Deep Interpretable Non-rigid Structure from Motion

Chen kong

November 26

The Robotics Institue
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Simon Lucey, Carnegie Mellon University, Chair
David Held, Carnegie Mellon University
Ashwin Sankaranarayanan, Carnegie Mellon University
Hongdong Li, Australian National University

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

Keywords: non-rigid, structure from motion, 3D reconstruction, single image, deep learning, compressive sensing, sparse coding, dictionary learning, optimization

Abstract

Current non-rigid structure from motion (NRSfM) algorithms are limited with respect to: (i) the number of images, and (ii) the type of shape variability they can handle. This has hampered the practical utility of NRSfM for many applications within vision. Deep Neural Networks (DNNs) are an obvious candidate to help with such issue. However, their use has not been explored in recovering poses and 3D shapes from an ensemble of vector-based 2D landmarks. In this proposal, we present a novel deep neural network to recover camera poses and 3D points solely from an ensemble of 2D image coordinates. The proposed neural network is built upon our prior work on compressible structure form motion – extending the original single-layer sparsity constraint to a multi-layer one. The network architecture is mathematically interpretable as a multi-layer block sparse dictionary learning problem.

Our network is capable of handling problems of unprecedented scale in terms of samples and parameterization – allowing us to effectively recover 3D shapes deemed too complex by previous state-of-the-art. We further propose a generalization measure (based on the network weights) for guiding training to efficiently avoid overfitting, circumventing the need for 3D ground-truth. Once the networks weights are estimated (for a non-rigid object) we show how our approach can be used to recover 3D shape from a single image without 3D supervision. We shall propose how to extend our current framework to handle missing data, and identify new applications – such as Structure from Category (SfC) – where our approach can have substantial impact within computer vision.

Contents

Chapter 1

Introduction

Building an AI capable of inferring the 3D structure and pose of an object from a single image is a problem of immense importance. Training such a system using supervised learning requires a large number of labeled images – how to obtain these labels is currently an open problem for the vision community. Rendering [50] is problematic as the synthetic images seldom match the appearance and geometry of the objects we encounter in the real-world. Hand annotation is preferable, but current strategies rely on associating the natural images with an external 3D dataset (*e.g*. ShapeNet [13], ModelNet [64]), which we refer to as *3D supervision*. If the 3D shape dataset does not capture the variation we see in the imagery, then the problem is inherently ill-posed.

Non-Rigid Structure from Motion (NRS*f*M) offers computer vision a way out of this quandary – by recovering the pose and 3D structure of an object category *solely* from hand annotated 2D landmarks with no need of 3D supervision. Classically [9], the problem of NRS*f*M has been applied to objects that move non-rigidly over time such as the human body and face. But NRS*f*M is not restricted to non-rigid objects; it can equally be applied to rigid objects whose object categories deform non-rigidly [32]. Consider, for example, chairs. Each chair in isolation represents a rigid shape, but the set of all shapes describing "chair" is non-rigid. In other words, each object instance can be modeled as a deformation from its category's general shape.

Current NRS*f*M algorithms [15, 34, 35] all suffer from the difficulty of processing large-scale image sequences, limiting their ability to reliably model complex shape variations. This additionally hinders their ability to generalize to unseen images. Deep Neural Networks (DNNs) are an obvious candidate to help with such issue. However, the influence of DNNs has been most noticeable when applied to raster representations (*e.g*. raw pixel intensities [20]). While DNNs have recently exhibited their success to 3D point representations (*e.g*. point clouds) [29, 44], their use has not been explored in recovering poses and 3D shapes from an ensemble of vector-based 2D landmarks.

In this proposal, we present a novel DNN to solve the problem of NRS*f*M. Our employment of DNNs moves from an opaque black-box to a transparent "glass-box" in terms of its interpretability. The term "black-box" is often used as a critique of DNNs with respect to the general lack of understanding surrounding the inner workings. We demonstrate how the problem of NRS*f*M can be cast as a multi-layer block sparse dictionary learning problem. Through recent theoretical innovations [42], we then show how this problem can be reinterpreted as a

feed-forward DNN auto-encoder that can be efficiently solved through modern deep learning environments.

The salient features of the proposed work are

- Our deep NRS*f*M is capable of handling hundreds of thousands of images and learning large parameterizations to model non-rigidity.
- Our proposed approach is completely unsupervised in a 3D sense, relying solely on the projected 2D landmarks of the non-rigid object or object category to recover the pose and 3D shape.
- The considerable capacity of modeling non-rigidity allows us to efficiently apply it to unseen data. This facilitates an accurate 3D reconstruction of objects from a single view with no aid of 3D ground-truth.
- We propose a measure of generalization quality (using coherence and trained parameters), which improves the practical utility of our model in the real world applications.
- Our proposed work can be easily implemented via modern deep learning packages (*e.g*. Tensorflow, PyTorch).

Chapter 2

Background

Reconstructing the 3D geometry of objects from 2D images is a fundamental task in computer vision. With the remarkable success in Structure from Motion (S*f*M), which is now capable of reconstructing entire cities using large-scale photo collections [2] and real-time visual SLAM on embedded and mobile devices [52], the computer vision community is starting to explore the possibility of constructing a 3D model of an object from a single image [46, 62, 63, 69, 70]. Since estimating 3D geometry from a single view is an inherently ill-posed problem, all of these approaches need to employ 3D supervision.

In this chapter, we first visit a methods of single image 3D reconstruction using dense CAD models in Section 2.1. This work as a representative shows how heavily the current solutions rely on the 3D supervision. That followed is the central problem of this proposal—NRS*f*M—in Section 2.2, where we define the problem and visit classical assumptions along with their corresponding algorithms.

2.1 Single Image 3D Reconstruction using CAD Models

Given a single image of a certain object, our method seeks to estimate camera position and reconstruct a 3D dense model by utilizing the 2D landmark and silhouette information. We assume that the landmark positions have been labeled/detected and the image has been segmented beforehand. To achieve the goal, we first leverage the 3D CAD models in the object category by building up a graph, which we refer to as the Local Dense Correspondence (LDC) graph, to describe the dense correspondence between each pair of CAD models. We then propose a two-step approach: (1) estimate a coarse camera position and select the CAD model from the LDC graph to best register the 2D landmarks; (2) refine the camera position and deform the best CAD model by linear combination with its neighbors in the LDC graph to fit both the landmarks and the silhouette. Figure 2.1 shows the proposed LDC graph and our two-step coarse-to-fine method.

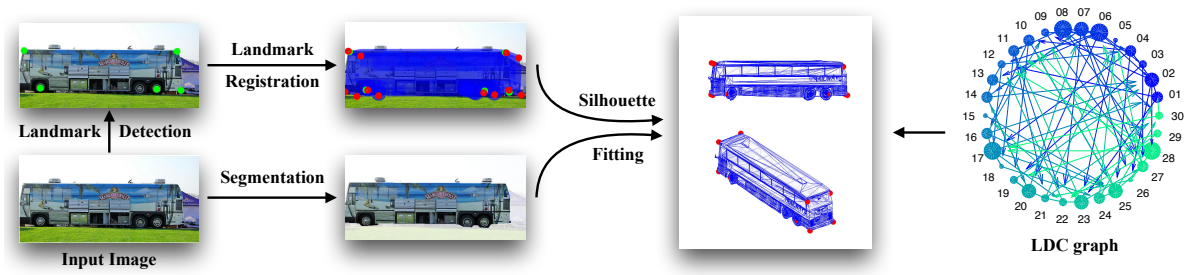


Figure 2.1: Overview of the method using 3D CAD models to solve single image 3D reconstruction problem. Given a single image with annotated/detected landmarks and silhouette, we build up a local dense correspondence graph (right), followed by a coarse estimate of camera position and CAD model by landmark registration, and final refinement creating a *deformable* dense model to fit both landmarks and silhouette

2.1.1 Local Dense Correspondence Graph

Local Dense Correspondence (LDC) graph is a directed graph with CAD models as nodes and the dense correspondence as edges. As each model here is manually and independently designed and does not necessarily share the same number of vertices or the same structure of meshes, dense correspondence based on vertex matching is not feasible. Instead, to build up the dense correspondence from model $\mathcal{S}_1$ to $\mathcal{S}_2$, we find a matching point on the surface of $\mathcal{S}_2$ for each vertices of $\mathcal{S}_1$. Therefore, such correspondence from $\mathcal{S}_1$ to $\mathcal{S}_2$ is not identical to that of $\mathcal{S}_2$ to $\mathcal{S}_1$, implying that the LDC graph is directed.

Creating graph

To create the LDC graph, we exploit the non-rigid ICP algorithm [6] to find a matching point for each vertex. We propose a distance metric to establish match quality. More specially, to build up dense correspondence from $\mathcal{S}_1(\mathbf{V}_1, \mathbf{E}_1)$ to $\mathcal{S}_2(\mathbf{V}_2, \mathbf{E}_2)$ where $\mathbf{V}, \mathbf{E}$ indicates the vertices and triangulation respectively, we warp the source, $\mathcal{S}_1$ in this case, to the target, $\mathcal{S}_2$ by non-rigid ICP, such that the warped $\mathcal{S}_1$ can represent the same shape as $\mathcal{S}_2$. For convenience, we denote the positions of warped vertices as $\mathbf{V}_1^2$, and the warped surface as $\mathcal{S}_1^2(\mathbf{V}_1^2, \mathbf{E}_1)$, since the triangulation should not change during the warp. We define that if the warped surface $\mathcal{S}_1^2$ represents the target $\mathcal{S}_2$ successfully, the warped vertices $\mathbf{V}_1^2$ are the dense correspondence from $\mathcal{S}_1$ to $\mathcal{S}_2$. To estimate the success of the warping, in other words, the similarity between the warped surface and the target, we propose the following metric[1]:

$$E_{12} = \frac{1}{|\mathbf{V}_1^2|} \sum_{\mathbf{v}_i \in \mathbf{V}_1^2} e(\mathbf{v}_i, \mathcal{S}_2; \theta) + \frac{1}{|\mathbf{V}_2|} \sum_{\mathbf{v}_i \in \mathbf{V}_2} e(\mathbf{v}_i, \mathcal{S}_1^2; \theta), \tag{2.1}$$

[1]As measuring the similarity between two surfaces is not a main contribution of our paper, we utilize this simple metric. More accurate metrics including 3D descriptors could be used to boost performance.

where function

$$e(\mathbf{v}, \mathcal{S}; \theta) = \begin{cases} 1 & \text{if dist}(\mathbf{v}, \mathcal{S}) > \theta \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

The value θ was chosen through a cross-validation such that consistent LDC graphs are formed across object categories.

Once we have measured warping quality, we establish dense correspondence according to a predefined threshold. Warps which score below the threshold are ignored. Note that, due to failure of nonrigid ICP in some cases, we found that indirect warping typically does not improve the performance. More specifically, given $\mathcal{S}_i, i = 1, 2, 3$, the indirect warping sequence $\mathcal{S}_1 \rightarrow \mathcal{S}_2 \rightarrow \mathcal{S}_3$ typically does not outperform the direct deformation $\mathcal{S}_1 \rightarrow \mathcal{S}_3$. Therefore, when creating our LDC graph, we only consider direct warping.

Figure 2.1 (right) shows an example of the proposed LDC graph, which has 30 nodes and 87 edges. The numbers besides the nodes are the indices of the corresponding CAD models. The size of nodes showed in the figure is proportional to the number of edges starting from that node. Note that due to the difficulty of non-rigid matching, not every edges are bidirectional. This can be caused by many factors: unbalanced numbers of vertices/meshes between two nodes, unbalanced sizes between two models and *etc*.

LDC subgraphs

Due to the nonexistence of global dense correspondence, the LDC graph is never fully connected. Therefore, we explore the local properties and sparsity structure of the LDC graph in this section. We divide the LDC graph into multiple subgraphs such that each subgraph has a node as center and contains all nodes that have dense correspondence from the center. Specifically, denote Ω as an index set pointing to the nodes in a certain subgraph with $\mathcal{S}_c$ as the center. The definition of subgraph implies that dense correspondence $\mathbf{V}_c^i$ always exist for any $i \in \Omega$. Therefore, a deformable model $\mathcal{S}(\mathbf{V}, \mathbf{E})$ can be created by linear combination:

$$\mathbf{V} = \omega_c \mathbf{V}_c + \sum_{i \in \Omega} \omega_i \mathbf{V}_c^i, \quad \mathbf{E} = \mathbf{E}_c, \tag{2.3}$$

where $\mathbf{V}$'s, $\mathbf{E}$'s are matrices containing the vertex position and triangulation respectively and ω's are combination weights. As a result, each LDC subgraph actually defines one deformable dense model controlled by the combination weights ω's. This insight is at the heart of our paper. Benefiting from this insight, the dense 3D reconstruction task could be addressed by first searching all subgraphs to find the best one, and then estimating the weights ω's.

Before visiting all possible subgraphs, we are curious how many subgraphs exist there, and how big these subgraphs are. From its definition, one can learn that the number of subgraphs equals to the number of nodes in the LDC graph[2], while the size of it varies from the smallest 1 (the center itself) to the number of nodes (the whole graph.)

[2]As the subgraph is the largest subset of nodes connected from its center, one node has and only has one subgraph expanded from it.

2.1.2 Landmark Registration

Given the LDC graph and a single image, we now want to decide which subgraph or which deformable dense model is the best option for a dense 3D reconstruction task. Even though exhaustively searching all possible subgraphs is a strategy to achieve the best performance, it is, however, computationally infeasible, especially when using large-scale 3D model dataset, like ShapeNet [13]. Therefore, instead of visiting all possible subgraphs, we propose a landmark registration algorithm to rapidly select the "closest" single node to the given image, and use the subgraph extended from this node as the optimal LDC subgraph.

Give the single image $\mathbf{I}$, we assume a certain landmark detection algorithm has been exploited, such that the 2D positions of landmarks on the image plane are known as $\mathbf{w}_p$, for $p = 1, ..., P$. Since some landmarks may not be visible, due to occlusion or self-occlusion, we denote an index set, $\mathcal{P}$, to indicate landmark visibility. By using the weak-perspective projection, we denote $\mathbf{R} \in \mathbb{R}^{2\times 3}$ as the first two rows of rotation matrix, $\mathbf{t}$ as the translation, s as the scale of camera. We define the i-th column in matrix $\mathbf{Y}_p \in \mathbb{R}^{3\times N}$, where N is the number of models, as the 3D position of p-th landmark in i-th model. Instead of trying all possible candidates exhaustively, we propose to use sparsity constraint for simultaneously selecting the best CAD model and estimating the camera parameters:

$$\begin{aligned} &\underset{\mathbf{R},s,\mathbf{t},\mathbf{c}}{\operatorname{argmin}} \frac{1}{2}\sum_{p\in\mathcal{P}} \Big\| s\mathbf{R}\mathbf{Y}_p\mathbf{c} + \mathbf{t} - \mathbf{w}_p \Big\|_2^2 \\ &\text{s.t. } \mathbf{R}\mathbf{R}^T = \mathbf{I}_2, \quad \|\mathbf{c}\|_0 = 1, \end{aligned} \tag{2.4}$$

where $\|\cdot\|_0$ is the ℓ_0 norm and $\mathbf{c}$ contains either zero or one, indicating which model is active. This objective can be minimized efficiently by Alternating Direction Method of Multipliers (ADMMs) [8].

From ADMMs, an auxiliary variable $\mathbf{Z}$ is introduced and the Equation 2.4 can be identically expressed as:

$$\begin{aligned} &\underset{\mathbf{M},\mathbf{Z},\mathbf{t},\mathbf{c}}{\operatorname{argmin}} \frac{1}{2}\sum_{p\in\mathcal{P}} \Big\| \mathbf{Z}\mathbf{Y}_p\mathbf{c} + \mathbf{t} - \mathbf{w}_p \Big\|_2^2 \\ &\text{s.t. } \mathbf{M}\mathbf{M}^T = s^2\mathbf{I}_2, \quad \|\mathbf{c}\|_0 = 1, \quad \mathbf{Z} = \mathbf{M}, \end{aligned} \tag{2.5}$$

where $\mathbf{M} = s\mathbf{R}$ for convenience. The augmented Lagrangian of Equation 2.5 is formulated as:

$$\begin{aligned} \mathcal{L} = \frac{1}{2}\sum_{p\in\mathcal{P}} \Big\| \mathbf{Z}\mathbf{Y}_p\mathbf{c} + \mathbf{t} - \mathbf{w}_p \Big\|_2^2 + \\ \langle \mathbf{\Lambda}, \mathbf{M} - \mathbf{Z} \rangle + \frac{\rho}{2}\big\|\mathbf{M} - \mathbf{Z}\big\|_F^2, \end{aligned} \tag{2.6}$$

where $\mathbf{\Lambda}$ is the lagrangian multiplier, ρ is a penalty factor to control the convergence behavior, and $< \cdot, \cdot >$ is Frobenius product of two matrices. ADMMs decomposes an objective into several sub-problems and iteratively solves them till convergence occurs [8]. We update $\mathbf{Z}$ by:

$$\begin{aligned} \mathbf{Z}^+ &= \underset{\mathbf{Z}}{\operatorname{argmin}}\, \mathcal{L} \\ &= \Big(\sum_{p\in\mathcal{P}} (\mathbf{w}_p - \mathbf{t})\mathbf{c}^T\mathbf{Y}_p^T + \mathbf{\Lambda} + \rho\mathbf{M} \Big) \Big(\sum_{p\in\mathcal{P}} \mathbf{Y}_p\mathbf{c}\mathbf{c}^T\mathbf{Y}_p^T + \rho\mathbf{I} \Big)^{\dagger}, \end{aligned} \tag{2.7}$$

and update $\mathbf{M}$ by:

$$\mathbf{M}^{+} = \underset{\mathbf{M}}{\operatorname{argmin}} \mathcal{L} = \mathbf{U} \begin{bmatrix} (\sigma_1 + \sigma_2)/2 & \\ & (\sigma_1 + \sigma_2) \end{bmatrix} \mathbf{V}^T, \tag{2.8}$$

where

$$\mathbf{Z} - \frac{1}{\rho}\mathbf{\Lambda} = \mathbf{U} \begin{bmatrix} \sigma_1 & \\ & \sigma_2 \end{bmatrix} \mathbf{V}^T, \tag{2.9}$$

and update $\mathbf{c}$ by:

$$\begin{aligned} \mathbf{c} = \underset{\mathbf{c}}{\operatorname{argmin}} \mathcal{L} = \underset{\mathbf{c}}{\operatorname{argmin}} \frac{1}{2} \sum_{p \in \mathcal{P}} \left\| \mathbf{Z}\mathbf{Y}_p\mathbf{c} + \mathbf{t} - \mathbf{w}_p \right\|_2^2, \\ \text{s.t. } \|\mathbf{c}\|_0 = 1, \end{aligned} \tag{2.10}$$

which can be solved by Orthogonal Matching Pursuit (OMP) [57] efficiently, and update $\mathbf{t}$ by:

$$\mathbf{t} = \underset{\mathbf{t}}{\operatorname{argmin}} \mathcal{L} = \frac{\sum_{p \in \mathcal{P}} \mathbf{w}_p - \mathbf{Z}\mathbf{Y}_p\mathbf{c}}{|\mathcal{P}|}, \tag{2.11}$$

where $|\mathcal{P}|$ indicate the number of visible points, and update Lagrangian multipliers and penalty factor by:

$$\mathbf{\Lambda} = \mathbf{\Lambda} + (\mathbf{M} - \mathbf{Z}), \quad \rho = \min(\rho * \tau, \rho_{max}), \tag{2.12}$$

where τ is the updating rate, and ρ_{max} is the upper bound of ρ. The whole algorithm is shown in Algorithm 1.

Algorithm 1: Landmark registration by ADMMs

Initialize variables: $\mathbf{M} = \begin{bmatrix} 1,0,0 \\ 0,1,0 \end{bmatrix}, t = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathbf{Z} = \mathbf{M}, \mathbf{\Lambda} = \mathbf{0}$;

while *not converge* **do**

- Update $\mathbf{Z}$ by Equation 2.7;
- Update $\mathbf{M}$ by Equation 2.8;
- Update $\mathbf{c}$ by Equation 2.10;
- Update $\mathbf{t}$ by Equation 2.11;
- Update lagrangian multiplier $\mathbf{\Lambda}$ and penalty ρ;

end

2.1.3 Silhouette Fitting

After landmark registration, we now have a rough estimate of camera position and a selected node which is considered to be "closest" to the given image. By treating this node as center, we extend an LDC subgraph to undertake our silhouette fitting step. We assume that a certain segmentation method has been executed so that the given image $\mathbf{I}$ has been segmented into foreground and background which means the silhouette is known. The main idea of this step is to simultaneously refine camera position and estimate combination weights such that as many vertices of the created model as possible are projected inside the silhouette.

In particular, by denoting the center as $\mathcal{S}_c$, Ω as the index set pointing to the nodes in the LDC subgraph, the deformable model $\mathcal{S}(\mathbf{V}, \mathbf{E})$ can be represented by Equation 2.3 with landmark

positions $\mathbf{X} = \omega_c \mathbf{X}_c + \sum_{i\in\Omega} \omega_i \mathbf{X}_c^i$, where $\mathbf{X}_c, \mathbf{X}_c^i$ are the 3D position of landmarks on model $\mathcal{S}_c$ and $\mathcal{S}_c^i$ respectively. The silhouette fitting problem can then be written as minimizing the energy function, with respect to the camera estimate $\mathbf{R}, \mathbf{t}$[3] and combination weights ω's:

$$\begin{aligned} &E(\mathbf{R}, \mathbf{t}, \boldsymbol{\omega}) \\ &= \frac{1}{2} \sum_{p\in\mathcal{P}} \Big\| s\mathbf{R}\big(\omega_c[\mathbf{X}_c]_p + \sum_{i\in\Omega} \omega_i [\mathbf{X}_c^i]_p\big) + \mathbf{t} - \mathbf{w}_p \Big\|_2^2 + \\ &\mu \sum_{p=1}^{N} \mathbf{C}\Big(s\mathbf{R}\big(\omega_c[\mathbf{V}_c]_p + \sum_{i\in\Omega} \omega_i [\mathbf{V}_c^i]_p\big) + \mathbf{t}\Big) + \frac{\gamma}{2} \sum_{i\in\Omega} \omega_i^2, \end{aligned} \tag{2.13}$$

where $[\cdot]_p$ is the p-th column of the matrix, N is the number of vertices, and μ, γ are penalty weights. The first term is the reprojection error as in Equation 2.4, the second term penalizes the vertices whose projection is outside of silhouette, where $\mathbf{C}$ is the Chamfer distance map from the segmentation of $\mathbf{I}$, and the third term is an ℓ_2 regularization.

