Abstract

We propose a new formulation to the non-rigid structure-from-motion problem that only requires the deforming surface to meaning that its differential structure is preserved. This is a much weaker assumption than the traditional ones of isometry or conformality. We show that it is nevertheless sufficient to establish local correspondences between the surface in two different images and therefore to perform point-wise reconstruction using only up to first-order derivatives. We formulate differential constraints and solve them algebraically using the theory of resultants. We will demonstrate that our approach is more widely applicable, more stable in noisy and sparse imaging conditions and much faster than earlier ones, while delivering similar accuracy.

1. Introduction

Reconstructing the 3D shape of deformable objects from monocular images, known as Non-Rigid Structure-from-Motion (NRSfM), has applications in domains ranging from entertainment [23] to medicine [20]. It was introduced in [4] by expressing shapes in terms of a low-rank shape-basis. Many variants of this idea have since been proposed with a view to improve reconstruction stability [5, 2, 32, 11, 21, 14]. Over the last decade, physically-inspired NRSfM models [33, 34, 30, 8, 9, 18, 24, 25] have emerged as an attractive alternative. Such models exploit local surface properties to perform 3D reconstruction. They can handle large deformations and generally outperform techniques relying on low-rank priors. Unfortunately, most methods in both categories become prohibitively expensive as the number of images increases, because of their non-linear complexity, and cannot handle missing data. This makes them impractical for real-world scenarios.

The method of [24, 25] bucks this trend. By expressing isometry or conformality constraints in terms of differential properties, local reconstruction constraints can be established between the deforming surface as seen in two different images. Thus, the surface 3D shape in any frame can be obtained by pairing that frame with all others and the complexity only grows linearly with the number of images. Furthermore, missing data for example due to occlusions, can be easily handled by using a parametric image registration warp. While effective in theory, this approach suffers from two main drawbacks. First, it requires the second-order derivatives of the image registration warps, which are usually noisy, and sometimes even downright wrong when given only sparse correspondences to compute them. In the first case, an expensive warp refinement [26] must be performed and in the second the approach simply becomes impractical. Second, a deformation model must be chosen a priori, which precludes using this method for surfaces of unknown properties.

In this paper, we introduce a framework that overcomes these drawbacks. To this end, we leverage the assumption that the deforming surface is locally diffeomorphic, that is, that the deformation preserves the local differential structure of the surface, which is a much more generic model than isometry or conformality and encompasses both as well as equiareality. We will show that it suffices to establish local reconstruction constraints between pairs of surfaces without requiring second-order derivatives or a priori knowledge about the surface properties. This makes our approach immune to the difficulties described above. Furthermore, if knowledge about the surface properties happens to be available, the corresponding metric-preserving constraints can be incorporated.

We will show that, when the deformations are equiareal instead of conformal and the correspondences sparse, our approach delivers good results whereas the one of [24, 25] cannot be used. Furthermore, in the conformal case with dense correspondences, our approach delivers a similar accuracy as [24, 25] and it is 10 times faster than these methods. In addition, we require only first order derivatives which is at least 20 times faster than computing second order derivatives for [24, 25]. We also compare with some of the best performing methods in state of the art and show that we outperform most of them in terms of both accuracy...
and computation time.

2. Related Work

NRSfM methods can be grouped into three broad classes depending on how deformations are modeled.

Low-Dimensional Deformations. These methods [4, 6, 1, 14, 10, 21] produce a global 3D shape by jointly reconstructing the points in all frames. This is an ill-posed problem that is solved by constraining the deformations to lie in a low-dimensional space. This makes these methods ill-suited to model complex deformations and to handle missing correspondences. Furthermore, it usually requires the shape-space dimension to be decided a priori.

Global Physical Deformations. These methods [30, 34, 33, 9, 18] aim to preserve physical properties of surfaces. Most of them assume deformations to be isometric (distance-preserving) and but they model an approximation of isometry such as inextensibility [9, 18], piece-wise inextensibility [30, 34] or piece-wise rigidity [33]. These methods usually find a globally optimal solution by solving for constraints over all the points altogether. They usually require a computationally expensive optimization which makes them impractical for handling large number of images.

Local Physical Deformations. Fewer methods only characterize local deformations. These methods formulate and solve isometric constraints locally. [8] formulates isometry as local rigidity and [24, 25] formulates the exact constraint for isometry using differential properties of surfaces. [24, 25] showed that their complexity scales linearly with the number of images, unlike that of the methods discussed above, which grows much faster. This is because they use differential properties that are preserved under isometry up to a change of variables. Due to this, they show that adding images to system does not increase the number of variables. In practice, the methods of [24, 25] have been shown to yield faster and more accurate reconstructions than existing methods. However, as discussed in the introduction, they rely on second-order derivatives, which are computationally expensive to compute and therefore, impractical. [24, 25] assume that the second-order derivatives are provided with the input. We discuss the problems with obtaining the second-order derivatives in our experiments. Furthermore, they still impose strong constraints on what the surface deformations may be. In this paper, we seek the minimalistic deformations constraints. Given that the surface deformations in nature are at least locally diffeomorphic, we show that it provides sufficient constraints to perform reconstruction. Thus we show that any deformation stronger than local diffeomorphism, (isometry, conformality and equiareality), is thus solvable. In addition, we get rid off the second-order derivatives and thus we obtain a highly reliable, fast and practical solution for NRSfM, just with the assumption of local diffeomorphism.

3. Method Outline

Fig 1 depicts our setup when using only two images \( I \) and \( \overline{I} \) acquired by a calibrated camera. In each one, we denote the deforming surface as \( S \) and \( \overline{S} \), respectively, and model it in terms of functions \( \phi, \overline{\phi} : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \) that associate a surface point to an image 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. In practice, it can be computed using standard image matching techniques such as optical flow [29, 28] or SIFT [22].

These functions can be composed to create a mapping \( \psi : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \), which we assume to be locally diffeomorphic, from 3D surface points seen in the two images. We use a parametric representation of \( \eta \) and \( \phi \) using a splines [3], which allows us to easily and accurately obtain first-order derivatives of these functions. Any other approach to obtain derivatives, such as finite difference methods, can be used alternatively.

