# A Closed-Form, Pairwise Solution to Local Non-Rigid Structure-from-Motion

Shaifali Parashar[1], Yuxuan Long[2], Mathieu Salzmann[3], Pascal Fua[3]
[1]CNRS-LIRIS, FRANCE [2]ZHEJIANG LAB, CHINA [2]EPFL, SWITZERLAND

A recent trend in Non-Rigid Structure-from-Motion (NRSfM) is to express local, differential constraints between pairs of images, from which the surface normal at any point can be obtained by solving a system of polynomial equations. While this approach is more successful than its counterparts relying on global constraints, the resulting methods face two main problems: First, most of the equation systems they formulate are of high degree and must be solved using computationally expensive polynomial solvers. Some methods use polynomial reduction strategies to simplify the system, but this adds some phantom solutions. In any event, an additional mechanism is employed to pick the best solution, which adds to the computation without any guarantees on the reliability of the solution. Second, these methods formulate constraints between a pair of images. Even if there is enough motion between them, they may suffer from local degeneracies that make the resulting estimates unreliable without any warning mechanism.
In this paper, we solve these problems for isometric/conformal NRSfM. We show that, under widely applicable assumptions, we can derive a new system of equations in terms of the surface normals, whose two solutions can be obtained in closed-form and can easily be disambiguated locally. Our formalism also allows us to assess how reliable the estimated local normals are and to discard them if they are not. Our experiments show that our reconstructions, obtained from two or more views, are significantly more accurate than those of state-of-the-art methods, while also being faster.

## 1 INTRODUCTION

Reconstructing the 3D shape of deformable objects from monocular image sequences is known as Non-Rigid Structure-from-Motion (NRSfM) and has applications in domains ranging from entertainment [35] to medicine [26]. Early methods relied on low-rank representations of the surfaces [4], [7], [10], [12], [17], [23], [25], [28], [49], while more recent ones exploit local surface properties to derive constraints and can handle larger deformations [8], [9], [20], [47], [50], [51]. Unfortunately, these constraints have to be enforced jointly on the entire set of reconstructed points for a whole sequence. Hence, the computational cost increases non-linearly with the number of images and quickly becomes prohibitive. Furthermore, a globally optimal solution is obtained using an iterative refinement, which is prone to getting trapped in local minima and adds to the computation if the initialization is not close to the actual solution. Finally, most of these *global* methods cannot handle missing data. In [13], [34], this is done by iteratively updating the missing entries, which adds to the computational complexity. We refer the interested reader to [19] for a detailed review of *global* methods.

In earlier work [37], [38], [39], we have shown that *local* methods constitute a powerful alternative. Expressing isometry, conformality, or equiareality constraints in terms of differential properties makes the number of local variables remain fixed. Unfortunately, the systems of equations that arise in these computations are bivariate of high degree. They can have up to five real solutions. In theory, a unique solution can be obtained from 3 images, but this requires either a complicated sum-of-squares formulation [37], [38] or reduction methodologies that add phantom solutions [36], [39]. Hence, in practice, it takes more than 3 images to produce reliable estimates. Furthermore, when the motion between the frames is too small, the system becomes ill-posed and the estimates unreliable, without any mechanism to flag such situations as problematic.

In this paper, we introduce a new local method. Instead of inferring the depth derivatives, we estimate surface normals. More specifically, given a 2D warp between two images, we consider tangent planes at corresponding points. For each pair of points, we compute the homography relating the two planes and decompose it to compute the normals by solving local differential constraints [37], [38]. This has two solutions, instead of five in our earlier approaches [36]. For each plane, we pick the right one by enforcing an easy-to-compute measure of local smoothness. Furthermore, our formulation lets us assess how well-conditioned the problem is and, hence, how usable the resulting normals are. In other words, we can derive from an image pair, a set of reliable normals and discard the others.

We will demonstrate on both synthetic and real data that we outperform state-of-the-art local and global methods at a fraction of the computational cost. Our contribution is therefore an approach to NRSfM that relies on solving in closed form a set of equations relating surface normals at corresponding points. Being entirely local, the computation is both fast and reliable. Although our solution is designed for isometric or conformal deformations, it yields good results for generic ones.

## 2 RELATED WORK

NRSfM was introduced in [7] and the ill-posedness of the problem was handled by constraining the deformations to lie on a low-dimensional manifold. Later variants introduced additional constraints for efficient low-rank factorization [4], [11], [12], [17],

*This work was carried out when Shaifali Parashar and Yuxuan Long were at EPFL.*
*E-mail: shaifali.parashar@gmail.com*

| Method | Deformation modeling | Variables | Assumptions | Constraints on | Degree | Solution strategy | Unique solution |
|---|---|---|---|---|---|---|---|
| [37] | Isomety | Depth | LP | Metric tensor, Connections | 3 | Sum-of-squares mimimisation of bivariates | >=3 images |
| [38] | Isometry, Conformality, Equiareality | Depth | LP+LL | Metric tensor, Connections | 3 | Sum-of-squares mimimisation of bivariates | >=3 images |
| [39] | Diffeomorphism | Depth | LL | Connections | 10 | Reduce to univariates using resultants | >=3 images |
| [36] | Isometry | Depth | LP | Metric tensor, Connections | 3 | Reduce to univariates using resultants or substiution | >=3 images |
| Ours | Isometry, Conformality | Normals | LP+LL | Metric tensor, Connections | 2 | Closed-form solution from univariates | >=2 images using local smoothness |

TABLE 1: Summary of the local methods we developed in earlier work.

[18], [48] or performed additional optimization [10], [14], [23], [24], [28], [33], [52] to improve the statistical modeling. Learning-based techniques have been used to tune the dimensionality of the deformation space [16], [22], [40] using a large amount of annotated data for supervision. [31], [44] formulated learning-based techniques in an unsupervised setting to reconstruct from sparse and dense data, respectively. However, this does not overcome the fundamental limitation of approaches relying on low-rank assumptions: they cannot model complex deformations. Furthermore, they do not naturally handle missing data and occlusions, and complex formulations [15] are required to overcome this. As a result, these methods have been limited to objects that deform in a relatively predictable way, such as human faces. Recently, these limitations have been addressed by imposing constraints between corresponding points across images in one of the following ways.

**Modeling Global Deformations.** Several methods seek to enforce physical properties on the deformation, such as isometry that preserves local distances on the deforming surface. They approximate isometry by inextensibility [9], [20], piece-wise inextensibility [41], [42], [51], local or piece-wise rigidity [8], [25], [47], [50]. A globally optimal solution is then found by jointly solving over all corresponding points. This requires a computationally expensive optimization, which makes this approach impractical for handling large numbers of images. To handle non-isometric surfaces, a mechanics-based approach is proposed in [1], [2], [3], introducing the forces required to compute the resulting shape. In any event, all these methods require an initialization, usually obtained using standard rigid-body reconstruction techniques. Furthermore, they are often inaccurate.

**Modeling Local Deformations.** In earlier works, we have proposed methods that rely on formulating local deformation constraints in terms of algebraic expressions. This makes it possible to reconstruct each surface point independently by solving algebraic equations, which reduces the computation cost. Being local, these methods inherently handle missing data and occlusions. In [37], we treated surfaces as locally planar (LP) and formulated local isometric constraints using metric tensors and connections representing the rate of change of metric tensors. In [38], we extended this deformation modeling to conformal and equiareal deformations by assuming the deformation to be locally linear (LL). For each pair of images, we obtained two cubic equations in two variables related to local depth derivatives with 9 possible solutions. In practice, up to 5 of them can be real. We found a unique solution by minimizing sum-of-squares of residuals over multiple images. In [36], we proposed two fast solutions to the equations of isometric NRSfM [37]. Using substitution and resultants, we converted the original bivariate equations to univariate ones that can be solved efficiently. However, this comes at the cost of adding phantom solutions that cannot be identified. We picked the solution that yields the smallest residual of the isometry constraints on the entire image set. In [39], we proposed an NRSfM solution for generic deformations. It uses only connections to formulate constraints to enforce surface smoothness.

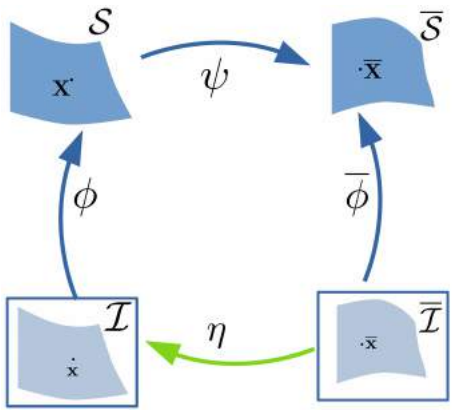


Fig. 1: A 2-view model for NRSfM. Assuming $\psi$ to be locally isometric/conformal, our goal is to find $\phi$, $\overline{\phi}$ given that $\eta$ is known.

Table 1 summarizes the characteristics of these local methods. As in [37], [38], [39], we use the metric tensors and connections to jointly formulate isometric and conformal deformation constraints. However, we formulate these constraints directly in terms of the surface normals rather than of the depth derivatives. As a consequence, the problem is significantly simplified, and we obtain a closed-form solution. By contrast, the approach of [37], [38] relies on a computationally expensive solver while that of [39] requires complex polynomial reduction techniques that add phantom solutions. Furthermore, while existing local methods tend to perform significantly better than their global counterparts but suffer from one key drawback: The local constraints are not always well-posed, leading to many-fold ambiguities or even degenerate solutions, without any mechanism for telling when this happens. In this paper, we address this problem by identifying and discarding the degenerate cases where all of these methods yield an unreliable estimate. This significantly boosts performance over earlier approaches.

## 3 Formalism and Assumptions

At the heart of our approach is the fact that the normals at two different instants at a point on a deforming 3D surface can be computed given the point's projections in two images and a 2D

warp between these images, under the sole assumptions of local surface planarity and deformation local linearity. In this section, we first introduce the NRSfM setup we will use in the rest of this paper, which is similar to the one of [38]. We then explain what our assumptions mean and why they are widely applicable. Finally, we formulate the constraints we will use for reconstruction purposes.

### 3.1 Setup

Fig. 1 depicts our setup when using only two images, $\mathcal{I}$ and $\overline{\mathcal{I}}$, acquired by a calibrated camera. In each one, we denote the deforming surface as $\mathcal{S}$ and $\overline{\mathcal{S}}$, respectively, and model it in terms of functions $\phi, \overline{\phi} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ that associate an image point to a surface point. Let us assume that we are given an image registration function $\eta : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ that associates points in the first image to points in the second. This is often referred to as a warp. In practice, it can be computed using standard image matching techniques, such as optical flow [45], [46] or SIFT [29]. These functions can be composed to create a mapping $\psi : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ between 3D surface points seen in the two images. We use a parametric representation of $\eta$ and $\phi$ with B-splines [6], which allows us to accurately obtain first- and second-order derivatives of these functions. A finite-difference approach could also be used.

Given a point $\mathbf{x} = (u, v)$ on $\mathcal{I}$ and its corresponding 3D point $\mathbf{X} = \phi(\mathbf{x})$ on $\mathcal{S}$, we write $\phi(\mathbf{x}) = \frac{1}{\beta(u,v)} \begin{pmatrix} u & v & 1 \end{pmatrix}^\top$, where $\beta$ represents the inverse of the depth. The Jacobian of $\phi$ is given by

$$\mathbf{J}_\phi = \frac{1}{\beta(u,v)} \begin{pmatrix} 1 - uk_1 & -uk_2 \\ -vk_1 & 1 - vk_2 \\ -k_1 & -k_2 \end{pmatrix}, \tag{1}$$

where $k_1 = \frac{\partial_u \beta}{\beta}$, $k_2 = \frac{\partial_v \beta}{\beta}$ express the surface depth derivatives. $\overline{u}$, $\overline{v}$, $\overline{\phi}$, $\overline{k_1}$, and $\overline{k_2}$ are defined similarly in $\overline{\mathcal{I}}$.