By using exponential map to depict the change of rotation, we can identically express the energy function as

$$\begin{aligned} &E(\boldsymbol{\xi}, \mathbf{t}, \boldsymbol{\omega}) \\ &= \frac{1}{2} \sum_{p\in\mathcal{P}} \Big\| s\mathbf{R} e^{[\boldsymbol{\xi}]_\times} \big(\omega_c[\mathbf{X}_c]_p + \sum_{i\in\Omega} \omega_i [\mathbf{X}_c^i]_p\big) + \mathbf{t} - \mathbf{w}_p \Big\|_2^2 + \\ &\mu \sum_{p=1}^{N} \mathbf{C}\Big(s\mathbf{R} e^{[\boldsymbol{\xi}]_\times} \big(\omega_c[\mathbf{V}_c]_p + \sum_{i\in\Omega} \omega_i [\mathbf{V}_c^i]_p\big) + \mathbf{t}\Big) + \frac{\gamma}{2} \sum_{i\in\Omega} \omega_i^2, \end{aligned} \tag{2.14}$$

where $[\cdot]_\times$ is the skew-symmetric matrix. To minimize the proposed energy, we use gradient descent. The gradient of the energy with respect to ω_i's is

$$\begin{aligned} &\sum_{p\in\mathcal{P}} \Big(s\mathbf{R}\big(\omega_c[\mathbf{X}_c]_p + \sum_{i\in\Omega} \omega_i [\mathbf{X}_c^i]_p\big) + \mathbf{t} - \mathbf{w}_p \Big)^T s\mathbf{R}[\mathbf{X}_c^i]_p + \\ &\quad \mu \sum_{p=1}^{N} \nabla\mathbf{C}^T s\mathbf{R}[\mathbf{V}_c^i]_p + \gamma\omega_i, \end{aligned} \tag{2.15}$$

where $\nabla\mathbf{C}$ is the derivative of Chamfer distance. The gradient of the energy with respect to $\boldsymbol{\xi}$ is

$$\begin{aligned} &\sum_{p\in\mathcal{P}} \Big(s\mathbf{R}\big(\omega_c[\mathbf{X}_c^*]_p + \sum_{i\in\Omega} \omega_i [\mathbf{X}_c^i]_p\big) + \mathbf{t} - \mathbf{w}_p \Big)^T \\ &\Big(s\mathbf{R} \frac{\partial [\boldsymbol{\xi}]_\times}{\xi_j} \big(\omega_c[\mathbf{X}_c]_p + \sum_{i\in\Omega} \omega_i [\mathbf{X}_c^i]_p\big)\Big) + \\ &\mu \sum_{p=1}^{N} \nabla\mathbf{C}^T \Big(s\mathbf{R} \frac{\partial [\boldsymbol{\xi}]_\times}{\xi_j} \big(\omega_c[\mathbf{V}_c]_p + \sum_{i\in\Omega} \omega_i [\mathbf{V}_c^i]_p\big)\Big). \end{aligned} \tag{2.16}$$

[3]The scale in camera position is absorbed by ω's

The gradient of the energy with respect to translation $\mathbf{t}$ is

$$\sum_{p\in\mathcal{P}}\Big(s\mathbf{R}\big(\omega_c[\mathbf{X}_c]_p+\sum_{i\in\Omega}\omega_i[\mathbf{X}_c^i]_p\big)+\mathbf{t}-\mathbf{w}_p\Big)+\mu\sum_{p=1}^{N}\nabla\mathbf{C}. \tag{2.17}$$

We use backtracking to decide step sizes in each iteration[4].

2.1.4 Experiments

We evaluate our method by three metrics: (i) 2D landmark reprojection error, (ii) pose error, and (iii) structure error. More specifically, the 2D landmark reprojection error measures the accuracy of reprojected landmarks, which is computed as mean Euclidean distance between the projected landmarks of estimated dense model and the labeled landmarks on the image plane: $\text{err}_{2d}=\frac{1}{|\mathcal{P}|}\sum_{p\in\mathcal{P}}\left\|s\mathbf{R}\big(\omega_c[\mathbf{X}_c]_p+\sum_{i\in\Omega}\omega_i[\mathbf{X}_c^i]_p\big)+\mathbf{t}-\mathbf{w}_p\right\|_2$,following the same notations in Section 6. The pose error measures the accuracy of estimated pose (rotation): $\text{err}_{rot}=\|\mathbf{R}^*-\mathbf{R}\|_F$, where $\mathbf{R}^*,\mathbf{R}$ are the estimated and ground truth rotation matrices respectively. The structure error measures the quality of reconstructed dense model against the ground truth, following the same metrics shown in Equation 2.1.

As described in previous sections, our method consists of two steps where the silhouetting fitting involves three key components: (i) ℓ_2 regularization, (ii) refining camera position, and (iii) estimating the weights of linear combination. To show the performance boost introduced by the silhouetting fitting against the landmark registration with/without each components, extensive experiments are conducted using both synthetic and real images. To our best knowledge, the most related work [30, 61] reconstruct 3D dense models purely from 2D images without any access to 3D CAD models. Therefore, a direct comparison of our method against theirs is not fair. However, from visual evaluation (Figure 2.5), one can clearly observe the deformation of CAD models and their detailed geometry, which outperforms the state-of-the-art dense reconstruction algorithms.

Learned LDC Graphs

To show the generalization of the proposed method in various object categories, we learn LDC graphs for eight categories: diningtable, bicycle, car, chair, motorbike, sofa, aeroplane, and bus. To learn the graph, we randomly sample approximately 30 CAD models from the ShapeNet dataset [13] in each object category and manually annotate landmarks on each CAD models.

The learned LDC graphs are shown in Figure 2.2. One can observe that the density of connection varies significantly among different object categories, *e.g*. bicycle is the sparsest and diningtable is the densest. This connection density actually reflects the intra-category variations. Moreover, by visualizing the size of nodes in the graph proportional to the number of edges starting from that node, one can see that in some categories, like aeroplane, some nodes have an obviously larger size against others. This implies that these categories are more likely to share the same basic structures which is consistent to the common sense that aeroplanes have similar structure (wings in the middle of body, *etc*.) due to the same functionality.

[4]To be general, we initialized $\omega_c=1$ and $\omega_i=0$ for $i\in\Omega$

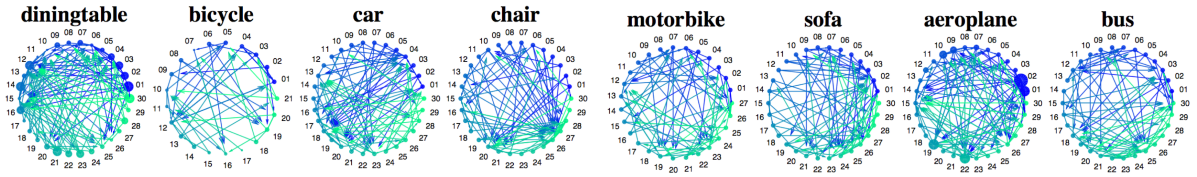


Figure 2.2: Learned LDC graphs from ShapeNet dataset for eight object categories.

Synthetic Experiments

We first evaluate the performance of our method using synthetic images projected by weak-perspective cameras. To generate these synthetic images, we visit all CAD models used as ground truth in PASCAL3D+ dataset [65]. By randomly generating weak-perspective camera positions, we project these CAD models into the image plane and estimate the corresponding segmentation and landmark positions. The results of this experiment is shown in Figure 2.3, demonstrating the performance increased by silhouette fitting and dense model combination. This evaluation shows that the silhouette fitting step with all components not only creates a deformable dense model closer to actual object geometry by LDC graph but also balances well between camera refinement and model combination.

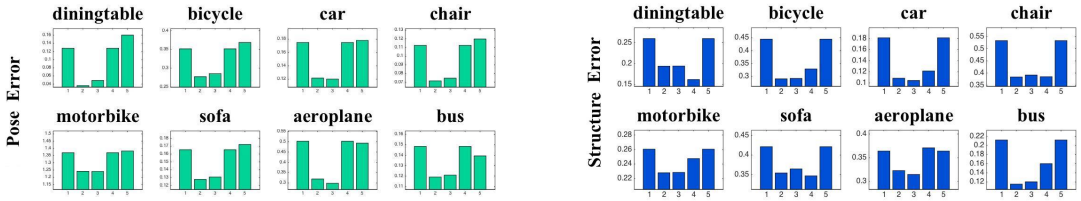


Figure 2.3: Evaluating our method using synthetic images in terms of pose error (top), and structure error (bottom). The x-axis shows the results of (1) landmark registration, (2) silhouette fitting with all components, (3) silhouette fitting without ℓ_2 regularization, (4) silhouette fitting without camera refinement, and (5) silhouette fitting without dense model combination.

Pascal3D+

To evaluate the performance of our framework over perspective projection and missing landmarks, we apply our proposed method to reconstruct 3D dense models of the PASCAL3D+ [65] natural images. For evaluation, we utilize the ground truth camera position, CAD models, and their annotated landmarks associated with the dataset to compute the pose, structure, and re-projection errors. The results are summarized in Figure 2.4 and Table 2.1. For all these eight categories except "bus", our method with full components achieves the best performance in terms of dense 3D models, camera positions and balancing between them.

Some qualitative results are shown in Figure 2.5. The models estimated by landmark registration (the second and forth columns) shows that landmark registration itself is not sufficient to select a correct model or estimate precise pose due to the limited information offered by sparse points. Further, the comparison between models reconstructed by landmark registration and silhouette fitting in both 2D and 3D shows that our proposed method not only refines the

	Component	din-ingtable	bicycle	car	chair	motor-bike	sofa	aero-plane	bus
Pose Error	LR	0.2227	**0.3216**	**0.2484**	0.1964	0.8674	0.3430	0.4527	**0.1699**
	Full SF	**0.1948**	0.3944	0.2777	**0.1858**	**0.8037**	**0.2682**	0.3507	0.2148
	SF-l2	0.1966	0.4036	0.2842	0.1901	0.8138	0.2918	**0.3401**	0.2116
	SF-Cam	0.2227	0.3216	0.2484	0.1964	0.8674	0.3430	0.4527	0.1699
	SF-Ome	0.2434	0.4081	0.3009	0.1917	0.9070	0.3495	0.5355	0.2198
Struct Error	LR	1.2936	0.3424	0.2541	0.3316	0.1954	0.4838	0.3709	**0.0998**
	Full SF	**0.6441**	**0.3314**	**0.2004**	**0.3046**	**0.1830**	**0.3872**	0.3098	0.1197
	SF-l2	0.7912	0.3409	0.2012	0.3130	0.1934	0.4831	0.3058	0.1164
	SF-Cam	0.9265	0.3479	0.2217	0.3137	0.1862	0.4486	**0.2997**	0.1267
	SF-Ome	1.2936	0.3424	0.2541	0.3236	0.1954	0.4839	0.3709	0.0998
Reproj Error	LR	30	**32**	45	21	**33**	29	**39**	**25**
	Full SF	23	41	40	19	33	22	44	38
	SF-l2	**22**	42	**39**	**17**	33	**20**	43	38
	SF-Cam	25	39	43	19	35	24	45	38
	SF-Ome	29	40	52	22	36	29	46	37

Table 2.1: Pose, structure and reprojection error obtained by landmark registration (LR), silhouette fitting with all components (Full SF), silhouette fitting without ℓ_2 regularization (SF-l2), silhouette fitting without refining camera position (SF-Cam), and silhouette fitting without model deformation (SF-Ome) for eight object categories.

object pose but also deforms the dense model to be consistent with 2D images, *e.g*. changing the length-width ratio of table, diminishing arms of a chair, and even warping a van into a sedan. We also compare our results again volumetric representation which is directly voxelized from ground truth 3D models. It is clear to see that the volumetric representation suffers from low resolution and is too coarse to represent any finer geometry. As shown in Figure 2.4 and Figure 2.5, our method fails in "bus" category. This is caused by either the strong perspective effect or the high occlusion of buses in PASCAL3D+ image set. Note that our method would fail if the large amount of object silhouette is broken or invisible.

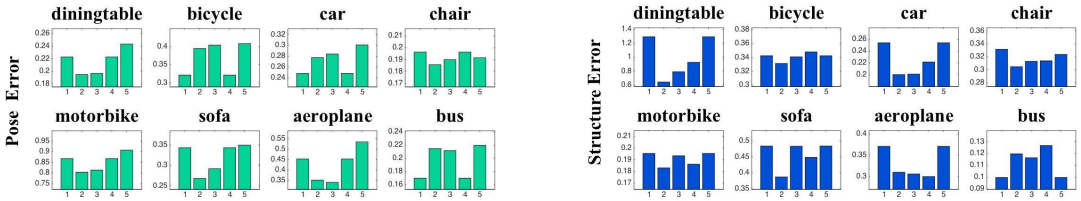


Figure 2.4: Evaluating our method using PASCAL3D+ natural images in terms of pose error (top), and structure error (bottom). The x-axis shows the results of (1) landmark registration, (2) silhouette fitting with all components, (3) silhouette fitting without ℓ_2 regularization, (4) silhouette fitting without camera refinement, and (5) silhouette fitting without dense model combination.

Figure 2.5: Visual evaluation of estimated 3D models by our proposed methods for eight object categories including diningtable, bicycle, car, chair, motorbike, sofa, aeroplane, and bus. We denote green nodes as labelled landmarks, red nodes as projected landmarks, and red number as landmarks index. The columns here shows respectively (1) the input images with landmarks and silhouette, (2) projection of dense model estimated by landmark registration, (3) projection of dense model estimated by silhouette fitting with all components, (4) the dense model estimated by landmark registration, (5) the dense model estimated by silhouette fitting with all components, (6) ground truth, and (7) volumetric representation of ground truth. The failure case is shown in red in the last row. Best viewed in color.

2.2 Non-rigid Structure from Motion

NRS*f*M under weak perspective deals with the problem of factorizing an image measurement matrix $\mathbf{W}$ as the product of camera motion (projection) matrix $\mathbf{M}$ and a shape $\mathbf{S}$, such that,

$$\mathbf{W} = \mathbf{MS} \tag{2.18}$$

where $\mathbf{S}$ is the 3D structure consisting of P points deforming over F frames, resulting in a $3F \times P$ concatenated matrix of points[9].

Weak perspective cameras is a reasonable assumption for objects whose variation in depth is small compared to their distance from the camera. In general, the measurement matrix is assumed to be already centered, so the camera matrix reduces to a $2F \times 3F$ block diagonal matrix whose blocks $\mathbf{M}_1 \cdots \mathbf{M}_F$ are each 2×3 matrices. The weak perspective camera assumption implies an orthonormal constraint such that $\mathbf{M}_f\mathbf{M}_f^T = \sigma^2\mathbf{I}_2$.

A commonly used reshape [9, 66] on the shape matrix $\mathbf{S}$ is $\mathbf{S}^\sharp = \mathbf{C}^\sharp\mathbf{B}^\sharp$ where $\mathbf{B}^\sharp \in \mathbb{R}^{K\times 3P}$, $\mathbf{C}^\sharp \in \mathbb{R}^{F\times K}$ and $\mathbf{S}^\sharp$ is a $F \times 3P$ reshape of $\mathbf{S}$ such that

$$\mathbf{W} = \mathbf{M}(\mathbf{C}^\sharp \otimes \mathbf{I}_3)\mathbf{B} = \mathbf{\Pi B} \tag{2.19}$$

where $\mathbf{I}_3$ is a 3×3 identity matrix, $\mathbf{B}$ is the $3K\times P$ reshape of the matrix $\mathbf{B}^\sharp$ and $\mathbf{\Pi} = \mathbf{M}(\mathbf{C}^\sharp\otimes\mathbf{I}_3)$.

2.2.1 Representative Shape Priors and algorithms

For a rigid 3D structure,

$$\mathbf{S}^\sharp = \begin{bmatrix} \mathbf{s}_x^T, & \mathbf{s}_y^T, & \mathbf{s}_z^T \\ \vdots & \vdots & \vdots \\ \mathbf{s}_x^T, & \mathbf{s}_y^T, & \mathbf{s}_z^T \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} \mathbf{s}_x, \mathbf{s}_y, \mathbf{s}_z & \dots & \mathbf{s}_x, \mathbf{s}_y, \mathbf{s}_z \end{bmatrix}^T \tag{2.20}$$

it is clear that the rank of $\mathbf{S}^\sharp$ *must* be one ($K = 1$) where $\mathbf{s}_x$, $\mathbf{s}_y$ and $\mathbf{s}_z$ are the P dimensional components of the $x-$, $y-$ and $z-$ coordinates of the rigid 3D structure. From Equation 2.20 this implies that $\mathbf{S}$ must have a rank of less than or equal to three due to the reshaping operation on $\mathbf{S}^\sharp$. This insight was used to great effect through the seminal work of Tomasi & Kanade [54] who demonstrated that one can compute the decomposition $\mathbf{W} = \hat{\mathbf{\Pi}}\hat{\mathbf{B}}$ via an SVD by preserving the first three modes of variation. Tomasi & Kanade also noted that the decomposition is non-unique, such that any nonsingular $\mathbf{G}$ can be inserted to form a valid factorization $\mathbf{W} = \hat{\mathbf{\Pi}}\hat{\mathbf{B}} = \hat{\mathbf{\Pi}}\mathbf{G}\mathbf{G}^{-1}\hat{\mathbf{B}} = \mathbf{\Pi B}$. The matrix $\mathbf{G}$ is referred to in literature as the corrective transformation [66].

Low-Rank assumption

Bregler *et al*. [9] extended the work of Tomasi & Kanade by assuming that $\mathbf{S}^\sharp$ must be of fixed rank $K > 1$ for non-rigid structure. From this insight, a recent work of Dai *et al*. [16], proposed an approach with a prior that the non-rigid 3D structure could be represented by a linear subspace of known rank K. In this work, Dai *et al*. proposed a strategy for estimating the corrective transformation matrix $\mathbf{G}$ whereby both the camera motion $\mathbf{M}$ and the 3D structure $\mathbf{S}$ can be

obtained. This approach offered a practical breakthrough to the problem of low-rank NRS*f*M, which had previously been touted [66] as being theoretically impossible to solve without additional prior/constraints. Further, from another perspective their algorithm answers the question: what are the minimal set of constraints/priors required to find a unique solution to the problem of low-rank NRS*f*M.

Based on the low-rank NRS*f*M, numerous innovations have followed, most of them centered around introducing additional "priors" to make the NRS*f*M problem less ambiguous. Notable examples of additional priors include: basis [66], temporal [4, 56, 72], articulation [43, 59], and camera motion [25] constraints. These priors are demonstrated useful for making the low-rank NRS*f*M problem tractable but considerably limit its applicability to scenarios where these constraints do not hold.

Manifold assumption

Another type of prior is manifold assumption [25, 45] which replace the low-rank assumption with learning a non-linear manifold. Most notable is the recent work of Gotardo and Martinez [25] who demonstrated how the "kernel trick" could be employed to model 3D shape as a non-linear subspace. A more recent work [34] proposed an objective from a Grassmannian perspective to solve NRS*f*M problem in dense scenario *i.e*. the number of points is large. A drawback to these approaches, however, was its reliance on additional priors except this manifold assumption, *e.g*. [25] further assumes k basis constraints and [34] assumes temporal consistency *i.e*. shapes move continuously along frames. These additional priors limit approaches' applicability to real world application.

It is worth mentioning that there is some overlap between this manifold assumption and the later proposed union of subspaces prior, as it has been demonstrated [18] that the field of manifold learning has a strong link to the recovery of compressed signals. Specifically, it has been demonstrated that a set of K sparse signals forms a K-dimensional Riemannian manifold. Further, it can be shown [18] that many manifold models can be expressed as an infinite union of subspaces.

Union-of-subspaces assumption

Recently, Zhu *et al*. [72] demonstrated a strategy for utilizing a union of local subspaces assumption within NRS*f*M. Specifically, the authors utilized an adaptation of Dai *et al*. [16] approach - which simultaneously reconstruct the 3D structure and affinity matrix. The affinity matrix is of importance as it naturally encodes the cluster/subspace membership of each projected shape sample. Experiments exhibit superior performance to Dai *et al*.'s approach for 3D structures that do not adhere to the low-rank assumption. However, their method relies on a sequence of other priors (i) the 3D structure is in a known temporal order, (ii) the camera motions are known, and (iii) the sparse basis is known a priori.

Chapter 3

Compressible Structure from Motion

One can view much of the literature of low-rank NRS*f*M drawing heavily upon the fact that one can obtain a solution to the rank constrained factorization problem

$$\underset{\mathbf{\Pi},\mathbf{B}}{\operatorname{argmin}} ||\mathbf{W}-\mathbf{\Pi}\mathbf{B}||_F^2, \quad \text{s.t. } \operatorname{rank}(\mathbf{\Pi}) = 3K \tag{3.1}$$

through an Singular Value Decomposition (SVD). Even though the SVD returns a unique solution $\{\hat{\mathbf{\Pi}}, \hat{\mathbf{B}}\}$ it is easy to demonstrate that this solution is just one of many possible solutions to $\mathbf{W} = \hat{\mathbf{\Pi}}\hat{\mathbf{B}} = \hat{\mathbf{\Pi}}\mathbf{G}\mathbf{G}^{-1}\hat{\mathbf{B}} = \mathbf{\Pi}\mathbf{B}$, where the corrective matrix $\mathbf{G}$ is any non-singular matrix. The ambiguity of this factorization is problematic for NRS*f*M problems as additional constraints are required to obtain a unique solution.

For rigid NRS*f*M (*i.e.* $K = 1$) the application of camera constraints [54] is typically sufficient in order to find a correction matrix $\mathbf{G}$ that gives a unique solution. Xiao *et al*. [66] famously demonstrated for $K > 1$ that one cannot determine a unique $\mathbf{G}$ since the space of solutions lies in a nullspace of rank $2K^2 - K$. Akhter *et al*. [3] additionally demonstrated that even though $\mathbf{G}$ is not unique, any solution to $\mathbf{G}$ that satisfies the camera constraints returns a valid 3D shape and camera motion pair. In this chapter, we want to explore whether moving away from canonical rank constraints and instead assuming that $\mathbf{\Pi}$ is block-sparse could result in a far less ambiguous factorization thus resulting in an NRS*f*M algorithm that can circumvent current theoretical and practical limitations.

3.1 Uniqueness of Block Sparse Dictionary Learning

3.1.1 Uniqueness of Sparse Dictionary Learning

The uniqueness of Sparse Dictionary Learning (SDL) is explored in literature [28]. In general terms, the problem of SDL can be described as

$$\underset{\mathbf{D},\mathbf{Z}}{\operatorname{argmin}} \|\mathbf{X}-\mathbf{D}\mathbf{Z}\|_F^2 \quad \text{s.t. } \|\mathbf{z}_i\|_0 = K, \quad i = 1, ..., N \tag{3.2}$$

where we are trying to recover the concatenation of a sparse coefficient matrix $\mathbf{Z}$ and dictionary basis $\mathbf{D}$ from a known set of signals in $\mathbf{X} \in \mathbb{R}^{D\times N}$. Specifically, the sparse coefficient matrix is

the concatenation of $K-$sparse coefficient vectors $\mathbf{Z} = \left[\mathbf{z}_1, \ldots, \mathbf{z}_D\right]$, and concatenation of $\mathbf{D} = \left[\mathbf{d}_1, \ldots, \mathbf{d}_M\right]$ dictionary basis vectors. An important question to ask in the context of applying SDL to NRS*f*M is how unique is the solution to Equation 3.2?

Hillar *et al*. [28] recently characterized theoretically the answer to this question. The authors define that if any valid solution $\{\hat{\mathbf{D}}, \hat{\mathbf{Z}}\}$ to the SDL objective in Equation 3.2 is ambiguous up to a $M \times M$ permutation matrix $\mathbf{P}$ and a diagonal invertible weighting matrix $\mathbf{\Lambda}$ such that $\hat{\mathbf{D}} = \mathbf{DP\Lambda}$, and $\hat{\mathbf{Z}} = \mathbf{\Lambda}^{-1}\mathbf{P}^T\mathbf{Z}$, they say that $\mathbf{X}$ has a *unique SDL*. Moreover, they proved theoretically that, given large enough N, the uniqueness of SDL is achieved if and only if the dictionary $\mathbf{D}$ satisfies the spark condition[1]:

$$\mathbf{Dz}_1 = \mathbf{Dz}_2 \quad \text{for } K\text{-sparse } \mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^M \Rightarrow \mathbf{z}_1 = \mathbf{z}_2. \tag{3.3}$$

Coherence as a proxy

The spark condition provides a complete characterization on the uniqueness of SDL. However, verifying whether a matrix $\mathbf{D}$ satisfies the spark condition is an NP-hard problem, which has to visit all $\binom{M}{K}$ subspaces. It is preferable in practice to use properties of $\mathbf{D}$ that are easily computable such as mutual coherence—which measures the largest absolute inner product between any two column vectors in the matrix—and with high probability is indicative of the spark condition of the matrix. In the experimental portion of this chapter we shall demonstrate how the coherence of a matrix can be utilized to predict the reconstructibility of a 3D structure solely from its 2D projections.