At the heart of our approach is the fact that, under the assumption that the two surfaces are locally diffeomorphic, some differential properties of corresponding 3D points should match. These properties can be expressed in terms of connections. They are generic properties of a differentiable surface that express intrinsic relationship between a point on the surface and its local neighborhood [7, 16]. In particular, the well known first and second fundamental forms on surfaces can be derived from them. Crucially, they are preserved under diffeomorphism [25], which we prove formally. Furthermore, assuming the surfaces to be locally planar, we show that we can use connections computed using only first derivatives. Thus, we can express...
depth and its derivatives at \( S \) in terms of the same quantities at \( S \). As a consequence, the 3D coordinates of corresponding points on the surface are strongly constrained, thanks to multi-view constraints and the 3D reconstruction problem becomes sufficiently constrained. This approach has several strengths:

- Because all the constraints can be expressed in terms of first derivatives of \( \phi, \tilde{\phi} \) and \( \eta \), which, unlike the second derivatives that are required by the formulation of [25], can be estimated even if the points for which we have correspondences are relatively sparse.
- If we happen to know that the deformation is isometric, conformal, or equiareal, we can easily incorporate these additional constraints into our framework.
- If we take \( S \) to be the reference image in which we wish to recover the shape, we can write the constraints for as many surfaces \( \tilde{S} \) as we want to increase robustness and the cost only grows linearly with the number of such images. The shape at \( \tilde{S} \) can then be expressed in terms of the recovered shape at \( S \).

We now turn to defining connections formally and then show how we use them.

### 4. Connections and Local Diffeomorphisms

In this section, we formalize connections and show their invariance under diffeomorphic deformations. We will use these concepts in Section 5 to implement our Diff-NRSfM framework. We use the notation introduced at the beginning of Section 3 and depicted by Fig. 1.

**Moving Frames.** Given the projection of a surface point \( x = [u, v]^{\top} \) in \( I \) and the corresponding 3D point \( X \) on \( S \), we write

\[
X = \phi(x),
\]

\[
E(\phi) = (e_1 = \partial \phi / \partial u, e_2 = \partial \phi / \partial v, e_3 = e_1 \times e_2). \tag{1}
\]

\( E(\phi) \) is a moving reference frame for \( S \), and we define \( E(\tilde{\phi}) \) similarly for \( \tilde{S} \).

**Connections.** We can now define the *connections* that encode differential surface properties that are invariant under diffeomorphic deformations and are at the heart of our approach. Assuming \( S \) to be locally planar, we can rewrite \( \phi(u, v) \) as \( \beta(u, v)[u, v, 1]^{\top} \), where \( \beta \) is a linear function representing depth, within a small neighborhood around the projection of any surface point \( x \). Injecting this definition in Eq. 1 yields

\[
e_1 = \beta(u, v)[1 + ux_1, vx_1, x_1]^{\top},
\]

\[
e_2 = \beta(u, v)[ux_2, 1 + vx_2, x_2]^{\top}, \tag{2}
\]

where \( x_1 = \frac{1}{\beta(u, v)} \frac{\partial \beta}{\partial u} \) and \( x_2 = \frac{1}{\beta(u, v)} \frac{\partial \beta}{\partial v} \). The connections \( \Gamma^i_{jk} \) are then taken to be the solutions of the linear system

\[
\frac{\partial e_j}{\partial u} = \Gamma^1_{j1} e_1 + \Gamma^2_{j2} e_2 + \Gamma^3_{j3} e_3, \quad j = [1, 2, 3]
\]

\[
\frac{\partial e_j}{\partial v} = \Gamma^1_{j2} e_1 + \Gamma^2_{j3} e_2 + \Gamma^3_{j3} e_3. \tag{3}
\]

Because \( \beta \) is assumed to be linear, its partial derivatives that appear in the definition of \( x_1 \) and \( x_2 \) in Eq. 2 are constant and its second order derivatives are 0. Thus, solving the linear system of Eq. 3 yields

\[
\begin{pmatrix}
\Gamma^1_{11} & \Gamma^1_{12} & \Gamma^1_{13} \\
\Gamma^2_{11} & \Gamma^2_{12} & \Gamma^2_{13} \\
\Gamma^3_{11} & \Gamma^3_{12} & \Gamma^3_{13}
\end{pmatrix} = \begin{pmatrix}
\beta \\
\frac{\partial \phi}{\partial u} \\
\frac{\partial \phi}{\partial v}
\end{pmatrix} \tag{4}
\]

where

\[
D = \beta^4(x_1^2 + x_2^2 + (1 + ux_1 + vx_2)^2),
\]

\[
T_1 = x_1 + vx_1 x_2 - 2ux_2, T_2 = x_2 + ux_1 x_2 - vx_1,
\]

\[
T_3 = v + (1 + u^2)x_2 + u v x_1, T_4 = u + (1 + u^2)x_1 + u v x_2,
\]

\[
T_5 = 1 + (u^2)x_1^2 - (1 + u^2)x_2^2 + 2ux_1,
\]

\[
T_6 = 1 + (1 + u^2)x_2^2 - (1 + u^2)x_1^2 + 2ux_2,
\]

\[
T_7 = 1 + (1 + u^2)x_1^2 + 2ux_1 + vx_2 + u v x_1 x_2,
\]

\[
T_8 = 1 + (1 + u^2)x_2^2 + ux_1 + 2ux_1 + 2ux_2 x_2.
\]

From the above equation we can verify that \( \Gamma^i_{jk} = \Gamma^i_{kj} \) always holds and that \( \Gamma^3_{3k} = \Gamma^1_{1k} + \Gamma^2_{2k} \). This leaves us with a set of 13 distinct \( \Gamma^i_{jk}(\phi) \) in terms of image observations, depths, and the depth first-order derivatives. Their formulation may seem complex but this is the price to pay to achieve invariance to diffeomorphic deformations, which we prove below.

