v_i(z_i' + t_z) \,, \tag{23}$$

which simply states that the product of the depth and 2D coordinates is equivalent to the x-y coordinates in 3D.

In the object-centric coordinate system, translation can be expressed as the mean of back-projection of the 2D points

$$t_x = \frac{1}{P}\sum_{i=1}^{P} u_i(z_i' + t_z), \quad t_y = \frac{1}{P}\sum_{i=1}^{P} v_i(z_i' + t_z) \,. \tag{24}$$

Substituting (24) and (22) into (23), we have the following linear relationships in matrix form:

$$\underbrace{\begin{pmatrix} \vdots \\ (u_i - \frac{1}{P}\sum_{j=1}^{P} u_j)t_z \\ (v_i - \frac{1}{P}\sum_{j=1}^{P} v_j)t_z \\ \vdots \end{pmatrix}}_{\tilde{\mathbf{W}}\text{:normalized 2D projection}} = \tilde{\mathbf{D}}_1 \underbrace{\begin{pmatrix} \boldsymbol{\psi}_x \\ \boldsymbol{\psi}_y \\ \boldsymbol{\psi}_z \end{pmatrix}}_{\tilde{\mathbf{\Psi}}_1} \,, \tag{25}$$

$$\tilde{\mathbf{D}}_1 = \begin{pmatrix} \vdots & \vdots & \vdots \\ \mathbf{d}_i^\top & 0 & -u_i\mathbf{d}_i^\top + \frac{1}{P}\sum_{j=1}^{P} u_j\mathbf{d}_j^\top \\ 0 & \mathbf{d}_i^\top & -v_i\mathbf{d}_i^\top + \frac{1}{P}\sum_{j=1}^{P} v_j\mathbf{d}_j^\top \\ \vdots & \vdots & \vdots \end{pmatrix}. \quad (26)$$

In this case, $\tilde{\mathbf{W}}$ is formed not only by shifting $\mathbf{W}$ with its mean but also scaled by t_z, the depth of the object center to the camera. t_z is simply a scalar that normalizes the 2D input and controls the scale of the 3D reconstruction, which is similar to the weak perspective case. However, $\tilde{\boldsymbol{\Psi}}_1$ is now a vectorization of $\boldsymbol{\Psi}_1$ by concatenating its columns.

Handling missing data. As in the weak perspective case, translation can be expressed using the average of visible and occluded points multiplied by their projective depth as

$$\begin{aligned} t_x &= \frac{1}{P}\sum_{i=1}^{P} \underbrace{m_i u_i (z_i' + t_z)}_{\text{visible points}} + \underbrace{(1-m_i)(x_i' + t_x)}_{\text{occluded points}} \\ &= \frac{\sum_{i=1}^{P} m_i u_i (z_i' + t_z) + (1-m_i)x_i'}{\sum_{i=1}^{P} m_i} \end{aligned} \quad (27)$$

and similarly for t_y. Substituting the new expressions of the translational components t_x and t_y into (23), we have

$$\mathbf{M} \underbrace{\begin{pmatrix} \vdots \\ (u_i - \frac{\sum_{j=1}^{P} m_j u_j}{\sum_{j=1}^{P} m_j})t_z \\ (v_i - \frac{\sum_{j=1}^{P} m_j v_j}{\sum_{j=1}^{P} m_j})t_z \\ \vdots \end{pmatrix}}_{\tilde{\mathbf{W}}\text{:normalized 2D projection}} = \mathbf{M}\tilde{\mathbf{D}}_1 \begin{pmatrix} \boldsymbol{\psi}_x \\ \boldsymbol{\psi}_y \\ \boldsymbol{\psi}_z \end{pmatrix}, \quad (28)$$

$$\tilde{\mathbf{D}}_1 = \begin{pmatrix} \vdots & \vdots & \vdots \\ \tilde{\mathbf{d}}_i^\top & 0 & -u_i\mathbf{d}_i^\top + \frac{\sum_{j=1}^{P} m_j u_j \mathbf{d}_j^\top}{\sum_{j=1}^{P} m_j} \\ 0 & \tilde{\mathbf{d}}_i^\top & -v_i\mathbf{d}_i^\top + \frac{\sum_{j=1}^{P} m_j v_j \mathbf{d}_j^\top}{\sum_{j=1}^{P} m_j} \\ \vdots & \vdots & \vdots \end{pmatrix} \quad (29)$$

where $\tilde{\mathbf{d}}_i^\top = \mathbf{d}_i^\top + \frac{\sum_{j=1}^{P}(1-m_j)\mathbf{d}_j^\top}{\sum_{j=1}^{P} m_j}$, which is the same as normalized dictionary in (21).