3.1.2 Block Sparse Dictionary Learning and Uniqueness

As we will discuss in the next section, there is a strong connection between compressible NRS*f*M and Block Sparse Dictionary Learning (BSDL). BSDL is a generalization of the SDL objective in Equation 3.2:

$$\underset{\mathbf{D},\mathbf{Z}}{\operatorname{argmin}} ||\mathbf{X} - \mathbf{DZ}||_F^2 \quad \text{s.t. } ||\mathbf{Z}_i||_{0,\alpha} = K, \quad i = 1, ..., N/\beta, \tag{3.4}$$

where $\mathbf{Z}_i \in \mathbb{R}^{D\times\beta}$ is a submatrix of $\mathbf{Z}$, *i.e.* $\mathbf{Z} = \left[\mathbf{Z}_1, ..., \mathbf{Z}_{N/\beta}\right]$. Each $\mathbf{Z}_i$ is divided into M/α blocks of size $\alpha \times \beta$ and $\|\mathbf{Z}_i\|_{0,\alpha}$ counts the number of blocks of which at least one element is non-zero. α and β need to be chosen such that D and M are perfectly divisible. One of particular importance in our compressible NRS*f*M problem is 3×2 block-sparsity which we will describe in more detail in the next section on compressible NRS*f*M.

Definition 1 *If any valid solution* $\{\hat{\mathbf{D}}, \hat{\mathbf{Z}}\}$ *to the objective in Equation 3.4 is ambiguous only up to a* $M \times M$ *block permutation matrix* $\mathbf{P}_\alpha$ *and a block-diagonal invertible weighting matrix* $\mathbf{\Lambda}_\alpha$ *such that*

$$\hat{\mathbf{D}} = \mathbf{DP}_\alpha\mathbf{\Lambda}_\alpha, \quad \hat{\mathbf{Z}} = \mathbf{\Lambda}_\alpha^{-1}\mathbf{P}_\alpha^T\mathbf{Z}, \tag{3.5}$$

we say $\mathbf{X}$ *has a unique BSDL.*

[1] Refer to [28] for the proof and a lower bound of N

The block permutation matrix is actually defined as $\mathbf{P}_\alpha = \mathbf{P} \otimes \mathbf{I}_\alpha$ where $\mathbf{P}$ is an arbitrary $(M/\alpha) \times (M/\alpha)$ permutation matrix and $\mathbf{I}_\alpha$ is a $\alpha \times \alpha$ identity matrix. The block-diagonal invertible weighting matrix $\mathbf{\Lambda}_\alpha$ has a $\alpha \times \alpha$ block structure. We now ask the same question: what is the sufficient and necessary condition for the uniqueness of BSDL?

Theorem 1 *There exist* $K\binom{M/\alpha}{K}^2$ *K-block-sparse vectors* $\mathbf{Z}_1, ..., \mathbf{Z}_{N/\beta}$, i.e. $N = \beta K\binom{M/\alpha}{K}^2$, *such that the uniqueness of BSDL holds if and only if the matrix* $\mathbf{D}$ *satisfies the block spark condition:*

$$\mathbf{D}\mathbf{Z}_1 = \mathbf{D}\mathbf{Z}_2 \quad \textit{for } K\textit{-block-sparse } \mathbf{Z}_1, \mathbf{Z}_2 \in \mathbb{R}^{M \times \beta} \Rightarrow \mathbf{Z}_1 = \mathbf{Z}_2. \tag{3.6}$$

3.1.3 Proof

Let's first prove Theorem 1 in the case when $\beta = 1$ and once it is proven, the general case $\beta > 1$ is simple to handle: We can split sparse causes $\mathbf{Z}^i$ into $[\mathbf{z}_1^i, \cdots, \mathbf{z}_\beta^i]$, where $\mathbf{z}_j^i \in \mathbb{R}^{D \times 1}$ and then

$$\mathbf{D}\mathbf{Z}^i = \mathbf{D}[\mathbf{z}_1^i, \cdots, \mathbf{z}_\beta^i] = \hat{\mathbf{D}}\hat{\mathbf{Z}}^i = \hat{\mathbf{D}}[\hat{\mathbf{z}}_1^i, \cdots, \hat{\mathbf{z}}_\beta^i] \tag{3.7}$$

is equivalent to $\mathbf{D}\mathbf{z}_j^i = \hat{\mathbf{D}}\mathbf{z}_j^i$, which degenerates to the situation where $\beta = 1$.

A simple case when $K = 1$

To better understand Theorem 1 and prepare for the proof in full generality, let us start from a simple case when $K = 1$. Denote $\mathbf{e}_i^L$ as a L-dimensional column vector that has one in its i-th coordinate and zeros elsewhere. For convenience, let $L = M/\alpha$. Now let us produce M block vectors

$$\mathbf{z}_j^i = (\mathbf{e}_i^L \otimes \mathbf{e}_j^\alpha), \quad i = 1, \cdots, L, \quad j = 1, \cdots, \alpha, \tag{3.8}$$

which denotes that its j-th coordinate in i-th block is one and zeros elsewhere, and $L\binom{\alpha}{2}$ block vectors $\mathbf{z}_{jk}^i = \mathbf{z}_{jk}^i + \mathbf{z}_{jk}^i$, for any i and $j \neq k$.

Now we claim that the uniqueness of BSDL in this simple case can be achieved by these $M + L\binom{\alpha}{2}$ block vectors, which is less than $K\binom{M/\alpha}{K}^2$ assuming $M \gg \alpha$.

Proof: There exists a matrix $\hat{\mathbf{D}}$ and 1-block-sparse vector $\hat{\mathbf{z}}_j^i = (\mathbf{e}_{\pi(i,j)}^L \otimes \mathbf{I}_\alpha)\boldsymbol{\lambda}_{ij}$, for some mapping $\pi : \{1, ..., L\} \times \{1, ..., \alpha\} \to \{1, ..., L\}$ and $\boldsymbol{\lambda}_{ij} \in \mathbb{R}^\alpha$, such that

$$\mathbf{D}\mathbf{z}_j^i = \mathbf{D}(\mathbf{e}_i^L \otimes \mathbf{e}_j^\alpha) = \hat{\mathbf{D}}\hat{\mathbf{z}}_j^i = \hat{\mathbf{D}}(\mathbf{e}_{\pi(i,j)}^L \otimes \mathbf{I}_\alpha)\boldsymbol{\lambda}_{ij}, \tag{3.9}$$

We claim that $\pi(i, j)$ is only dependent on i, not j. From Equation 3.9, we know that for any $j \neq k$,

$$\mathbf{D}\mathbf{z}_{jk}^i = \mathbf{D}(\mathbf{z}_j^i + \mathbf{z}_k^i) = \mathbf{D}\mathbf{z}_j^i + \mathbf{D}\mathbf{z}_k^i = \hat{\mathbf{D}}\hat{\mathbf{z}}_j^i + \hat{\mathbf{D}}\hat{\mathbf{z}}_k^i = \hat{\mathbf{D}}(\hat{\mathbf{z}}_j^i + \hat{\mathbf{z}}_k^i). \tag{3.10}$$

Since $\mathbf{z}_{jk}^i$ is 1-block-sparse, this implies that $\hat{\mathbf{z}}_j^i + \hat{\mathbf{z}}_k^i$ should also be 1-block-sparse. Therefore, $\pi(i, j) = \pi(i, k)$, that is, $\pi : \{1, ..., L\} \to \{1, ..., L\}$ and

$$\mathbf{D}(\mathbf{e}_i^L \otimes \mathbf{e}_j^\alpha) = \hat{\mathbf{D}}(\mathbf{e}_{\pi(i)}^L \otimes \mathbf{I}_\alpha)\boldsymbol{\lambda}_{ij}. \tag{3.11}$$

Let us now prove that $\mathbf{\Lambda}_i = [\boldsymbol{\lambda}_{i1}, \ldots, \boldsymbol{\lambda}_{i\alpha}]$ is invertible. Let $\mathbf{Z}^i = [\mathbf{z}_1^i, \ldots, \mathbf{z}_\alpha^i]$ and $\hat{\mathbf{Z}}^i = [\hat{\mathbf{z}}_1^i, \ldots, \hat{\mathbf{z}}_\alpha^i]$. From Equation 3.11, it follows that

$$\mathbf{D}\mathbf{Z}^i = \mathbf{D}[\mathbf{z}_1^i, \ldots, \mathbf{z}_\alpha^i] = \mathbf{D}[(\mathbf{e}_i^L \otimes \mathbf{e}_1^\alpha), ..., (\mathbf{e}_i^L \otimes \mathbf{e}_\alpha^\alpha)] = \mathbf{D}(\mathbf{e}_i^L \otimes \mathbf{I}_\alpha) \tag{3.12}$$

and

$$\mathbf{D}\mathbf{Z}^i = \hat{\mathbf{D}}\hat{\mathbf{Z}}^i = \hat{\mathbf{D}}(\mathbf{e}_{\pi(i)}^L \otimes \mathbf{I}_\alpha) \left[\boldsymbol{\lambda}_{i1}, ..., \boldsymbol{\lambda}_{i\alpha}\right] = \hat{\mathbf{D}}(\mathbf{e}_{\pi(i)}^L \otimes \mathbf{I}_\alpha)\mathbf{\Lambda}_i. \tag{3.13}$$

Therefore,

$$\mathbf{D}(\mathbf{e}_i^L \otimes \mathbf{I}_\alpha) = \hat{\mathbf{D}}(\mathbf{e}_{\pi(i)}^L \otimes \mathbf{I}_\alpha)\mathbf{\Lambda}_i. \tag{3.14}$$

Due to the fact that $\mathbf{D}$ satisfies the block spark condition, $\mathrm{rank}(\mathbf{D}(\mathbf{e}_i^L \otimes \mathbf{I}_\alpha)) = \alpha$. From Equation 3.14, $\mathrm{rank}(\hat{\mathbf{D}}(\mathbf{e}_{\pi(i)}^L \otimes \mathbf{I}_\alpha)\mathbf{\Lambda}_i) = \alpha$. We know that $\mathrm{rank}(\mathbf{XY}) \leq \min(\mathrm{rank}(\mathbf{X}), \mathrm{rank}(\mathbf{Y}))$, for any matrix $\mathbf{X}, \mathbf{Y}$. So $\mathrm{rank}(\mathbf{\Lambda}_i) \geq \alpha$. As $\mathbf{\Lambda}_i \in \mathbb{R}^{\alpha\times\alpha}$, $\mathrm{rank}(\mathbf{\Lambda}_i) = \alpha$.

Now, let us show π is necessarily injective. Suppose $\pi(i) = \pi(j)$, with $i \neq j$, then from Equation 3.14,

$$\mathbf{D}(\mathbf{e}_i^L \otimes \mathbf{I}_\alpha) = \hat{\mathbf{D}}(\mathbf{e}_{\pi(i)}^L \otimes \mathbf{I}_\alpha)\mathbf{\Lambda}_i = \hat{\mathbf{D}}(\mathbf{e}_{\pi(j)}^L \otimes \mathbf{I}_\alpha)\mathbf{\Lambda}_j\mathbf{\Lambda}_j^{-1}\mathbf{\Lambda}_i = \mathbf{D}(\mathbf{e}_j^L \otimes \mathbf{I}_\alpha)\mathbf{\Lambda}_j^{-1}\mathbf{\Lambda}_i. \tag{3.15}$$

Since $\mathbf{D}$ satisfies the block spark condition, which implies $\mathbf{D}$ can never map two different 1-block-sparse vectors to the same measurement, this is possible only if $i = j$. Thus, π is injective.

Let $\mathbf{P}_\pi$ and $\mathbf{D}$ be generated by

$$\mathbf{P}_\pi = \begin{bmatrix}\mathbf{e}_{\pi(1)}^L & \cdots & \mathbf{e}_{\pi(K)}^L\end{bmatrix}, \mathbf{\Lambda} = \begin{bmatrix}\mathbf{\Lambda}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathbf{\Lambda}_L\end{bmatrix}. \tag{3.16}$$

Since π is injective, $\mathbf{P}_\pi$ is a permutation matrix. Let us stack Equation 3.14 from left-to-right on both sides, and it follows that on left sides,

$$[\mathbf{D}(\mathbf{e}_1^L \otimes \mathbf{I}_\alpha), \ldots, \mathbf{D}(\mathbf{e}_L^L \otimes \mathbf{I}_\alpha)] = \mathbf{D}, \tag{3.17}$$

and on right sides,

$$[\hat{\mathbf{D}}(\mathbf{e}_{\pi(1)}^L \otimes \mathbf{I}_\alpha)\mathbf{\Lambda}_1, \ldots, \hat{\mathbf{D}}(\mathbf{e}_{\pi(L)}^L \otimes \mathbf{I}_\alpha)\mathbf{\Lambda}_L] = \hat{\mathbf{D}}(\mathbf{P}_\pi \otimes \mathbf{I}_\alpha)\mathbf{\Lambda}. \tag{3.18}$$

Hence, we proved Theorem 1 for the simple case, where $K = 1$. ■

Preparation

We use the same notation reported in [28]: Denote $[L]$ as the set $\{1, \ldots, L\}$ and $\binom{[L]}{K}$ as the K-element subset of $[L]$. Moreover, let the dictionary $\mathbf{D} = [\mathbf{D}_1, \ldots, \mathbf{D}_L]$ with $\mathbf{D}_i \in \mathbb{R}^{D\times\alpha}$, and denote $\mathrm{span}\{\mathbf{D}_\mathcal{S}\}$ as a subspace expanded by $\mathbf{D}_i$, $i \in \mathcal{S}$.

To prove Theorem 1 in general situations, we offer a lemma at first.

Lemma 1 *Suppose that* $\mathbf{D}$ *satisfies the block spark condition and*

$$\kappa : \binom{[L]}{K} \to \binom{[L]}{K} \tag{3.19}$$

is a mapping with the following property: for all $\mathcal{S} \in \binom{[L]}{K}$,

$$\text{span}\{\mathbf{D}_{\mathcal{S}}\} = \text{span}\{\hat{\mathbf{D}}_{\kappa(\mathcal{S})}\}. \tag{3.20}$$

Then, there exist a permutation matrix $\mathbf{P}_\kappa \in \mathbb{R}^{L\times L}$ *and an invertible block diagonal matrix* $\mathbf{\Lambda} \in \mathbb{R}^{M\times M}$ *such that* $\mathbf{D} = \hat{\mathbf{D}}(\mathbf{P}_\kappa \otimes \mathbf{I}_\alpha)\mathbf{\Lambda}$.

Proof: Here we demonstrate, through induction, that if our $K = 1$ case holds, then, $K > 1$ case should also hold. First, let us show function κ is injective. Suppose that $\mathcal{S}, \mathcal{S}' \in \binom{[L]}{K}$ are different and $\kappa(\mathcal{S}) = \kappa(\mathcal{S}')$ holds. Then by Equation 3.20,

$$\text{span}\{\mathbf{D}_{\mathcal{S}}\} = \text{span}\{\hat{\mathbf{D}}_{\kappa(\mathcal{S})}\} = \text{span}\{\hat{\mathbf{D}}_{\kappa(\mathcal{S}')}\} = \text{span}\{\mathbf{D}_{\mathcal{S}'}\}. \tag{3.21}$$

As $\mathbf{D}$ satisfies the block spark condition, every $K+1$ block columns of $\mathbf{D}$ are linearly independent. From Lemma 2 (see below), it turns out that $\mathcal{S} = \mathcal{S}'$, which implies κ is injective.

Denote $\eta = \kappa^{-1}$ as the inverse of κ. Fix $\mathcal{S} = \{i_1, ..., i_{K-1}\} \in \binom{[L]}{K-1}$, and set $\mathcal{S}_1 = \mathcal{S} \cup \{p\}$ and $\mathcal{S}_2 = \mathcal{S} \cup \{q\}$ for some fixed $p, q \notin \mathcal{S}$ with $p \neq q$. Since $K < L$, $L - (K-1) > 1$, thus, it is always possible to find such p and q. From Equation 3.20, we obtain:

$$\text{span}\{\mathbf{D}_{\eta(\mathcal{S}_1)}\} = \text{span}\{\hat{\mathbf{D}}_{\mathcal{S}_1}\}, \tag{3.22}$$

$$\text{span}\{\mathbf{D}_{\eta(\mathcal{S}_2)}\} = \text{span}\{\hat{\mathbf{D}}_{\mathcal{S}_2}\}. \tag{3.23}$$

Let us intersect Equation 3.22 and Equation 3.23, and from Lemma 3 (see below) it follows that

$$\text{span}\{\hat{\mathbf{D}}_{\mathcal{S}_1}\} \cap \text{span}\{\hat{\mathbf{D}}_{\mathcal{S}_2}\} = \text{span}\{\mathbf{D}_{\eta(\mathcal{S}_1)\cap\eta(\mathcal{S}_2)}\}. \tag{3.24}$$

Since $\text{span}\{\hat{\mathbf{D}}_{\mathcal{S}}\} \subseteq \text{span}\{\hat{\mathbf{D}}_{\mathcal{S}_1}\} \cap \text{span}\{\hat{\mathbf{D}}_{\mathcal{S}_2}\}$, it follows that $\text{span}\{\hat{\mathbf{D}}_{\mathcal{S}}\} \subseteq \text{span}\{\mathbf{D}_{\eta(\mathcal{S}_1)\cap\eta(\mathcal{S}_2)}\}$. The number of the elements in $\eta(\mathcal{S}_1) \cap \eta(\mathcal{S}_2)$ is $K-1$, since $\eta(p) \neq \eta(q)$, with $p \neq q$, by injectivity of η. Moreover the number of the elements in $\mathcal{S}$ is also $K-1$, which implies that

$$\text{span}\{\hat{\mathbf{D}}_{\mathcal{S}}\} = \text{span}\{\mathbf{D}_{\eta(\mathcal{S}_1)\cap\eta(\mathcal{S}_2)}\}. \tag{3.25}$$

The association $\mathcal{S} \to \eta(\mathcal{S}_1) \cap \eta(\mathcal{S}_2)$ from Equation 3.25 defines a function $\sigma : \binom{[L]}{K-1} \to \binom{[L]}{K-1}$, with property that $\text{span}\{\hat{\mathbf{D}}_{\mathcal{S}}\} = \text{span}\{\mathbf{D}_{\sigma(\mathcal{S})}\}$.

Finally, let's show that σ is injective. Suppose $\mathcal{S}, \mathcal{S}' \in \binom{[L]}{K-1}$, and $\sigma(\mathcal{S}) = \sigma(\mathcal{S}')$, it follows that

$$\text{span}\{\hat{\mathbf{D}}_{\mathcal{S}}\} = \text{span}\{\mathbf{D}_{\sigma(\mathcal{S})}\} = \text{span}\{\mathbf{D}_{\sigma(\mathcal{S}')}\} = \text{span}\{\hat{\mathbf{D}}_{\mathcal{S}'}\}. \tag{3.26}$$

As every K block columns of $\mathbf{D}$ are linear independent, and κ is injective, every K block columns of $\hat{\mathbf{D}}$ are also linear independent. From Lemma 2, it follows that $\mathcal{S} = \mathcal{S}'$, which implies σ is injective. Hence, let $\xi = \sigma^{-1}$, with properties: for all $\mathcal{S} \in \binom{[L]}{K-1}$, $\text{span}\{\mathbf{D}_{\mathcal{S}}\} = \text{span}\{\hat{\mathbf{D}}_{\xi(\mathcal{S})}\}$. ■

Lemma 2 *If any set of $K+1$ block columns of matrix $\mathbf{D} = [\mathbf{D}_1, \ldots, \mathbf{D}_L]$ are linear independent, then for $\mathcal{S}, \mathcal{S}' \in \binom{[L]}{K}$,*

$$\mathrm{span}\{\mathbf{D}_{\mathcal{S}}\} = \mathrm{span}\{\mathbf{D}_{\mathcal{S}'}\} \quad \Rightarrow \quad \mathcal{S} = \mathcal{S}'. \tag{3.27}$$

Proof: Suppose that $\mathcal{S} \neq \mathcal{S}' \in \binom{[L]}{K}$ satisfying $\mathrm{span}\{\mathbf{D}_{\mathcal{S}}\} = \mathrm{span}\{\mathbf{D}_{\mathcal{S}'}\}$. Then without loss of generality, there is an $i \in \mathcal{S}$ with $i \notin \mathcal{S}'$, but atoms $\mathbf{D}_i \in \mathrm{span}\{\mathbf{D}_{\mathcal{S}'}\}$, which implies that the $K+1$ block columns indexed by $\mathcal{S}' \cup \{i\}$ are not linear independent, a contradiction to the assumption. ■

Lemma 3 *If matrix $\mathbf{D}$ satisfies the block spark condition, then for $\mathcal{S}, \mathcal{S}' \in \binom{[L]}{K}$,*

$$\mathrm{span}\{\mathbf{D}_{\mathcal{S}\cap\mathcal{S}'}\} = \mathrm{span}\{\mathbf{D}_{\mathcal{S}}\} \cap \mathrm{span}\{\mathbf{D}_{\mathcal{S}'}\}. \tag{3.28}$$

Proof: The inclusion "$\subseteq$" is trivial, so let us prove "$\supseteq$". Suppose a block vector $\mathbf{x} \in \mathrm{span}\{\mathbf{D}_{\mathcal{S}}\} \cap \mathrm{span}\{\mathbf{D}_{\mathcal{S}_2}\}$. Express $\mathbf{x}$ as a linear combination of K atoms of $\mathbf{D}$ indexed by $\mathbf{S}$ and, separately, as a combination of K atoms of $\mathbf{D}$ indexed by $\mathcal{S}'$. By the block spark condition, these linear combinations must be identical. In particular, $\mathbf{x}$ was expressed as a linear combination of atoms of $\mathbf{D}$ indexed by $\mathcal{S} \cap \mathcal{S}'$, and thus is in $\mathrm{span}\{\mathbf{D}_{\mathcal{S}\cap\mathcal{S}'}\}$ ■

Proof of Theorem 1 when $\beta = 1$

First, we produce a set of $N = K\binom{M/\alpha}{K}^2$ vectors $\mathbf{s}_i \in \mathbb{R}^{\alpha K}$ in general linear position (*i.e.* any subset of K of them are linearly independent). One possible strategy is to produce a "Vandermonde" matrix [58]. Next, we form K-block-sparse vectors $\mathbf{z}_1, ..., \mathbf{z}_N$ by taking $\mathbf{s}_i$ for the support value of $\mathbf{z}_i$ where each possible support set is represented $K\binom{M/\alpha}{K}$ times. We claim that these $\mathbf{z}_i$ always guarantee the uniqueness of BSDL.