**Invariance under Local Diffeomorphism.** In the previous paragraph, we defined the connections \( \Gamma^i_{jk}(\phi) \). We can similarly define the connections \( \Gamma^i_{jk}(\tilde{\phi}) \), which we will denote as \( \Gamma^i_{jk} \), for \( \tilde{S} \). We now discuss their invariance to diffeomorphic deformations, that is,

\[
\Gamma^i_{jk}(\phi \circ \eta) = \Gamma^i_{jk}(\tilde{\phi}). \tag{5}
\]

As can be seen in Fig. 1, \( \tilde{\phi} = \psi \circ \phi \circ \eta \). We show in the supplementary material that it follows that

\[
\overline{E}(\tilde{\phi}) = \text{diag}(\lambda_1, \lambda_2, \lambda_3) R E(\phi) \text{diag}(J_\eta, |J_\eta|), \tag{6}
\]
where $J_{\eta} = \left( \frac{\partial u}{\partial \eta}, \frac{\partial v}{\partial \eta} \right)$ is the Jacobian of $\eta$, $\lambda_i$ are scalars and $R$ is a rotation matrix. As we also show in the supplementary material, injecting Eq. 6 into the definition of the $\Gamma_{jk}$ yields

$$
\begin{align*}
\Gamma_{11} \equiv \frac{x_1^2 \beta^3}{D} = \frac{\beta^3 \lambda_2}{|J_{\eta}|D}, \\
\Gamma_{12} \equiv \frac{x_1 x_2 \beta^3}{D} = \frac{\beta^3 \lambda_2}{|J_{\eta}|D},
\end{align*}
$$

(8)

$$
\begin{align*}
\Gamma_{11} \equiv \frac{x_1^2 \beta^3}{D} = \frac{\beta^3 \lambda_2}{|J_{\eta}|D}, \\
\Gamma_{12} \equiv \frac{x_1 x_2 \beta^3}{D} = \frac{\beta^3 \lambda_2}{|J_{\eta}|D},
\end{align*}
$$

(9)

where $t_1 = \frac{\partial u}{\partial x_1} + \frac{\partial v}{\partial x_2}$, and $t_2 = \frac{\partial u}{\partial x_1} + \frac{\partial v}{\partial x_2}$.

$$
\begin{align*}
t_3 &= \left( t_2 (vx_1 - ux_2) + \frac{\partial u}{\partial x_1} - \frac{\partial v}{\partial x_1} \right), \\
t_4 &= \left( t_1 (ux_1 - vx_2) - \frac{\partial u}{\partial x_1} + \frac{\partial v}{\partial x_1} \right).
\end{align*}
$$

The above relation shows that connections are preserved up to a change of variable. In other words, we can compute the connections of $S$ from those of $\tilde{S}$ using $\eta$. In the next section, we exploit this to perform NRSfM.

5. Surface Reconstruction Under Local Diffeomorphism

In this section, we use connections and their preservation relations (7) to derive reconstruction equations. We first express the depth and its derivatives at $\tilde{S}$ in terms of the ones at $S$ and $\eta$. We show that this helps in constraining the complexity of the problem. Then we derive constraints to perform reconstruction from a local diffeomorphism and other metric-preserving mappings.

5.1. Relating Depths

(7) expresses $\Gamma_j(\phi)$ in terms of $\Gamma_j(\eta)$ and the first- and second-order derivatives of $\eta$. The expanded expressions are shown in the supplementary material. However, not all of the $\Gamma_j$ depend on the second-order derivatives. In particular, some of the non-diagonal $\Gamma_j$ are expressed only in terms of $\Gamma_j$ and of the first-order derivatives of $\eta$. By considering only these and equating their definition from (4) with that from (7), we can write

$$
\Gamma_{11} \equiv \frac{x_1^2 \beta^3}{D} = \frac{\beta^3 \lambda_2}{|J_{\eta}|D}, \\
\Gamma_{12} \equiv \frac{x_1 x_2 \beta^3}{D} = \frac{\beta^3 \lambda_2}{|J_{\eta}|D},
$$

(10)

where $t_1 = \frac{\partial u}{\partial x_1} + \frac{\partial v}{\partial x_2}$, and $t_2 = \frac{\partial u}{\partial x_1} + \frac{\partial v}{\partial x_2}$.

Computing $(\tilde{x}_1, \tilde{x}_2, \tilde{x}_3)$ from the first three equations in (8), and substituting the results in (9) yields

$$
\begin{align*}
\Gamma_{11} \equiv \beta^2 (t_1 + (m_1 - m_2) \tilde{x}_2) = \beta^2 |\eta| t_3, \\
\Gamma_{12} \equiv \beta^2 (t_2 + (m_2 - m_1) \tilde{x}_1) = \beta^2 |\eta| t_4, \\
\Gamma_{31} \equiv \beta^2 (t_1 + t_2 (\tilde{x}_3 - \tilde{x}_1)) = \beta^2 |\eta| t_3, \\
\Gamma_{32} \equiv \beta^2 (t_2 + t_1 (\tilde{x}_3 - \tilde{x}_1)) = \beta^2 |\eta| t_4.
\end{align*}
$$

(11)

Multiplying $\Gamma_{12}$ and $\Gamma_{32}$ with $\tilde{t}_1$ and $\tilde{t}_2$, respectively, and adding the results yields

$$
\beta^2 (\tilde{t}_1 + \tilde{t}_2) = \beta^2 |\eta| \tilde{x}_1 + \tilde{x}_2).
$$

(12)

In (11) and (12), we have obtained the unknowns $\Gamma_{12}(\tilde{x}_1, \tilde{x}_2)$ on $\tilde{S}$, in terms of $(x_1, x_2)$ on $S$ and a known $\eta$. Therefore, the unknown quantities, depth and its derivatives, for any additional surface can be expressed in terms of the unknown quantities of $S$. Due to this property, adding additional views yields additional constraints without increasing the number of unknowns in the system, which limits the complexity of the system.

5.2. Reconstruction Equations

We first derive the reconstruction equations under the assumption of local diffeomorphism. Then, we show how to
add the explicit metric-preserving constraints of isometry, conformality and equiareality.