Scale correction. Properly normalizing the input has positive impact on both classical NRS*f*M and deep learning methods. Since we do not assume oracle 3D information to be accessible, we keep our algorithm general by using available 2D information such as bounding boxes. In practice, we utilize a strategy to estimate and recorrect the scale t_z in an iterative fashion. Specifically, we use the detected 2D object bounding box to provide an initial estimate of t_z; subsequently, we update the scale estimation with either the Frobenius norm of the reconstructed shape $\mathbf{S}$ or the average bone length of a skeleton model, if available. Once we have updated the scale estimation, namely t_z's, we rerun Deep NRS*f*M++ and update the reconstruction. This scale correction procedure is applied iteratively, such that the 3D reconstruction and scale estimation improves each other.

5. Experiments

Implementation details. The only hyper-parameter for our approach is the number of layers and the size of the dictionaries. Among those we find the most important hyper parameter is the size of dictionary at the last level, which is chosen depending on the amount of shape variation in the data. For human skeleton data, we use 8-10, and for rigid objects, we use 2-4. Compared to the residual networks used in other deep learning approaches, our model is much more compact. We use standard optimizers *e.g.* Adam to train our method. Detailed description is provided in supplementary.

Evaluation metrics. We employ the following metrics to evaluate the accuracy of 3D reconstruction: **MPJPE**: before calculating the mean per joint position, we first normalize the scale of the prediction to match against ground truth (GT). To account for the ambiguity due to weak perspective cameras, we also 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.* [30] is a metric invariant to camera pose and scale. **Normalized 3D error**: 3D error normalized by the scale of GT, used in prior NRS*f*M works [4, 9, 22, 15].

Block-soft vs ReLU thresholding. We first compare our approach against Deep NRS*f*M [22] on orthogonal projection data with perfect point correspondences, *e.g.* **CMU motion capture** dataset [1]. We follow the settings in Deep NRS*f*M– train a separate model for each of the human subject, and report the normalized 3D error on the training set. Table 2 shows that our approach with closer proximity to solving the true block sparse coding objective, namely using block soft thresholding instead of ReLU, achieves better accuracy compared to our own re-implementation (ReLU) of Deep NRS*f*M as well as the numbers reported in the original paper. We also compare against other NRS*f*M methods and show dramatic improvement.

Weak perspective projection with missing data. We evaluate the weak perspective version of our method on two benchmarks with high amount of missing data: **Synthetic Up3D** is a large synthetic dataset with dense human keypoints based on the Unite the People 3D(Up3D) dataset [26]. The data is generated by the orthographic projection SMPL body shape with 6890 vertices. The visibility of each point is computed using a ray tracer. The goal is to reconstruct 3D shapes from the rendered 2D keypoints. We follow the exact

same settings as C3DPO [30], we apply the learned model on the test set, and evaluate the metric on the 79 representative vertices of the SMPL model. **Pascal 3D+** [45] consists of images from PASCAL VOC and ImageNet images for 12 rigid object categories with sparse keypoints annotations. Each categories are associated with up to 10 CAD models. To ensure consistence between 2D keypoint and 3D GT., we follow [30] to use the orthographic projections of the aligned CAD model. The visibility of each point is then updated according to the original 2D annotations. Like C3DPO, we also train a single model to account for all 12 object categories. In addition, we include the result of testing our methods using detected 2D keypoints by HRNet [37].

Our method demonstrates state-of-the-art performance compared to other NRS*f*M methods and also deep learning method, namely C3DPO (see Table 3 & 4). On the one hand, we show over 32% error reduction due to our novel formulation of handling missing data by comparing to Deep NRS*f*M (see Table 4). On the other hand, ours compares favorably against C3DPO. Our method shows even greater advantage compared to C3DPO-base while both are learnt using the 2D reprojection loss. This advocates the elegance and effectiveness of our method.

Subject	CNS	NLO	SPS	Deep NRSfM	ReLU	Ours
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: Results on **CMU Motion Capture**. Compared with NRS*f*M methods: CNS [27], NLO [10], SPS [21] and Deep NRS*f*M [22]. 'ReLU' is our re-implementation of Deep NRS*f*M.

method	MPJPE	STRESS
EM-SfM [39]	0.107	0.061
GbNRSfM [14]	0.093	0.062
C3DPO-base [30]	0.160	0.105
C3DPO [30]	0.067	0.040
Ours	**0.062**	**0.037**

Table 3: Results on **Synthetic Up3D**.