Proof: Suppose there exists an alternate dictionary $\hat{\mathbf{D}}$ and a set of K-block-sparse vectors $\hat{\mathbf{z}}_1, ..., \hat{\mathbf{z}}_N$ such that $\mathbf{D}\mathbf{z}_i = \mathbf{x}_i = \hat{\mathbf{D}}\hat{\mathbf{z}}_i$. As there are $K\binom{M/\alpha}{K}$ $\mathbf{x}_i$ for each support indexed by $\mathcal{S}$, the "pigeon-hole principle"[2] implies that there are at least K vectors $\hat{\mathbf{z}}_{i_1}, ..., \hat{\mathbf{z}}_{i_K}$ using the same support $\mathcal{S}'$. Thus, $\mathrm{span}\{\mathbf{x}_{i_1}, ..., \mathbf{x}_{i_K}\} \subseteq \mathrm{span}\{\hat{\mathbf{D}}_{\mathcal{S}'}\}$. By the general linear position and the block spark condition, $\mathrm{span}\{\mathbf{x}_{i_1}, ..., \mathbf{x}_{i_K}\} = \mathrm{span}\{\mathbf{D}_{\mathcal{S}}\}$. Therefore $\mathrm{span}\{\mathbf{D}_{\mathcal{S}}\} \subseteq \mathrm{span}\{\hat{\mathbf{D}}_{\mathcal{S}'}\}$. As the dimension of $\mathrm{span}\{\hat{\mathbf{D}}_{\mathcal{S}'}\}$ is less and equal to K, $\mathrm{span}\{\mathbf{D}_{\mathcal{S}}\} = \mathrm{span}\{\hat{\mathbf{D}}_{\mathcal{S}'}\}$.

By Lemma 1, Theorem 1 is proved. ■

3.2 Modeling via Block Sparsity

Let us assume that the unknown 3D structures $\mathbf{S}^\sharp$ are compressible, that is, the 3D structure in each frame (each row of $\mathbf{S}^\sharp$) can be approximated by only K basis shapes (K rows of $\mathbf{B}^\sharp$.)

[2]The pigeon-hole principle states that if n items are put into m containers, with $n > m$, then at least one container must contain more than one item [11].

Therefore, the factorization $\mathbf{S}^\sharp = \mathbf{C}^\sharp\mathbf{B}^\sharp$ results in a set of coefficients $\mathbf{C}^\sharp \in \mathbb{R}^{F\times L}$ whose rows are each K-sparse.

$$\mathbf{S}^\sharp = \mathbf{C}^\sharp\mathbf{B}^\sharp, \quad \text{s.t. } \|\mathbf{C}_i^\sharp\|_0 < K, \tag{3.29}$$

where $\|\cdot\|_0$ counts the number of active elements of argument vector/matrix and $\mathbf{C}_i^\sharp$ is the i-th row of $\mathbf{C}^\sharp$. Note that one never has access to the 3D structure $\mathbf{S}^\sharp$ a priori only the 2D projections $\mathbf{W}$. Interestingly, however, if we know $\mathbf{S}^\sharp$ is compressible then from Equation 2.19 (*i.e.* $\mathbf{\Pi} = \mathbf{M}(\mathbf{C}^\sharp \otimes \mathbf{I}_3)$), $\mathbf{\Pi}$ *must* be 2×3 block sparse as the camera matrix $\mathbf{M}$ is 2×3 block-diagonal. It is this insight that forms the crucial component of our algorithm. From a known measurement matrix $\mathbf{W}$ and desired K, L, one can factorize $\mathbf{W}^T$ through a 3×2 block sparse dictionary learning process. Note: for NRS*f*M $\mathbf{W} = \mathbf{\Pi}\mathbf{B}$, whereas for BSDL this would be expressed as $\mathbf{W}^T = \mathbf{B}^T\mathbf{\Pi}^T$ where $\mathbf{X} = \mathbf{W}^T, \mathbf{D} = \mathbf{B}^T$, and $\mathbf{Z} = \mathbf{\Pi}^T$.

Theorem 2 *If one can recover $\hat{\mathbf{B}}$ using a 3×2 BSDL such that $\mathbf{D} = \hat{\mathbf{B}}^T$ satisfies the block spark condition, then it can be shown that the transpose of $\hat{\mathbf{B}}^\sharp$ satisfies the canonical spark condition, where $\hat{\mathbf{B}}^\sharp$ is an $L \times 3P$ reshape of $\hat{\mathbf{B}}$. Further, for such BSDL to be unique, K must be less than or equal to $P/3 - 1$.*

Proof: Suppose two K-sparse vectors $\mathbf{z}_1$ and $\mathbf{z}_2$ such that $(\hat{\mathbf{B}}^\sharp)^T\mathbf{z}_1 = (\hat{\mathbf{B}}^\sharp)^T\mathbf{z}_2$. Then from the reshape, it follows that $\hat{\mathbf{B}}^T(\mathbf{z}_1 \otimes \mathbf{I}_3) = \hat{\mathbf{B}}^T(\mathbf{z}_2 \otimes \mathbf{I}_3)$. As $\hat{\mathbf{B}}^T$ satisfies the block spark condition, it follows that $\mathbf{z}_1 = \mathbf{z}_2$, therefore, $(\mathbf{B}^\sharp)^T$ satisfies the canonical spark condition. Further, the uniqueness of the BSDL factorization requires $\hat{\mathbf{B}}^T$ to satisfy the block spark condition. This implies that any $P \times 3(K + 1)$ submatrices generated by concatenating $K + 1$ block columns of $\hat{\mathbf{B}}^T$ needs to be full column rank. Consider, a counterexample for contradiction: if $K = 2$, and $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ are 3 linear dependent block columns of $\hat{\mathbf{B}}^T$. In addition, suppose any 2 of them are linear independent. Then subspace spanned by $\{\mathbf{b}_1, \mathbf{b}_2\}$ are identical to one by $\{\mathbf{b}_1, \mathbf{b}_3\}$, which breaks the block spark condition. Therefore K need to be less than or equal to $P/3 - 1$. ■

Theorem 2 actually tells us that the uniqueness of the BSDL factorization on 2D projections automatically guarantees the uniqueness of the SDL factorization on the unknown 3D structures. Interestingly, the converse is not always true. This result highlights a drawback in our proposed approach, that is, we cannot recover all compressible structures but the subsets where $\hat{\mathbf{\Pi}}$ is sufficiently sparse ($K \leq P/3 - 1$) and $\hat{\mathbf{B}}$ satisfies the block spark condition. In the experiments section, we show a strategy that can be utilized in practice to improve the incoherence of $\hat{\mathbf{B}}$ and push it to satisfy the block spark condition.

3.3 Solving via Block Sparse Dictionary Learning

3.3.1 BSDL algorithms

In this section, we describe our BSDL algorithm that adapts K-SVD [49], OMP [57] and FOCUSS [23] to the block sparse situation respectively. However, any valid BSDL method can be employed here as long as it returns a valid factorization $\mathbf{W} = \hat{\mathbf{\Pi}}\hat{\mathbf{B}}$.

Block K-SVD

Similar to regular K-SVD, block K-SVD is an iterative algorithm with 2 steps: 1) Fixing dictionary, solve block-sparse representation by block OMP or block FOCUSS, and 2) Fixing block-sparse pattern, update dictionary by SVD. The only alternation from regular K-SVD is to keep the first α singular values instead of one when updating each block columns of the dictionary. For compressible NRS*f*M, $\alpha = 3$. The techniques to get rid of local minimal reported in [49] are also valid and serve in block K-SVD.

Block OMP

To solve block sparse approximation problem, we extend regular Orthogonal Matching Pursuit (OMP) [57] to block OMP. Both of them are greedy algorithms picking the first K atoms in dictionary describing the signal best. Specifically, in each iteration, block OMP computes the inner product of residual and each dictionary atoms left, and picked the atoms corresponding to least inner product value. Then it computes coefficients, associates with chosen atoms, updates residual and repeats until the number of chosen atoms hits the known number K. Block OMP is efficient compared to block FOCUSS, but it succeeds only when the dictionary is sufficient incoherent.

Block FOCUSS

Serving the same function as block OMP, we adapt FOcal Underdetermined System Solver (FOCUSS) [23] to block FOCUSS to estimate the block sparse approximation. Block FOCUSS and FOCUSS are iterative algorithms solving the ℓ_p-norm ($p < 1$) relaxation of block sparse approximation and regular sparse approximation respectively. The only difference between them is the design of the weight matrix $\mathbf{W}_{pk}$ (refer to [23] for more detail.) Other than letting $\mathbf{W}_{pk} = \text{diag}(\mathbf{x}^{k-1})$, block FOCUSS updates $\mathbf{W}_{pk}$ by the Frobenius norm of each block in $\mathbf{x}^{k-1}$, which promotes elements in one block to be either all active or all zeros. A regularization technique [47] serves also in block FOCUSS balancing the approximation error and sparsity of estimated coefficients. Block FOCUSS can often achieve successful block-sparse estimation even in circumstances where block OMP fails. One drawback, however, is its speed as it is dramatically slower than block OMP.

Initialization

The BSDL factorization itself is inherently an NP-hard problem, therefore it is important to have a good initialization. We relax the BSDL objective using a block ℓ_1-norm, and solve the relaxed problem by Alternating Direction Method of Multipliers (ADMM) [1, 8, 10, 21]. Even though the relaxed problem is not convex either, ADMM splits the objective into several small *convex* sub-problems by introducing several auxiliary variables. A stationary point can be achieved for our ADMM initialization through the judicious choice of parameters [10].

3.3.2 Camera and Structure Recovery

As the scale of camera and size of structures are inherently relative, we simply set the camera scale σ to unity, such that $\mathbf{M}_f\mathbf{M}_f^T = \mathbf{I}_2$. Assuming that $\mathbf{W} = \hat{\Pi}\hat{\mathbf{B}}$ has a unique BSDL, from Definition 1, the corrective matrix $\mathbf{G}$ must be of form $\mathbf{G} = (\mathbf{P} \otimes \mathbf{I}_3)\mathbf{\Lambda}$. As the permutation ambiguity has no bearing on camera motion and 3D structure, we set $\mathbf{P}$ to identity, therefore $\mathbf{G} = \mathbf{\Lambda}$.

Denote $\mathbf{G}_j$ as j-th block on diagonal of $\mathbf{G}$, and $\hat{\mathbf{\Pi}}_j, \mathbf{\Pi}_j \in \mathbb{R}^{2F\times 3}$ as the j-th coloumn-triplet of $\hat{\mathbf{\Pi}}, \mathbf{\Pi}$ respectively. From the structure of corrective matrix, it follows that $\mathbf{\Pi}_j = \hat{\mathbf{\Pi}}_j\mathbf{G}_j$, for $j = 1, ..., L$. Define Ω_j as the set of indices pointing to the block $\hat{\mathbf{\Pi}}_{ij} \in \mathbb{R}^{2\times 3}$ that is active, *i.e.* $\Omega_j = \text{supp}(\hat{\mathbf{\Pi}}_j) = \{i | 1 \le i \le F, \hat{\mathbf{\Pi}}_{ij} \neq \mathbf{0}\}$. If a certain Ω_j is empty, it is implied that the corresponding atom in the dictionary has never been used. We can then decrease L, and re-learn the dictionary so that Ω_j is never empty.

From Equation 2.19 (*i.e.* $\mathbf{W} = \mathbf{M}(\mathbf{C}^\sharp \otimes \mathbf{I}_3)\mathbf{B} = \mathbf{\Pi}\mathbf{B}$), it is known that $\mathbf{\Pi}_{ij} = c_{ij}\mathbf{M}_i$, where c_{ij} is $ij-$th elements of $\mathbf{C}^\sharp$. Thus, since Ω_j can never be empty, $\hat{\mathbf{\Pi}}_{ij}\mathbf{G}_j = \mathbf{\Pi}_{ij} = c_{ij}\mathbf{M}_i$, for each $i \in \Omega_j$. From camera constraints, it follows that

$$\hat{\mathbf{\Pi}}_{ij}\mathbf{G}_j\mathbf{G}_j^T\hat{\mathbf{\Pi}}_{ij}^T = c_{ij}^2\mathbf{M}_i\mathbf{M}_i^T = c_{ij}^2\mathbf{I}_2, \quad i \in \Omega_j, \tag{3.30}$$

and for convenience, let $\mathbf{Q}_j = \mathbf{G}_j\mathbf{G}_j^T$. Since c_{ij} is unknown, let us eliminate it and rewrite Equation 3.30 as

$$(\hat{\mathbf{\Pi}}_{ij}\mathbf{Q}_j\hat{\mathbf{\Pi}}_{ij}^T)_{11} = (\hat{\mathbf{\Pi}}_{ij}\mathbf{Q}_j\hat{\mathbf{\Pi}}_{ij}^T)_{22}, (\hat{\mathbf{\Pi}}_{ij}\mathbf{Q}_j\hat{\mathbf{\Pi}}_{ij}^T)_{12} = 0, \tag{3.31}$$

where $(\cdot)_{ij}$ denotes the (i, j)-th elements. Now, denote $\mathbf{q}_j = \text{vec}(\mathbf{Q}_j)$ as the vectorization of $\mathbf{Q}_j$. Let us rewrite Equation 3.31 in a compact way with the fact that $\text{vec}(\hat{\mathbf{\Pi}}_{ij}\mathbf{Q}_j\hat{\mathbf{\Pi}}_{ij}) = (\hat{\mathbf{\Pi}}_{ij} \otimes \hat{\mathbf{\Pi}}_{ij})\mathbf{q}_j$:

$$\begin{bmatrix} \hat{\mathbf{\Pi}}_{ij} \otimes \hat{\mathbf{\Pi}}_{ij}(1,:) - \hat{\mathbf{\Pi}}_{ij} \otimes \hat{\mathbf{\Pi}}_{ij}(4,:) \\ \hat{\mathbf{\Pi}}_{ij} \otimes \hat{\mathbf{\Pi}}_{ij}(2,:) \end{bmatrix} \mathbf{q}_j = \mathbf{A}_{ij}\mathbf{q}_j = 0, \tag{3.32}$$

where $\hat{\mathbf{\Pi}}_{ij} \otimes \hat{\mathbf{\Pi}}_{ij}(k,:)$ denotes k-th row of $\hat{\mathbf{\Pi}}_{ij} \otimes \hat{\mathbf{\Pi}}_{ij}$. Stacking all such equations for all $i \in \Omega_j$, we obtain

$$\mathbf{A}_j\mathbf{q}_j = 0. \tag{3.33}$$

Circumventing the nullspace

One benefit of Equation 3.33 is that $\mathbf{A}_j \in \mathbb{R}^{2|\Omega_j|\times 9}$, where $|\Omega_j|$ is the number of elements in set Ω_j, with high possibility will be overcomplete as $F \gg L$. This result is important as it circumvents the nullspace issue faced by low-rank NRS*f*M. This null space issue can be problematic in many practical scenarios due to its sensitivity to noise. Similar to Tomasi-Kanade's method [54], we simply pick up the eigenvector corresponding to the least eigenvalue of $\mathbf{A}_j^T\mathbf{A}_j$ and then $\mathbf{Q}_k \in \mathbb{S}_+^3$ holds automatically.

Once $\mathbf{Q}_j$ is estimated, the absolute value of c_{ij} can be computed by Equation 3.30. The sign of c_{ij}, however, is not able to be determined, which actually is an inherent ambiguity without assuming any temporal prior of camera or structures. Considering equation $\mathbf{W} = \mathbf{MS}$, any block diagonal matrix blkdiag$(\pm\mathbf{I}_3)$ can be inserted between $\mathbf{M}$ and $\mathbf{S}$, but the compressibility

assumption and camera constraint still hold. Dai *et al.* [16] *breaks* their "prior-free" assertion by restricting the camera movement between frames to at most $\pm 90^\circ$ to determine the sign of c_{ij}. In our paper, however, we claim that the absolute sign of c_{ij} cannot be determined by current assumption, but the relative sign in each column can. Thus, the camera matrix and structures can be recovered but up to a sign ambiguity.

Enforcing camera consistency

Let us consider the submatrix $\mathbf{G}_j$ in isolation,

$$\hat{\mathbf{\Pi}}_{ij}\mathbf{G}_j = c_{ij}\mathbf{M}_i, \quad \text{for } i \in \Omega_j. \tag{3.34}$$

One can recover the camera matrices $\{\mathbf{M}_i\}_{i\in\Omega_j}$ by solving the system of equations above. Further, if one was to then choose another $\mathbf{G}_k$ where $j \neq k$, such that one or more indexes in Ω_j are shared with Ω_k, one can equally recover the camera matrices $\{\mathbf{M}_i^*\}_{i\in\Omega_k}$. An inconsistency arises, however, such that we cannot guarantee that

$$\mathbf{M}_i^* = \mathbf{M}_i, \quad \text{for} \quad i \in \Omega_j \cap \Omega_k. \tag{3.35}$$

This inconsistency does not just occur across pairs of submatrices within $\mathbf{G}$, but actually across all possible submatrices of $\mathbf{G}$ with overlapping active blocks. We attempt to resolve this inconsistency in a recursive manner by solving for an orthonormal matrix $\mathbf{H}_k$ such that $\mathbf{M}_i^*\mathbf{H}_k = \mathbf{M}_i$. First, we choose an arbitrary $\mathbf{G}_j$ (typically the one with most active blocks) and solve for the cameras $\{\mathbf{M}_i\}_{i\in\Gamma}$, where we initially set $\Gamma = \Omega_j$. Then we choose a $\mathbf{G}_k$ whose $|\Gamma \cap \Omega_k|$ is largest. We solve for the cameras $\{\mathbf{M}_i^*\}_{i\in\Omega_k}$, and then find an orthonormal $\mathbf{H}_k$ such that,

$$\underset{\mathbf{H}_k,\boldsymbol{\eta}}{\operatorname{argmin}} \sum_{i\in\Gamma\cap\Omega_k} \|\mathbf{M}_i - \eta_i\mathbf{M}_i^*\mathbf{H}_k\|_F \quad \text{s.t. } \mathbf{H}_k^T\mathbf{H}_k = \mathbf{I}, \eta_i = \{+1, -1\}, \tag{3.36}$$

where η_i contains the relative sign of elements in $\mathbf{C}^\sharp$ for Γ. For the element in $\mathbf{C}^\sharp$ that are not explicitly defined through $\boldsymbol{\eta}$, we set them arbitrarily to be positive. We then update $\Gamma \leftarrow \Gamma \cup \Omega_k$ and repeat the process until all cameras and relative signs in $\mathbf{C}^\sharp$ are known. The structure matrix $\mathbf{S}$ is then recovered by $(\mathbf{C}^\sharp \otimes \mathbf{I}_3)\mathbf{H}^{-1}\mathbf{G}^{-1}\mathbf{B}$, where $\mathbf{H}$ is a matrix with $\mathbf{H}_1, ..., \mathbf{H}_L$ on main diagonal.

3.4 Experiments

3.4.1 Compressibility

Our first experiment explores the compressibility of real 3D structures from the CMU Motion Capture dataset, where we learned various dictionaries with different dictionary size L and sparsity level K. Figure 3.1 clearly shows that the real 3D structures are modeled well by our compressibility assumption and the coherence of the learned dictionary is being controlled by balancing the approximation error. This result offers a strategy to achieve a unique BSDL factorization at the cost of approximating structures less precisely, which extends the application of our method.

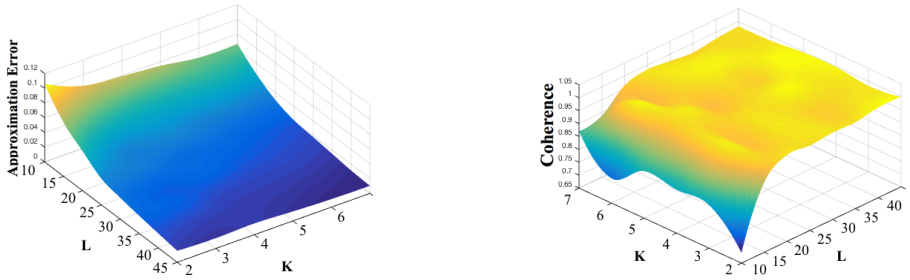


Figure 3.1: The results of SDL factorization for Motion-4 by Subject-5 in CMU Motion Capture. **Left:** The approximation error. **Right:** The coherence of learned dictionary. With the decrease of K and L, the coherence of learned dictionary becomes better at the cost of approximating structures less precisely.

3.4.2 Recovering temporal order

In Figure 3.2 we demonstrated that the sparse codes recovered using our method have a natural temporal coherence. This indicates our prior-less approach could be useful for the recovery of the temporal order of 3D structures in future applications. The full analysis of this phenomena is outsize of the scope of this paper.

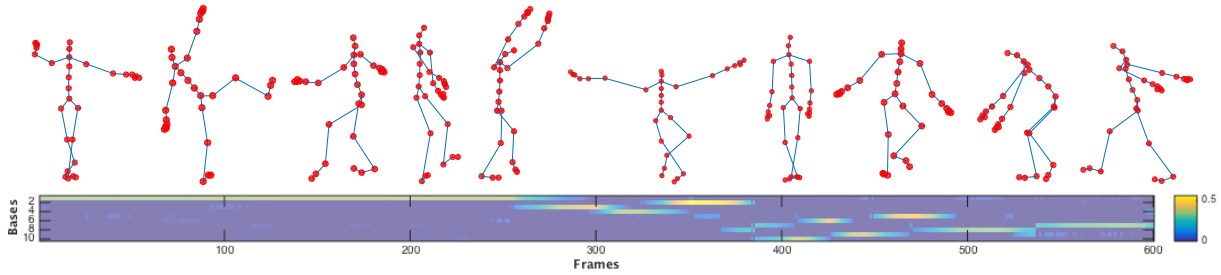


Figure 3.2: **Top:** 10 learned basis structures for Motion-4 by Subject-5 in CMU Motion Capture when $K = 2, L = 10$. These bases are learned from 3D shape sequences and identical to those learned from 2D image sequences, due to the uniqueness of BSDL. **Bottom:** The visualization of coefficients. The coefficients of each atoms varies gradually in a shape of Gaussian distribution, which reveals the temporal information of video sequence. It is not used in NRSfM, but may be useful for recovering the temporal order of 3D structure in future applications.

3.4.3 High-rank performance

To verify the performance of the proposed method on high-rank and full-rank structures, we conducted experiments with synthetic data where the rank of structures is easily controlled. We utilized Dai *et al.*'s work as a baseline, which demonstrated that it outperforms other low-rank NRSfM methods in [16].Note that for a fair comparison, we visit all possible rank k for [16] to ensure a best baseline estimation.

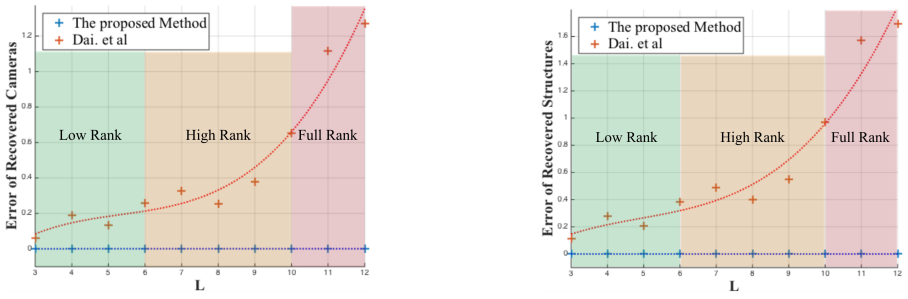


Figure 3.3: **Left:** The error of estimated camera matrix. **Right:** The error of estimated structures. The error matrices follows [4, 16, 24]. Our methods obtained nearly perfectly results irrespective to rank of structures.