### 3.2 Local Planarity and Linearity

In this work, we assume local planarity of the 3D surfaces and local linearity of the deformations as described in [21], [27]. We now describe these two assumptions and argue that they are weak ones that are generally applicable.

**Surface Local Planarity.** Let $\mathbf{x}_0$ be an image point with surface normal $\mathbf{n}$ at $\phi(\mathbf{x}_0)$. All points $\mathbf{x} = (u, v)$ sufficiently close to $\mathbf{x}_0$ can be accurately described as lying on the tangent plane. Hence, they satisfy $\mathbf{n}^\top \phi(\mathbf{x}) + d = 0$, where $d$ is a scalar, which we can rewrite as $\beta = -\frac{\mathbf{n}^\top}{d} \begin{pmatrix} u & v & 1 \end{pmatrix}^\top$. Therefore the inverse depth $\beta$ that appears in Eq. 1 is a linear function of $\mathbf{x}$ even though $\phi$ is not. Nevertheless, all higher-order derivatives of $\phi$ can be expressed in terms of $\beta$ and its first-order derivatives. This is widely viewed as a weak assumption that applies to most smooth manifolds [27]. For example, our planet is a sphere that can be treated as locally planar.

**Deformation Local Linearity.** According to [21], every non-linear function can be approximated with an infinite number of linear functions. This assumption has been successfully used in shape-matching [32]. We assume the deformation $\psi$ that relates locally two planes to be smooth enough to be well described locally by its first-order approximation, so that we can ignore its second derivatives. In other words, we use a first-order approximation for the local deformations but a second-order one for the surface depth to allow for globally non-planar shapes. This is a looser set of assumptions than what is normally used in NRSfM. For example, [10], [28] and other low-rank methods assume the deformation space to be small; physics-based methods that use inextensibility [9], [51] or piecewise-rigidity [47], [50] make a much stronger assumption.

Under the assumption of local planarity, we have $\mathbf{X}$ and $\overline{\mathbf{X}}$ lying on a planar surface. A generic transformation between these two surfaces, which defines the deformation $\psi$, can be expressed as $\overline{\mathbf{X}} = \mathbf{S}\mathbb{R}\mathbf{X} + \mathbf{T}$, where $\mathbb{R}$ and $\mathbf{T}$ are rotation and translation and $\mathbf{S}$ is a scaling matrix. If $\mathbf{S}$ happens to be a purely diagonal matrix with equal entries, $\psi$ is a planar homography, and the resulting deformation is purely isometric or conformal. Nevertheless, $\psi$ is linear. Therefore, local planarity of surfaces implies local linearity of deformations. However, the reverse is not true.

### 3.3 Differential Constraints across Images

To express constraints between quantities computed in $\mathcal{I}$ and $\overline{\mathcal{I}}$, we define *metric tensors* and *connections* as described in [27].

**Metric Tensors.**

The metric tensors $\mathbf{g}$ in $\mathcal{I}$ and $\overline{\mathbf{g}}$ in $\overline{\mathcal{I}}$ are first-order differential quantities that capture local distances and angles. They can be written as

$$\mathbf{g} = \mathbf{J}_\phi^\top \mathbf{J}_\phi \text{ and } \overline{\mathbf{g}} = \mathbf{J}_{\overline{\phi}}^\top \mathbf{J}_{\overline{\phi}}\,, \tag{2}$$

where $\mathbf{J}_\phi$ and $\mathbf{J}_{\overline{\phi}}$ are local surface jacobians computed according to Eq. 1. These tensors can be used to impose isometry, conformality, and equiareality constraints by forcing the scalars $k_1$ and $k_2$ of Eq. 1 to satisfy one of the three conditions below:

$$\begin{cases} \overline{\mathbf{g}} = \mathbf{J}_\eta^\top \mathbf{g} \mathbf{J}_\eta, & \text{Isometry} \\ \overline{\mathbf{g}} = \lambda^2 \mathbf{J}_\eta^\top \mathbf{g} \mathbf{J}_\eta, \lambda^2 \in \mathbb{R}^+ - \{1\}, & \text{Conformality} \\ \sqrt{\det(\overline{\mathbf{g}})} = \sqrt{\det(\mathbf{J}_\eta^\top \mathbf{g} \mathbf{J}_\eta)}, & \text{Equiareality} \end{cases} \tag{3}$$

where $\mathbf{J}_\eta$ is the Jacobian of the warp $\eta$.

**Linear Relation between Surface Derivatives.**

Given $\mathbf{J}_\phi$, a local reference frame on the surfaces can be expressed with the column vectors as tangents and their cross product as normal. Connections are second-order differential quantities that express the rate of change of this local frame. Using connections under the assumption of local linearity as stated above, it can be shown [38] that

$$\begin{pmatrix} \overline{k}_1 \\ \overline{k}_2 \end{pmatrix} = \mathbf{J}_\eta^\top \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \mathbf{J}_\eta^{-1} \frac{\partial^2 \eta}{\partial \overline{u} \overline{v}}, \tag{4}$$

where $\frac{\partial^2 \eta}{\partial \overline{u} \overline{v}}$ are the second-order derivatives of the warp. Solutions to isometric, conformal and equiareal NRSfM can be obtained by solving the metric tensor preservation equations in Eq. 3 under the constraints of Eq. 4.

### 3.4 Method Overview

Our method, described in the remainder of the paper, is organized as follows. In Section 4, we reformulate the metric tensor preservation constraints for isometric and conformal deformations (Eq. (3)), as well as the linear relation between surface derivatives (Eq. (4)), in terms of normals. Using these two relations derived from an image pair, we define isometric/conformal NRSfM as a system of quadratic equations in two variables. These quadratic equations yield two normals at each point on each image. We use a simple heuristic to obtain a unique solution. Furthermore, for a given image pair, our formulation allows us to assess how well-conditioned the derived NRSfM constraints are. We discuss the

image transformations that lead to ill-conditioned or degenerate data and devise strategies to identify them. Finally, Section 5 proposes an algorithm that uses multiple image pairs to obtain reliable normals from the well-conditioned data only.

## 4 Computing Normals from Two Images

In earlier approaches [38], the NRSfM problem was addressed by solving the system of Eq. 3 under the isometry, conformality, and equiareality constraints of Eq. 3 with respect to the variables $k_1$ and $k_2$ of Eq. 1. Here, we solve this system of equations directly in terms of the surface normals. We will show that, not only can this be done in closed form, but it also allows us to identify degenerate situations that result in unreliable estimates.

**Differentiating the Warp.**

Let us consider a point $\overline{\mathbf{x}} = (\overline{u}, \overline{v})^T$ in $\overline{\mathcal{I}}$ and its corresponding point $(u, v)^T = \eta(\overline{u}, \overline{v})$ in $\mathcal{I}$, with corresponding points on surfaces $\mathbf{X}$ and $\overline{\mathbf{X}}$. Assuming the surfaces to be locally planar means that there is a $3 \times 3$ homography matrix $\mathbf{H} = [h_{ij}]_{1 \leq i,j \leq 3}$ such that $\mathbf{X} = \lambda \mathbf{H} \overline{\mathbf{X}}$. Since we assume a perspective projection for the camera, we write

$$\mathbf{x} = \frac{1}{\overline{s}} \mathbf{H} \overline{\mathbf{x}} \implies \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{\overline{s}} \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} \overline{u} \\ \overline{v} \\ 1 \end{pmatrix}, \tag{5}$$

where $\overline{s} = h_{31}\overline{u} + h_{32}\overline{v} + h_{33}$. The first- and second-order derivatives of $\eta$ can be computed as

$$\mathbf{J}_\eta = \begin{pmatrix} \frac{\partial \eta}{\partial \overline{u}} & \frac{\partial \eta}{\partial \overline{v}} \end{pmatrix} = \frac{1}{\overline{s}} \begin{pmatrix} h_{11} - h_{31}u & h_{12} - h_{32}u \\ h_{21} - h_{31}v & h_{22} - h_{32}v \end{pmatrix},$$
$$\begin{pmatrix} \frac{\partial^2 \eta}{\partial \overline{u}^2} & \frac{\partial^2 \eta}{\partial \overline{u}\partial \overline{v}} & \frac{\partial^2 \eta}{\partial \overline{v}^2} \end{pmatrix} = -\frac{1}{\overline{s}} \mathbf{J}_\eta \begin{pmatrix} 2h_{31} & h_{32} & 0 \\ 0 & h_{31} & 2h_{32} \end{pmatrix}. \tag{6}$$

**Image Embedding and Local Normal.**

The unit normal $\mathbf{n}$ at $\mathbf{x}$ is the cross product of the columns of the matrix $\mathbf{J}_\phi$ from Eq. 1. This lets us write

$$\mathbf{n} = \frac{1}{\beta^2 \sqrt{\det \mathbf{g}}} \begin{pmatrix} k_1 \\ k_2 \\ 1 - uk_1 - vk_2 \end{pmatrix} \tag{7}$$
$$= \frac{1}{\beta^2 \sqrt{\det \mathbf{g}}} \begin{pmatrix} \mathbf{I}_{2\times 2} & 0 \\ -\mathbf{x}^\top & 1 \end{pmatrix} \begin{pmatrix} k_1 \\ k_2 \\ 1 \end{pmatrix}.$$
$$\Rightarrow \begin{pmatrix} k_1 \\ k_2 \\ 1 \end{pmatrix} = \beta^2 \sqrt{\det \mathbf{g}} \begin{pmatrix} \mathbf{I}_{2\times 2} & 0 \\ \mathbf{x}^\top & 1 \end{pmatrix} \mathbf{n}. \tag{8}$$

Given the normal $\mathbf{n}$ of Eq. 7, we rewrite the matrix $\mathbf{J}_\phi$ of Eq. 1 as

$$\mathbf{J}_\phi = \frac{1}{\beta} \begin{pmatrix} 0 & uk_1 + vk_2 - 1 & k_2 \\ 1 - uk_1 - vk_2 & 0 & -k_1 \\ -k_2 & k_1 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 0 \\ v & -u \end{pmatrix}$$
$$= \beta \sqrt{\det \mathbf{g}} [\mathbf{n}]_\times \mathbf{E}. \tag{9}$$

We can now rewrite the differential constraints across images introduced in Section 3.3 in terms of the normals.

**Linear Relation between Surface Normals.**

Given the $\eta$ derivatives from Eq. 6, the linear relation of Eq. 4 becomes

$$\begin{pmatrix} \overline{k}_1 \\ \overline{k}_2 \end{pmatrix} = \mathbf{J}_\eta^\top \begin{pmatrix} k_1 \\ k_2 \end{pmatrix} + \frac{1}{\overline{s}} \begin{pmatrix} h_{31} \\ h_{32} \end{pmatrix}. \tag{10}$$

Defining $\mathbf{m} = \frac{1}{\overline{s}} \begin{pmatrix} h_{31} \\ h_{32} \end{pmatrix}$ lets us rewrite the above equation as

$$\begin{pmatrix} \overline{k}_1 \\ \overline{k}_2 \\ 1 \end{pmatrix} = \begin{pmatrix} \mathbf{J}_\eta^\top & \mathbf{m} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} k_1 \\ k_2 \\ 1 \end{pmatrix}. \tag{11}$$

Using Eq. 8, we reformulate the above expression as

$$\begin{aligned} \overline{\mathbf{n}} &= \frac{\beta^2}{\overline{\beta}^2} \sqrt{\frac{\det \mathbf{g}}{\det \overline{\mathbf{g}}}} \mathbf{T}\mathbf{n} \\ &= \frac{\beta^2}{\overline{\beta}^2} \sqrt{\frac{\det \mathbf{g}}{\det \overline{\mathbf{g}}}} \begin{pmatrix} \mathbf{I}_{2\times 2} & 0 \\ -\overline{\mathbf{x}}^\top & 1 \end{pmatrix} \begin{pmatrix} \mathbf{J}_\eta^\top & \mathbf{m} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{I}_{2\times 2} & 0 \\ \mathbf{x}^\top & 1 \end{pmatrix} \mathbf{n} \\ &= \frac{\beta^2}{\overline{s}\overline{\beta}^2} \sqrt{\frac{\det \mathbf{g}}{\det \overline{\mathbf{g}}}} \begin{pmatrix} \mathbf{I}_{2\times 2} & 0 \\ -\overline{\mathbf{x}}^\top & 1 \end{pmatrix} \begin{pmatrix} h_{11} & h_{21} & h_{31} \\ h_{12} & h_{22} & h_{32} \\ \overline{s}u & \overline{s}v & \overline{s} \end{pmatrix} \mathbf{n} \\ &= \frac{\beta^2}{\overline{s}\overline{\beta}^2} \sqrt{\frac{\det \mathbf{g}}{\det \overline{\mathbf{g}}}} \mathbf{H}^\top \mathbf{n}, \end{aligned} \tag{12}$$

which directly relates the two normals.