Strong perspective projection We evaluate our approach on two datasets with strong perspective effects: **H3.6M** [19] is a large scale human pose dataset annotated by MoCap systems. We closely follow the commonly used evaluation protocol – we use 5 subjects (S1, S5, S6, S7, S8) for training, and 2 subjects (S9, S11) for testing. **Apollo 3D Car** [35] has 5277 images featuring cars. Each car instance is annotated with 3D pose by running PnP with 3D CAD models. 2D keypoint annotations are provided without 3D ground truth. To evaluate our method, we render 2D keypoints by projecting

method	MPJPE	STRESS
EM-SfM [39]	131.0	116.8
GbNRSfM [14]	184.6	111.3
Deep NRS*f*M [22]	51.3	44.5
C3DPO-base [30]	53.5	46.8
C3DPO [30]	36.6	31.1
Ours	**34.8**	**27.9**
Deep NRS*f*M	65.3	47.7
CMR [20]	74.4	53.7
C3DPO	57.5	41.4
Ours	**53.0**	**36.1**

Table 4: Results on **Pascal3D+**. The first row section is the testing result with GT 2D kepoints as input. The second row section is the testing result with detected kepoints by HRNet.

Method	MV/T	Ext3D	MPJPE	PA-MPJPE
Martinez *et al.* [29]	-	-	45.5	37.1
Zhao *et al.* [47]	-	-	43.8	-
3DInterpreter [44]		✓	-	88.6
AIGN [13]		✓	-	79.0 2
RepNet [40]		✓	50.9	38.2
Drover *et al.* [11]		✓	-	38.2
Pose-GAN [24]			130.9	-
C3DPO [30]			95.6	-
Chen *et al.* [7]			-	58
+ DA, TD	✓		-	51
Ours(weak persp)			104.2	72.9
+ persp			60.5	51.8
+ scale corr itr1			57.0	51.3
+ scale corr itr2			**56.6**	**50.9**

Table 5: Results on **H3.6M**. For each method we indicate their training supervision. MV/T means multi-view or temporal constraint. Ext3D means using external 3D data. The first row section lists two state-of-the-art supervised methods as reference. The 2nd section lists weakly supervised methods that use external 3D data. The bottom section lists unsupervised methods. The different versions of our method are: + persp: using perspective projection model, +scale corr itr: applying different number of scale correction iterations.

	no missing pts.		with missing pts.	
method	train	test	train	test
Consensus [27]	1.30	-	-	-
Ours(weak persp)	0.596	0.591	0.679	0.681
Ours(persp)	0.152	0.145	0.182	0.185
+ scale correction	0.131	0.124	0.165	0.168

Table 6: Results on **Apollo 3D Car dataset**. The evaluation metric is MPJPE in meters.