The compressible structure $\mathbf{S}$, with 100 frames and 30 points in each frames, are generated by random dictionary of size L, such that $\text{rank}(\mathbf{S}) = 3L$. We repeat the proposed method as well as Dai *et al.*'s method 50 times for each L from 3 to 12. The results are summarized in Figure 3.3. It is seen that our method works perfectly and robustly on structures with any rank, while the low-rank NRSfM fails in high-rank and full-rank situations. Moreover, even in low-rank situation, the proposed method outperforms the Dai *et al.*'s method.

3.4.4 Noise performance

To evaluate the performance under noise, we repeat the experiments on low-rank structures (with $L = 5$) at different noise ratios, defined as $\frac{\|W - W_0\|_F}{\|W_0\|_F}$. The Figure 5.3 demonstrates that our method is sensitive to noise. However, it still works no worse than Dai *et al.*'s method even at high noise ratios.

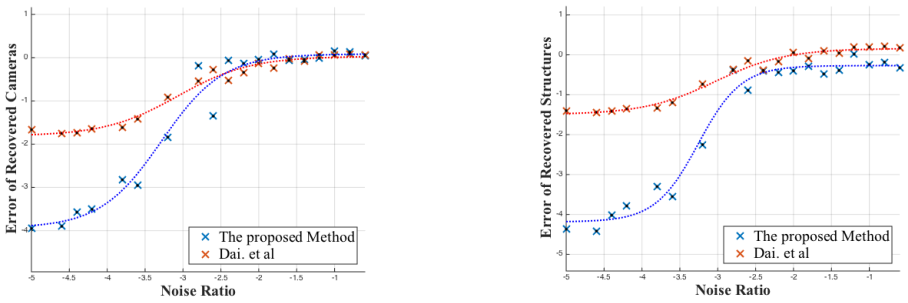


Figure 3.4: **Left:** The error of estimated camera matrix. **Right:** The error of estimated structures. Both x- and y-axis are in logarithm space. Our method is sensitive to noise, while it still works no worse than the baseline even at high noise.

3.4.5 Practical performance

The proposed method is evaluated on real compressible structures: Motion-4, -5, -6, -7, -8 by Subject-5, and Motion-2, -4 by Subject-1, Motion-5 by Subject-2, Motion-3, -4 by Subject-3 and Motion-13 by Subject-6 in CMU Motion Captures, and a Shark sequence in [56]. The visual

evaluation shows that our method obtains impressive results in Figure 3.5, 3.6, 3.7, while it fails in Figure 3.8. Actually, this failure is able to be forecast even without ground truth. The coherence of the learned dictionary for sequence Shark is too poor to guarantee the uniqueness of the BSDL factorization. This insight offers an effective way to predict the reconstructibility of 3D structure when the ground truth structure are not available in practice.

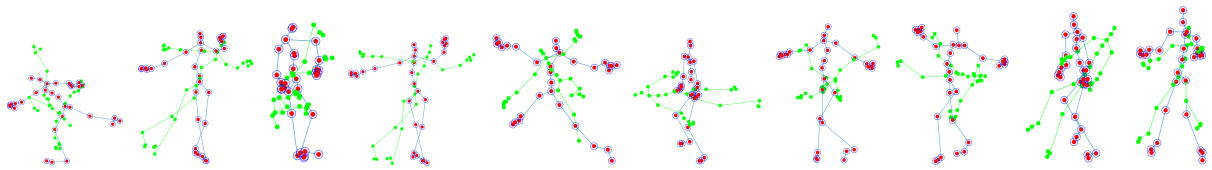


Figure 3.5: Random Sampled frames from Motion-4,-5,-6 by Subject-5.

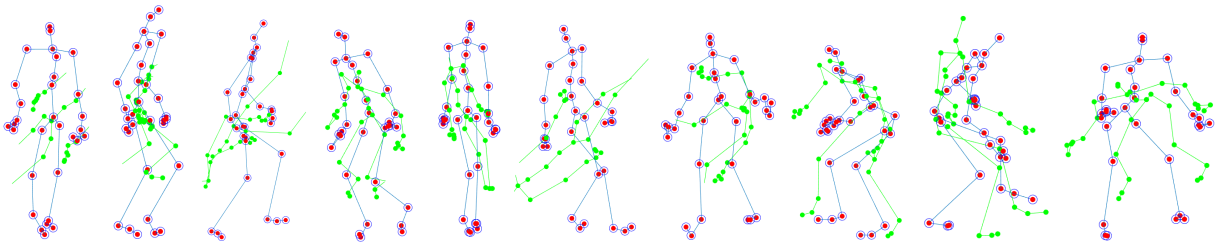


Figure 3.6: Random Sampled frames from Motion-3,-4 by Subject-3 and Motion-13 by Subject-6.

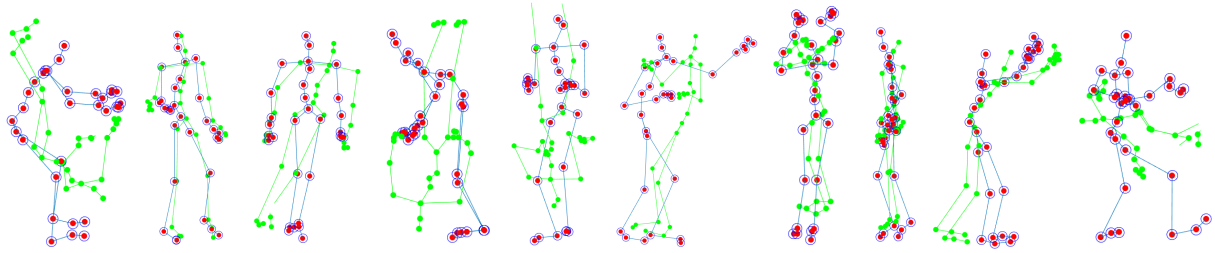


Figure 3.7: Random Sampled frames from Motion-2,-4 by Subject-1 and Motion-5 by Subject-2.

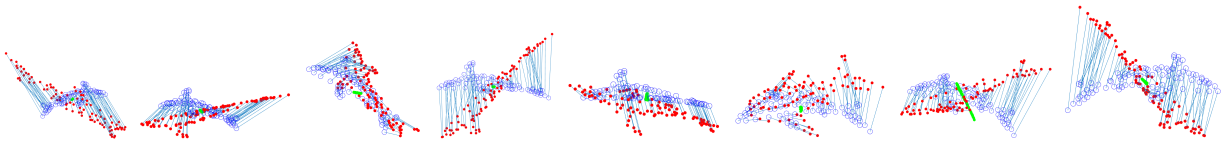


Figure 3.8: Random Sampled frames from Shark sequence.

Chapter 4

Structure from Categories

This chapter introduces the method of Structure from Category (S*f*C) to infer 3D structures of objects in images stemming from the same object category. S*f*C is built upon the insight that the shape space describing an object category (*e.g*. aeroplane) is inherently non-rigid, even though individual instances of the category may be rigid. In other words, the shape of each instance can be modeled as a deformation from its category's general shape. Based on this observation, we frame S*f*C through an augmented sparse shape-space model that estimates the 3D shape of an object as a sparse linear combination of a set of rotated shape bases.

The proposed S*f*C is a *generic* and *prior-less* 3D reconstruction algorithm. Unlike current NRS*f*M methods which are mainly limited to very few deformable objects (*e.g*. human body and face), S*f*C can be generally applied on any object category, due to the non-rigid assumption of objects shape space. Moreover, all parameters including shape bases, sparse coefficients and (scaled) camera motion are jointly learned though an iterative manner, with *no* constraint on camera motion, 3D shape structure, temporal order and deformation patterns (prior-less). Being generic and prior-less with no learning procedure in advance offers robust large scale 3D reconstruction for unseen object images and categories.

4.1 Problem Formulation

Inspired by the augmented sparse shape-space model [69], the 3D shape of instance f, $\mathbf{S}_f \in \mathbb{R}^{3\times P}$, can be well-approximated as a linear combination of a set of L rotated 3D *shape bases* $\{\mathbf{B}_l\}_{l=1}^{L}$:

$$\mathbf{S}_f = \sum_{l=1}^{L} c_{fl}\mathbf{R}_{fl}\mathbf{B}_l, \tag{4.1}$$

where $\mathbf{B}_l \in \mathbb{R}^{3\times P}$, represented by the location of P key points in the 3D space, describe the object's shape space. $\mathbf{R}_{fl} \in \mathbb{R}^{3\times 3}$ and c_{fl} respectively refer to the rotation matrix and the coefficient of the l-th shape base and the f-th instance.

Given a set of F instances of the same object category, Eq(4.1) can be written as :

$$\begin{bmatrix} \mathbf{S}_1 \\ \vdots \\ \mathbf{S}_F \end{bmatrix} = \begin{bmatrix} c_{11}\mathbf{R}_{11} & \cdots & c_{1L}\mathbf{R}_{1L} \\ \vdots & \vdots & \vdots \\ c_{F1}\mathbf{R}_{F1} & \cdots & c_{FL}\mathbf{R}_{FL} \end{bmatrix} \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_L \end{bmatrix}. \tag{4.2}$$

The projection of $\{\mathbf{S}_f\}_{f=1}^F$ into the image plane, $\{\mathbf{W}_f\}_{f=1}^F$, is computed by:

$$\begin{bmatrix} \mathbf{W}_1 \\ \vdots \\ \mathbf{W}_F \end{bmatrix} = \begin{bmatrix} \mathbf{K}\mathbf{S}_1 \\ \vdots \\ \mathbf{K}\mathbf{S}_F \end{bmatrix} + \begin{bmatrix} \mathbf{T}_1 \\ \vdots \\ \mathbf{T}_F \end{bmatrix} = \begin{bmatrix} c_{11}\mathbf{K}\mathbf{R}_{11} & \cdots & c_{1L}\mathbf{K}\mathbf{R}_{1L} \\ \vdots & \vdots & \vdots \\ c_{F1}\mathbf{K}\mathbf{R}_{F1} & \cdots & c_{FL}\mathbf{K}\mathbf{R}_{FL} \end{bmatrix} \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_L \end{bmatrix} + \begin{bmatrix} \mathbf{T}_1 \\ \vdots \\ \mathbf{T}_F \end{bmatrix}, \tag{4.3}$$

where we denote translation by $\mathbf{T}_f$, and projection matrix by $\mathbf{K}$. $\mathbf{W}_f \in \mathbb{R}^{2\times P}$ contains the 2D locations of P key points projected into the image plane. We consider weak-perspective cameras, which is a reasonable assumption for objects whose variation in depth is small compared to their distance from the camera, *i.e.* $\mathbf{K} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$.

Denoting $\mathbf{M}_{fl} = c_{fl}\mathbf{K}\mathbf{R}_{fl}$, Eq(4.3) can be written as:

$$\begin{bmatrix} \mathbf{W}_1 \\ \vdots \\ \mathbf{W}_F \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{11} & \cdots & \mathbf{M}_{1L} \\ \vdots & \vdots & \vdots \\ \mathbf{M}_{F1} & \cdots & \mathbf{M}_{FL} \end{bmatrix} \begin{bmatrix} \mathbf{B}_1 \\ \vdots \\ \mathbf{B}_L \end{bmatrix} + \begin{bmatrix} \mathbf{T}_1 \\ \vdots \\ \mathbf{T}_F \end{bmatrix} \tag{4.4}$$

and more concisely in the matrix form as,

$$\mathbf{W} = \mathbf{M}\mathbf{B} + \mathbf{T} \tag{4.5}$$

The goal of S*f*C is to jointly compute $\mathbf{M}$ (projected rotation matrix), $\mathbf{B}$ (shape bases), and $\mathbf{T}$ (translation), using $\mathbf{W}$ (location of corresponding key points in a set of 2D images). This is performed by minimizing the *projection error* subject to the scaled orthogonality constraint on each $\mathbf{M}_{fl}$ and the sparsity constraint on the number of shape bases activated for each instance, which is framed as:

$$\begin{aligned} \min_{\mathbf{M},\mathbf{B},\mathbf{T}} \quad & \frac{1}{2}\Big\|\mathbf{\Gamma} \odot \big(\mathbf{M}\mathbf{B} + \mathbf{T}\big) - \mathbf{W}\Big\|_F^2 + \lambda\|\mathbf{C}\|_1 \\ \text{s.t.} \quad & \mathbf{M}_{fl}\mathbf{M}_{fl}^T = c_{fl}^2\mathbf{I}_2,\ f = 1, ..., F,\ l = 1, ..., L, \\ & \|\mathbf{B}_l\|_F = 1,\ f = 1, ..., F, \end{aligned} \tag{4.6}$$

where $\mathbf{C} = [c_{fl}]$ and $\|\mathbf{C}\|_1$ computes the summation of ℓ_1-norm of each row in $\mathbf{C}$. $\|.\|_F$ denotes the Frobenius norm of a matrix, and $\mathbf{\Gamma}$ is a binary matrix that encodes the visibility (1) and occlusion (0) of each key point. The objective in Eq(4.6) is non-convex due to the multiplication of $\mathbf{M}$ and $\mathbf{B}$ and the orthogonality constraint on each $\mathbf{M}_{fl}$. To make the problem more convex, we utilize the relaxation strategy proposed by Zhou *et al*. [69] that eliminates the orthogonality constraint by replacing it with a spectral norm regularization. In such case, Eq(4.6) is relaxed as:

$$\begin{aligned} \min_{\mathbf{M},\mathbf{B},\mathbf{T}} \quad & \frac{1}{2}\Big\|\mathbf{\Gamma} \odot \big(\mathbf{M}\mathbf{B} + \mathbf{T}\big) - \mathbf{W}\Big\|_F^2 + \lambda \sum_{l,f} \|\mathbf{M}_{fl}\|_2 \\ \text{s.t.} \quad & \|\mathbf{B}_l\|_F = 1,\ l = 1, ..., L, \end{aligned} \tag{4.7}$$

where $\|.\|_2$ here is the spectral norm of a matrix. The Alternating Direction Method of Multipliers (ADMM) [8] will be utilized to solve the objective in Eq(4.7).

4.2 Optimization via ADMM

Our proposed approach for solving Eq(4.7) involves the introduction of two auxiliary variables $\mathbf{Z}$ and $\mathbf{A}$. In this case, Eq(4.7) can be identically expressed as:

$$\begin{aligned} \min_{\mathbf{M},\mathbf{B},\mathbf{T},\mathbf{Z},\mathbf{A}} \quad & \frac{1}{2}\Big\|\mathbf{\Gamma} \odot \big(\mathbf{ZB}+\mathbf{T}\big)-\mathbf{W}\Big\|_F^2 + \lambda \sum_{f,l} \|\mathbf{M}_{fl}\|_2 \\ \text{s.t.} \quad & \mathbf{M}=\mathbf{Z},\ \mathbf{A}=\mathbf{B}, \\ & \|\mathbf{A}_l\|_F = 1,\ l=1,...,L. \end{aligned} \tag{4.8}$$

The augmented Lagrangian of Eq(4.8) is formulated as:

$$\begin{aligned} \mathcal{L}(\mathbf{M},\mathbf{Z},\mathbf{B},\mathbf{A},\mathbf{T},\mathbf{\Lambda},\mathbf{\Pi}) = & \frac{1}{2}\Big\|\mathbf{\Gamma} \odot \Big(\mathbf{ZB}+\mathbf{T}\Big)-\mathbf{W}\Big\|_F^2 \\ & + \lambda \sum_{f,l} \|\mathbf{M}_{fl}\|_2 + \frac{\mu}{2}\Big\|\mathbf{M}-\mathbf{Z}\Big\|_F^2 + \frac{\rho}{2}\Big\|\mathbf{A}-\mathbf{B}\Big\|_F^2 \\ & + \Big\langle \mathbf{\Lambda}, \mathbf{M}-\mathbf{Z}\Big\rangle_F + \Big\langle \mathbf{\Pi}, \mathbf{A}-\mathbf{B}\Big\rangle_F \\ \text{s.t.} \quad & \|\mathbf{A}_l\|_F = 1,\ l=1,...,L, \end{aligned} \tag{4.9}$$

where $\mathbf{\Pi}, \mathbf{\Lambda}$ are Lagrangian multipliers, and μ, ρ are penalty factors to control the convergence behavior, and $< \cdot, \cdot >_F$ is Frobenius product of two matrices.

Particularly, we utilize the Alternating Direction Method of Multipliers (ADMM) to optimize Eq(4.9). ADMM decomposes an objective into several sub-problems, and iteratively solves them till convergence occurs [8]. We detail each of the sub-problem as follows.

Sub-problem $\mathbf{M}$

$$\begin{aligned} \mathbf{M}^* &= \operatorname{argmin} \mathcal{L}(\mathbf{M};\mathbf{Z},\mathbf{B},\mathbf{A},\mathbf{T},\mathbf{\Lambda},\mathbf{\Pi}) \\ &= \operatorname{argmin} \lambda \sum_{f,l} \|\mathbf{M}_{fl}\|_2 + \frac{\mu}{2}\Big\|\mathbf{M}-\mathbf{Z}\Big\|_F^2 + \Big\langle \mathbf{\Lambda}, \mathbf{M}-\mathbf{Z}\Big\rangle_F \end{aligned} \tag{4.10}$$

Following [69], each $\mathbf{M}_{fl}$ can be computed by using soft-thresholding:

$$\mathbf{M}^*_{fl} = \mathcal{D}_{\lambda/\mu}\left(\mathbf{Z}_{fl} - \frac{1}{\mu}\mathbf{\Lambda}_{fl}\right) \tag{4.11}$$

Sub-problem $\mathbf{Z}$

$$\begin{aligned} \mathbf{Z}^* = \operatorname{argmin} \quad & \mathcal{L}(\mathbf{Z};\mathbf{M},\mathbf{B},\mathbf{A},\mathbf{T},\mathbf{\Lambda},\mathbf{\Pi}) \\ = \operatorname{argmin} \quad & \frac{1}{2}\Big\|\mathbf{\Gamma} \odot \Big(\mathbf{ZB}+\mathbf{T}\Big)-\mathbf{W}\Big\|_F^2 + \frac{\mu}{2}\Big\|\mathbf{M}-\mathbf{Z}\Big\|_F^2 + \Big\langle \mathbf{\Lambda}, \mathbf{M}-\mathbf{Z}\Big\rangle_F \end{aligned} \tag{4.12}$$

$\mathbf{Z}^*$ is updated iteratively by gradient descent several times, where the gradient is $\Big(\mathbf{\Gamma} \odot \mathbf{\Gamma} \odot \big(\mathbf{ZB} + \mathbf{T}\big) - \mathbf{W}\Big)\mathbf{B}^T - \mathbf{\Lambda} + \mu(\mathbf{Z} - \mathbf{M})$. If $\mathbf{\Gamma}$ is all ones (all key points are visible), we can compute $\mathbf{Z}^*$ easily by pseudo-inverse:

$$\mathbf{Z}^* = \Big(\mathbf{B}\mathbf{B}^T + \mu\mathbf{I}\Big)^\dagger \Big(\big(\mathbf{W} - \mathbf{T}\big)\mathbf{B}^T + \mathbf{\Lambda} + \mu\mathbf{M}\Big) \tag{4.13}$$

Sub-problem B

$$\begin{aligned} \mathbf{B}^* = \operatorname{argmin} \quad & \mathcal{L}(\mathbf{B}; \mathbf{M}, \mathbf{Z}, \mathbf{A}, \mathbf{T}, \mathbf{\Lambda}, \mathbf{\Pi}) \\ = \operatorname{argmin} \quad & \frac{1}{2}\Big\|\mathbf{\Gamma} \odot \big(\mathbf{ZB} + \mathbf{T}\big) - \mathbf{W}\Big\|_F^2 \\ & + \Big\langle \mathbf{\Pi}, \mathbf{A} - \mathbf{B} \Big\rangle_F + \frac{\rho}{2}\Big\|\mathbf{A} - \mathbf{B}\Big\|_F^2 \end{aligned} \tag{4.14}$$

Each column of $\mathbf{B}$, corresponded to each key point p, can be independently optimized as:

$$\begin{aligned} \mathbf{B}_p^* = \operatorname{argmin} \quad & \frac{1}{2}\Big\|\operatorname{diag}\big(\mathbf{\Gamma}_p\big)\mathbf{Z}\mathbf{B}_p + \mathbf{\Gamma}_p \odot \mathbf{T}_p - \mathbf{W}_p\Big\|_2^2 \\ & + \Big\langle \mathbf{\Pi}_p, \mathbf{A}_p - \mathbf{B}_p \Big\rangle_F + \frac{\rho}{2}\Big\|\mathbf{A}_p - \mathbf{B}_p\Big\|_2^2 \end{aligned} \tag{4.15}$$

We utilized a gradient descent solver to optimize Eq(4.15) when ρ is small (Eq(4.15) is poorly conditioned). Once ρ becomes big enough, we solve $\mathbf{B}_p$ directly using a least square solver. If all entries of $\mathbf{\Gamma}$ are one, *i.e.* all key points are visible, $\mathbf{B}^*$ can efficiently computed by:

$$\mathbf{B}^* = \Big(\mathbf{Z}^T\mathbf{Z} + \rho\mathbf{I}\Big)^\dagger \Big(\mathbf{Z}^T\big(\mathbf{W} - \mathbf{T}\big) + \mathbf{\Pi} + \rho\mathbf{A}\Big) \tag{4.16}$$

Sub-problem A

$$\begin{aligned} \mathbf{A}^* = \quad & \operatorname{argmin} \mathcal{L}(\mathbf{A}; \mathbf{M}, \mathbf{Z}, \mathbf{B}, \mathbf{T}, \mathbf{\Lambda}, \mathbf{\Pi}) \\ = \quad & \operatorname{argmin} \Big\langle \mathbf{\Pi}, \mathbf{A} - \mathbf{B} \Big\rangle_F + \frac{\rho}{2}\Big\|\mathbf{A} - \mathbf{B}\Big\|_F^2 \\ \text{s.t.} \quad & \big\|\mathbf{A}_l\big\|_F = 1,\ l = 1, ..., L. \end{aligned} \tag{4.17}$$

The optimal solution for Eq(4.17) can be obtained as [10],

$$\mathbf{A}_l^* = \frac{\mathbf{B}_l - 1/\rho\mathbf{\Pi}_l}{\big\|\mathbf{B}_l - 1/\rho\mathbf{\Pi}_l\big\|_F} \tag{4.18}$$

Sub-problem T

$$\mathbf{T}^* = \operatorname{argmin} \mathcal{L}(\mathbf{T}; \mathbf{M}, \mathbf{Z}, \mathbf{B}, \mathbf{A}, \mathbf{\Lambda}, \mathbf{\Pi}) = \operatorname{argmin} \frac{1}{2}\Big\|\mathbf{\Gamma} \odot \big(\mathbf{ZB} + \mathbf{T}\big) - \mathbf{W}\Big\|_F^2. \tag{4.19}$$

Since all columns of $\mathbf{T} \in \mathbb{R}^{2F\times P}$, $\boldsymbol{\tau}$'s, are identical, we compute a $\boldsymbol{\tau} \in \mathbb{R}^{2F\times 1}$ by minimizing the above objective:

$$\boldsymbol{\tau}^* = \quad \operatorname{argmin} \frac{1}{2}\sum_{p=1}^{P} \left\| \boldsymbol{\Gamma}_p \odot \big(\mathbf{Z}\mathbf{B}_p + \boldsymbol{\tau}\big) - \mathbf{W}_p \right\|_2^2, \tag{4.20}$$

and optimal $\boldsymbol{\tau}$ is computed by:

$$\boldsymbol{\tau}^* = \Big(\sum_{p=1}^{P} \mathbf{W}_p - \sum_{p=1}^{P} \boldsymbol{\Gamma}_p \odot \boldsymbol{\Gamma}_p \odot \mathbf{Z}\mathbf{B}_p\Big) \oslash \Big(\sum_{p=1}^{P} \boldsymbol{\Gamma}_p \odot \boldsymbol{\Gamma}_p\Big) \tag{4.21}$$

where $\oslash$ denotes the element-wise division.