**Diffeomorphic NRSfM.** In the last section, we used few of the connections to derive \((\tilde{x}, \tilde{x_1}, \tilde{x_2})\) in terms of \((x_1, x_2)\). These relations were expressed with only first-order derivatives of \(\eta\). The remaining connections are sufficient to obtain reconstruction constraints but they all contain second-order derivatives of \(\eta\) and therefore, we cannot use them directly. According to [26], the image of two given planes should satisfy 2D Schwarzschild equations given by

\[
\frac{\partial u}{\partial \tau} \frac{\partial^2 v}{\partial \tau^2} - \frac{\partial v}{\partial \tau} \frac{\partial^2 u}{\partial \tau^2} = 0, \quad \frac{\partial v}{\partial \tau} \frac{\partial^2 v}{\partial \tau^2} - \frac{\partial v}{\partial \tau} \frac{\partial^2 v}{\partial \tau^2} = 0.
\]  

(13)

The above two expressions appear in \((\Gamma^1_{22}, \Gamma^2_{11})\) and setting them to zero cancels out second-order derivatives. Since we assume that surfaces are locally planar, we can use (13) to write

\[
\Gamma^2_{11} = \frac{\partial^2 T_2}{\partial \tau^2} = \frac{\partial^4 t_2}{\partial \tau^2} \frac{\partial T_3}{\partial \tau} - \frac{\partial^4 T_2}{\partial \tau^2} \frac{\partial T_3}{\partial \tau},
\]

(14)

Computing \((\tilde{x}_1^2, \tilde{x}_1, \tilde{x}_2, \tilde{x}_2^2)\) from (8), as before, and substituting the resulting expressions in (14) yields

\[
\beta \tilde{T}_3 = \beta \left( \frac{\partial u}{\partial \tau} T_3 - \frac{\partial v}{\partial \tau} T_4 \right), \quad \beta \tilde{T}_4 = -\beta \left( \frac{\partial u}{\partial \tau} T_3 - \frac{\partial v}{\partial \tau} T_4 \right).
\]

(15)

Substituting \((\tilde{x}_1, \tilde{x}_2)\) from (12) to the above equation gives

\[
\beta (\tilde{x}_1 + (1 + \tilde{\tau}^2) t_2 + \tilde{\tau} \tilde{t}_1) N_1 = \beta N_2 \left( \frac{\partial u}{\partial \tau} T_3 - \frac{\partial v}{\partial \tau} T_4 \right),
\]

(16)

\[
\beta (\tilde{x}_2 + (1 + \tilde{\tau}^2) t_1 + \tilde{\tau} \tilde{t}_2) N_1 = -\beta N_2 \left( \frac{\partial u}{\partial \tau} T_3 - \frac{\partial v}{\partial \tau} T_4 \right).
\]

Squaring (16) and substituting \((\tilde{\tau}^2)^2\) from (11) gives

\[
|J_\eta|^2 (x_1^2 + x_2^2) (\tilde{x}_1 N_2 + (1 + \tilde{\tau}^2) t_2 + \tilde{\tau} \tilde{t}_1) N_1 = N_2 (t_1^2 + t_2^2) \left( \frac{\partial u}{\partial \tau} T_3 - \frac{\partial v}{\partial \tau} T_4 \right),
\]

\[
|J_\eta|^2 (x_1^2 + x_2^2) (\tilde{x}_2 N_2 + (1 + \tilde{\tau}^2) t_1 + \tilde{\tau} \tilde{t}_2) N_1 = -N_2 (t_1^2 + t_2^2) \left( \frac{\partial u}{\partial \tau} T_3 - \frac{\partial v}{\partial \tau} T_4 \right).
\]

(17)

Thus we obtain two polynomials in two variables. With the solution to \((x_1, x_2)\), we can write the normals at \(S\) using (2). We can then obtain \((\tilde{x}_1, \tilde{x}_2)\) at \(S\) using \((x_1, x_2)\) in (12). Thus, we obtain normal on all the surfaces. We can obtain an up-to-scale depth by integrating the normals.

**Solving polynomial equations.** Our strategy for solving (17) consists of using resultants [12] to convert these equations to univariate polynomials, which can then be easily solved. A resultant is defined as an expression written in terms of the coefficients of two polynomials. If the polynomials have a common root, their resultant evaluates to zero. We write the equations (17) as \(A(x_1, x_2) = 0\) and \(B(x_1, x_2) = 0\). Their resultant with respect to \(x_1\) is given by \(R(x_2)\). Since these equations must bear a common root, we get \(R(x_2) = 0\). \(R(x_2)\) is a univariate equation of degree 10. We show the structure of \(R(x_2)\) in the supplementary equation. We substitute \(x_2\) into \(A(x_1, x_2)\) and \(B(x_1, x_2)\) and solve for \(x_1\).

**Metric-preserving from local diffeomorphism.** Under a local diffeomorphism, the moving frames \(\tilde{\mathcal{E}}\) and \(\mathcal{E}\) are related by (6). Since all metric-preserving deformations (including isometry, conformality and equiareality), fall under the category of locally diffeomorphic mappings, we write

\[
\tilde{\mathcal{E}}^T \mathcal{E} = \text{diag}(\lambda_1^2, \lambda_2^2, \lambda_3^2) \text{diag}(\mathcal{J}_\eta, |\mathcal{J}_\eta|)^T \mathcal{E}^T \text{diag}(\mathcal{J}_\eta, |\mathcal{J}_\eta|),
\]

where,

Isometry \(\equiv \lambda_1 = \lambda_2 = \lambda_3 = 1\),

Conformity \(\equiv \lambda_1 = \lambda_2 = \lambda_3 = \lambda\),

Equiareality \(\equiv \lambda_1 = \lambda_2 = \lambda_3 = 1\).