**Metric Tensor.**

As shown in Fig. 1, we can write $\overline{\phi} = \psi \circ \phi \circ \eta$. Differentiating this expression and multiplying it by its transpose yields

$$\overline{\mathbf{g}} = \mathbf{J}_{\overline{\phi}}^\top \mathbf{J}_{\overline{\phi}} = \mathbf{J}_\eta^\top \mathbf{J}_\phi^\top \mathbf{J}_\psi^\top \mathbf{J}_\psi \mathbf{J}_\phi \mathbf{J}_\eta. \tag{13}$$

Using Eq 9, we write $\mathbf{J}_\phi \mathbf{J}_\eta = \beta \sqrt{\det g} [\mathbf{n}]_\times \mathbf{E} \mathbf{J}_\eta$. Given the $\eta$ derivatives of Eq. 6, we simplify $\mathbf{E}\mathbf{J}_\eta$ to $\frac{1}{\overline{s}} \begin{pmatrix} \mathbf{h}_1 \times \hat{\mathbf{x}} & \mathbf{h}_2 \times \hat{\mathbf{x}} \end{pmatrix}$, where $\mathbf{h}_1, \mathbf{h}_2$ are the first two columns of the homography matrix $\mathbf{H}$, and $\hat{\mathbf{x}} = \begin{pmatrix} u & v & 1 \end{pmatrix}^\top$. By writing $\mathbf{z}_1 = \mathbf{n} \times (\mathbf{h}_1 \times \hat{\mathbf{x}})$ and $\mathbf{z}_2 = \mathbf{n} \times (\mathbf{h}_2 \times \hat{\mathbf{x}})$, Eq. 3 reduces to

$$\begin{cases} \overline{\mathbf{g}} = \frac{\lambda^2 \beta^2 \det(\mathbf{g})}{\overline{s}^2} \begin{pmatrix} \mathbf{z}_1^\top \mathbf{z}_1 & \mathbf{z}_1^\top \mathbf{z}_2 \\ \mathbf{z}_1^\top \mathbf{z}_2 & \mathbf{z}_2^\top \mathbf{z}_2 \end{pmatrix}, \\ \sqrt{\det(\overline{\mathbf{g}})} = \sqrt{\det(\mathbf{J}_\eta^\top \mathbf{g} \mathbf{J}_\eta)}. \end{cases} \tag{14}$$

**NRSfM from Isometric/Conformal Constraints.**

So far, we have expressed the metric preservation conditions in terms of the normals of the two surfaces under consideration. The only unknown left in the system is therefore $\mathbf{n}$. We now show that this unknown can in fact be computed in closed form.

Given the multiplicative nature of the cross product, the constraints on the normals of Eq. 12 imply that

$$[\overline{\mathbf{n}}]_\times = \frac{\beta^2}{\overline{s}\overline{\beta}^2} \sqrt{\frac{\det \mathbf{g}}{\det \overline{\mathbf{g}}}} \det(\mathbf{H}^\top) \mathbf{H}^{-1} [\mathbf{n}]_\times \mathbf{H}^{-\top}. \tag{15}$$

This lets us rewrite the matrix $\mathbf{J}_{\overline{\phi}}$ of Eq. 9 as

$$\begin{aligned} \mathbf{J}_{\overline{\phi}} &= \frac{\beta^2}{\overline{\beta}} \sqrt{\det \mathbf{g}} \mathbf{H}^{-1} [\mathbf{n}]_\times \left( \frac{\det \mathbf{H}^\top}{\overline{s}} \mathbf{H}^{-\top} \overline{\mathbf{E}} \right) \\ &= \frac{\beta^2 \sqrt{\det \mathbf{g}}}{\overline{\beta}} \mathbf{H}^{-1} [\mathbf{n}]_\times \begin{pmatrix} \mathbf{h}_1 \times \hat{\mathbf{x}} & \mathbf{h}_2 \times \hat{\mathbf{x}} \end{pmatrix} \\ &= \frac{\beta^2 \sqrt{\det \mathbf{g}}}{\overline{\beta}} \mathbf{H}^{-1} \begin{pmatrix} \mathbf{z}_1 & \mathbf{z}_2 \end{pmatrix}. \end{aligned} \tag{16}$$

Injecting this expression into the isometric/conformal metric tensor preservation relation of Eq. 14 yields

$$\begin{pmatrix} \mathbf{z}_1^\top \mathbf{H}^{-\top} \mathbf{H}^{-1} \mathbf{z}_1 & \mathbf{z}_1^\top \mathbf{H}^{-\top} \mathbf{H}^{-1} \mathbf{z}_2 \\ \mathbf{z}_1^\top \mathbf{H}^{-\top} \mathbf{H}^{-1} \mathbf{z}_2 & \mathbf{z}_2^\top \mathbf{H}^{-\top} \mathbf{H}^{-1} \mathbf{z}_2 \end{pmatrix} = \frac{\lambda^2 \overline{\beta}^2}{\overline{s}^2 \beta^2} \begin{pmatrix} \mathbf{z}_1^\top \mathbf{z}_1 & \mathbf{z}_1^\top \mathbf{z}_2 \\ \mathbf{z}_1^\top \mathbf{z}_2 & \mathbf{z}_2^\top \mathbf{z}_2 \end{pmatrix},$$

Fig. 2: Two images with ill-conditioned data shown in red. The estimated normals are highly erroneous in this region. To compute the normals in this region, third image is considered.

$$\Rightarrow \mathbf{z}_i^\top \left( \overline{\mathbf{H}}^\top \overline{\mathbf{H}} - \frac{\lambda^2 \overline{\beta}^2}{\overline{s}^2 \beta^2} \mathbf{I}_{3\times 3} \right) \mathbf{z}_j = 0, \forall i,j \in \{1,2\}, \tag{17}$$

where $\overline{\mathbf{H}} = \mathbf{H}^{-1}$. Assuming $\mathbf{H}$ to be normalized, that is, its second singular value to be 1, the relation between a 3D point observed in the two input images is given by $\phi(\mathbf{x}) = \mathbf{H}\phi(\overline{\mathbf{x}})$. Using Eq. 5 yields $\overline{\beta} = \beta\overline{s}$. By writing $\mathbf{z}_i = [\mathbf{n}]_\times [\mathbf{h}_i]_\times \hat{\mathbf{x}}$, the above constraints further simplify to

$$[\mathbf{n}]_\times^\top (\overline{\mathbf{H}}^\top \overline{\mathbf{H}} - \lambda^2 \mathbf{I}_{3\times 3}) [\mathbf{n}]_\times = 0. \tag{18}$$

Since $\mathbf{H} \sim \alpha\mathbf{H}$, we divide the above expression by $\lambda^2$ and, with a slight abuse of notation, write $\frac{1}{\lambda}\overline{\mathbf{H}}$ as $\overline{\mathbf{H}}$. This simplifies the above expressions to

$$[\mathbf{n}]_\times^\top (\overline{\mathbf{H}}^\top \overline{\mathbf{H}} - \mathbf{I}_{3\times 3}) [\mathbf{n}]_\times = [\mathbf{n}]_\times^\top \mathbf{S} [\mathbf{n}]_\times = 0. \tag{19}$$

**Degenerate Cases.** The system of Eq. 19 holds as long as $\mathbf{S}$ is a non-null matrix, which means $\overline{\mathbf{H}}^\top \overline{\mathbf{H}} \neq \mathbf{I}_{3\times 3}$. Therefore, $\overline{\mathbf{H}}$ should not be an orthogonal matrix. $\overline{\mathbf{H}}$ will be orthogonal if the relative transformation between the two images is 1) non-existent (zero relative motion); 2) purely translational; 3) purely rotational; or 4) purely reflective. Therefore, given two distinct images, reconstruction is not possible if one of them is a rotated, translated or flipped version of the other. In a local framework, each point correspondence must avoid these four traps to yield a normal. The chances of facing degenerate data are therefore much higher for local methods than for global ones. For example, consider the first two images in Fig. 2. While these images are globally distinct, the central portions (shown in red) are very close to being related by a pure translation. The normals computed in this region by other local methods [36], [38], [39] are thus unreliable. We classify this region to be degenerate and ignore the computed normals. However, when each of these images are paired with the third image, there are no degeneracies encountered. Therefore, the normals in red regions can be reconstructed by considering the third image.

**Affine Stability.** Under affine imaging conditions, $h_{31} = h_{32} = 0$, and $h_{33} = 1$. In this case, $\mathbf{z}_i$ and $\mathbf{S}$ remain non-null, and thus the system in Eq. 19 does *not* become degenerate, and we can still compute the normal.

**Solution.** The solution to the system in Eq. 19 can be obtained by homography decomposition [30]. We give an overview of the solution here but recommend reading [30] for more detail.

$\mathbf{S} = \{s_{ij}\}$ is a symmetric matrix expressed in terms of $\overline{\mathbf{H}}$. It can be numerically computed using $\eta$ and image observations $(\mathbf{x}, \overline{\mathbf{x}})$. Specifically, Eq. 12 gives the closed-form definition $\mathbf{H}^\top = \begin{pmatrix} \mathbf{I}_{2\times 2} & 0 \\ -\overline{\mathbf{x}}^\top & 1 \end{pmatrix} \begin{pmatrix} \mathbf{J}_\eta^\top & \mathbf{m} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{I}_{2\times 2} & 0 \\ \mathbf{x}^\top & 1 \end{pmatrix}$. Let us write $\mathbf{n} = \begin{pmatrix} n_1 & n_2 & n_3 \end{pmatrix}^\top$. Since $n_3 \neq 0$, we define $y_1 = \frac{n_1}{n_3}$ and $y_2 = \frac{n_2}{n_3}$ and expand the system in Eq. 19 accordingly. This yields 6 constraints, out of which only 3 are unique. They are given by

$$\begin{aligned} s_{33}y_2^2 - 2s_{23}y_2 + s_{22} &= 0\,, \\ s_{33}y_1^2 - 2s_{13}y_1 + s_{11} &= 0\,, \\ s_{22}y_1^2 - 2s_{12}y_1y_2 + s_{11}y_2^2 &= 0\,. \end{aligned} \tag{20}$$

By solving the first two, we obtain $y_1 = \dfrac{s_{13} \pm \sqrt{s_{13}^2 - s_{33}s_{11}}}{s_{33}}$ and $y_2 = \dfrac{s_{23} \pm \sqrt{s_{23}^2 - s_{33}s_{22}}}{s_{33}}$. We use the third expression to disambiguate the solutions. Ultimately, this gives us closed-form expressions for the two potential solutions for the normal, written as

$$\begin{aligned} \mathbf{n}_a &= \begin{pmatrix} s_{13} + s\sqrt{s_{13}^2 - s_{33}s_{11}} & s_{23} + \sqrt{s_{23}^2 - s_{33}s_{22}} & s_{33} \end{pmatrix}^\top, \\ \mathbf{n}_b &= \begin{pmatrix} s_{13} - s\sqrt{s_{13}^2 - s_{33}s_{11}} & s_{23} - \sqrt{s_{23}^2 - s_{33}s_{22}} & s_{33} \end{pmatrix}^\top, \\ &\text{where } s = sign(s_{23}s_{13} - s_{12}s_{33}). \end{aligned} \tag{21}$$

**Normal Validation.** The normals thus obtained must be visible to the camera. Given the analytical normal in Eq. 7, $\mathbf{n}_a$ and $\mathbf{n}_b$ are visible if $\frac{s_{33}}{1-uk_1-vk_2} > 0$, i.e., they have a similar orientation towards the camera. We discard the normals that do not meet the visibility constraint.

