Robust Isometric Non-Rigid Structure-from-Motion

Shaifali Parashar, Daniel Pizarro and Adrien Bartoli

Abstract—Non-Rigid Structure-from-Motion (NRSfM) reconstructs a deformable 3D object from keypoint correspondences established between monocular 2D images. Current NRSfM methods lack statistical robustness, which is the ability to cope with correspondence errors. This prevents one to use automatically established correspondences, which are prone to errors, thereby strongly limiting the scope of NRSfM. We propose a three-step automatic pipeline to solve NRSfM robustly by exploiting isometry. Step *(i)* computes the optical flow from correspondences, step *(ii)* reconstructs each 3D point's normal vector using multiple reference images and integrates them to form surfaces with the best reference and step *(iii)* rejects the 3D points that break isometry in their local neighborhood. Importantly, each step is designed to discard or flag erroneous correspondences. Our contributions include the robustification of optical flow by warp estimation, new fast analytic solutions to local normal reconstruction and their robustification, and a new scale-independent measure of 3D local isometric coherence. Experimental results show that our robust NRSfM method consistently outperforms existing methods on both synthetic and real datasets.

Index Terms—3D computer vision, NRSfM, robustness, isometry

1 Introduction

The 3D reconstruction of a deformable object from monocular images is a key computer vision problem to which NRSfM forms a promising approach. NRSfM takes keypoint correspondences as inputs and reconstructs a 3D point cloud for each image. The correspondences are typically obtained from either temporally-organized, short-baseline images extracted from a video or temporally-unorganized, wide-baseline images. Most NRSfM methods are for short-baseline data [8], [13], [15], [21], [41] as the wide-baseline case was studied later [9], [10], [44]. Most of the recent methods exploit isometry, which has been established as a widely applicable object deformation model [10], [31], [32], [36], [44]. The recent methods have brought considerable improvements in accuracy and computation time [10], [13], [21], [31], [36]. However, the majority of existing methods are non-robust: they may tolerate noise in image point positions up to some level but do not cope with erroneous correspondences. This strongly limits their scope as any automatic correspondence computation method is likely to make mistakes. This is mitigated in the short-baseline case, where correspondences may be obtained from optical flow methods such as [19], [38] and successfully used in NRSfM. Failure will however occur when the amplitude of correspondence drift, which grows with the number of images, reaches the maximal noise level tolerance of NRSfM. The wide-baseline case is however far worse, for it relies on establishing correspondences, using for instance SIFT [27], which contain far more errors than optical flow and a large amount of missing data.

A robust NRSfM method must cope with erroneous correspondences by classifying each image point as an inlier or an outlier while performing the reconstruction. It is important to classify each image point and not each correspondence, as a correspondence spanning multiple images is generally erroneous due to a few points only but still provides useful constraints. Developing a robust NRSfM method is extremely challenging, mainly because the constraints one may use are much weaker than rigidity in SfM [22] and SLAM [14]. The high level of robustness reached in the rigid case is largely based on the principle of random sampling, as popularized by RANSAC [18]. As deformation is a local phenomenon, this principle can unfortunately not be used in NRSfM.

We propose a three-step robust NRSfM pipeline for isometric deformations (see figure 1). The idea is to leverage our correspondence-wise solution to NRSfM [31], which is based on the local optical flow derivatives. Such a solution is less vulnerable to errors as it does not use the entire set of correspondences together at the early stages of reconstruction. However, [31] falls in the category of NRSfM methods [9], [39], [44], [45] that use a single reference image to perform reconstruction, which may substantially degrade the performance. Each step in our robust NRSfM pipeline is carefully designed to ensure stability, statistical robustness and rejection of the inconsistent image points: *(i) Computation of the optical flow derivatives at the correspondences.* This step interpolates the correspondences using robust warp fitting and differentiation. *(ii) Up-to-scale correspondence-wise reconstruction.* This step reconstructs the surface normal at each 3D point independently for each correspondence using multiple reference images. It keeps the normals corresponding to the best reference image, which is the one that yields the most coherent normals. It finally integrates the normal field to obtain up-to-scale 3D point clouds. *(iii) Isometry consistent filtering and rescaling.* This step filters out the 3D points incoherent with their neighborhood according to isometry, finds the relative scale between the 3D point clouds and rescales them.