(18)

Using (2) in the above equation lets us write

\[
\tilde{\mathcal{E}}^2 \text{diag}(\mathcal{G}, \tilde{\mathcal{G}}) = \mathcal{E}^2 \text{diag}(\lambda_1^2, \lambda_2^2, \lambda_3^2) \text{diag}(\mathcal{P}, \mathcal{J}_\eta) |\mathcal{J}_\eta|,\]

with \(\mathcal{P} = \mathcal{J}_\eta^2 \mathcal{G} \mathcal{J}_\eta\)

\[
\mathcal{G} = \left( \frac{\epsilon x_1^4}{x_1^2 + 1} + 2u x_1 \frac{\epsilon x_1 x_2 + u x_2 + v x_1}{x_1^2 + 1} \right),
\]

\[
\tilde{\mathcal{G}} = \left( \frac{\epsilon \tilde{x}_1^4}{\tilde{x}_1^2 + 1} + 2\tilde{u} \tilde{x}_1 \frac{\epsilon \tilde{x}_1 \tilde{x}_2 + \tilde{u} \tilde{x}_2 + \tilde{v} \tilde{x}_1}{\tilde{x}_1^2 + 1} \right),
\]

\[
\epsilon = 1 + u^2 + v^2, \quad \tilde{\tau} = 1 + \tilde{u}^2 + \tilde{v}^2.
\]

(19)

Substituting \((\tilde{x}_1, \tilde{x}_2)\) from (12) in the above equation gives

\[
\sqrt{\tilde{\mathcal{G}}} \text{diag}(\tilde{\mathcal{G}}, \tilde{\mathcal{G}}) = \tilde{\mathcal{E}}^2 \text{diag}(\lambda_1^2, \lambda_2^2, \lambda_3^2) \text{diag}(\mathcal{P}, \mathcal{J}_\eta) |\mathcal{J}_\eta|,\]

with \(\tilde{\mathcal{G}}|1,1| = \left( \frac{1}{N_2} \right)^2 \left( \epsilon \tilde{x}_1^4 N_1^2 + \epsilon \tilde{x}_2^4 N_1^2 + 2\epsilon \tilde{u}_1 N_1 N_2 \right)

\[
\tilde{\mathcal{G}}|1,2| = \left( \frac{1}{N_2} \right)^2 \left( \epsilon \tilde{x}_1 \tilde{x}_2 N_1^2 + \epsilon \tilde{u}_2 N_1 N_2 + \epsilon \tilde{v}_1 N_1 N_2 \right)
\]

\[
\tilde{\mathcal{G}}|2,2| = \left( \frac{1}{N_2} \right)^2 \left( \epsilon \tilde{x}_1 \tilde{x}_2 N_1^2 + \epsilon \tilde{u}_2 N_1 N_2 + \epsilon \tilde{v}_1 N_1 N_2 \right).
\]

(20)

Our Diff-NRSIM solution in (17) was derived using only the assumption of locally diffeomorphic deformations. In a scenario where we know a priori the specific surface properties, we can incorporate the corresponding constraints explicitly in Diff-NRSIM using (19).

**Conformal constraints.** We obtain the constraints by taking the ratios of the components of \(\mathcal{P}\) and \(\tilde{\mathcal{G}}\) in (20) to remove \(\lambda\) and \(\tilde{\lambda}\). The equations are given by

\[
\frac{\tilde{\mathcal{G}}|1,1|}{\tilde{\mathcal{G}}|2,2|} = \frac{\mathcal{G}|1,1|}{\mathcal{G}|2,2|} \quad \frac{\tilde{\mathcal{G}}|1,2|}{\tilde{\mathcal{G}}|2,2|} = \frac{\mathcal{G}|1,2|}{\mathcal{G}|2,2|}
\]

(21)
Thus we obtain two relations of degree 9 in two variables.

**Equiareal constraints.** We obtain the constraints by comparing areas, which are given by determinants, in (20). Substituting \((\frac{\pi}{2})^2\) from (11) gives

\[
(t_1^2 + t_2^2)^2|G| = (x_1^2 + x_2^2)^2|G|.
\]

(22)

Thus we obtain a relation of degree 8 in two variables.

**Isometric constraints.** Isometry is expressed as a combination of conformity and equiareality. We use (21) and (22) to define isometric constraints.

### 5.3. Algorithm

Let \(\{x_j^i\}, i \in [1, M], j \in [1, N]\), denote a set of \(N\) point correspondences between \(M\) images. Our goal is to find the 3D point and the normal corresponding to each \(x_j^i\). We take an arbitrary image \(I\) as our reference and use a standard algorithm such as optical flow [29, 28] or SIFT [22] to compute an \(\eta\) mapping between each remaining images and \(I\). Our point-wise Diff-NRSfM algorithm then goes through the following steps:

- **Solve for \(x_2\).** For each image paired with the reference, compute the resultant \(R(x_2)\) from \(A(x_1, x_2) = 0, B(x_1, x_2) = 0\), defined in (17). Find \(x_2\) by minimizing the sum of squares of \(R(x_2) = 0\) computed over all available image pairs.

- **Solve for \(x_1\).** Substitute \(x_2\) obtained from the previous step into \(A(x_1, x_2) = 0, B(x_1, x_2) = 0\) and find \(x_1\) by minimizing their sum of squares.

- **(optional) Add metric preserving constraints.** If the deformation model is known \(a priori\), minimize the sum of squares of (21), or (22), or both to add conformal, equiareal, or isometric constraints. Use \((x_1, x_2)\) obtained from previous steps to initialize this solution.

- **Find local normals.** Use \((x_1, x_2)\) to express local normals as in (2).

After obtaining a local normal for each \(x_j^i\), we integrate them to compute depth up to a scale factor.

### 6. Experiments

We first evaluate our method on the NRSfM challenge dataset [17] and then show results on one synthetic and three real datasets.

In the remainder of this section, we denote our Diff-NRSfM framework as Diff and its variants with isometric, conformal or equiareal constraints as DiffC, DiffF and DiffE. We compare it to competing NRSfM approaches that assume isometry/conformity, Pa17 [24, 25], inextensibility, Ch17 [9], soft-inextensibility, Vi12 [34], local rigidity Ch14 [8] and low-rank constraints Go11 [13]. We report mean shape \(E_s\) and mean depth \(E_d\) errors that are computed as RMSEs between reconstructed and ground-truth normals and 3D points.

**NRSfM Challenge dataset.** This dataset consists of 5 image sequences depicted by Fig. 2. They feature 5 kinds of non-rigid motions: articulated (piecewise-rigid), balloon (conformal), paper bending (isometric), rubber (elastic), and paper being torn. 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. Fig. 3 compares the performance of Diff with that of other methods on this dataset, with Best being the one that does best to date and that we beat by a large margin in the perspective case. Note that Diff still does well in the orthographic case, even though it is explicitly designed for the perspective case. Note also that Pa17 does not appear in this table, presumably because the sparsity of the correspondences are not ideal to compute the second derivatives that it needs.