**Normal Selection.** Using Eq. 8, the local depth derivatives $(k_1, k_2)$ at $\mathbf{X}$ are given by $k_i = \frac{n_i}{un_1+vn_2+n_3}$. From the solution in Eq. 21, we thus obtain two possible solutions for the local depth derivatives $(k_{1a}, k_{2a})$ and $(k_{1b}, k_{2b})$. We pick the normal that minimizes the corresponding sum of squares of depth derivatives. That is, we compute the normal $\mathbf{n}$ as

$$\mathbf{n} = \begin{cases} \mathbf{n}_a & \text{if } k_{1a}^2 + k_{2a}^2 \leq k_{1b}^2 + k_{2b}^2 \\ \mathbf{n}_b & \text{otherwise.} \end{cases} \tag{22}$$

Following Eq. 5, $\overline{\mathbf{n}}$ is then obtained as $\mathbf{H}^\top \mathbf{n}$.

**Measure of Degeneracy.** In degenerate situations, the singular values $(\sigma_1, \sigma_2, \sigma_3)$ of $\mathbf{H}$ are all one. We use the ratio $\frac{\sigma_1}{\sigma_3}$ to quantify the degeneracy. Thus, we only reconstruct from $\mathbf{S}$ if $\frac{\sigma_1}{\sigma_3} > \tau$, and we set $\tau = 1.05$.

**Surface Reconstruction.** We consider a planar surface and bend it to match the normals obtained using the homography decomposition mentioned above, as opposed to [36], [38], [39] which integrate the normals on each surface. The upside of surface bending is that it does not require to set a smoothness parameter, which needs to be tuned for the normal integration. Furthermore, surface bending is much faster than its normal integration counterpart in the presence of dense data. It is also less affected by the noise in the normals corresponding to high-perspective image regions.

## 5 Normals from Multiple Images

Methods such as those of [36], [38], [39] pick a reference image and formulate reconstruction constraints between it and the other images, which are then solved by solving a least-squares problem over the entire set of images. We use the same strategy, except that we reconstruct from all image pairs, with each image acting in turn as the reference image for the other ones. Therefore, given $N$ images, for each reference image, we obtain $N-1$ estimates for the reference image and 1 estimate for each of the non-reference images. By considering all image pairs, we obtain $2(N-1)$ estimates for the normals on each image. In other words, the use

of multiple image pairs yields more estimates for each normals, which in turns allows us to obtain more reliable estimates, particularly in degenerate regions such as those highlighted in Fig. 2.

More formally, let $\{\mathbf{x}_j^i\}, i \in [1, M], j \in [1, N]$, be a set of $N$ point correspondences between $M$ images. Our goal is to find the 3D point $\mathbf{X}_j^i$ and the normal $\mathbf{n}_j^i$ corresponding to each $\mathbf{x}_j^i$. Using Eq. 12, we write the local homography for each point correspondence $\mathbf{H}_j^{ik}$ between image pairs $(i, k) \in [1, M], i \neq k$, using the warp $\eta$. Each local homography $\mathbf{H}_j^{ik}$ is normalized by dividing it by its second singular value. We compute $\mathbf{Hc}_j^{ik}$ given by the ratio of the first and third singular value, and the normals for each local homography $\mathbf{H}_j^{ik}$ using Eq. 21. We then pick a unique solution using Eq. 22. The solution on the reference and non-reference image is given by $\mathbf{n}_j^{kk}$ and $\mathbf{n}_j^{ki}$, respectively. For non-degenerate cases, where $\frac{\sigma_1}{\sigma_3} \geq 1.05$, we compute the normal $\mathbf{n}_j^i$ by taking the median of the $\mathbf{n}_j^{ik}$s computed over $k$ reference images. We obtain a 3D surface by bending a planar surface to match the obtained normals on each surface.

We summarize our complete pipeline in Algorithm 1.

**Algorithm 1:** Our NRSfM Algorithm

**Data:** $\mathbf{x}_j^i, \mathbf{H}_j^{ik}$ and $\mathbf{Hc}_j^{ik}$
**Result:** $\mathbf{n}_j^i$
$\frac{\sigma_1}{\sigma_3} = 1.05$;
**for** *each reference image* $k = [1, M]$ **do**
  **for** *each point* $j = [1, N]$ **do**
    **for** *images* $i = [1, M], i \neq k$ **do**
      **if** $\mathbf{Hc}_j^{ik} > \frac{\sigma_1}{\sigma_3}$ **then**
        Compute normals using (21);
        Pick a solution $\mathbf{n}_j^{kk}$ using (22);
        Write $\mathbf{n}_j^{ik} = (\mathbf{Hc}_j^{ik})^\top \mathbf{n}_j^{kk}$;
      **else**
        Set $\mathbf{n}_j^{ik}, \mathbf{n}_j^{kk}$ to zero;
      **end**
    **end**
  **end**
**end**
**for** *each point* $j = [1, N]$ **do**
  **for** *images* $i = [1, M]$ **do**
    Obtain $\mathbf{n}_j^i$ by as the median of the non-zero $\mathbf{n}_j^{ik}$s;
  **end**
**end**

## 6 Experiments

We compare our method against state-of-the-art ones on both synthetic and real datasets with available ground truth.

### 6.1 Datasets

Our datasets include the ones we used in our previous work, the one of [44] and the NRSfM challenge dataset [19]. Note that Kinect, 3D scanners, and 3D reconstruction toolboxes provide noisy depth observations which cannot be corrected, even manually, beyond a limited extent. The performance on these dataset is therefore slightly approximative. The NRSfM Challenge Dataset has been synthetically created using 3D creation software such as Blender and is thus accurate. Therefore, the relative performance of the methods evaluated on our datasets (and the ones of [44]) can be slightly different than on the NRSfM Challenge Dataset.

**Synthetic Datasets.** We created 3 smooth surfaces: a plane, a cylindrical surface and a stretched surface with 400 tracked correspondences, as shown in Figure 3.

**Real Datasets from our Previous Work.**

These include the **Paper** [43], **Rug** [37] and **Tshirt** [8] datasets. **Paper** comprises 191 images from a video of a deforming sheet of paper with 1500 point correspondences. **Rug** comprises 159 images from a video of a deforming rug with 3900 point correspondences. **Tshirt** has 10 wide-baseline images with 85 point correspondences. The correspondences in the **Paper** dataset were obtained using SIFT with a manual supervision of accuracy and are thus highly accurate. By contrast, those in the **Rug** dataset were computed using the dense optical flow method of [14] and contain errors due to optical drift and regional mismatches due to the lack of texture. The correspondences in **Tshirt** were computed manually. The ground truth for **Paper** and **Rug** was obtained using a kinect, and is thus very noisy, jittery and contains large, inconsistent depth variations. We manually checked each frame for inaccuracies, and used B-spline warps to fit a smooth surface onto the noisy data and obtained a surface representation of the ground truth. The ground truth for **Tshirt** was computed using rigid reconstruction of each image from multiple views.

**NRSfM Challenge Dataset [19].** It consists of 5 image sequences depicted by Fig. 6. They feature 5 kinds of non-rigid motions: articulated (piecewise-rigid) with 207 images and 69 point correspondences, balloon (conformal) with 51 images and 211 point correspondences, paper bending (isometric) with 40 images and 153 point correspondences, rubber (elastic) with 40 images and 481 point correspondences, and paper being torn with 432 images and 405 point correspondences. The dataset features images from 6 different camera motions and provides image points captured assuming both a perspective and an orthographic projection. It provides only one ground-truth surface for each of the sequences. The correspondences are sparse and not well-distributed across the images.

**Datasets used by [44].** [44] released the **Paper**, **Tshirt**, **Actor** and **Expressions** datasets, which have been widely used by many physics-based and low-rank constraints based methods. The **Paper** images are the same as the one used by us. [44] uses 60K dense correspondences computed using optical flow [14] and the raw depth data from the kinect is considered as the ground truth. The **Tshirt** data has 300 images with 70K dense correspondences computed using [14], with the kinect raw depth data as ground truth. To deal with the inconsistent depth variations of the raw kinect data, [44] refines the raw data and focuses on small portions of theses datasets where the inconsistent depth variations are minimal, as shown in Fig 4. **Actor** contains 100 images of a deforming human face with 36K dense correspondences, and **Expressions** includes 384 3D shapes of a deforming human faces with 1000 point correspondences. The ground truth for both these datasets is synthetic. Fig 4 shows some samples.

Additionally, [44] released **Back**, **Owl** and **Heart** video sequences with dense correspondences computed using [14]. The ground truth for these datasets is not available. **Back** contains 150 images of large deformations of the back with 20K dense correspondences. **Owl** contains 202 images of an owl with 20K dense correspondences. **Heart** contains 80 images of a beating heart under surgery with 68K dense correspondences.

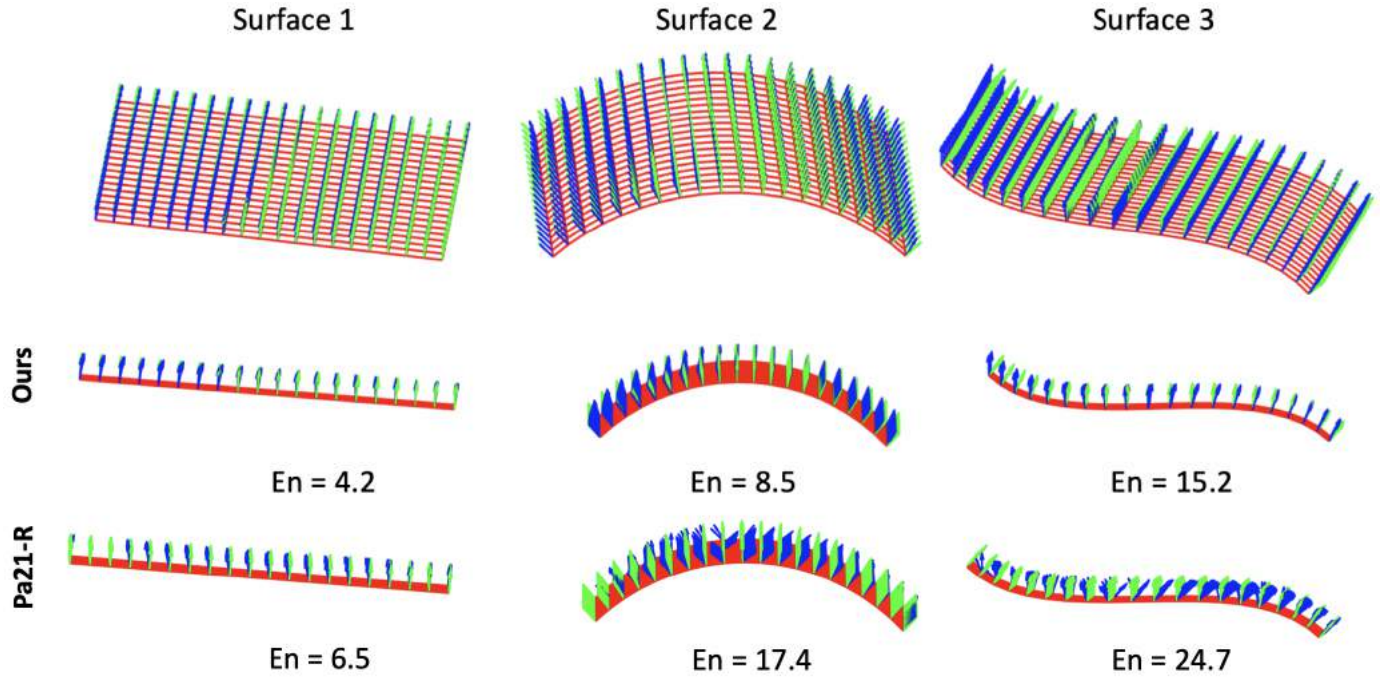


Fig. 3: **Reconstructed normals**. A synthetic deforming surface reconstructed in three different frames. The predicted normals are shown in blue and the ground-truth ones in green.