Lagrange Multiplier Update

The lagrange multipliers $\boldsymbol{\Pi}, \boldsymbol{\Lambda}$ at each iteration are updated as,

$$\begin{aligned} \boldsymbol{\Lambda}^{[i+1]} &= \boldsymbol{\Lambda}^{[i]} + \mu\big(\mathbf{M}^{[i+1]} - \mathbf{Z}^{[i+1]}\big) \\ \boldsymbol{\Pi}^{[i+1]} &= \boldsymbol{\Pi}^{[i]} + \rho\big(\mathbf{A}^{[i+1]} - \mathbf{B}^{[i+1]}\big) \end{aligned} \tag{4.22}$$

Penalty Update

Superlinear convergence of ADMM may be achieved by $\mu, \rho \to \infty$. In practice, we limit the value of μ, ρ to avoid poor condition and numerical errors. Specifically, we adopt the following update strategy:

$$\begin{aligned} \mu^{[i+1]} &= \min(\mu_{max}, \beta_1\mu^{[i]}) \\ \rho^{[i+1]} &= \min(\rho_{max}, \beta_2\rho^{[i]}) \end{aligned} \tag{4.23}$$

We found experimentally $\mu^{[0]} = 10^{-2}$, $\rho^{[0]} = 10^{-1}$, $\beta_1(\beta_2) = 1.1$, and $\mu_{max}(\rho_{max}) = 10^5$ to perform well.

4.3 Experiments

4.3.1 Evaluation setup

We compare the proposed method against the most notable NRS*f*M algorithms: Tomasi-Kanade factorization [54], and the state-of-the-art Dai *et al.*'s prior-less NRS*f*M method [16], in terms of reprojection and reconstruction errors. The reprojection error measures the accuracy of reprojected key points: $\frac{1}{F}\sum_{i=1}^{F} \|\mathbf{W}_i - \hat{\mathbf{W}}_i\|_F$. The reconstruction error, on the other hand, evaluates the quality of estimated 3D shapes: $\frac{1}{F}\sum_{i=1}^{F} \min_\kappa \|\mathbf{S}_i - \kappa\hat{\mathbf{S}}_i\|_F$. κ (scalar) handles the scale ambiguity in camera projection.

Extensive experiments are conducted to evaluate the performance of our framework using both synthetic and natural images. For the synthetic images, we downloaded 70 CAD models

of aeroplane category from Sketchup 3D warehouse [1], and manually annotated their 3D key points. The synthetic images are simply generated by projecting random poses of these 3D models under weak-perspective camera into the image plane. The PASCAL3D+ dataset [65] is used for the natural image experiment, which consists of 12 object categories, and each category comes with a set of annotated 3D CAD models and corresponding natural images. We utilize most of images from all categories except those displaying highly occluded objects. More details of the PASCAL3D+ dataset can be found in [65].

The main differences between synthetic and PASCAL3D+ images come from the camera projection and object occlusion. We utilize random *weak*-perspective projection to generate the synthetic images of the aeroplane dataset, which follows the weak-projection assumption in this paper, whilst, the camera projection in the PASCAL3D+ is perspective. Moreover, all key points in synthetic images are visible, while, some key points in the PASCAL3D+ may be occluded by object itself or other objects.

4.3.2 3D reconstruction from synthetic images

The first experiment evaluates the performance of the proposed method on synthetic images, comparing with the Tomasi-Kanade factorization [54] and Dai *et al.*'s prior-less NRS*f*M approaches [16]. The synthetic images are randomly generated from all 3D CADs of the aeroplane dataset under weak perspective projection, and these approaches are applied to reconstruct the 3D shape of each image. The predicted shapes, then, are projected into the 2D plane to compute the key points reprojection error. The result of this experiment is shown in Fig. 4.1 (top), demonstrating the superior performance of our method to the other approaches. This evaluation shows that the 3D shapes reconstructed by the proposed S*f*C not only represent the actual geometry of the objects in 3D space, but also preserve the objects' spatial configuration when projected in the image plane. The result also verifies the sensitivity of the low-rank factorization NRS*f*M algorithm, *e.g.* Dai *et al.*'s method in the real world uncontrolled circumstances, when the shape of an object can not be modeled by very few shape bases [61].

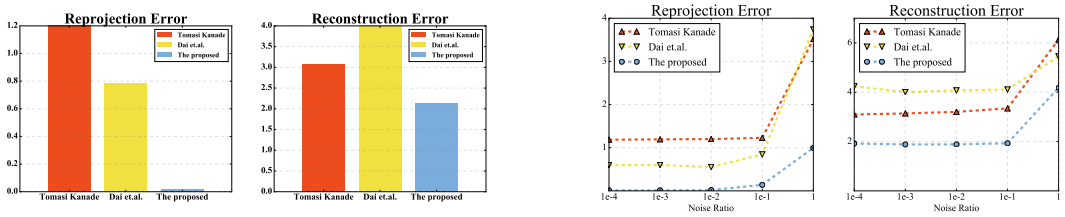


Figure 4.1: Comparing our method with Tomasi-Kanade [54] and Dai *et al.* [16] methods using the synthetic images. (left) The reconstruction and reprojection errors. (right) Noise performance.

[1] https://3dwarehouse.sketchup.com/

4.3.3 Noise performance

To analyse the robustness of our method against inaccurate key point detection, which is inevitable in real-world circumstances, we repeat the first experiment (using synthetic aeroplane images) with different levels of Gaussian noise added to the ground truth 2D locations. The average reconstruction and reprojection errors of ten random runs for each noise ratio is reported in Fig. 4.1 (bottom), showing that, compared to the other methods, the SfC method is more robust against inaccurate key point detections.

4.3.4 3D reconstruction of PASCAL3D+ dataset

To evaluate the performance of our framework over perspective projection and missing key points, we apply the proposed SfC approach to reconstruct 3D shapes of the PASCAL3D+ natural images. There is no additional shape and camera motion assumption given in this experiment, and images of all 12 object categories are taken under uncontrolled real-world circumstances. All images and their corresponding ground truth 3D CAD models are represented by a set of 2D and 3D annotated key points, respectively, which together with the predicted 3D structures and their reprojected 2D key points will be used to compute the reconstruction and reprojection errors. Since the Tomasi-Kanade factorization and Dai *et al.*'s method are not capable of handling occluded objects, we utilize the non-convex matrix completion via iterated soft thresholding [39] to predict the missing points for these approaches. This experiment is conducted over two different settings. In the first setting, we use the ground truth key points of each image provided by the PASCAL3D+. In the other setting, however, we adapt the SDM [67] approach for key point detection, and the predicted points are used for 3D reconstruction.

Using ground truth key points

The reprojection and reconstruction errors for each object category are summarized in Table 4.1 and showed by Fig. 4.2, where our approach outperforms the competitors and achieves the lowest reconstruction and reprojection error for each object category.

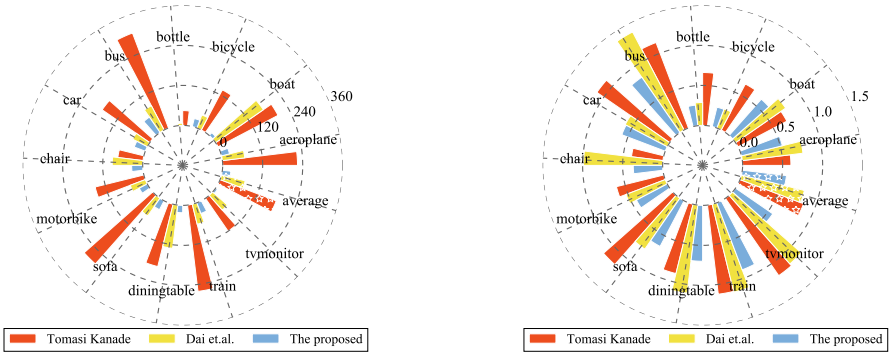


Figure 4.2: The reprojection (left) and reconstruction (right) performance of the proposed method, Tomasi-Kanade factorization [54] and Dai *et al.*'s method [16] on natural images (the PASCAL3D+ dataset) with ground truth key points.

category	key points	Reprojection Error			Reconstruction Error		
		Tomasi Kanade	Dai *et al*.	Our method	Tomasi Kanade	Dai *et al*.	Our Method
aeroplane	GT	224.3925	67.7078	**24.5695**	0.6035	0.7631	**0.5257**
	detected	364.7172	282.5179	**251.0064**	0.7986	0.7465	**0.6223**
boat	GT	202.9794	174.2009	**11.1862**	0.6892	0.7609	**0.6061**
	detected	150.7320	171.5790	**133.1670**	0.7844	0.8531	**0.7497**
bicycle	GT	135.9651	41.8621	**24.7112**	0.6490	0.2568	**0.2495**
	detected	295.2249	223.5721	**207.6959**	0.7327	0.6695	**0.6351**
bottle	GT	44.4231	6.4836	**2.8315**	0.6609	0.2865	**0.2590**
	detected	108.4824	**68.6833**	69.7238	0.7087	0.4220	**0.3812**
bus	GT	304.8072	82.1719	**56.0355**	1.1427	1.3839	**0.8396**
	detected	564.3329	311.0550	**264.9117**	1.4164	1.3924	**1.1562**
car	GT	173.6506	49.5333	**35.4720**	1.1062	0.5943	**0.5808**
	detected	265.4429	173.8730	**138.6603**	0.9959	0.9636	**0.8242**
chair	GT	75.9437	91.5107	**33.0905**	0.3958	0.9887	**0.3671**
	detected	194.7178	136.9023	**117.6726**	1.0985	1.0511	**0.9338**
motorbike	GT	150.6358	48.3516	**27.1717**	0.6096	0.5252	**0.4344**
	detected	464.8820	280.3500	**264.5549**	0.7333	0.7185	**0.6887**
sofa	GT	274.9890	64.2714	**30.0575**	1.1561	0.7727	**0.6438**
	detected	416.9723	253.0140	**196.6783**	1.1198	1.1617	**1.0126**
diningtable	GT	192.5072	130.5157	**21.9391**	0.8924	1.1084	**0.6982**
	detected	258.3700	110.2296	**103.4765**	1.2404	1.1124	**1.0107**
train	GT	260.5996	61.7900	**34.2347**	1.1215	1.1316	**0.8957**
	detected	457.0754	296.3881	**213.2750**	1.2568	1.2728	**1.1799**
tvmonitor	GT	119.8794	59.2110	**6.6706**	1.1740	1.1454	**0.5653**
	detected	277.1977	100.6167	**60.0780**	0.9307	1.0412	**0.7516**
average	GT	180.0644	73.1342	**25.6642**	0.8501	0.8098	**0.5554**
	detected	318.1790	200.7318	**168.4084**	0.9847	0.9504	**0.8288**

Table 4.1: Reprojection and Reconstruction errors obtained by Tomasi Kanade factorization [54], Dai *et al*.'s method [16], and our method using ground truth key points (GT) and detected key points (detected).

Using predicted key points

We adapt the Supervised Descent Method (SDM) [67], originally proposed for the task of facial landmarks alignment, to detect key points of generic objects within natural images. The main assumption of the SDM is that training samples fall into a Domain of Homogeneous Descent (DHD)[2], due to their limited pose space and appearance variation [68]. This assumption, however, is rarely valid in an object category with large intra-class appearance and pose variations that lies in multiple DHDs. To deal with this situation, we propose to employ a subset of training images with homogeneous gradient directions to train an SDM in an "on-the-fly" manner. Particularly, given a test image, we use fc_7 feature from the ConvNet [50] to retrieve its M most similar samples from training images and use them to train an SDM. The training set is generated by adding Gaussian noise to the ground truth locations. After training the SDM regressors, we run them independently from M different initializations (the ground truth landmark locations of the M retrieved samples). This returns M sets of predicted key points, which will be further pruned by the mean-shift algorithm. More details of SDM training/testing can be found in [67].

The results are shown in Fig. 4.3 and Table 4.1. For both two settings, using ground truth and predicted key points, our method achieves the best reconstruction and reprojection performance.

[2]A DHD refers to optimization spaces of a function that share similar directions of gradients.

The results also state that the performance of using ground truth key points is much better than the detected key points. Some qualitative results are shown in Fig. 4.4, illustrating the 3D reconstruction of two instances of each object category using ground truth key points and detected key points respectively. During the experiments, we observed that most of the failure cases are caused by severe perspective effect (*e.g.* train), missing key points (*e.g.* sofa), and inaccurate key point detection (*e.g.* chair).

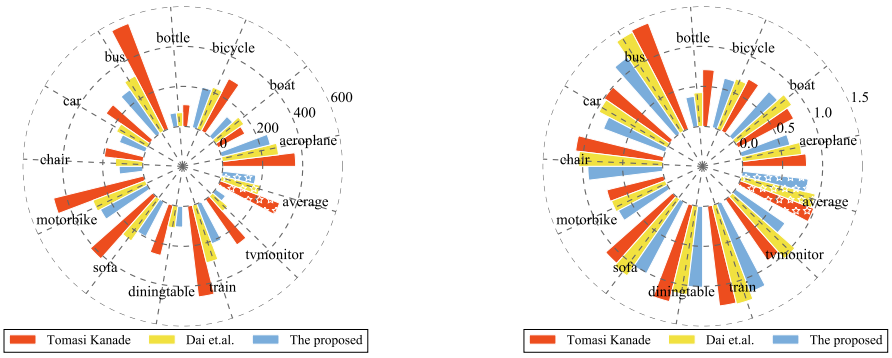


Figure 4.3: The reprojection (left) and reconstruction (right) performance of the proposed method, Tomasi-Kanade factorization [54] and Dai *et al.*'s method [16] on natural images (the PASCAL3D+ dataset) with detected key points.

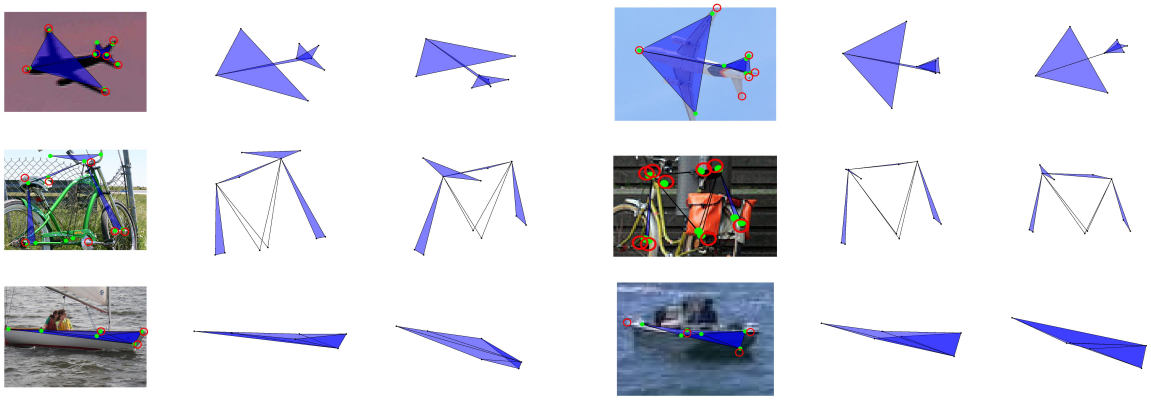


Figure 4.4: Visual evaluation of estimated structures for every category including aeroplane, bicycle, boat, bottle, bus, car, chair, diningtable, motorbike, sofa, train, and tvmonitor. The first 3 columns use ground truth key points, while the last 3 columns use detected key points. In each triplet columns, the left columns show the images, projection of estimated 3D shapes, projection of estimated landmarks (green), and the ground truth landmarks (red, some are missing due to occlusion); The middle ones show the estimated 3D shapes in the same viewpoint as camera; The right ones show a new viewpoint of the estimated 3D shapes. Two failure cases are shown in red. Best viewed in color.

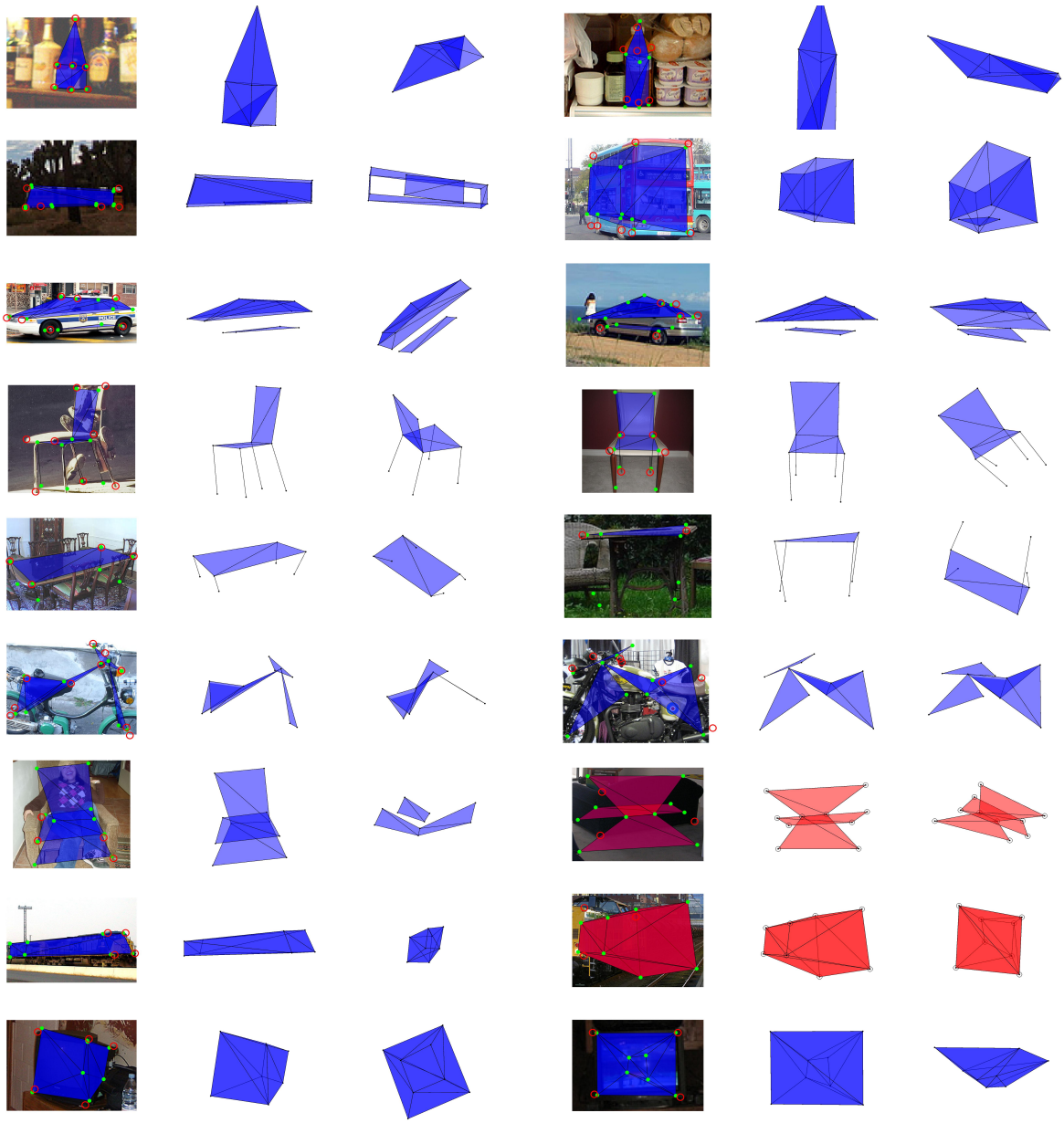


Figure 4.5: Continue Figure 4.4.

Chapter 5

Proposed Work and Extensions

From the previous chapters, it is clear that current NRS*f*M algorithms [15, 34, 35] are mainly utilizing classical optimization algorithms (*e.g*. ADMMs, K-SVD) to find the optimal solution. This results in the difficulty of processing large-scale image sequences, limiting their ability to reliably model complex shape variations. This additionally hinders their ability to generalize to unseen images. In this chapter, we start with a recent innovation to build conduit between the classical sparse dictionary learning and Deep Neural Networks (DNNs). Then we extend this to the block sparse scenario. Finally we show how to use a DNN to solve NRS*f*M problem.

5.1 Sparse Dictionary Learning and Deep Neural Network

To be consistent with symbols within this chapter, we revisit the sparse dictionary learning problem. Sparse dictionary learning can be considered as an unsupervised learning task and divided into two sub-problems: (i) dictionary learning, and (ii) sparse code recovery. Let us consider sparse code recovery problem, where we estimate a sparse representation $\mathbf{z}$ for a measurement vector $\mathbf{x}$ given the dictionary $\mathbf{W}$ *i.e*.

$$\min_{\mathbf{z}} \|\mathbf{x} - \mathbf{W}\mathbf{z}\|_2^2 \quad \text{s.t. } \|\mathbf{z}\|_0 < \lambda, \tag{5.1}$$

where λ related to the trust region controls the sparsity of recovered code. One classical algorithm to recover the sparse representation is Iterative Shrinkage and Thresholding Algorithm (ISTA) [7, 17, 48]. ISTA iteratively executes the following two steps with $\mathbf{z}^{[0]} = \mathbf{0}$:

$$\mathbf{v} = \mathbf{z}^{[i]} - \alpha\mathbf{W}^T(\mathbf{W}\mathbf{z}^{[i]} - \mathbf{x}), \tag{5.2}$$

$$\mathbf{z}^{[i+1]} = \operatorname*{argmin}_{\mathbf{u}} \frac{1}{2}\|\mathbf{u} - \mathbf{v}\|_2^2 + \tau\|\mathbf{u}\|_1, \tag{5.3}$$

which first uses the gradient of $\|\mathbf{x} - \mathbf{W}\mathbf{z}\|_2^2$ to update $\mathbf{z}^{[i]}$ in step size α and then finds the closest sparse solution using an ℓ_1 convex relaxation. It is well known in literature that the second step has a closed-form solution using soft thresholding operator. Therefore, ISTA can be summarized as the following recursive equation:

$$\mathbf{z}^{[i+1]} = h_\tau\big(\mathbf{z}^{[i]} - \alpha\mathbf{W}^T(\mathbf{W}\mathbf{z}^{[i]} - \mathbf{x})\big), \tag{5.4}$$

where h_τ is a soft thresholding operator and τ is related to λ for controlling sparsity.