**Cylinder dataset.** It consists of 400 points tracked across 10 images (640×480p) of a cylindrical surface deforming isometrically created synthetically. We add a gaussian noise of 3 pixels to the image points. The results are shown in Figure 4a. Pa17 shows the best performance, with our method,
**Diff**, being close to it. The results are slightly improved with the additional constraints of **DiffI**, **DiffC** and **DiffE**. The constraints obtained for **Diff**, **DiffC** and **DiffE** are high order polynomials which do not lead to strong constraints in practical terms. **Ch14** and **Vi12** also show a decent performance, but it is not consistent throughout the dataset. **Ch17** does not perform well on this dataset, it yields a very high normal error which indicates flattening. This is because it requires the constraints to be very distinct especially if the images are as few as 10. We do not show the results of **Go11**.
as it is a low-rank method and requires a lot more than 10 images to give a stable performance.

**Tshirt dataset.** [8] It consists of 85 manually computed points correspondences across 10 images of a tshirt deforming isometrically from very different views. Figure 4b shows the results. Ch17 shows the best performance on this dataset. Pa17 also has a very stable performance. The performance of our methods, Diff, DiffI, DiffC and DiffE, is also very stable and similar to that of Pa17. Ch14 and Vi12 do not show a consistent performance but they get decent results for most of the images.

**Paper dataset.** [27] It consists of 190 images of a paper deforming isometrically with 1500 point correspondences on them. For such a high number of images, Ch14 and Vi12 are highly impractical. We used only 150 point correspondences on this dataset. Figure 4c shows the 3D reconstruction error for all compared methods. Amongst our methods, we show the results of Diff only as the remaining ones perform very close to it, as seen in Table 1. Ch17 shows the best performance on this dataset. The performance of our methods is very close to Pa17. Go11 shows the worst performance. In supplementary material, we show the visual performance on few images.

**Rug dataset.** [24] It consists of 160 images of a rug deforming isometrically with 1500 point correspondences obtained using optical flow [29]. We consider 350 points for evaluating all methods. This data contains correspondence errors arising from the lack of texture on the object. Figure 4d shows the comparison of the 3D error of Diff with the rest. We see that Diff performs the best on this dataset. Pa17’s performance is degraded due to poor computation of second-order derivatives in noisy conditions. Ch17 degrades even more and its performance is worse than Diff and Pa17. Go11 does not perform well. In supplementary material, we compare the visual performance on few images.

**Discussion.** In short, when the correspondences are sparse, as in the NRSfM Challenge dataset, our approach dominates as shown in Fig. 2 and when they are dense and of high quality Pa17 does best but our approach comes very close behind in terms of accuracy. However, when met with noisy data, such as the Rug dataset, Pa17 degrades due to high error in second order derivatives and Diff becomes noticeably better, as shown in Table 1.

**Computation time.** Ch14 and Vi12 usually take 15-20 minutes to reconstruct 10 images. Therefore, we did not evaluate them on large datasets. Ch17 takes almost 15 minutes for 30 images, therefore, we split the large sequences into sets of 30 images and evaluated this method. Pa17 is a local method with linear complexity, like ours, but it uses an expensive polynomial solver which takes 1.5 seconds to evaluate normals at a point constrained from 10 images. In similar conditions, Diff takes only 10 ms, thus, it is $10 \times$ faster. In addition, it only requires first order derivatives which can be computed within 20-30 ms for each image pair. Refined second-order derivatives that are required by Pa17 may take anything between 2-5 seconds.

### Table 1: RMSE for all methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Kinect paper</th>
<th>Rug</th>
<th>Tshirt</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Es</td>
<td>Ed</td>
<td>Es</td>
</tr>
<tr>
<td>Go11</td>
<td>18.65</td>
<td>28.88</td>
<td>19.87</td>
</tr>
<tr>
<td>Vi12</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Ch14</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Ch17</td>
<td>6.40</td>
<td>5.38</td>
<td>16.77</td>
</tr>
<tr>
<td>Pa17</td>
<td>10.50</td>
<td>8.26</td>
<td>16.70</td>
</tr>
<tr>
<td>Diff</td>
<td>20.75</td>
<td>10.75</td>
<td>22.06</td>
</tr>
<tr>
<td>DiffI</td>
<td>19.51</td>
<td>10.39</td>
<td>21.67</td>
</tr>
<tr>
<td>DiffC</td>
<td>19.01</td>
<td>9.91</td>
<td>21.29</td>
</tr>
<tr>
<td>DiffE</td>
<td>19.55</td>
<td>10.39</td>
<td>21.79</td>
</tr>
</tbody>
</table>

#### 7. Conclusions and Future directions

In this paper, we have explored the limiting case for NRSfM: What minimum assumptions on surface deformations are required to solve the problem. We have shown that the assumption of local diffeomorphism, which is a generic surface property of surfaces, yields enough constraints to perform reconstruction. Our experiments have validated our theoretical formulation and demonstrated that it compares favorably to methods that rely on much stronger constraints such as isometry or conformality. Furthermore, it is faster and applicable in more general settings because it only requires first-order derivatives instead of second-order. In the future, we plan to exploit the strengths of this method to develop refinement methods for NRSfM, to which Diff-NRSfM can serve as an initialization.

### References


Appendix

We prove the invariance under local diffeomorphism that was used in Section 4 of the main paper and we then provide qualitative results on the NRSIM challenge dataset.

A1. Invariance under Local Diffeomorphism

In figure 1, \( \eta \) is the image registration function, which means that we have \( x = \eta(x) \). We can then write

\[
\left( \frac{dx}{dv} \right) = \begin{pmatrix} \frac{\partial x}{\partial v} & \frac{\partial y}{\partial v} \end{pmatrix},
\]

where \( \frac{dx}{dv} \) is the Jacobian of the function \( \eta \). Thus, under a change of variable from \( x \) to \( \mathbf{X} \), we write \( E(\phi \circ \eta) \) in terms of \( E(\psi) \) (written as (2)) as

\[
E(\phi \circ \eta) = \left( (e_1 \quad e_2) \quad J_\eta \quad [J_\eta]e_3 \right) = E(\phi) \text{diag}(J_\eta, [J_\eta]) .
\]

We now introduce a theorem that describes the relation between moving frames related by a local diffeomorphism.