Fig. 4: **Datasets used by [44].** Sample images and region of interest used for depth computation for **Paper**, **Tshirt** and **Actor** sequences. For **Expressions**, sample ground truth shape is shown.

**Blue Sheet Dataset.** Additionally, we recorded a video sequence featuring a textureless blue sheet deforming isometrically using a kinect. We used B-spline warps to fit a smooth surface onto the ground truth. The sequence comprises 60 images and 7K point correspondences that were tracked using dense optical flow [14]. Optical flow on textureless surfaces is prone to large errors, and the flow we obtained confirms this.

### 6.2 Baselines and Metrics

We compare our method to local linearity-based diffeomorphic NRSfM **Pa20** [39], jointly solving isometric/conformal NRSfM **Pa19** [38], two fast solutions **Pa21-R** and **Pa21-S** [36] that transform the original constraints to univariate polynomials, which can be easily solved, and local and piecewise homography decomposition, **Ch14** [8] and **Va09** [50], respectively. These are methods that, like ours, reconstruct local/piecewise surface normals and integrate them to obtain depth. Note that the solution to isometric NRSfM in [37] is the same as the one in **Pa19**. Therefore, there is no need for additional comparison.

We report errors in terms of accuracy of the normals *En* and 3D points *Ed*. *En* is computed as the average dot product between ground-truth and computed normals. The normal integration done in the above methods yields a smooth reconstruction by enforcing a local smoothness on the normals. As a consequence, it improves

TABLE 2: **Synthetic experiments results**. 'X' indicates that the method does not return a result because we are not using enough images.

| Method | Surfaces 1, 2 | | | Surfaces 1,3 | | | Surfaces 1,2,3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | En | En (s) | Ed | En | En (s) | Ed | En | En (s) | Ed |
| **Lee16** | X | X | X | X | X | X | X | X | X |
| **An17** | X | X | X | X | X | X | X | X | X |
| **Va09** | 16.4 | 12.3 | 10.2 | 24.5 | 16.7 | 12.1 | 17.3 | 16.0 | 9.8 |
| **Ch14** | X | X | X | X | X | X | 28.1 | 24.1 | 20.2 |
| **Ch17** | X | X | X | X | X | X | X | X | 14.5 |
| **Ji17** | X | X | X | X | X | X | X | X | 15.6 |
| **Pa19** | X | X | X | X | X | X | 17.4 | 13.6 | 4.3 |
| **Pa20** | X | X | X | X | X | X | 24.7 | 15.6 | 9.3 |
| **Pa21-S** | X | X | X | X | X | X | 16.4 | 13.3 | 4.2 |
| **Pa21-R** | X | X | X | X | X | X | 16.2 | 13.3 | 4.2 |
| **Ours** | **4.0** | **3.5** | **2.1** | **8.3** | **7.6** | **4.3** | **9.3** | **8.4** | **3.2** |

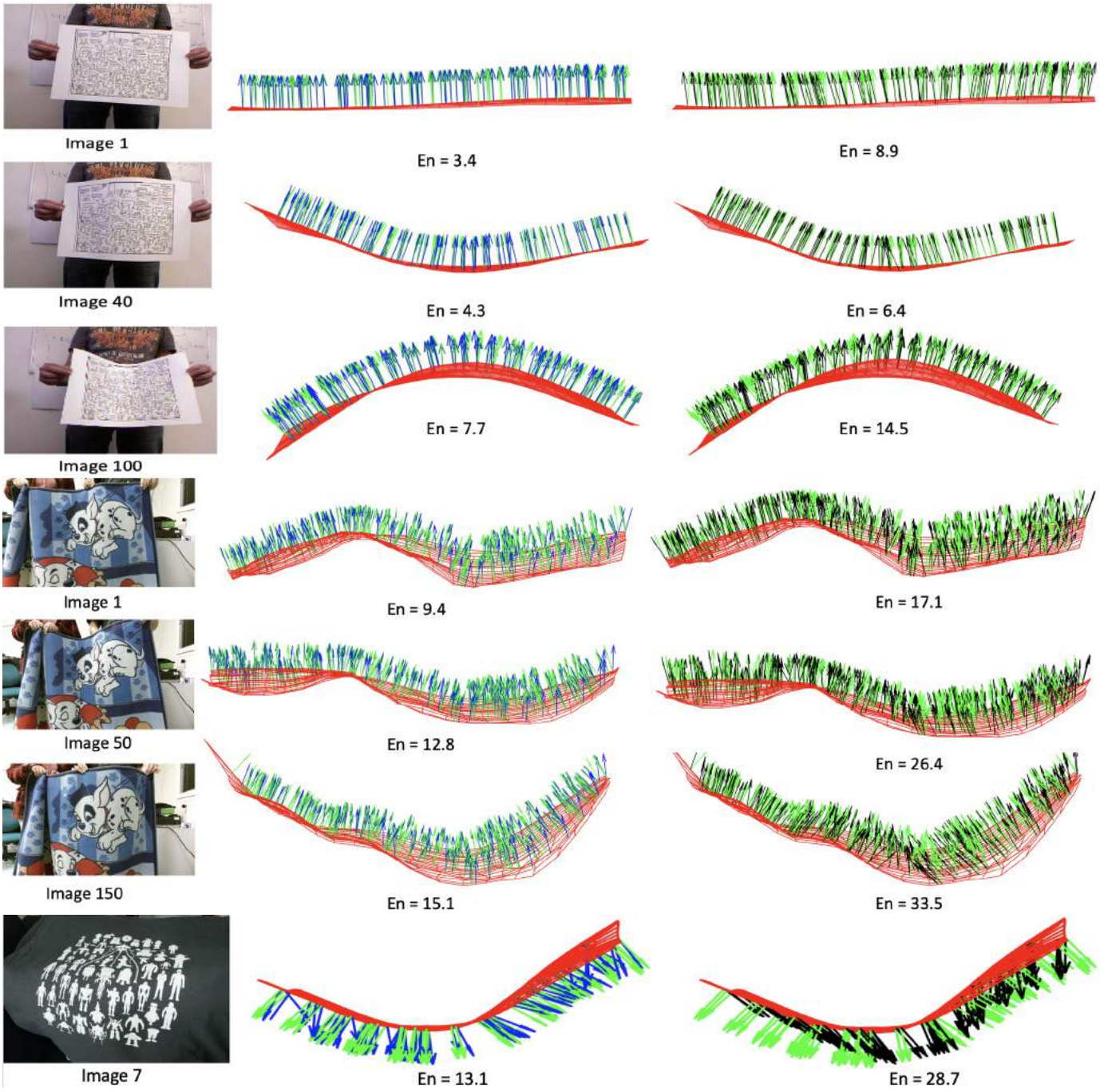


Fig. 5: **Datasets used in our previous work.** Reconstructed normals on three images. The ground-truth normals are shown in green, the ones predicted by **Ours** in blue, and those by **Pa21-R** in black. Note that our normals are far less noisy.

the quality of the reconstructed normals. Therefore, we also report *En (s)*, which is the error between the smoothened and the ground-truth normals. *Ed* is the mean RMSE between the ground-truth and computed 3D points.

We also compare our approach against three of the best global methods, **Ch17** [9], **Ji17** [20] and **Lee16** [28], along with a dense method, **An17** [5]. They directly return 3D points. Hence, we only report *Ed* for these methods.

While comparing on the datasets used by [44], we report $Ed$ as the mean 3D error, as computed in this method. Therefore, $Ed = \frac{1}{N}\sum_t \frac{||P_{recon} - P_{GT}||_2}{||P_{GT}||_2}$, where $P_{recon}$ is the obtained reconstruction, $P_{GT}$ is the ground truth and $N$ is the number of images in the dataset.

In the remainder of this section we will refer to the method described in this paper as **Ours**.

### 6.3 Comparative Results

**Results on Synthetic Data.** Fig. 3 shows the generated surfaces. The performance of all methods is averaged over 10 trials with added gaussian noise with a 3 pixels standard deviation. As **Ours** can reconstruct from two images only, we perform both pairwise reconstructions and joint reconstruction from the image

TABLE 3: (left) **RMSE results on the datasets used in our previous work**. 'X' indicates that the method does not evaluate normals. '—-' indicates that method failed to return a result due to its high computational complexity.

| Method | Paper (partial) | | | Rug (partial) | | | Paper (full) | | | Rug (full) | | | Tshirt (full) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | En | En (s) | Ed | En | En (s) | Ed | En | En (s) | Ed | En | En (s) | Ed | En | En (s) | Ed |
| **Lee16** | X | X | 21.6 | X | X | 89.8 | X | X | 21.9 | X | X | 90.7 | X | X | X |
| **An17** | X | X | 14.7 | X | X | 60.6 | X | X | 14.7 | X | X | 63.7 | X | X | X |
| **Ch14** | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | 23.4 | 16.5 | 12.6 |
| **Va09** | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | 27.1 | 16.8 | 14.6 |
| **Ch17** | X | X | 5.4 | X | X | 63.5 | ------ | ------ | ------ | ------ | ------ | ------ | X | X | **3.7** |
| **Ji17** | X | X | 5.7 | X | X | 67.1 | ------ | ------ | ------ | ------ | ------ | ------ | X | X | 5.2 |
| **Pa19** | 17.3 | 10.5 | 8.3 | 34.5 | 16.7 | 52.4 | 16.8 | 8.6 | 7.2 | 35.8 | 18.2 | 54.3 | 35.8 | 17.2 | 8.9 |
| **Pa20** | 20.7 | 19.4 | 10.2 | 28.1 | 21.5 | 46.1 | 24.8 | 19.0 | 11.3 | 29.4 | 22.1 | 47.1 | 29.4 | 27.0 | 13.0 |
| **Pa21-S** | 15.6 | 9.3 | 5.9 | 26.6 | 15.5 | 40.1 | 18.4 | 8.8 | 5.3 | 31.0 | 17.5 | 43.4 | 31.0 | 17.2 | 7.1 |
| **Pa21-R** | 15.3 | 9.0 | 5.9 | 26.8 | 15.7 | 40.8 | 19.8 | 8.0 | 5.2 | 32.2 | 17.8 | 44.5 | 30.2 | 16.5 | 7.1 |
| **Ours** | **8.9** | **8.3** | **3.9** | **18.3** | **15.5** | **25.9** | **9.1** | **8.4** | **4.1** | **19.3** | **16.3** | **31.1** | **18.4** | **16.3** | 6.6 |

TABLE 4: **Computation times** as a function of the number of images and points used.