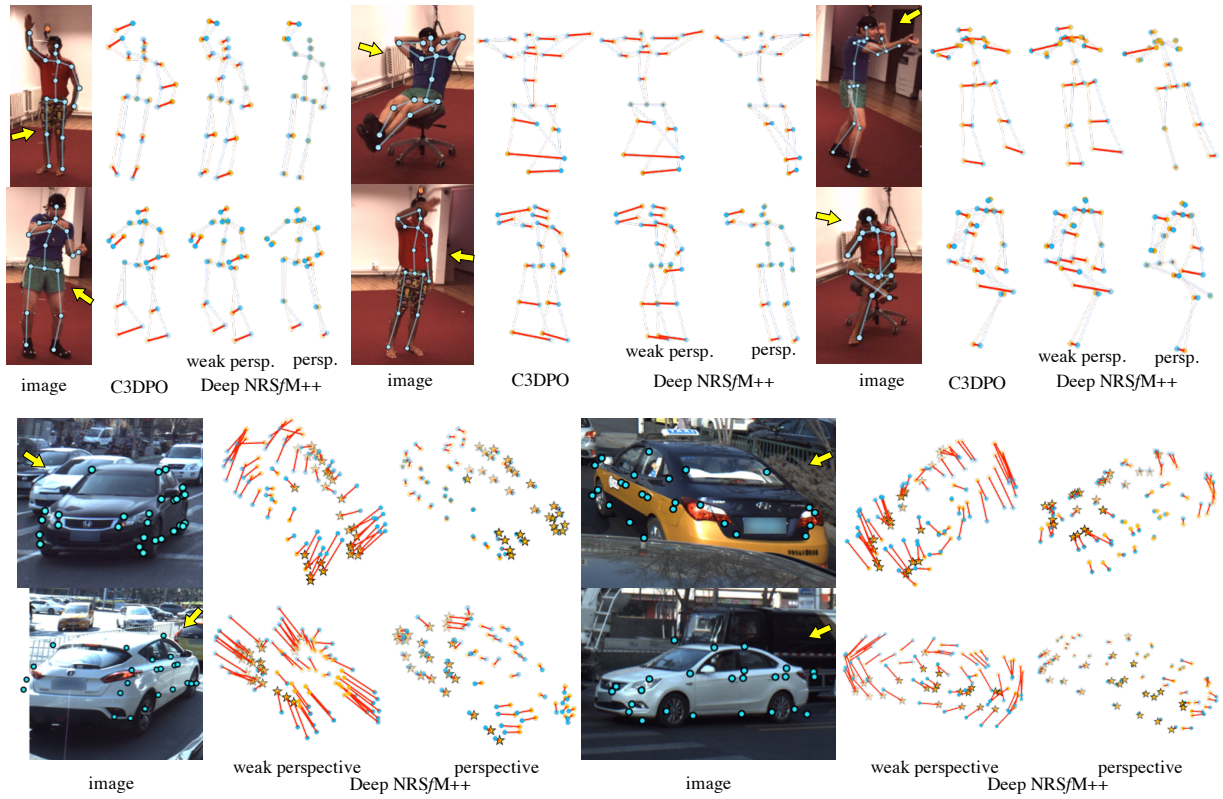

Figure 3: Qualatitative comparison. Blue points: GT, yellow points: prediction, red lines: distance between prediction and GT. The first two rows are visual results from H3.6M. In each sample, results from left to right are: C3DPO, ours with weak perspective camera model, and ours with perspective projection. The bottom two rows are results from Apollo dataset. In each sample from left to right are: ours with weak perspective camera model, and ours with perspective projection.

	Missing pts. (%)		
Noise(σ)	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 7: Results of our method on Apollo 3D Car dataset under different occlusion rate and noise in 2D keypoints.

34 car models according to the 3D car pose labels. Visibility of each keypoints are marked according to the original 2D keypoint annotations. To demonstrate strong perspective effects, we select cars within 30 meters with no less than 25 visible points (out of 58 in total) for our experiment. This gives us 2848 samples for training and 1842 for testing.

We evaluate different variants of our approach. We find that modeling perspective projection (Ours persp) leads to significant improvement over weak perspective model (Ours weak persp) and applying scale correction further improves accuracy. Our method shows robustness under different level of noise and occlusion (see Table 7) and achieves the best result compared to other unsupervised learning method. We outperforms the previous leading GAN-based method [7] by significant margin when using the same training set (50.9 v.s. 58) and Chen *et al*. [7] have to utilize external training source and temporal constraints to reach our level of performance.

6. Conclusion

We propose Deep NRS*f*M++, a general NRS*f*M framework for joint recovery of 3D shapes and camera poses solely from 2D landmarks. Deep NRS*f*M++generalizes to both weak and strong perspective camera models with the ability to handle missing/occluded landmark data, making it applicable to datasets captured in the wild. Furthermore, we establish a closer theoretical connection of an end-to-end learning framework to the true objective of the NRS*f*M problem. We demonstrate state-of-the-art performances across numerous benchmarks against various classical NRS*f*M techniques as well as deep learning based approaches, indicating the effectiveness of our approach.

Appendix

I: Implementation details

Dictionary sizes. The dictionary size in each layer of the block sparse coding is listed in Table 8. We tried two strategies to set the dictionary sizes: (i) exponentially decrease the dictionary size, i.e. 512, 256, 128, ..., (ii) linearly decrease, i.e. 125, 115, 104, ... Both strategies give reasonably good results. However, the hack we need to perform is to pick the size of the first and last layer dictionaries. We find that the size of the first layer would not have a major impact on accuracy as long as it is sufficiently large, and the corresponding hierarchical dictionary is deep enough (e.g. $256+$ for exponential decrease, $125+$ for linear decrease). The major performance factor is the size of the last layer dictionary. The rule of thumb we discovered is: 2-4 for rigid objects and 8-10 for articulated objects such as human body.