**Theorem 1** (Moving frames under local diffeomorphism). Given two surfaces \( \mathcal{S} \) and \( \mathcal{S}' \) related by a local diffeomorphic mapping \( \psi \), their respective moving frames \( \mathcal{E} \) and \( \mathcal{E}' \) are related by a linear transformation.

**Proof.** From figure 1, we can write \( \mathcal{E}' = \psi \circ \phi \circ \eta \) and therefore,

\[
J_{\mathcal{E}'} = J_{\psi \circ \phi \circ \eta} J_{\phi \circ \eta} J_{\eta}
\]

(A3)

According to inverse function theorem, \( J_{\psi} \) is a linear function if \( \psi \) is locally diffeomorphic. Thus, we have \( J_{\psi} = \text{diag}(\lambda_1, \lambda_2, \lambda_3) \mathbf{R} \) where \( \lambda_i \) are scalars and \( \mathbf{R} \) is a rotation matrix. Using this result and the expression of \( E(\phi) \) in terms of \( J_\phi \) in (\( \phi \)), we write (A3) as

\[
\mathcal{E}(\mathcal{E}') = \text{diag}(\lambda_1, \lambda_2, \lambda_3) \mathbf{R} E(\phi \circ \eta),
\]

(A4)

where \( E(\phi \circ \eta) \) is given by (A2).

\( \square \)

Given the relation (A3) obtained between moving frames under local diffeomorphism, we now derive the relation between connection components \( \Gamma^1_{jk}(\mathcal{E}) \) and \( \Gamma^1_{jk}(\mathcal{E}') \) in the next theorem.

We write \( w_j^i = \Gamma^i_{jk} du + \Gamma^i_{j2} dv \), and thus, the linear system in (3) can be written as

\[
de_j = \frac{\partial e_j}{\partial u} du + \frac{\partial e_j}{\partial v} dv = w_j^1 e_1 + w_j^2 e_2 + w_j^3 e_3, \quad j = [1, 2, 3]
\]

(A5)

**Theorem 2** (Connection preservation under local diffeomorphism). Given two surfaces \( \mathcal{S} \) and \( \mathcal{S}' \) related by a local diffeomorphic mapping \( \psi \), their respective connection components \( \Gamma^i_{jk}(\phi) \) and \( \Gamma^i_{jk}(\mathcal{E}) \) are preserved, i.e., \( \Gamma^i_{jk}(\mathcal{E}) = \Gamma^i_{jk}(\phi \circ \eta) \).

**Proof.** Using the relation between \( \mathcal{E} \) and \( \mathcal{E}' \) obtained in (A4) in (A5), we get

\[
\begin{pmatrix}
\lambda_1 e_1^1 \\
\lambda_2 e_2^1 \\
\lambda_3 e_3^1
\end{pmatrix}
\begin{pmatrix}
\mathbf{R}^T \\
\mathbf{d}(\mathbf{R})
\end{pmatrix}
\begin{pmatrix}
\eta_1^1 \\
\eta_2^1 \\
\eta_3^1
\end{pmatrix} =
\begin{pmatrix}
\lambda_1 e_1^1 \\
\lambda_2 e_2^1 \\
\lambda_3 e_3^1
\end{pmatrix}
\begin{pmatrix}
\mathbf{R}^T \\
\mathbf{d}(\mathbf{R})
\end{pmatrix}
\begin{pmatrix}
\eta_1^1 \\
\eta_2^1 \\
\eta_3^1
\end{pmatrix} + \mathbf{d}(\mathbf{J}_\eta, \mathbf{d}[\mathbf{J}_\eta])
\]

(A6)

The above relation is thus independent of \( \lambda_i \) and \( \mathbf{R} \). We write \( \eta_j^1 = w_j^1(\phi \circ \eta) \) and thus, \( \Gamma^1_{jk}(\mathcal{E}) = \Gamma^1_{jk}(\phi \circ \eta) \). \( \square \)

Upon expanding the relation between the connections (A6), we get

\[
\begin{pmatrix}
\Gamma^1_{11} \\
\Gamma^1_{21} \\
\Gamma^1_{31}
\end{pmatrix}
\begin{pmatrix}
\eta_1^1 \\
\eta_2^1 \\
\eta_3^1
\end{pmatrix} = \begin{pmatrix}
0 & 0 & -1 \\
0 & 1 & 0 \\
1 & 0 & 0
\end{pmatrix}
\begin{pmatrix}
\eta_1^1 \\
\eta_2^1 \\
\eta_3^1
\end{pmatrix} + \mathbf{d}(\mathbf{J}_\eta, \mathbf{d}[\mathbf{J}_\eta])
\]

(A7)