350 points

| Images | Lee16 | An17 | Va09 | Ch14 | Ch17 | Ji17 | Pa19 | Pa20 | Pa21-S | Pa21-R | Ours |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 17.8 | 10 | 69.4 | 75.3 | 31.3 | 341 | 9.7 | 24.5 | 4.1 | 4.1 | 0.2 |
| 30 | 23.4 | 12 | 3103 | 3407 | 129 | ------ | 12.5 | 32.7 | 9.3 | 9.4 | 0.6 |
| 60 | 45.6 | 19 | ------ | ------ | ------ | ------ | 14.8 | 45.3 | 11.4 | 11.4 | 1.9 |

1500 points

| Images | Lee16 | An17 | Va09 | Ch14 | Ch17 | Ji17 | Pa19 | Pa20 | Pa21-S | Pa21-R | Ours |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 256 | 12 | 1435 | 1256 | 995 | 2532 | 103 | 745 | 15.7 | 15.6 | 0.5 |
| 30 | 987 | 14 | ------ | ------ | 3400 | ------ | 118 | 1205 | 73 | 75 | 2.0 |
| 60 | 1705 | 22 | ------ | ------ | ------ | ------ | 124 | 1807 | 90 | 93 | 5.8 |

triplet available for each surface. We report the results in Table 2. For methods that perform normal integration, we report errors of both computed and smoothened normals. The improvement in the normals due to smoothing is huge for **Ch14** and **Va09**, substantial for **Pa19**, **Pa20**, **Pa21-S** and **Pa21-R** and minor for our method. To truly compare the NRSfM techniques themselves, we therefore report the accuracy of the computed normals rather than the smoothed ones. We obtain a very accurate reconstruction from 2 images only. Beside **Ours**, **Va09** is the only baseline that can reconstruct from 2 images. However, it does not perform well on this data. **Lee16** and **An17** are designed for video sequences, and thus need more than 3 images to perform effectively. The remaining methods can operate on three images, but their accuracy is lower than ours, especially in terms of normal accuracy. Since we can discard the normals that have a low reliability, the accuracy of our reconstruction is strengthened using multiple images. Fig. 3 further confirms the quality of our reconstructions by depicting the normals we obtain *without* any smoothing.

**Results on the Datasets used in our Previous Work.** Because the computational complexity of the global baselines grows rapidly with the number of correspondences, we evaluated all methods on the full set of correspondences and on a subset of 350 correspondences on **Paper** and **Rug**. For example, **Ch17**, **Ji17** have a cubic complexity and hence they yield a very high computation time when there are many correspondences. Their Matlab implementation crashes when using all correspondences, and using only 1000 correspondences still takes hours on a modern CPU. Similarly **Ch14** and **Va09** take almost 1 hour to reconstruct 20 images and we therefore did not evaluate them on these datasets. The **Tshirt** dataset has only 10 wide-baseline images. **Lee16** and **An17** are not designed to work on wide-baseline data, therefore we did not evaluate them on this dataset.

We report our quantitative results in Table 3, and Figure 5 depict qualitative ones. We outperform all baselines in terms of *Ed* on the **Paper** and **Rug** dataset with partial and full correspondences. On the **Tshirt** dataset, **Ch17** and **Ji17** perform better. Crucially, our performance is achieved at a much reduced computational cost by solving a set of equations in closed form, as opposed to invoking a complex solver. As a result, our approach is about 150 times faster than **Ch17** on 350 correspondences and can handle thousands whereas **Ch17** cannot. Furthermore, our approach is also 50 times faster than **Pa19**, the counterpart local approach which uses expensive polynomial solvers, because we do not have to derive a complicated formulation to obtain a unique solution for each correspondence.

Table 4 provides a detailed analysis of the run-times of all the methods on 350 and 1500 points. We assume that the input point correspondences and their derivatives are pre-computed. Therefore, the timings only encode the computation of the normals or 3D points. Our approach yields the fastest run-times, seconded by **An17**. Note, however, that **An17** has a parallel implementation and is computationally optimized. By contrast, our approach, as all the other ones, is implemented in Matlab and not optimized for speed.

The relative slowness of the other local method arises from the local normal estimators of **Pa19** and **Pa20** having to minimize the sum of squares of polynomials, which is expensive even if it has linear complexity. **Pa20** is further slowed down by having to transform polynomials into univariate expressions. **Pa21-S** and **Pa21-R** obtain analytical solutions but require a fairly expensive disambiguation. By contrast, our local normal estimator is computationally cheap as it has a closed-form solution.

**Testing validity of LP and LL.**

Table 3 compares the performance of all methods on the **Paper** and **Rug** datasets with full and partial data. While considering the **Paper** dataset partially, we have uniformly sub-sampled 350 points, that is, $\approx 25\%$ of the original data. The performance on the full and partial data are (*Ed*=4.1, *En*=9.1) and (*Ed*=3.9, *En*=8.9), respectively. The performance is quite similar, slightly better for the partial data. This is because the uniformly sampled 350 points (which are evenly spread across the sheet of paper) are sufficient for the **LP** and **LL** assumptions to hold on a smooth object such as a sheet of paper. The slight performance improvement can be attributed to the smaller impact of noise on the partial data. The main takeaway is that, for a smooth object, we do not need dense data to achieve good results. However, with only $10\%$ of the data (150 points) chosen uniformly, the performance drops to (*Ed*=8.1, *En*=16.3). This is a significant drop in performance, showing that **LP** and **LL** do not hold on such sparsely sampled data.

We repeated the experiment with $10\% - 90\%$ of the data by choosing points randomly. Since the points are chosen randomly, some regions may not be well covered, which causes **LP** and **LL** to be rather distant approximations. Table 5 shows the results. The performance degrades significantly with less than $50\%$ of the data. This is because there are more chances that the **LP** and **LL** approximations fail when we rely on sparse data chosen randomly. Therefore, to obtain a good performance using **Ours**, a well-distributed set of correspondences across the object of interest should be used. However, dense data is not required.

**Results on the NRSfM Challenge Dataset.** Fig. 6 compares the performance of **Ours** with that of other methods in terms of $Ed$, measured in mm, with **Best** being the one that does best

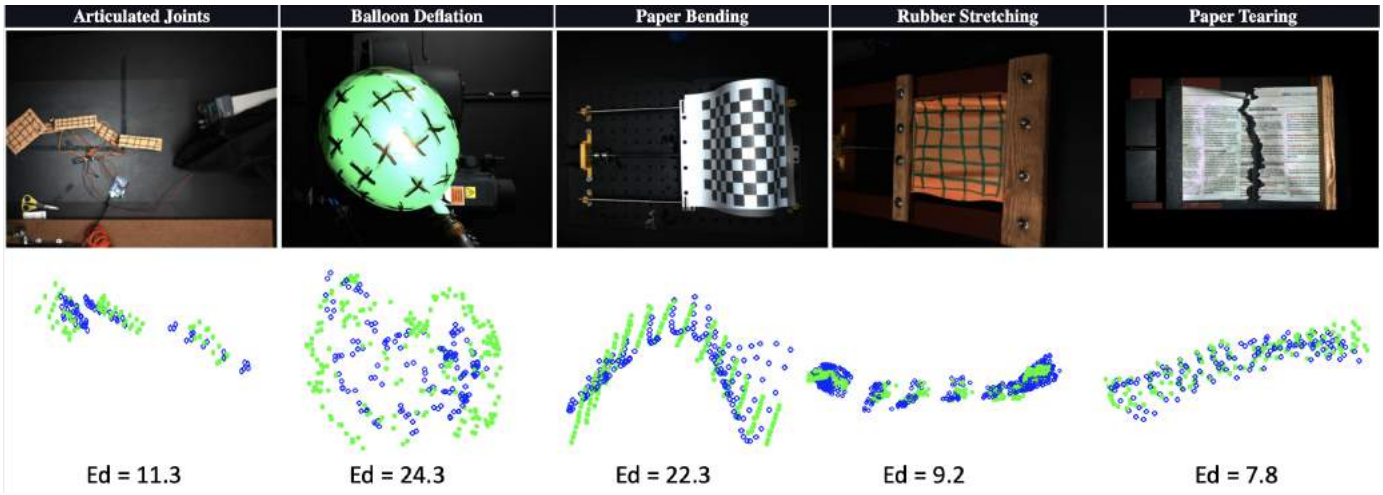


Fig. 6: NRSfM challenge dataset and some reconstructions using **Ours**. Green indicates the ground truth and blue indicates our reconstruction.

| % Data | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| Ed | 7.8 | 6.5 | 5.6 | 4.9 | 4.5 | 4.3 | 4.1 | 4.0 | 4.2 | 4.1 |
| En | 16.2 | 13.7 | 11.8 | 10.7 | 9.9 | 9.5 | 9.2 | 9.1 | 9.3 | 9.1 |

TABLE 5: Performance of **Ours** on $10\%-100\%$ correspondences chosen randomly from **Paper** dataset.

as reported in the benchmark statistics provided on the website. The local methods show a significant performance improvement compared to the other ones. **Pa19** uses second-order derivatives of the image registration $\eta$, which can be highly erroneous on this dataset. It uses an expensive polynomial solver, which cannot handle such large noise and fails on a large number of cases. **Pa21-S** and **Pa21-R** find an analytical solution to the isometric/conformal NRSfM posed in **Pa19**, which requires a non-linear refinement to obtain a unique solution; they obtain decent results on this dataset. **Pa20** solves NRSfM using diffeomorphic constraints, which uses only first-order derivatives of $\eta$, and is thus less impacted by the sparsity of the data and performs better than **Pa21-S** and **Pa21-R**. **Ours** requires second-order derivatives of the image registration, but it is equipped with a measure to compute the well-conditioning of the data. This lets us identify and discard the non-isometric/non-conformal data and reconstruct from as-isometric(or conformal)-as-possible data. As a result, **Ours** yields better results than **Pa20**. Fig. 6 shows some reconstructions obtained with our method.

**Results on the Blue Sheet Dataset and on the Datasets used by [44].** These datasets are large in terms of the number of either point correspondences or images they contain. We compare the performance of **Ours** with **An17**, which is designed for reconstructing dense objects, however, it takes several hours to reconstruct. Additionally, we report the performance of our other local methods **Pa19**, **Pa21-S** and **Pa21-R**. In this case, we report the mean 3D error to be able to compare with the performance of [44], which has demonstrated best results on these datasets. Table 7 summarizes the results. **Ours** performs better than most of the methods on these datasets. The **Actor** and **Expressions** sequences are relatively simple, with small relative motion across images. All local methods therefore perform similarly on these sequences. **An17** performs better than **Ours** on the **Actor** sequence. However, the visual performance is quite similar as the error margin is very low, to the third decimal place. Fig. 9 shows some reconstructions. Fig.s 7, 10 show the results on the **Blue Sheet**, **Paper** and **Tshirt** datasets where **Ours** performs significantly better than the compared methods.

We also evaluated our method on **Back**, **Owl** and **Heart** datasets. Fig. 8 shows the reconstructed surfaces of some images from these datasets.

## 7 Conclusion and Future Works

We have proposed an approach to NRSfM that can estimate normals from image pairs given a 2D warp and point correspondences between the two images. It does so in closed form from individual correspondences and is therefore fast. Furthermore, it can estimate if these normals are reliable given the motion from one image to the next. When they are found to be, our experiments show that they are indeed very accurate. As a result, our method performs well with various deformation types and can reconstruct large and small deformations at a low computational cost. Local methods require the first and second order derivatives of the image registration, which are computed using image warps. The computation of second order derivatives through warps is computationally expensive and can be adversely impacted by noise. Furthermore, depth is computed by integrating local normals on each surface, which is another expensive step. Our next goal will be to remove the dependency on expensive methods to compute warps and integrate normals so that a truly real-time application can be developed.

## References

[1] A. Agudo, L. Agapito, B. Calvo, and J. M. M. Montiel. Good Vibrations: A Modal Analysis Approach for Sequential Non-Rigid Structure from Motion. In *Conference on Computer Vision and Pattern Recognition*, 2014.