Recently, Papyan [42] proposed to use ISTA and sparse coding to reinterpret feed-forward neural networks. They argue that feed-forward passing a single-layer neural network $\mathbf{z} = \mathrm{ReLU}(\mathbf{W}^T\mathbf{x} - b)$ can be considered as one iteration of ISTA when $\mathbf{z} \geq 0, \alpha = 1$ and $\tau = b$. Based on this insight, the authors extend this interpretation to feed-forward neural network with n layers

$$\begin{aligned} \mathbf{z}_1 &= \mathrm{ReLU}(\mathbf{W}_1^T\mathbf{x} - b_1) \\ \mathbf{z}_2 &= \mathrm{ReLU}(\mathbf{W}_2^T\mathbf{z}_1 - b_2) \\ &\vdots \\ \mathbf{z}_n &= \mathrm{ReLU}(\mathbf{W}_n^T\mathbf{z}_{n-1} - b_n) \end{aligned} \tag{5.5}$$

as executing a sequence of single-iteration ISTA, serving as an approximate solution to the multi-layer sparse coding problem: find $\{\mathbf{z}_i\}_{i=1}^n$, such that

$$\begin{aligned} \mathbf{x} = \mathbf{W}_1\mathbf{z}_1, \quad & \|\mathbf{z}_1\|_0 < \lambda_1, \mathbf{z}_1 \geq 0, \\ \mathbf{z}_1 = \mathbf{W}_2\mathbf{z}_2, \quad & \|\mathbf{z}_2\|_0 < \lambda_2, \mathbf{z}_2 \geq 0, \\ \vdots \quad , \quad & \vdots \\ \mathbf{z}_{n-1} = \mathbf{W}_n\mathbf{z}_n, \quad & \|\mathbf{z}_n\|_0 < \lambda_n, \mathbf{z}_n \geq 0, \end{aligned} \tag{5.6}$$

where the bias terms $\{b_i\}_{i=1}^n$ (in a similar manner to τ) are related to $\{\lambda_i\}_{i=1}^n$, adjusting the sparsity of recovered code. Furthermore, they reinterpret back-propagating through the deep neural network as learning the dictionaries $\{\mathbf{W}_i\}_{i=1}^n$. This connection offers a novel breakthrough for understanding DNNs.

5.2 Deep Non-Rigid Structure from Motion

Under weak-perspective projection, NRS*f*M deals with the problem of factorizing a 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 matrix $\mathbf{M} \in \mathbb{R}^{3\times 2}$. Formally,

$$\mathbf{W} = \mathbf{S}\mathbf{M}, \tag{5.7}$$

$$\mathbf{W} = \begin{bmatrix} u_1 & v_1 \\ u_2 & v_2 \\ \vdots & \vdots \\ u_p & v_p \end{bmatrix}, \ \mathbf{S} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_p & y_p & z_p \end{bmatrix}, \ \mathbf{M}^T\mathbf{M} = \mathbf{I}_2, \tag{5.8}$$

where $(u_i, v_i), (x_i, y_i, z_i)$ are the image and world coordinates of the i-th point. Due to the scale ambiguity between camera focal length and shape size, we ignore camera scale. The goal of NRS*f*M is to recover simultaneously the shape $\mathbf{S}$ and the camera $\mathbf{M}$ for each projection $\mathbf{W}$ in a given set $\mathbb{W}$ of 2D landmarks. In a general NRS*f*M including S*f*C, this set $\mathbb{W}$ could contain deformations of a non-rigid object or various instances from an object category.

5.2.1 Modeling via multi-layer sparse coding

To alleviate the ill-posedness of NRS*f*M and also guarantee sufficient freedom on shape variation, we propose a novel prior assumption on 3D shapes via multi-layer sparse coding: The vectorization of $\mathbf{S}$ satisfies

$$\begin{aligned} \mathbf{s} = \mathbf{D}_1\boldsymbol{\psi}_1, &\quad \|\boldsymbol{\psi}_1\|_0 < \lambda_1, \boldsymbol{\psi}_1 \geq 0, \\ \boldsymbol{\psi}_1 = \mathbf{D}_2\boldsymbol{\psi}_2, &\quad \|\boldsymbol{\psi}_2\|_0 < \lambda_2, \boldsymbol{\psi}_2 \geq 0, \\ \vdots \quad , &\quad \vdots \\ \boldsymbol{\psi}_{n-1} = \mathbf{D}_n\boldsymbol{\psi}_n, &\quad \|\boldsymbol{\psi}_n\|_0 < \lambda_n, \boldsymbol{\psi}_n \geq 0, \end{aligned} \tag{5.9}$$

where $\mathbf{D}_1 \in \mathbb{R}^{3p\times k_1}, \mathbf{D}_2 \in \mathbb{R}^{k_1\times k_2}, \ldots, \mathbf{D}_n \in \mathbb{R}^{k_{n-1}\times k_n}$ are hierarchical dictionaries. In this prior, each non-rigid shape is represented by a sequence of hierarchical dictionaries and corresponding sparse codes. Each sparse code is determined by its lower-level neighbor and affects the next-level. Clearly this hierarchy adds more parameters, and thus more freedom into the system. We now show that it paradoxically results in a more constrained global dictionary and sparse code recovery.

More constrained code recovery

In a classical single dictionary system, the constraint on the representation is element-wise sparsity. Further, the quality of its recovery entirely depends on the quality of the dictionary. In our multi-layer sparse coding model, the optimal code not only minimizes the difference between measurements $\mathbf{s}$ and $\mathbf{D}_1\boldsymbol{\psi}_1$ along with sparsity regularization $\|\boldsymbol{\psi}_1\|_0$, but also satisfies constraints from its subsequent representations. This additional joint inference imposes more constraints on code recovery, helps to control the uniqueness and therefore alleviates its heavy dependency on the dictionary quality.

More constrained dictionary

When all equality constraints are satisfied, the multi-layer sparse coding model degenerates to a single dictionary system. From Equation 5.9, by denoting $\mathbf{D}^{(l)} = \prod_{i=1}^{l}\mathbf{D}_i$, it is implied that $\mathbf{s} = \mathbf{D}_1\mathbf{D}_2\ldots\mathbf{D}_n\boldsymbol{\psi}_n = \mathbf{D}^{(n)}\boldsymbol{\psi}_n$. However, this differs from other single dictionary models [31, 32, 69, 71, 72] in terms that a unique structure is imposed on $\mathbf{D}^{(n)}$ [51]. The dictionary $\mathbf{D}^{(n)}$ is composed by simpler atoms hierarchically. For example, each column of $\mathbf{D}^{(2)} = \mathbf{D}_1\mathbf{D}_2$ is a linear combination of atoms in $\mathbf{D}_1$, each column of $\mathbf{D}^{(3)} = \mathbf{D}^{(2)}\mathbf{D}_3$ is a linear combination of atoms in $\mathbf{D}^{(2)}$ and so on. Such a structure results in a more constrained global dictionary and potentially leads to higher quality with lower mutual coherence [22].

5.2.2 Multi-layer block sparse coding

Given the proposed multi-layer sparse coding model, we now build a conduit from the proposed shape code $\{\boldsymbol{\psi}_i\}_{i=1}^{k}$ to the 2D projected points. From Equation 5.9, we reshape vector $\mathbf{s}$ to a matrix $\mathbf{S} \in \mathbb{R}^{p\times 3}$ such that $\mathbf{S} = \mathbf{D}_1^{\sharp}(\boldsymbol{\psi}_1 \otimes \mathbf{I}_3)$, where $\otimes$ is Kronecker product and $\mathbf{D}_1^{\sharp} \in \mathbb{R}^{p\times 3k_1}$

is a reshape of $\mathbf{D}_1$ [16]. From linear algebra, it is well known that $\mathbf{AB} \otimes \mathbf{C} = (\mathbf{A} \otimes \mathbf{C})(\mathbf{B} \otimes \mathbf{C})$ given three matrices $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$. Based on this lemma, we can derive that

$$\begin{aligned} \mathbf{S} = \mathbf{D}_1^\sharp(\boldsymbol{\psi}_1 \otimes \mathbf{I}_3), &\quad \|\boldsymbol{\psi}_1\|_0 < \lambda_1, \boldsymbol{\psi}_1 \geq 0, \\ \boldsymbol{\psi}_1 \otimes \mathbf{I}_3 = (\mathbf{D}_2 \otimes \mathbf{I}_3)(\boldsymbol{\psi}_2 \otimes \mathbf{I}_3), &\quad \|\boldsymbol{\psi}_2\|_0 < \lambda_2, \boldsymbol{\psi}_2 \geq 0, \\ \vdots \quad , &\quad \vdots \\ \boldsymbol{\psi}_{n-1} \otimes \mathbf{I}_3 = (\mathbf{D}_n \otimes \mathbf{I}_3)(\boldsymbol{\psi}_n \otimes \mathbf{I}_3), &\quad \|\boldsymbol{\psi}_n\|_0 < \lambda_n, \boldsymbol{\psi}_n \geq 0. \end{aligned} \tag{5.10}$$

Further, from Equation 5.7, by right multiplying the camera matrix $\mathbf{M} \in \mathbb{R}^{3\times 2}$ to the both sides of Equation 5.10 and denote $\boldsymbol{\Psi}_i = \boldsymbol{\psi}_i \otimes \mathbf{M}$, we obtain that

$$\begin{aligned} \mathbf{W} = \mathbf{D}_1^\sharp \boldsymbol{\Psi}_1, &\quad \|\boldsymbol{\Psi}_1\|_0^{(3\times 2)} < \lambda_1, \\ \boldsymbol{\Psi}_1 = (\mathbf{D}_2 \otimes \mathbf{I}_3)\boldsymbol{\Psi}_2, &\quad \|\boldsymbol{\Psi}_2\|_0^{(3\times 2)} < \lambda_2, \\ \vdots \quad , &\quad \vdots \\ \boldsymbol{\Psi}_{n-1} = (\mathbf{D}_n \otimes \mathbf{I}_3)\boldsymbol{\Psi}_n, &\quad \|\boldsymbol{\Psi}_n\|_0^{(3\times 2)} < \lambda_n, \end{aligned} \tag{5.11}$$

where $\|\cdot\|_0^{(3\times 2)}$ divides the argument matrix into blocks with size 3×2 and counts the number of active blocks. Since $\boldsymbol{\psi}_i$ has active elements less than λ_i, $\boldsymbol{\Psi}_i$ has active blocks less than λ_i, that is $\boldsymbol{\Psi}_i$ is block sparse. This derivation demonstrates that if the shape vector $\mathbf{s}$ satisfies the multi-layer sparse coding prior described by Equation 5.9, then its 2D projection $\mathbf{W}$ must be in the format of multi-layer *block* sparse coding described by Equation 5.11. We hereby interpret NRS*f*M as a hierarchical *block* sparse dictionary learning problem *i.e*. factorizing $\mathbf{W}$ as products of hierarchical dictionaries $\{\mathbf{D}_i\}_{i=1}^n$ and block sparse coefficients $\{\boldsymbol{\Psi}_i\}_{i=1}^n$.

5.2.3 Block ISTA and DNNs solution

Before solving the multi-layer block sparse coding problem in Equation 5.11, we first consider the single-layer problem:

$$\min_{\mathbf{Z}} \|\mathbf{X} - \mathbf{W}\mathbf{Z}\|_F^2 \quad \text{s.t. } \|\mathbf{Z}\|_0^{(3\times 2)} < \lambda. \tag{5.12}$$

Inspired by ISTA, we propose to solve this problem by iteratively executing the following two steps:

$$\mathbf{V} = \mathbf{Z}^{[i]} - \alpha \mathbf{W}^T(\mathbf{W}\mathbf{Z}^{[i]} - \mathbf{X}), \tag{5.13}$$

$$\mathbf{Z}^{[i+1]} = \operatorname*{argmin}_{\mathbf{U}} \frac{1}{2}\|\mathbf{U} - \mathbf{V}\|_F^2 + \tau \|\mathbf{U}\|_{F1}^{(3\times 2)}, \tag{5.14}$$

where $\|\cdot\|_{F1}^{(3\times 2)}$ is defined as the summation of Frobenius norm of each 3×2 block, serving as a convex relaxation of block sparsity constraint. It is derived in [21] that the second step has a closed-form solution computing each block separately by $\mathbf{Z}_j^{[i+1]} = (h_\tau(\|\mathbf{V}_j\|_F)/\|\mathbf{V}_j\|_F)\mathbf{V}_j$, where the subscript j represents the j-th block and h_τ is a soft thresholding operator. This solution keeps a certain portion of the block according to its Frobenius norm and discards the block if

its Frobenius norm is sufficiently small. However, soft thresholding the Frobenius norms for every block brings unnecessary computational complexity. Therefore, instead of applying the soft thresholding to the Frobenius norm, we propose a relaxation applying the soft thresholding to the block itself and achieving a similar functionality. Specifically, we introduce auxiliary parameters $\mathbf{b}$, whose j-th element b_j is somehow negatively proportional to the Frobenius norm of $\mathbf{V}_j$. We then compute $\mathbf{Z}_j^{[i+1]} = h_{b_j}(\mathbf{V}_j)$ such that the block with greater Frobenius norm are shrunk by a smaller threshold and the block with smaller Frobenius norm are shrunk by a greater threshold. This relaxation aligns different thresholds to different blocks, lying between the closed-form solution and sharing one threshold over all blocks (degenerating to regular sparsity). Therefore, this aligning different thresholds to blocks maintains the structure of block sparsity. Based on this relaxation, a single-iteration block ISTA with step size $\alpha = 1$ can be represented by :

$$\mathbf{Z} = h_{\mathbf{b}}\big(\mathbf{W}^T\mathbf{X}\big) = \mathrm{ReLU}(\mathbf{W}^T\mathbf{X} - \mathbf{b} \otimes \mathbf{1}_{3\times 2}), \tag{5.15}$$

where $h_{\mathbf{b}}$ is a soft thresholding operator using the j-th element b_j as threshold of the j-th block and the second equality holds if $\mathbf{Z}$ is non-negative.

Encoder

Recall from Section 5.1 that the feed-forward pass through a deep neural network can be considered as a sequence of single ISTA iterations and thus provides an approximate recovery of multi-layer sparse codes. We follow the same scheme: we first relax the multi-layer block sparse coding to be non-negative and then sequentially use single-iteration block ISTA to solve it *i.e*.

$$\begin{aligned}
\mathbf{\Psi}_1 &= \mathrm{ReLU}((\mathbf{D}_1^\sharp)^T\mathbf{W} - \mathbf{b}_1 \otimes \mathbf{1}_{3\times 2}),\\
\mathbf{\Psi}_2 &= \mathrm{ReLU}((\mathbf{D}_2 \otimes \mathbf{I}_3)^T\mathbf{\Psi}_1 - \mathbf{b}_2 \otimes \mathbf{1}_{3\times 2}),\\
&\vdots\\
\mathbf{\Psi}_n &= \mathrm{ReLU}((\mathbf{D}_n \otimes \mathbf{I}_3)^T\mathbf{\Psi}_{n-1} - \mathbf{b}_n \otimes \mathbf{1}_{3\times 2}),
\end{aligned} \tag{5.16}$$

where thresholds $\mathbf{b}_1, ..., \mathbf{b}_n$ are learned, controlling the block sparsity. This learning is crucial because in previous NRS*f*M algorithms utilizing low-rank [16], subspaces [72] or compressible [31] priors, the weight given to this prior (*e.g*. rank or sparsity) is hand-selected through a cumbersome cross validation process. In our approach, this weighting is learned simultaneously with all other parameters removing the need for any irksome cross validation process. This formula composes the encoder of our proposed DNN.

Decoder

Let us for now assume that we can extract camera $\mathbf{M}$ and regular sparse hidden code $\boldsymbol{\psi}_n$ from $\mathbf{\Psi}_n$ by some functions *i.e*. $\mathbf{M} = \mathcal{F}(\mathbf{\Psi}_n)$ and $\boldsymbol{\psi}_n = \mathcal{G}(\mathbf{\Psi}_n)$, which will be discussed in the next

section. Then we can compute the 3D shape vector $\mathbf{s}$ by:

$$\begin{aligned} \boldsymbol{\psi}_{n-1} &= \mathrm{ReLU}(\mathbf{D}_n\boldsymbol{\psi}_n - \mathbf{b}'_n), \\ &\vdots \\ \boldsymbol{\psi}_1 &= \mathrm{ReLU}(\mathbf{D}_2\boldsymbol{\psi}_2 - \mathbf{b}'_2), \\ \mathbf{s} &= \mathbf{D}_1^\sharp\boldsymbol{\psi}_1, \end{aligned} \tag{5.17}$$

Note we preserve the ReLU and bias term during decoding to further enforce sparsity and improve robustness. These portion forms the decoder of our DNN.

Variation of implementation

The Kronecker product of identity matrix $\mathbf{I}_3$ dramatically increases the time and space complexity of our approach. To eliminate it and make parameter sharing easier in modern deep learning environments (*e.g.* TensorFlow, PyTorch), we reshape the filters and features and show that the matrix multiplication in each step of the encoder and decoder can be equivalently computed via multi-channel 1×1 convolution ($*$) and transposed convolution ($*^T$) *i.e.*

$$(\mathbf{D}_1^\sharp)^T\mathbf{W} = \mathsf{d}_1^\sharp *^T \mathsf{w}, \tag{5.18}$$

where $\mathsf{d}_1^\sharp \in \mathbb{R}^{3\times1\times k_1\times p}, \mathsf{w} \in \mathbb{R}^{1\times2\times p1}$.

$$(\mathbf{D}_{i+1} \otimes \mathbf{I}_3)^T\boldsymbol{\Psi}_i = \mathsf{d}_{i+1} *^T \Psi_i, \tag{5.19}$$

where $\mathsf{d}_{i+1} \in \mathbb{R}^{1\times1\times k_{i+1}\times k_i}, \Psi_i \in \mathbb{R}^{3\times2\times k_i}$.

$$\mathbf{D}_i\boldsymbol{\psi}_i = \mathsf{d}_i * \psi_i, \tag{5.20}$$

where $\mathsf{d}_i \in \mathbb{R}^{1\times1\times k_i\times k_{i-1}}, \psi_i \in \mathbb{R}^{1\times1\times k_i}$.

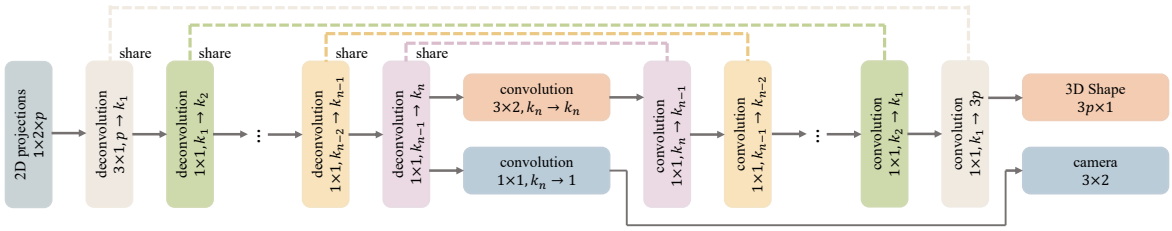


Figure 5.1: Deep NRS*f*M architecture. The network can be divided into two parts: encoder and decoder that are symmetric and share convolution kernels (*i.e.* dictionaries). The symbol $a \times b, c \to d$ refers to the operator using kernel size $a \times b$ with c input channels and d output channels.

[1]The filter dimension is height×width×# of input channel×# of output channel. The feature dimension is height×width×# of channel.

Code and camera recovery

Estimating $\boldsymbol{\psi}_n$ and $\mathbf{M}$ from $\boldsymbol{\Psi}_n$ is discussed in [31] and solved by a closed-form formula. Due to its differentiability, we could insert the solution directly within our pipeline. An alternative solution is using a relaxation *i.e.* a fully connected layer connecting $\boldsymbol{\Psi}_n$ and $\boldsymbol{\psi}_n$ and a linear combination among each blocks of $\boldsymbol{\Psi}_n$ to estimate $\mathbf{M}$, where the fully connected layer parameters and combination coefficients are learned from data. In our experiments, we will use the relaxed solution and represent them via convolutions, as shown in Figure 5.1, for conciseness and maintaining proper dimensions. Since the relaxation has no way to force the orthonormal constraint on the camera, we seek help from the loss function.

Loss function

The loss function must measure the reprojection error between input 2D points $\mathbf{W}$ and reprojected 2D points $\mathbf{SM}$ while simultaneously encouraging orthonormality of the estimated camera $\mathbf{M}$. One solution is to use spectral norm regularization of $\mathbf{M}$ because spectral norm minimization is the tightest convex relaxation of the orthonormal constraint [69]. An alternative solution is to hard code the singular values of $\mathbf{M}$ to be exact ones with the help of Singular Value Decomposition (SVD). Even though SVD is generally non-differentiable, the numeric computation of SVD is differentiable and most deep learning packages implement its gradients (*e.g.* PyTorch, TensorFlow). In our implementation and experiments, we will use SVD to ensure the success of the orthonormal constraint and a simple Frobenius norm to measure reprojection error,

$$Loss = \|\mathbf{W} - \mathbf{S}\tilde{\mathbf{M}}\|_F, \quad \tilde{\mathbf{M}} = \mathbf{U}\mathbf{V}^T, \tag{5.21}$$

where $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \mathbf{M}$ is the SVD of the camera matrix.

5.3 Experiments

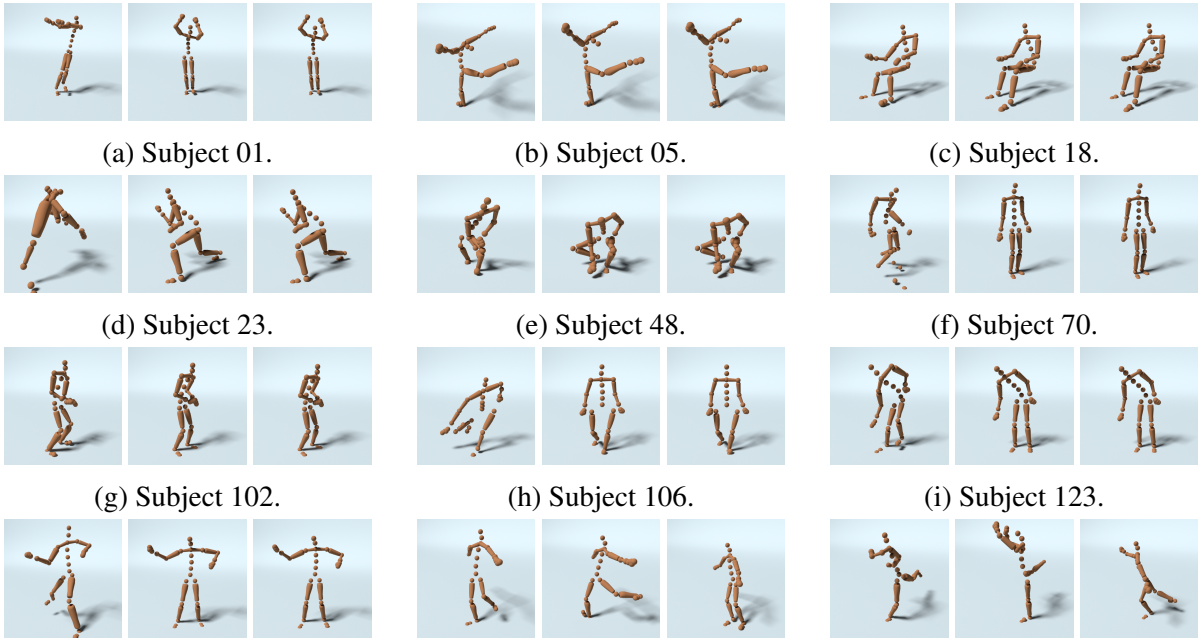
We conduct extensive experiments to evaluate the performance of our deep solution for solving NRS*f*M and S*f*C problems. Further, for evaluating generalizability, we conduct an experiment applying the pre-trained DNN to unseen data and reconstruct 3D human pose from a single view. Note that in all experiments, our model has no access to 3D ground-truth except qualitative and quantitative evaluations for comparison against the state-of-art methods. A detailed description of our architectures is in the supplementary material.