Beyond the proposed pipeline, we address several important problems required to implement its steps, leading to the follow-

- *S. Parashar did this work with EnCoV-Institut Pascal-CNRS/Université Clermont Auvergne, Clermont-Ferrand, France.*
- *D. Pizarro is with GEINTRA, Universidad de Alcalá, Alcalá de Henares, Spain.*
- *A. Bartoli is with EnCoV-Institut Pascal- CNRS/Université Clermont Auvergne, Clermont-Ferrand, France.*

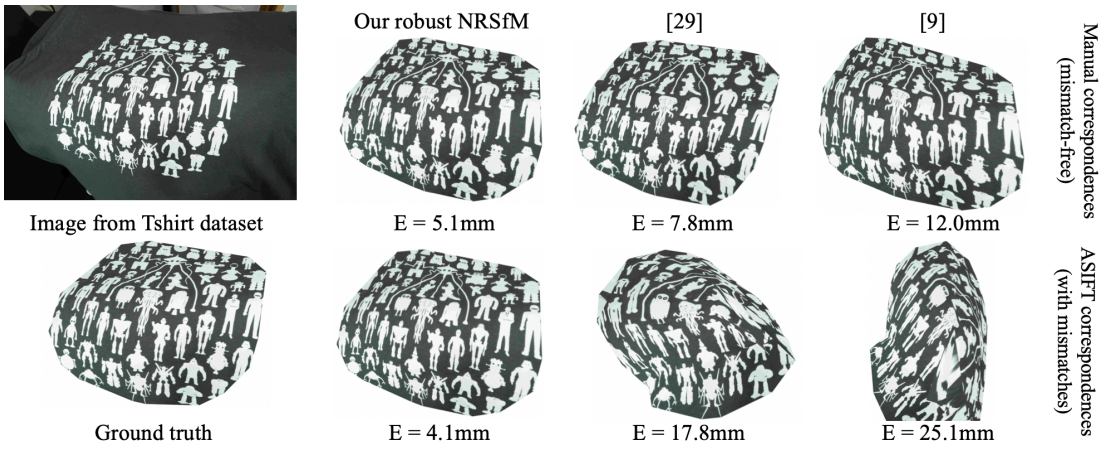


Fig. 1: Reconstruction of an image from the Tshirt dataset using *automatic correspondences*. This dataset was introduced in [9] with *manually clicked correspondences*. With these 'perfect' correspondences, both [9], [30] achieved good 3D reconstruction quality. However, they both fail when the correspondences are obtained automatically using ASIFT, because these contain mismatches. Our robust method, on the other hand, handles both the manual and automatic correspondences well.

ing four contributions. Our first contribution, in step *(i)*, is a robustification of Schwarps [34], a stable but non-robust warp estimation method. Our second and third contributions, in step *(ii)*, are respectively a tremendous acceleration and a robustification of the normal reconstruction method [31]. This method takes a correspondence with its local differential structure as input and reconstructs the surface normal for each image. The first problem with [31] is that it is prohibitively slow due to the use of a computationally expensive generic polynomial system solver [23] for multivariate equations. Relatively faster polynomial solvers such as [25] cannot be used as they need both linear and non-linear constraints, some of which independent of the image observations. On the other hand, solving univariate equations has been known to be extremely fast and reliable. Our second contribution is to convert the multivariate equations in [31] to a single univariate equation which guarantees a very fast solution. We propose two ways to obtain the univariate equation: by substitution and by resultants. The method of substitution does not require additional tools to transform the equations and is therefore extremely fast. However, it does not guarantee a real solution. The method of resultants is time-consuming when directly applied. We propose an offline computation which leads to a gain in speed of two orders of magnitude compared to [31] and brings it almost on par with the method of substitution. Importantly, the method of resultants guarantees at least one real solution. The second problem with [31] is that it is not robust, as it gives the same weight to all images and relies on an arbitrary selected reference image. Our third contribution is a robust estimation method based on splitting the input images into small subsets. For each subset, we select the reference image as the image giving the most coherent normal reconstruction, as measured with a robust statistic, and discard the incoherent points. Our fourth contribution, in step *(iii)*, is a scale-independent isometric consistency measure, which we use to reject the locally inconsistent 3D points.

We experimentally compared our method with existing ones on synthetic and real datasets, showing that it leads to more accurate results with and without correspondence errors, while coping with beyond 50% correspondence errors, an unprecedented achievement in NRSfM.

2 Previous Work

The first NRSfM method was probably [8], which introduced the low-rank shape model that imposes the time-varying 3D shape to follow a linear combination of shape bases. Numerous improvements were then proposed, including non-linear refinement [15], spatial smoothness [41] and a quadratic deformation model [17]. Inspired by the shape bases, the trajectory basis model imposes that each point trajectory follows a pre-computed basis [4]. This was improved by using a DCT basis to cope with large deformations [21]. [13] proposed a convex relaxation of the shape bases model and solved it using convex optimization. Recently, [3] expressed NRSfM using forces acting on the shape and a force basis. In parallel, template-based methods were developed successfully [2], [6], [37], achieving robustness due to the stronger constraints provided by the template [11], [40]. The success of these methods inspired the development of NRSfM with physics-based deformation models, in particular isometry [9], [31], [36], [39], [44], [45]. [10], [24] proposed an NRSfM method using the inextensibility relaxation of isometry. In contrast to most of the existing methods which use the calibrated perspective or orthographic camera, [29], [35] solve NRSfM with the uncalibrated perspective camera. [35] exhaustively searches for a focal length which maintains the best possible isometric consistency of reconstruction across views. [29] computes resultants to eliminate the surface from first- and second-order constraints, leaving only the focal length as unknown.