We write \( \mathbf{J}_\eta = \begin{pmatrix} a & b & c \\
d & e & f \end{pmatrix} \) and the above expression can be written as
\[
\begin{align*}
\left( \begin{array}{c}
\Gamma_{11}^1 \\
\Gamma_{21}^1 \\
\Gamma_{31}^1
\end{array} \right) & = \left( \begin{array}{c}
\mathbf{J}_y \\
0 \\
0
\end{array} \right) - 1 \left( \begin{array}{c}
\frac{\partial J_y}{\partial \eta} \\
0 \\
0
\end{array} \right) + \left( \begin{array}{c}
\frac{\partial J_y}{\partial \eta} \\
0 \\
0
\end{array} \right) + \left( \begin{array}{c}
\frac{\partial J_y}{\partial \eta} \\
0 \\
0
\end{array} \right), \\
\left( \begin{array}{c}
\Gamma_{12}^1 \\
\Gamma_{22}^1 \\
\Gamma_{32}^1
\end{array} \right) & = \left( \begin{array}{c}
\mathbf{J}_y \\
0 \\
0
\end{array} \right) - 1 \left( \begin{array}{c}
\frac{\partial J_y}{\partial \eta} \\
0 \\
0
\end{array} \right) + \left( \begin{array}{c}
\frac{\partial J_y}{\partial \eta} \\
0 \\
0
\end{array} \right) + \left( \begin{array}{c}
\frac{\partial J_y}{\partial \eta} \\
0 \\
0
\end{array} \right), \\
\left( \begin{array}{c}
\Gamma_{13}^1 \\
\Gamma_{23}^1 \\
\Gamma_{33}^1
\end{array} \right) & = \left( \begin{array}{c}
\mathbf{J}_y \\
0 \\
0
\end{array} \right) - 1 \left( \begin{array}{c}
\frac{\partial J_y}{\partial \eta} \\
0 \\
0
\end{array} \right) + \left( \begin{array}{c}
\frac{\partial J_y}{\partial \eta} \\
0 \\
0
\end{array} \right) + \left( \begin{array}{c}
\frac{\partial J_y}{\partial \eta} \\
0 \\
0
\end{array} \right).
\end{align*}
\]

Substituting \(\Gamma_{jk}^i\) and \(\Gamma_{jk}^i\) from (4) gives

\[
\begin{align*}
\Gamma_{11}^1 & = \frac{2\pi \beta^4}{\mathcal{D}} (1 + (1 + \mathbf{v}_1^2) \mathbf{x}_2^2 + \mathbf{w}_x \mathbf{w}_y + 2 \mathbf{w}_x \mathbf{u}_x + \mathbf{w}_y \mathbf{u}_y - \mathbf{u}_x^2 + 2 \mathbf{w}_y \mathbf{u}_x + \mathbf{w}_x \mathbf{u}_x^2) \\
\Gamma_{12}^1 & = \frac{2\pi \beta^4}{\mathcal{D}} (1 + (1 + \mathbf{v}_1^2) \mathbf{x}_2^2 + \mathbf{w}_x \mathbf{w}_y + 2 \mathbf{w}_x \mathbf{u}_x + \mathbf{w}_y \mathbf{u}_y - \mathbf{u}_x^2 + 2 \mathbf{w}_y \mathbf{u}_x + \mathbf{w}_x \mathbf{u}_x^2) \\
\Gamma_{13}^1 & = \frac{2\pi \beta^4}{\mathcal{D}} (1 + (1 + \mathbf{v}_1^2) \mathbf{x}_2^2 + \mathbf{w}_x \mathbf{w}_y + 2 \mathbf{w}_x \mathbf{u}_x + \mathbf{w}_y \mathbf{u}_y - \mathbf{u}_x^2 + 2 \mathbf{w}_y \mathbf{u}_x + \mathbf{w}_x \mathbf{u}_x^2).
\end{align*}
\]
\[ \Gamma^3_{12} = \Gamma^3_{21} = \frac{\frac{\partial \eta}{\partial x_1} \frac{\partial \eta}{\partial x_2} \beta^3}{D} = \frac{\beta^3(ax_1 + bx_2)(cx_1 + dx_2)}{|J_\eta|D}, \]
\[ \Gamma^3_{22} = \frac{\frac{\partial \eta}{\partial x_1} \frac{\partial \eta}{\partial x_2} \beta^3}{D} = \frac{\beta^3(cx_1 + dx_2)^2}{|J_\eta|D}. \]
\[ \Gamma^1_{31} = \frac{\frac{\partial \eta}{\partial x_1} \frac{\partial \eta}{\partial x_2} \beta^3}{D} = \frac{\beta^3(ax_1 + bx_2)}{D} \left( d(x_1 + vx_1 x_2 - ux_2) - c(x_2 + ux_1 x_2 - vx_1^2) \right), \]
\[ \Gamma^2_{31} = \frac{\frac{\partial \eta}{\partial x_1} \frac{\partial \eta}{\partial x_2} \beta^3}{D} = \frac{\beta^3(ax_1 + cx_2)}{D} \left( -b(x_1 + vx_1 x_2 - ux_2^2) + a(x_2 + ux_1 x_2 - vx_1^2) \right), \]
\[ \Gamma^1_{32} = \frac{\frac{\partial \eta}{\partial x_1} \frac{\partial \eta}{\partial x_2} \beta^3}{D} = \frac{\beta^3(cx_1 + dx_2)}{D} \left( d(x_1 + vx_1 x_2 - ux_2) - c(x_2 + ux_1 x_2 - vx_1^2) \right), \]
\[ \Gamma^2_{32} = \frac{\frac{\partial \eta}{\partial x_1} \frac{\partial \eta}{\partial x_2} \beta^3}{D} = \frac{\beta^3(cx_1 + dx_2)}{D} \left( -b(x_1 + vx_1 x_2 - ux_2^2) + a(x_2 + ux_1 x_2 - vx_1^2) \right). \]

From the last expressions, we can see that the last 8 connections 
\((\Gamma^3_{11}, \Gamma^3_{12}, \Gamma^3_{12}, \Gamma^3_{21}, \Gamma^3_{22}, \Gamma^1_{31}, \Gamma^1_{32}, \Gamma^2_{32})\), do not contain second order derivatives of \(\eta\).
A. Results on NRSfM Challenge Dataset

In Fig. A1, we provide qualitative results on the stretch sequence of the NRSfM challenge dataset. We remind that the data provided in this dataset is very limited and it is not known how the orthographic or perspective images were created. The images generated by these projections look similar but we did find that the data was numerically different. The camera matrices have not been provided which limits further inspection.

We show reconstruction on the two sequences. These sequences indicate camera motion flyby and semicircle. We show the reconstructions obtained by Diff (in green) on the orthographic and perspective projection and compare with the ground truth available (in blue). We can see that Diff reconstructs well even with orthographic images. The stretching of the rubber is well-preserved except for certain corners. This is because Diff does not depend on strong constraints such as isometry, it can reconstruct a wide range of deformations including elasticity up to an extent.

![Figure A1: Results on stretch dataset. Blue indicates ground truth and green indicates the reconstruction obtained using Diff. The mean depth error is reported for each reconstruction.](image-url)