[2] A. Agudo and F. Moreno-Noguer. Simultaneous Pose and Non-Rigid Shape with Particle Dynamics. In *Conference on Computer Vision and Pattern Recognition*, 2015.

[3] A. Agudo, F. Moreno-Noguer, B. Calvo, and J. M. M. Montiel. Sequential Non-Rigid Structure from Motion Using Physical Priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(5):979–994, 2016.

[4] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade. Nonrigid Structure from Motion in Trajectory Space. In *Advances in Neural Information Processing Systems*, 2008.

TABLE 6: **Results on the NRSfM challenge datasets.**

NRSFM Challenge Dataset

| Method | Perspective | | | | | | | | | | Orthographic | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Articulated | | Balloon | | Paper | | Stretch | | Tearing | | Articulated | | Balloon | | Paper | | Stretch | | Tearing | |
| | (full) | (missing) | (full) | (missing) | (full) | (missing) | (full) | (missing) | (full) | (missing) | (full) | (missing) | (full) | (missing) | (full) | (missing) | (full) | (missing) | (full) | (missing) |
| **Lee16** | 105.5 | ------ | 88.4 | ------ | 65.4 | ------ | 70.3 | ------ | 59.5 | ------ | 105.3 | ------ | 54.7 | ------ | 64.2 | ------ | 69.2 | ------ | 59.1 | ------ |
| **An17** | 65.1 | 73.7 | 35.5 | 55.2 | 48.9 | 64.7 | 48.1 | 59.8 | 40.3 | 50.8 | 58.1 | 58.9 | 31.7 | 46.4 | 45.4 | 50.3 | 38.9 | 42.8 | 35.0 | 38.4 |
| **Va09** | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ |
| **Ch14** | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ |
| **Ch17** | 91.6 | 75.5 | 58.0 | 53.5 | 66.5 | 63.8 | 62.5 | 65.9 | 53.9 | 51.9 | 88.7 | 82.0 | 58.3 | 52.6 | 67.0 | 65.0 | 66.3 | 67.2 | 56.7 | 57.2 |
| **Pa19** | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ | ------ |
| **Pa20** | **21.3** | **22.0** | 27.7 | 37.8 | 31.0 | 39.0 | 17.3 | 33.5 | **15.7** | 23.8 | **18.7** | **16.7** | 28.5 | 33.6 | 31.8 | 34.0 | **17.1** | **21.1** | 19.7 | 18.8 |
| **Pa21-S** | 26.0 | 25.2 | 29.2 | 40.7 | 30.1 | 40.4 | 20.6 | 26.8 | 23.7 | 27.8 | 22.0 | 23.3 | 29.0 | 27.3 | 31.2 | 32.2 | 33.0 | 28.9 | 20.8 | 21.0 |
| **Pa21-R** | 25.1 | 25.6 | 29.4 | 41.6 | 29.0 | 40.5 | 20.6 | 26.6 | 23.8 | 28.7 | 21.8 | 24.0 | 28.7 | 27.2 | 31.5 | 32.3 | 23.1 | 28.9 | 20.7 | 20.7 |
| **Best** | 40.7 | 46.6 | 28.0 | 35.7 | 35.7 | 39.0 | 30.3 | 28.9 | 23.0 | 24.9 | 35.5 | 43.7 | 14.5 | 33.8 | 22.9 | 37.0 | 22.9 | 23.2 | **18.1** | 18.3 |
| **Ours** | 21.8 | 22.4 | **27.6** | 38.1 | **28.3** | **38.2** | 17.2 | **26.5** | 19.0 | **23.1** | 20.1 | 22.9 | **28.0** | **26.8** | **30.5** | **32.1** | 17.2 | 23.3 | 18.8 | **18.3** |

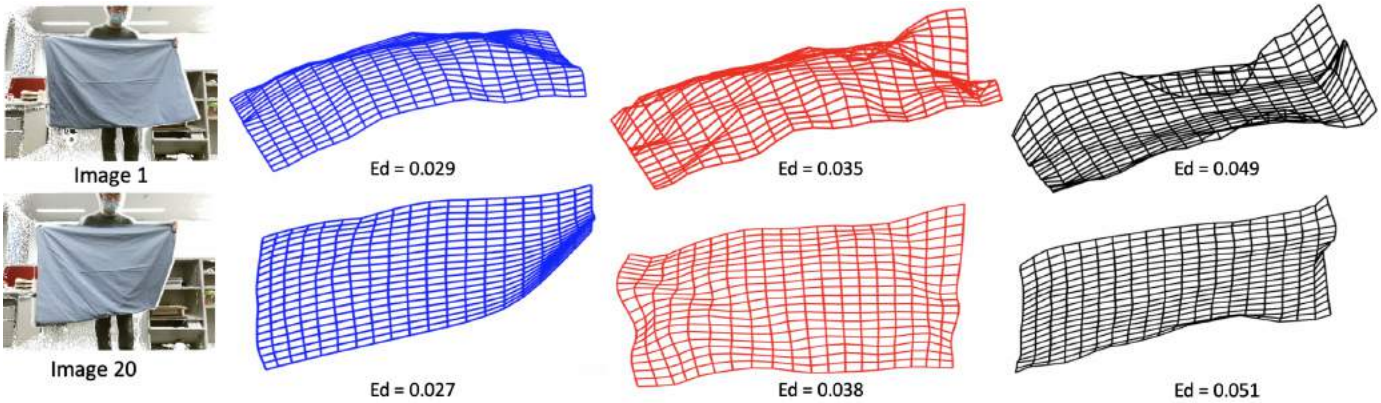


Fig. 7: **Blue Sheet dataset.** Reconstructed surfaces for two images. The predictions of **Ours** are shown in blue, of **Pa21-R** in red, and of **An17** in black. Note that our reconstructions are less noisy and match the surface 3D shape much better.

TABLE 7: **Performance on dense datasets.**

| Method | Blue sheet | Paper | Tshirt | Actor | Expressions |
|---|---|---|---|---|---|
| **An17** | 0.0558 | 0.0448 | 0.0276 | **0.0010** | 0.1352 |
| **Si20** | ------ | 0.0332 | 0.0309 | 0.0181 | 0.0260 |
| **Pa21-S** | 0.0462 | 0.0552 | 0.0402 | 0.0077 | 0.0180 |
| **Pa21-R** | 0.0463 | 0.0533 | 0.0399 | 0.0075 | 0.0180 |
| **Pa19** | 0.0479 | 0.0547 | 0.0391 | 0.0079 | 0.0180 |
| **Ours** | **0.0404** | **0.0313** | **0.0263** | 0.0072 | **0.0171** |


[5] M. D. Ansari, V. Golyanik, and D. Stricker. Scalable Dense Monocular Surface Reconstruction. In *International Conference on 3D Vision*, 2017.

[6] F. Bookstein. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[7] C. Bregler, A. Hertzmann, and H. Biermann. Recovering Non-Rigid 3D Shape from Image Streams. In *Conference on Computer Vision and Pattern Recognition*, 2000.

[8] A. Chhatkuli, D. Pizarro, and A. Bartoli. Non-Rigid Shape-From-Motion for Isometric Surfaces Using Infinitesimal Planarity. In *British Machine Vision Conference*, 2014.

[9] A. Chhatkuli, D. Pizarro, T. Collins, and A. Bartoli. Inextensible Non-Rigid Structure-From-Motion by Second-Order Cone Programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(20):2428–2441, 2017.

[10] Y. Dai, H. Li, and M. He. A Simple Prior-Free Method for Non-Rigid Structure-From-Motion Factorization. *International Journal of Computer Vision*, 107(2):101–122, 2014.

[11] A. Del Bue. A Factorization Approach to Structure from Motion with Shape Priors. In *Conference on Computer Vision and Pattern Recognition*, June 2008.

[12] A. Del Bue, X. Lladó, and L. Agapito. Non-Rigid Metric Shape and Motion Recovery from Uncalibrated Images Using Priors. In *Conference on Computer Vision and Pattern Recognition*, 2006.

[13] A. Del Bue, J. Xavier, L. Agapito, and M. Paladini. Bilinear modeling via augmented Lagrange multipliers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8):1496–1508, 2012.

[14] R. Garg, A. Roussos, and L. Agapito. Dense Variational Reconstruction of Non-Rigid Surfaces from Monocular Video. In *Conference on Computer Vision and Pattern Recognition*, 2013.

[15] V. Golyanik, T. Fetzer, and D. Stricker. Accurate 3D Reconstruction of Dynamic Scenes from Monocular Image Sequences with Severe Occlusions. In *IEEE Winter Conference on Applications of Computer Vision*, 2017.

[16] V. Golyanik, S. Shimada, K. Varanasi, and D. Stricker. Hdm-Net: Monocular Non-Rigid 3D Reconstruction with Learned Deformation Model. In *EuroVR*, 2018.

[17] P. F. Gotardo and A. M. Martinez. Computing Smooth Time Trajectories for Camera and Deformable Shape in Structure from Motion with Occlusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):2051–2065, 2011.

[18] P. F. Gotardo and A. M. Martinez. Kernel Non-Rigid Structure from Motion. In *International Conference on Computer Vision*, 2011.

[19] S. H. N. Jensen, A. Del Bue, M. E. B. Doest, and H. Aanæs. A Benchmark and Evaluation of Non-Rigid Structure from Motion. *International Journal of Computer Vision*, 129:882–899, 2021.

[20] P. Ji, H. Li, Y. Dai, and I. Reid. Maximizing Rigidity Revisited: A Convex Programming Approach for Generic 3D Shape Reconstruction from Multiple Perspective Views. In *International Conference on Computer Vision*, 2017.

[21] A. Kock. *Synthetic Geometry of Manifolds.* Cambridge University Press, 2010.

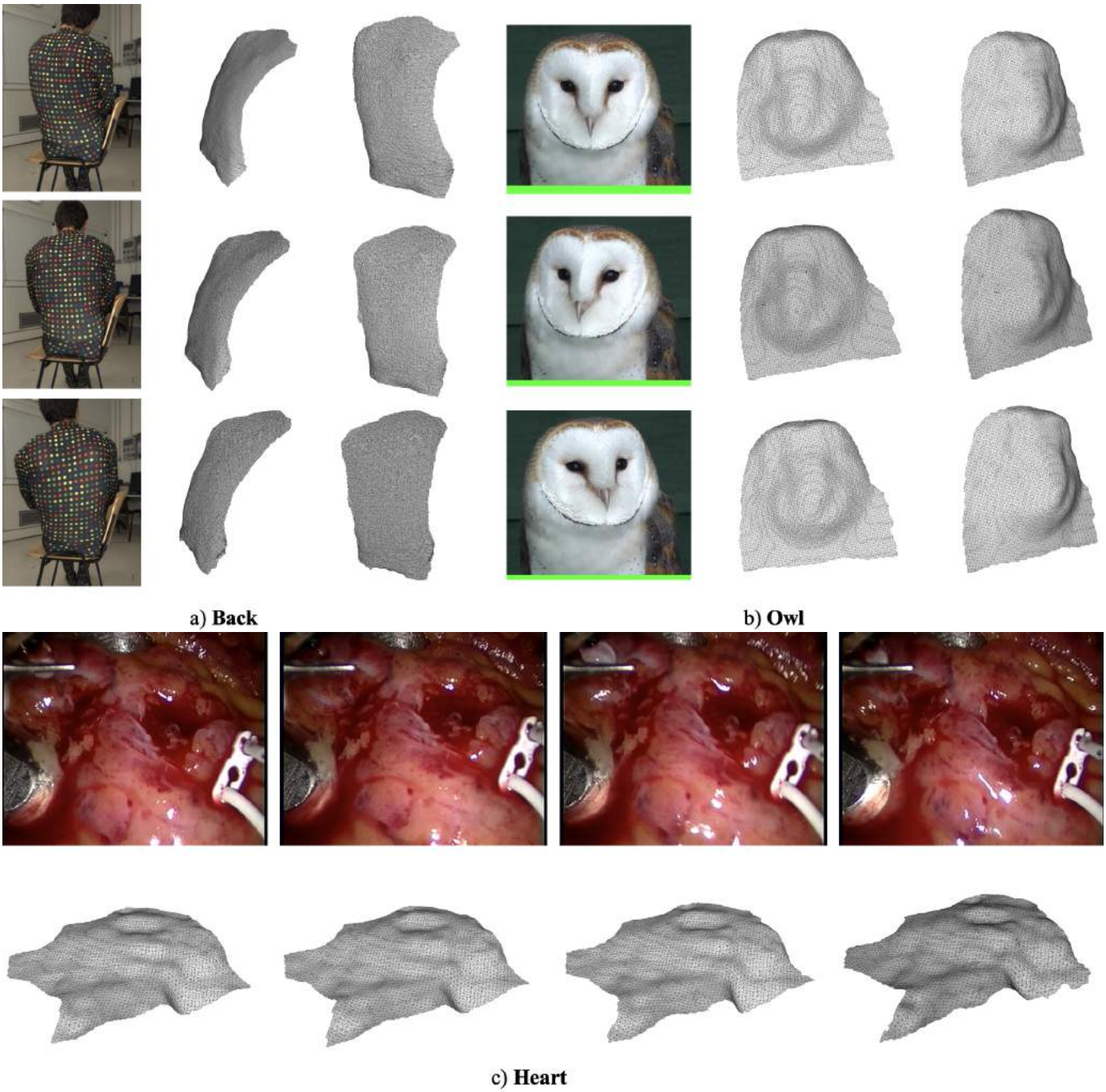


Fig. 8: **Back**, **Owl** and **Heart** datasets. Reconstructed surfaces for few images usig our method.