5.3.1 NRS*f*M on CMU Motion Capture

We first apply our method to solving the problem of NRS*f*M using the CMU motion capture dataset[2]. For evaluation on complex sequences, we concatenate all motions of the same subject and select ten subjects from CMU MoCap so that each subject contains tens of thousands of frames. We randomly create orthonormal cameras for each frame to project the 3D human joints onto images. We compare our method against state-of-the-art NRS*f*M works with code released

[2]http://mocap.cs.cmu.edu/

online[3] [16, 31, 55]. Since none of them are capable of scaling up to this number of frames, we shuffle each sequence, divide them into mini batches each containing 500 frames, feed each mini batch into baselines, and then compute the mean error. Our model is trained on the entire sequence. For error metrics, we use the shape error ratio defined as $\frac{1}{|\mathcal{S}|}\sum_{\mathcal{S}}\frac{\|\mathbf{S}-\hat{\mathbf{S}}\|_F}{\|\hat{\mathbf{S}}\|_F}$, where $\hat{\mathbf{S}}$ is the 3D ground-truth and $\mathcal{S}$ is the set of all shapes; as well as the mean point distance defined as $\frac{1}{|\mathcal{S}|}\sum_{\mathcal{S}}\sum_i\frac{\|\mathbf{S}_i-\hat{\mathbf{S}}_i\|_2}{p}$, where $\mathbf{S}_i$ is 3D coordinates of i-th point on shape $\mathbf{S}$ and p is the number of points. Note that shapes are normalized to real-world sizes so that each human skeleton is around 1.8 meters high, and the mean point distance is computed in centimeters. The results are summarized in Table 5.1. One can see that our method obtains impressive reconstruction performance and outperforms others in every sequences. We randomly select a frame for each subject and render the reconstructed human skeleton in Figure 5.2 (a) to 5.2 (j). To give a sense of the quality of reconstructions when our method fails, we go through all ten subjects in a total of 140,606 frames and select the frames with the largest errors as shown in Figure 5.2(k) and 5.2 (l). Even in the worst cases, our method grasps a rough 3D geometry of human body instead of completely diverging.



(a) Subject 01. (b) Subject 05. (c) Subject 18.

(d) Subject 23. (e) Subject 48. (f) Subject 70.

(g) Subject 102. (h) Subject 106. (i) Subject 123.

(j) Subject 127. (k) Failure case. (l) Failure case.

Figure 5.2: Qualitative evaluation on reconstructed human skeletons. (a) to (j) are randomly selected from each subject. (k) and (l) are two failure cases with the largest errors among all 140,606 images. In each sub-figure, the left is the reconstruction of [16], the middle is the ground-truth, and the right is ours.

[3] Paladini *et al*. [41] fails on all sequences and therefore removed from the table. Works [15, 19, 25, 26, 27, 35, 53, 60] did not release code. Works [5, 24, 33, 34] use additional priors, say temporal continuity, and thus not applicable.

	Subject	1	5	18	23	64	70	102	106	123	127
	# of frames	45025	13773	10024	10821	11621	10788	5929	12335	10788	9502
Shape Error (%)	EM-SfM [55]	110.23%	119.97%	111.05%	110.94%	114.04%	127.11%	111.60%	113.81%	107.67%	108.07%
	Simple [16]	16.45%	14.07%	13.85%	20.03%	18.13%	18.91%	18.78%	18.63%	19.32%	23.70%
	Sparse [31]	71.23%	66.30%	46.72%	52.44%	70.83%	39.42%	74.12%	47.00%	44.46%	73.85%
	Ours	**10.74%**	**13.40%**	**4.73%**	**3.24%**	**4.38%**	**2.17%**	**7.32%**	**6.83%**	**2.23%**	**6.00%**
Point Error (cm)	EM-SfM [55]	53.1818	60.5971	53.0413	52.2671	50.3960	56.3713	48.5891	50.3306	47.7355	50.8183
	Simple [16]	7.9905	6.9406	6.6340	9.5139	8.1784	8.4294	8.0171	8.1782	8.6922	10.9473
	Sparse [31]	35.0283	35.3014	22.6930	25.3302	32.4681	17.7433	30.8274	21.2735	20.3565	32.4896
	Ours	**5.0638**	**6.6717**	**2.2664**	**1.5138**	**2.2909**	**0.9622**	**3.0240**	**2.9130**	**0.9844**	**2.6820**

Table 5.1: Quantitative comparison of our method against the state-of-the-art methods in NRS*f*M task. Human skeletons are scaled to real-world sizes, around 1.8 meters high, and the mean point distance is measured in centimeters.

Noise performance

To analyze the robustness of our method, we re-train the neural network for Subject 70 using projected points with Gaussian noise perturbation. The results are summarized in Figure 5.3. The noise ratio is defined as $\|\text{noise}\|_F / \|\mathbf{W}\|_F$. One can see that our method gets far more precise reconstructions even when adding up to 20% noise to our image coordinates compared to baselines with no noise perturbation. This experiment clearly demonstrates the robustness of our model and its high accuracy against state-of-the-art works.

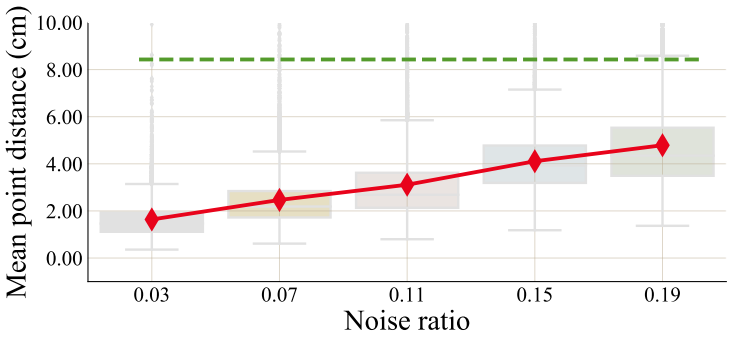


Figure 5.3: NRS*f*M with noise perturbation. The red solid line is ours while the green dashed line is the lowest error achieved by baselines with *no* noise perturbation.

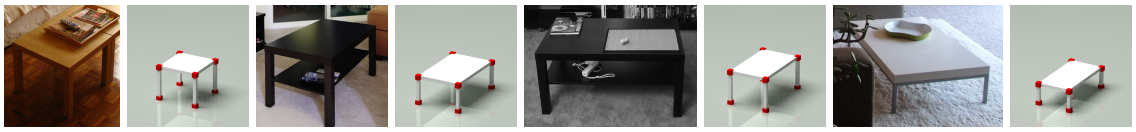
Missing data

Landmarks are not always visible from the camera owing to the occlusion by other objects or itself. In the present paper, we focus on a complete measurement situation not accounting for invisible landmarks. However, thanks to recent progress in deep-learning-based depth map reconstruction from sparse observations [12, 14, 36, 37, 40], our central pipeline of DNN can be easily adapted to handling missing data.

5.3.2 S*f*C on IKEA furnitures

We now apply our method to the application of S*f*C using IKEA dataset [38, 63]. The IKEA dataset contains four object categories: bed, chair, sofa, and table. For each object category, we

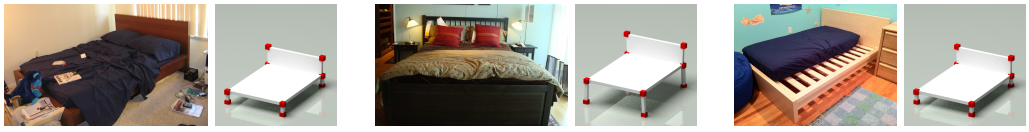
employ all annotated 2D point clouds and augment them with 2K ones projected from the 3D ground-truth using randomly generated orthonormal cameras[4]. We compare our method against the baselines [16, 32] again using the shape error ratio metric. The error evaluated on real images are reported and summarized into Table 5.2. One can observe that our method outperforms baselines with a large margin, clearly showing the superiority of our model. Table 5.2 from another perspective reveals the dilemma suffered by baselines of restricting ill-possedness and modeling high variance of object category. For qualitative evaluation, we randomly select frames from each object category and show them in Figure 5.4. It shows that our model successfully learns the intra-category shape variation and reconstructed landmarks effectively depict the 3D geometry of objects.



(a) Object table.

(b) Object sofa.

(c) Object chair.



(d) Object bed.

Figure 5.4: Qualitative results of S*f*C task. Reconstructions are randomly selected from each object category. Red cubes are reconstructed points while the planes and bars are manually added for descent rendering.

	Bed	Chair	Sofa	Table
Simple [16]	17.81%	33.32%	14.78%	12.40%
SfC [32]	22.51%	27.58%	13.35%	11.78%
Ours	**0.23%**	**1.15%**	**0.35%**	**0.81%**

Table 5.2: Quantitative comparison against state-of-the-art algorithms in S*f*C task. Results are evaluated by shape error ratio. Our method outperforms others in all four object categories with a large margin.

[4]Augmentation is utilized due to limited valid frames, because the ground-truth cameras are partially missing.

5.3.3 Shape from single-view landmarks

Even though almost all NRS*f*M algorithms learn a shape dictionary from 2D projections, none of them apply the learned dictionary to unseen data. This is because all of them are facing the difficulty of handling large amount of images and thus cannot generalize well. In this experiment, we show the generalization of our learned dictionary by evaluating it using sequences invisible to training. Specifically, we follow the same training and evaluation scheme in [69], training with Subject 86 in CMU MoCap and evaluating on Subject 13, 14 and 15. We compare our model to methods for human pose estimation [46, 69] following the same error metrics in [69]. It is worth mentioning that all baselines learn shape dictionaries directly from 3D ground-truth, but our method learns such dictionaries purely from 2D projections (*i.e*. no 3D supervision). Even in such an unfair scenario, our method achieves competitive results as summarized in Table 5.3. This clearly demonstrates that our method effectively learns the underlying geometry from pure 2D projections with no need for 3D supervision, and the learned dictionaries generalize well to unseen data.

	PMP	Alternate	Convex	Ours
Subject 13	0.390	0.293	0.259	**0.229**
Subject 14	0.393	0.308	**0.258**	0.261
Subject 15	0.340	0.286	0.204	**0.200**

Table 5.3: Comparison of our method against the state-of-the-art algorithms in single image human pose estimation task. Our method achieves competitive results using solely 2D projections while all others learn from 3D ground truth.

5.3.4 Coherence as guide

As explained in Section 5.2.1, every sparse code $\boldsymbol{\psi}_i$ is constrained by its subsequent representation and thus the quality of code recovery depends less on the quality of the corresponding dictionary. However, this is not applicable to the final code $\boldsymbol{\psi}_n$, making it least constrained with the most dependency on the final dictionary $\mathbf{D}_n$. From this perspective, the quality of the final dictionary measured by mutual coherence [22] could serve as a lower bound of the entire system. To verify this, we compute the error and coherence in a fixed interval during training in NRS*f*M experiments. We consistently observe strong correlations between 3D reconstruction error and the mutual coherence of the final dictionary. We plot this relationship in Figure 5.5. We thus propose to use the coherence of the final dictionary as a measure of model quality for guiding training to efficiently avoid over-fitting especially when 3D evaluation is not available. This improves the utility of our deep NRS*f*M in future applications without 3D ground-truth.

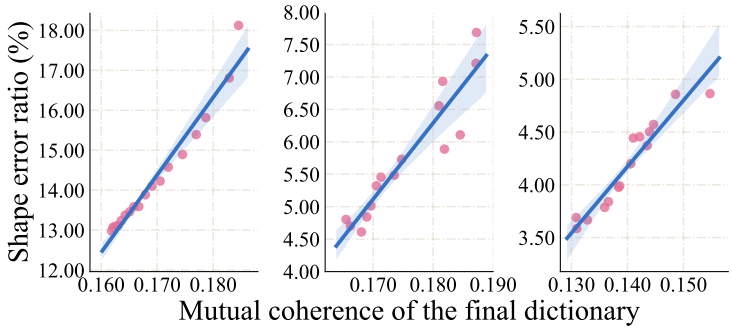


Figure 5.5: A scatter plot of the shape error ratio in percentage against the final dictionary coherence. A line is fitted based on the data. The left comes from subject 05, the middle from subject 18, the right from subject 64.

5.4 Current Status

In this paper, we proposed multi-layer sparse coding as a novel prior assumption for representing 3D non-rigid shapes and designed an innovative encoder-decoder neural network to solve the problem of NRS*f*M using no 3D supervision. The proposed DNN was derived by generalizing the classical sparse coding algorithm ISTA to a block sparse scenario. The proposed DNN architecture is mathematically interpretable as a NRS*f*M multi-layer sparse dictionary learning problem. Extensive experiments demonstrated our superior performance against the state-of-the-art methods and the impressive generalization to unseen data. Finally, we propose to use the coherence of the final dictionary as a generalization measure, offering a practical way to avoid over-fitting and selecting the best model without 3D ground-truth.

5.5 Proposed Timeline

- December 1 - February 1: Explore the theoretical benefits of recurrent NRSfM; Implement a unified template for feed-forward and recurrent NRSfM; Conduct experiments evaluating practical benefits.
- February 1 - April 1: Explore methods of handling missing data. Implement the method and apply it to real images with annotated landmarks. Conduct experiments evaluating its performance against 3DV baseline work.
- April 1 - May 1: Write and defend thesis.

Bibliography

[1] Alekh Agarwal, Animashree Anandkumar, Prateek Jain, and Praneeth Netrapalli. Learning sparsely used overcomplete dictionaries via alternating minimization. *arXiv preprint arXiv:1310.7991*, 2013. 3.3.1

[2] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10): 105–112, 2011. 2

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

[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. 2.2.1, 3.3

[5] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Trajectory space: A dual representation for nonrigid structure from motion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(7):1442–1456, 2011. 3

[6] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid icp algorithms for surface registration. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 2.1.1

[7] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 693–696. IEEE, 2009. 5.1

[8] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011. 2.1.2, 2.1.2, 3.3.1, 4.1, 4.2

[9] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, volume 2, pages 690–696. IEEE, 2000. 1, 2.2, 2.2.1

[10] Hilton Bristow, Anders Eriksson, and Simon Lucey. Fast convolutional sparse coding. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 391–398. IEEE, 2013. 3.3.1, 4.2

[11] Richard A Brualdi. *Introductory combinatorics*. New York, 1992. 2

[12] Cesar Cadena, Anthony R Dick, and Ian D Reid. Multi-modal auto-encoders as joint estimators for robotics scene understanding. In *Robotics: Science and Systems*, 2016. 5.3.1

[13] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. URL `http://arxiv.org/abs/1512.03012`. 1, 2.1.2, 2.1.4

[14] Zhao Chen, Vijay Badrinarayanan, Gilad Drozdov, and Andrew Rabinovich. Estimating depth from rgb and sparse sensing. *European Conference on Computer Vision (ECCV)*, 2018. 5.3.1

[15] Ajad Chhatkuli, Daniel Pizarro, Toby Collins, and Adrien Bartoli. Inextensible non-rigid shape-from-motion by second-order cone programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1719–1727, 2016. 1, 5, 3

[16] 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. 2.2.1, 2.2.1, 3.3.2, 3.4.3, 3.3, 4.3.1, 4.3.2, 4.1, 4.2, 4.1, 4.3, 5.2.2, 5.2.3, 5.3.1, 5.2, **??**, **??**, 5.3.2, **??**

[17] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004. 5.1

[18] Mark A Davenport, Marco F Duarte, Yonina C Eldar, and Gitta Kutyniok. Introduction to compressed sensing. *Preprint*, 93:1–64, 2011. 2.2.1

[19] 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. 3

[20] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 1

[21] Wei Deng, Wotao Yin, and Yin Zhang. Group sparse optimization by alternating direction method. In *SPIE Optical Engineering+ Applications*, pages 88580R–88580R. International Society for Optics and Photonics, 2013. 3.3.1, 5.2.3

[22] David L Donoho, Michael Elad, and Vladimir N Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on information theory*, 52(1):6–18, 2006. 5.2.1, 5.3.4

[23] Irina F Gorodnitsky and Bhaskar D Rao. Sparse signal reconstruction from limited data using focuss: A re-weighted minimum norm algorithm. *Signal Processing, IEEE Transactions on*, 45(3):600–616, 1997. 3.3.1, 3.3.1

[24] Paulo FU Gotardo and Aleix M Martinez. Computing smooth time trajectories for camera and deformable shape in structure from motion with occlusion. *Pattern Analysis and*

Machine Intelligence, IEEE Transactions on, 33(10):2051–2065, 2011. 3.3, 3

[25] Paulo FU Gotardo and Aleix M Martinez. Kernel non-rigid structure from motion. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 802–809. IEEE, 2011. 2.2.1, 2.2.1, 3

[26] Paulo FU Gotardo and Aleix M Martinez. Non-rigid structure from motion with complementary rank-3 spaces. 2011. 3

[27] 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. 3

[28] Christopher Hillar and Friedrich T Sommer. When can dictionary learning uniquely recover sparse data from subsamples? In *IEEE Transactions on Information Theory*, 2015. 3.1.1, 3.1.1, 1, 3.1.3

[29] Jing Huang and Suya You. Point cloud labeling using 3d convolutional neural network. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 2670–2675. IEEE, 2016. 1

[30] Abhishek Kar, Shubham Tulsiani, Joo Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1966–1974. IEEE, 2015. 2.1.4

[31] Chen Kong and Simon Lucey. Prior-less compressible structure from motion. *Computer Vision and Pattern Recognition (CVPR)*, 2016. 5.2.1, 5.2.3, 5.2.3, 5.3.1, **??**, **??**

[32] Chen Kong, Rui Zhu, Hamed Kiani, and Simon Lucey. Structure from category: a generic and prior-less approach. *International Conference on 3DVision (3DV)*, 2016. 1, 5.2.1, 5.3.2, **??**

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

[34] Suryansh Kumar, Anoop Cherian, Yuchao Dai, and Hongdong Li. Scalable dense non-rigid structure-from-motion: A grassmannian perspective. *arXiv preprint arXiv:1803.00233*, 2018. 1, 2.2.1, 5, 3

[35] 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. 1, 5, 3

[36] Yaoxin Li, Keyuan Qian, Tao Huang, and Jingkun Zhou. Depth estimation from monocular image and coarse depth points based on conditional gan. In *MATEC Web of Conferences*, volume 175, page 03055. EDP Sciences, 2018. 5.3.1

[37] Yiyi Liao, Lichao Huang, Yue Wang, Sarath Kodagoda, Yinan Yu, and Yong Liu. Parse geometry from a line: Monocular depth estimation with partial laser observation. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 5059–5066. IEEE, 2017. 5.3.1

[38] Joseph J. Lim, Hamed Pirsiavash, and Antonio Torralba. Parsing IKEA Objects: Fine Pose

Estimation. *ICCV*, 2013. 5.3.2

[39] Angshul Majumdar and Rabab Kreidieh Ward. Some empirical advances in matrix completion. *Signal Processing*, 91(5):1334–1338, 2011. 4.3.4

[40] Fangchang Mal and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018. 5.3.1

[41] Marco Paladini, Alessio Del Bue, Marko Stosic, Marija Dodig, Joao Xavier, and Lourdes Agapito. Factorization for non-rigid and articulated structure using metric projections. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2898–2905. IEEE, 2009. 3

[42] 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. 1, 5.1

[43] Hyun Soo Park and Yaser Sheikh. 3d reconstruction of a smooth articulated trajectory from a monocular image sequence. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 201–208. IEEE, 2011. 2.2.1

[44] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017. 1

[45] Vincent Rabaud and Serge Belongie. Re-thinking non-rigid structure from motion. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 2.2.1

[46] Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh. Reconstructing 3d human pose from 2d image landmarks. In *European Conference on Computer Vision*, pages 573–586. Springer, 2012. 2, 5.3.3

[47] BD Rao and Kenneth Kreutz-Delgado. Basis selection in the presence of noise. In *Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on*, volume 1, pages 752–756. IEEE, 1998. 3.3.1

[48] 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. 5.1

[49] Ron Rubinstein, Tomer Peleg, and Michael Elad. Analysis k-svd: A dictionary-learning algorithm for the analysis sparse model. *Signal Processing, IEEE Transactions on*, 61(3): 661–677, 2013. 3.3.1, 3.3.1

[50] Hao Su, Charles R Qi, Yangyan Li, and Leonidas J Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2686–2694, 2015. 1, 4.3.4

[51] Jeremias Sulam, Vardan Papyan, Yaniv Romano, and Michael Elad. Multi-layer convolutional sparse modeling: Pursuit and dictionary learning. *arXiv preprint arXiv:1708.08705*, 2017. 5.2.1

[52] Petri Tanskanen, Kalin Kolev, Lorenz Meier, Federico Camposeco, Olivier Saurer, and Marc Pollefeys. Live metric 3d reconstruction on mobile phones. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 65–72, 2013. 2

[53] Jonathan Taylor, Allan D Jepson, and Kiriakos N Kutulakos. *Non-rigid structure from locally-rigid motion*. IEEE, 2010. 3

[54] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992. 2.2.1, 3, 3.3.2, 4.3.1, 4.3.2, 4.1, 4.2, 4.1, 4.3

[55] Lorenzo Torresani, Aaron Hertzmann, and Christoph Bregler. Learning non-rigid 3d shape from 2d motion. In *Advances in Neural Information Processing Systems*, pages 1555–1562, 2004. 5.3.1, **??**, **??**

[56] Lorenzo Torresani, Aaron Hertzmann, and Christoph Bregler. Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(5):878–892, 2008. 2.2.1, 3.4.5

[57] Joel A Tropp and Anna C Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007. 2.1.2, 3.3.1, 3.3.1

[58] L Richard Turner. Inverse of the vandermonde matrix with applications. 1966. 3.1.3

[59] Jack Valmadre, Yingying Zhu, Sridha Sridharan, and Simon Lucey. Efficient articulated trajectory reconstruction using dynamic programming and filters. In *Computer Vision–ECCV 2012*, pages 72–85. Springer, 2012. 2.2.1

[60] Sara Vicente and Lourdes Agapito. Soft inextensibility constraints for template-free non-rigid reconstruction. In *European conference on computer vision*, pages 426–440. Springer, 2012. 3

[61] Sara Vicente, João Carreira, Lourdes Agapito, and Jorge Batista. Reconstructing pascal voc. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 41–48. IEEE, 2014. 2.1.4, 4.3.2

[62] Chunyu Wang, Yizhou Wang, Zhouchen Lin, Alan L Yuille, and Wen Gao. Robust estimation of 3d human poses from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2361–2368, 2014. 2

[63] Jiajun Wu, Tianfan Xue, Joseph J Lim, Yuandong Tian, Joshua B Tenenbaum, Antonio Torralba, and William T Freeman. Single image 3d interpreter network. *European Conference on Computer Vision (ECCV)*, 2016. 2, 5.3.2

[64] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 1

[65] 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. 2.1.4, 2.1.4, 4.3.1

[66] Jing Xiao, Jinxiang Chai, and Takeo Kanade. A closed-form solution to non-rigid shape and motion recovery. *International Journal of Computer Vision*, 67(2):233–246, 2006. 2.2, 2.2.1, 2.2.1, 3

[67] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 532–539, 2013. 4.3.4, 4.3.4

[68] Xuehan Xiong and Fernando De la Torre. Global supervised descent method. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2664–2673, 2015. 4.3.4

[69] Xiaowei Zhou, Spyridon Leonardos, Xiaoyan Hu, and Kostas Daniilidis. 3d shape estimation from 2d landmarks: A convex relaxation approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4447–4455, 2015. 2, 4.1, 4.1, 4.2, 5.2.1, 5.2.3, 5.3.3

[70] Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Kosta Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. *arXiv preprint arXiv:1511.09439*, 2015. 2

[71] Yingying Zhu and Simon Lucey. Convolutional sparse coding for trajectory reconstruction. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(3):529–540, 2015. 5.2.1

[72] Yingying Zhu, Dong Huang, Fernando De La Torre, and Simon Lucey. Complex non-rigid motion 3d reconstruction by union of subspaces. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1542–1549. IEEE, 2014. 2.2.1, 2.2.1, 5.2.1, 5.2.3