Training parameters. We use Adam optimizer to train. Learning rate = 0.0001 with linear decay rate 0.95 per 50k iterations. The total number of iteration is 400k to 1.2 million depending on the data. Batch size is set to 128. Larger or smaller batch sizes all lead to similar result.

Initial scale for input normalization. For weak perspective and strong perspective data, we need to estimate the scale so as to properly normalize the size of the 2D input shape. For Pascal3D+ and H3.6M datasets, we use the maximum length of the 2D bounding box edges, i.e. $t_z = 1/\max(\text{bbox\_height}, \text{bbox\_width})$. For Apollo 3D Car dataset, we choose the minimum length of the bounding box, i.e. $t_z = 1/\min(\text{bbox\_height}, \text{bbox\_width})$ by assuming that the height of each car is identical.

II: Additional empirical analysis

H3.6M. We add the test result using detected 2D keypoint as input in Table 9. Deep NRS*f*M ++ achieves state-of-the-art result compared to other unsupervised methods.

Apollo 3D Car dataset. Figure 4 shows additional analysis of Deep NRS*f*M ++ on the training set. Our method achieves $<$ 25cm error for over 80% testing samples with occlusions, while the compared baseline method, namely Consensus NRS*f*M [27] fails to produce meaningful reconstruction using perfect point correspondences. Average errors at different distances, rotation angles and occlusion rates are also reported. Overall, our method does not have a strong bias against a particular distance or occlusion rate. It does show larger error at 60° azimuth, most likely due to the data distribution, where most cars are in either front ($\approx 0^\circ$) or back ($\approx 180^\circ$) view.

dataset	dictionary sizes
CMU-Mocap	512, 256, 128, 64, 32, 16, 8
UP3D	512, 256, 128, 64, 32, 16, 8
Pascal3D+	256, 128, 64, 32, 16, 8, 4, 2
H3.6M	125, 115, 104, 94, 83, 73, 62, 52, 41, 31, 20, 10
Apollo	128, 100, 64, 50, 32, 16, 8, 4

Table 8: Dictionary sizes used in each experiment.

Method	MV/T	E3D	MPJPE	PA-MPJPE
Martinez *et al*. [29]	-	-	62.9	52.1
Zhao *et al*. [47]	-	-	57.6	-
3DInterpreter [44]		✓	-	98.4
AIGN [13]		✓	-	97.2
Tome *et al*. [38]	✓	✓	88.4	-
RepNet [40]		✓	89.9	65.1
Drover *et al*. [11]		✓	-	64.6
Pose-GAN [24]			173.2	-
C3DPO [30]			145.0	-
Wang *et al*. [41]			83.0	57.5
Chen *et al*. [7]	✓		-	68
Ours(persp proj)			68.9	59.4
+ scale corr itr1			67.3	59.2
+ scale corr itr2			**67.0**	**58.7**

Table 9: Result on **H3.6M** dataset with detected 2D kepoint input. In our result, we use detected points from cascaded pyramid network (CPN [8]) which is finetuned on H3.6M training set (excluding S9 and S11) by [33].

III: Additional discussion

One of the benefit of solving NRS*f*M by training a neural network is that, in addition to 2D reconstruction loss, we can easily employ other loss functions to further constrain the problem. In summary, our preliminary study finds that: (i) adding the canonicalization loss [30] does not noticeably improve result. (ii) adding Lasso regularization on φ_1 gives marginally better result in some datasets. (iii) adding symmetry constraint on the skeleton bone length helps to improve robustness against network initialization, but does not lead to better accuracy.

References

[1] Cmu motion capture dataset. available at http://mocap.cs.cmu.edu/. 6

[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

[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. 2

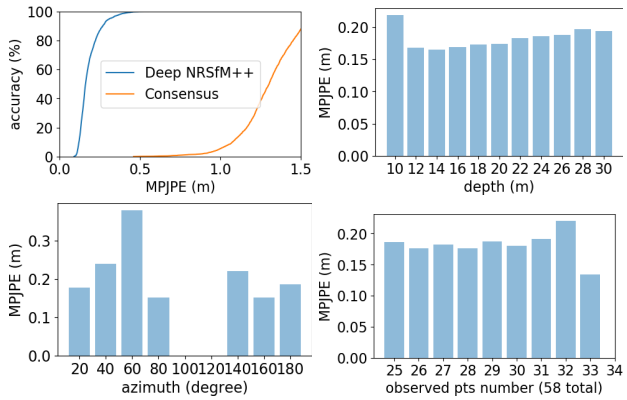

Figure 4: Additional result on **Apollo 3D Car** dataset. Top-left: percentage of sucess at different error thresholds. Rest: average error at different distances to the camera (top-right), azimuth angles of the car (bottom-left) and number of observed keypoints (bottom-right).