[22] C. Kong and S. Lucey. Deep Non-Rigid Structure from Motion. In *International Conference on Computer Vision*, 2019.
[23] S. Kumar. Jumping Manifolds: Geometry Aware Dense Non-Rigid Structure from Motion. In *Conference on Computer Vision and Pattern Recognition*, 2019.
[24] S. Kumar, A. Cherian, Y. Dai, and H. Li. Scalable Dense Non-Rigid Structure-From-Motion: A Grassmannian Perspective. In *Conference on Computer Vision and Pattern Recognition*, 2018.
[25] S. Kumar, Y. Dai, and H. Li. Superpixel Soup: Monocular Dense 3D Reconstruction of a Complex Dynamic Scene. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1705–1717, 2019.
[26] J. Lamarca, S. Parashar, A. Bartoli, and J. M. M. Montiel. DefSLAM: Tracking and Mapping of Deforming Scenes from Monocular Sequences. *IEEE Transactions on Robotics*, 37(1):291–298, 2020.
[27] J. M. Lee. *Riemannian Manifolds: An Introduction to Curvature*. Springer, 1997.
[28] M. Lee, J. Cho, and S. Oh. Consensus of Non-Rigid Reconstructions. In *Conference on Computer Vision and Pattern Recognition*, 2016.
[29] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 20(2):91–110, 2004.
[30] E. Malis and M. Vargas. Deeper Understanding of the Homography Decomposition for Vision-Based Control. Technical report, INRIA, 2007.
[31] D. Novotny, N. Ravi, B. Graham, N. Neverova, and A. Vedaldi. C3DPO: Canonical 3D Pose Networks for Non-Rigid Structure from Motion. In *International Conference on Computer Vision*, 2019.
[32] M. Ovsjanikov, M. Ben-Chen, J. Solomon, A. Butscher, and L. Guibas. Functional Maps: A Flexible Representation of Maps Between Shapes. *ACM Transactions on Graphics*, 31(4):1–11, 2012.
[33] M. Paladini, A. Del Bue, M. Dodig, J. Xavier, and L. Agapito. Factorization for Non-Rigid and Articulated Structure Using Metric Projections. In *Conference on Computer Vision and Pattern Recognition*, 2009.
[34] M. Paladini, A. Del Bue, M. Stosic, M. Dodig, J. Xavier, and L. Agapito. Optimal metric projections for deformable and articulated structure-

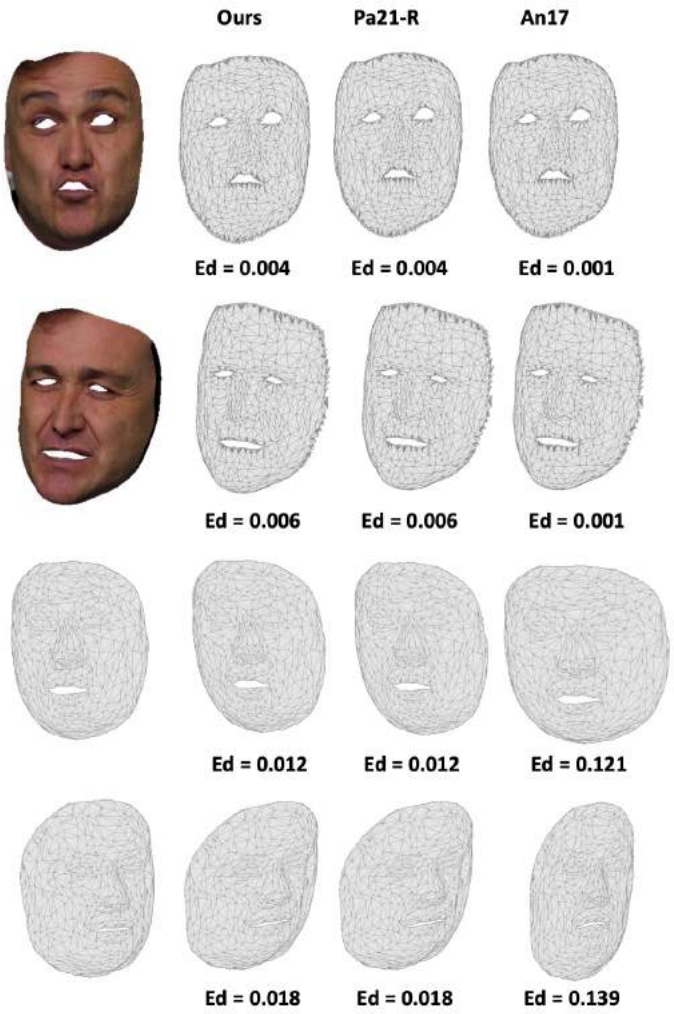


Fig. 9: **Actor and Expressions datasets.** Reconstructed surfaces for two images. The predictions of **Ours**, **Pa21-R** and of **An17** are quite similar.

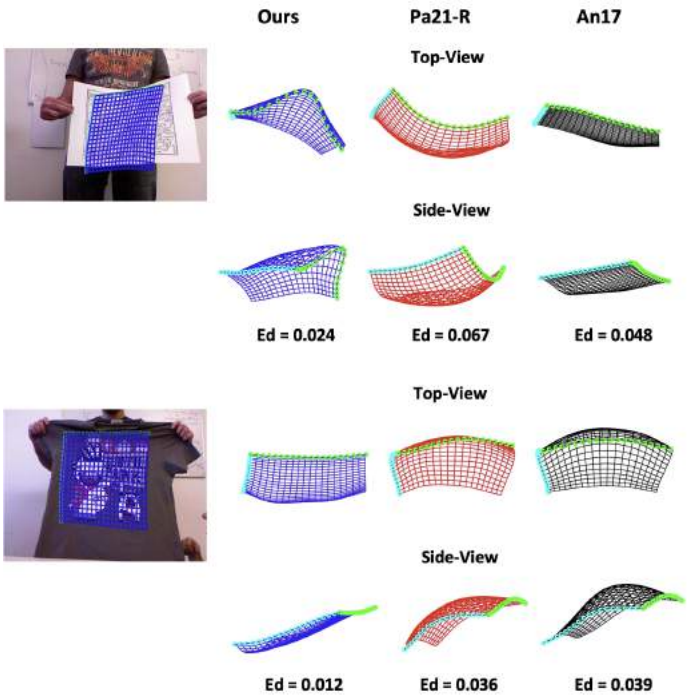


Fig. 10: **Paper and Tshirt datasets.** The predictions of **Ours** are shown in blue, of **Pa21-R** in red, and of **An17** in black. Note that our reconstructions are less noisy and match the surface 3D shape much better.

from-motion. *International Journal of Computer Vision*, 96(1):252–276, 2012.

[35] S. Parashar and A. Bartoli. 3DVFX: 3D Video Editing Using Non-Rigid Structure-From-Motion. In *Eurographics*, 2019.

[36] S. Parashar, A. Bartoli, and D. Pizarro. Robust Isometric Non-Rigid Structure-From-Motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6409–6423, 2021.

[37] S. Parashar, D. Pizarro, and A. Bartoli. Isometric Non-Rigid Shape-From-Motion with Riemannian Geometry Solved in Linear Time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2442–2454, 2017.

[38] S. Parashar, D. Pizarro, and A. Bartoli. Local Deformable 3D Reconstruction with Cartan's Connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(12):3011–3026, 2019.

[39] S. Parashar, M. Salzmann, and P. Fua. Local Non-Rigid Structure-From-Motion from Diffeomorphic Mappings. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[40] A. Pumarola, A. Agudo, L. Porzi, A. Sanfeliu, V. Lepetit, and F. Moreno-Noguer. Geometry-Aware Network for Non-Rigid Shape Prediction from a Single View. In *Conference on Computer Vision and Pattern Recognition*, June 2018.

[41] C. Russell, J. Fayad, and L. Agapito. Energy Based Multiple Model Fitting for Non-Rigid Structure from Motion. In *Conference on Computer Vision and Pattern Recognition*, 2011.

[42] C. Russell, R. Yu, and L. Agapito. Video Pop-Up: Monocular 3D Reconstruction of Dynamic Scenes. In *European Conference on Computer Vision*, 2014.

[43] M. Salzmann, R. Hartley, and P. Fua. Convex Optimization for Deformable Surface 3D Tracking. In *International Conference on Computer Vision*, October 2007.

[44] V. Sidhu, E. Tretschk, V. Golyanik, A. Agudo, and C. Theobalt. Neural Dense Non-Rigid Structure from Motion with Latent Space Constraints. In *European Conference on Computer Vision*, 2020.

[45] D. Sun, X. Yang, M. Liu, and J. Kautz. Pwc-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In *Conference on Computer Vision and Pattern Recognition*, 2018.

[46] N. Sundaram, T. Brox, and K. Keutzer. Dense Point Trajectories by Gpu-Accelerated Large Displacement Optical Flow. In *European Conference on Computer Vision*, 2010.

[47] J. Taylor, A. D. Jepson, and K. N. Kutulakos. Non-Rigid Structure from Locally-Rigid Motion. In *Conference on Computer Vision and Pattern Recognition*, June 2010.

[48] L. Torresani, A. Hertzmann, and C. Bregler. Nonrigid Structure-From-Motion: Estimating Shape and Motion with Hierarchical Priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):878–892, 2008.

[49] L. Torresani, D. B. Yang, E. Alexander, and C. Bregler. Tracking and Modeling Non-Rigid Objects with Rank Constraints. In *Conference on Computer Vision and Pattern Recognition*, pages 493–500, 2001.

[50] A. Varol, M. Salzmann, E. Tola, and P. Fua. Template-Free Monocular Reconstruction of Deformable Surfaces. In *International Conference on Computer Vision*, September 2009.

[51] S. Vicente and L. Agapito. Soft Inextensibility Constraints for Template-Free Non-Rigid Reconstruction. In *European Conference on Computer Vision*, 2012.

[52] Y. Zhu, D. Huang, F. De La Torre, and S. Lucey. Complex Non-Rigid Motion 3D Reconstruction by Union of Subspaces. In *Conference on Computer Vision and Pattern Recognition*, 2014.