[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] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. Citeseer. 3

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

[7] 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. 3, 7, 8, 9

[8] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7103–7112, 2018. 9

[9] 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. 2, 6

[10] Alessio Del Bue, Fabrizio Smeraldi, and Lourdes Agapito. Non-rigid structure from motion using ranklet-based tracking and non-linear optimization. *Image and Vision Computing*, 25(3):297–310, 2007. 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. 3, 7, 9

[12] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 1

[13] Hsiao-Yu Fish Tung, Adam W. Harley, William Seto, and Katerina Fragkiadaki. Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 7, 9

[14] 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

[15] Paulo FU Gotardo and Aleix M Martinez. Kernel non-rigid structure from motion. In *2011 International Conference on Computer Vision*, pages 802–809. IEEE, 2011. 2, 6

[16] Onur C Hamsici, Paulo FU Gotardo, and Aleix M Martinez. Learning spatially-smooth mappings in non-rigid structure from motion. In *European Conference on Computer Vision*, pages 260–273. Springer, 2012. 2

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

[18] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005. 3

[19] 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. 7

[20] Angjoo Kanazawa, Shubham Tulsiani, Alexei A Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 371–386, 2018. 7

[21] 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, 7

[22] 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, 5, 6, 7

[23] 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

[24] 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. 3, 7, 9

[25] Suryansh Kumar, Yuchao Dai, and Hongdong 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. 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. 6

[27] 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. 2, 7, 9

[28] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. Epnp: An accurate o (n) solution to the pnp problem. *International journal of computer vision*, 81(2):155, 2009. 1

[29] 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. 7, 9

[30] 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. 3, 5, 6, 7, 9

[31] Marco Paladini, Alessio Del Bue, João Xavier, Lourdes Agapito, Marko Stošić, and Marija Dodig. Optimal metric projections for deformable and articulated structure-from-motion. *International journal of computer vision*, 96(2):252–276, 2012. 2

[32] Vardan Papyan, Yaniv Romano, and Michael Elad. Convolutional neural networks analyzed via convolutional sparse coding. *The Journal of Machine Learning Research*, 18(1):2887–2938, 2017. 3, 4

[33] Dario Pavllo, Christoph Feichtenhofer, David Grangier, and Michael Auli. 3d human pose estimation in video with temporal convolutions and semi-supervised training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7753–7762, 2019. 9

[34] Christopher J Rozell, Don H Johnson, Richard G Baraniuk, and Bruno A Olshausen. Sparse coding via thresholding and local competition in neural circuits. *Neural computation*, 20(10):2526–2563, 2008. 3

[35] Xibin Song, Peng Wang, Dingfu Zhou, Rui Zhu, Chenye Guan, Yuchao Dai, Hao Su, Hongdong Li, and Ruigang 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. 1, 7

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

[37] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. *arXiv preprint arXiv:1902.09212*, 2019. 7

[38] Denis Tome, Chris Russell, and Lourdes Agapito. Lifting from the deep: Convolutional 3d pose estimation from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 9

[39] 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. 7

[40] 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. 3, 7, 9

[41] Chaoyang Wang, Chen Kong, and Simon Lucey. Distill knowledge from nrsfm for weakly supervised 3d pose learning. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 9

[42] Guanghui Wang, Hung-Tat Tsui, and Zhanyi Hu. Structure and motion of nonrigid object under perspective projection. *Pattern recognition letters*, 28(4):507–515, 2007. 2

[43] Guanghui Wang, Hung-Tat Tsui, and Q. M. Jonathan Wu. Rotation constrained power factorization for structure from motion of nonrigid objects. *Pattern Recognition Letters*, 29:72–80, 2008. 2

[44] Jiajun Wu, Tianfan Xue, Joseph J Lim, Yuandong Tian, Joshua B Tenenbaum, Antonio Torralba, and William T Freeman. Single image 3d interpreter network. In *European Conference on Computer Vision*, pages 365–382. Springer, 2016. 7, 9

[45] 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. 1, 7

[46] Jing Xiao and Takeo 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

[47] Long Zhao, Xi Peng, Yu Tian, Mubbasir Kapadia, and Dimitris N. Metaxas. Semantic graph convolutional networks for 3d human pose regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 7, 9

[48] 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

[49] 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. 2