However, the constraints available in NRSfM are much weaker than in SfM and in template-based reconstruction. Thus, NRSfM methods have not yet reached the same level of robustness, most of them being highly sensitive to correspondence errors. Some however have made a step towards robustness. [20] discards correspondences by testing compatibility with a shape prior. The need for a shape prior however strongly limits its applicability. [44] constrains the shape by learnt linear local models and [10] minimises the L_1 norm of slack variables modeling the deviation between the observed and predicted points. In practice, [44] breaks at about 10% and [10] at about 20% correspondence errors. The other NRSfM methods, which do not have a robust design, break at a few percents of correspondence errors, except [13],

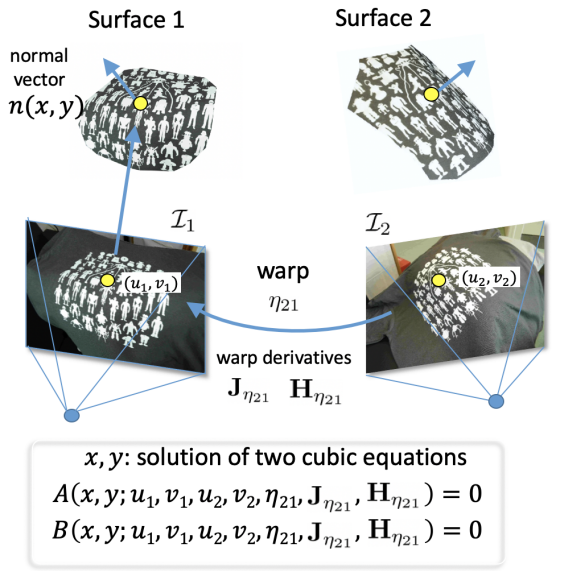


Fig. 2: Notation, illustrated in a two-view NRSfM example.

which we have found to break at 20-25%. In contrast, our robust NRSfM method uses a pipeline where each step was carefully designed to handle correspondence errors and breaks beyond 50% correspondence errors.

3 Fast Solutions to Isometric NRSfM

We consider a set of M images of a deforming object with point correspondences. We fix a reference image of index $i = 1$ and evaluate the η_{ji} warps, $j \in [2, M]$, using Schwarps [34]. Schwarps penalise the residual of the Schwarzian equations, which are second-order PDEs whose solutions are homographies. Schwarps thus preserve the local projective structure of the image transformation, leading to better performance than other solutions based on penalising the warp's second-order derivatives. Using η_{ji}, we hallucinate point correspondences by creating a grid on the reference image and transferring its points to the other images. Our goal is to retrieve the depth and normal for each of these points on each image. Following [31], the essential first step is to reconstruct the normal in the reference image, from the bi-cubic reconstruction equations. The normal and depth for all images are then easily found and can be jointly refined by iterative nonlinear minimisation. The reference normal reconstruction problem being non-convex, having a reliable initial estimate is essential. The generic polynomial optimiser [23] was used in [31] to minimise the sum of squares of the bicubic reconstruction equations, which results in extremely slow performance. In contrast, our new solutions rely on solving univariate polynomials, which is known to be very fast and stable. Our first method uses substitution to construct the univariates from the reconstruction equations, whereas our second approach uses the theory of resultants. We first recall the reconstruction equations from [31] and then present our two fast ad hoc initialisation solutions. Our notation is illustrated by figure 2.

3.1 Reconstruction Equations

We now explain how the isometric reconstruction equations from [31] are derived. These equations require concepts from Riemmanian differential geometry, mainly the metric tensor and the Christoffel Symbols (CS), to model the differential relationship between isometric surfaces and their image projections.

We describe a surface using an embedding function $\mathbf{P} = \varphi(\mathbf{p})$, where $\mathbf{p} = (u, v)^\top$ and $\mathbf{P} = (P_x, P_y, P_z)$. We assume that φ is at least twice differentiable. The surface tangent plane is generated by the embedding's 3×2 Jacobian matrix $\mathbf{J}$:

$$\mathbf{J} = \frac{\partial \varphi}{\partial \mathbf{p}} = \begin{pmatrix} \frac{\partial \varphi}{\partial u} & \frac{\partial \varphi}{\partial v} \end{pmatrix}. \tag{1}$$

Specifically, the tangent plane at any point $\mathbf{p}$, is the linear subspace generated by the column vectors of $\mathbf{J}$. The normal vector $\mathbf{n} = \frac{\partial \varphi}{\partial u} \times \frac{\partial \varphi}{\partial v}$ of the tangent plane coincides with the surface normal. Locally, the tangent plane induces a metric to measure differential distances between points on the surface. This metric is captured by the metric tensor $\mathbf{g}$, also known as the first fundamental form, given by:

$$\mathbf{g} = \mathbf{J}^\top . \mathbf{J} \tag{2}$$

The metric tensor is a 2×2 quadratic form which depends on the surface's embedding. For a change of coordinates $\mathbf{p} = \eta(\bar{\mathbf{p}})$, the surface embedding in the new coordinates $\bar{\varphi} = \varphi \circ \eta$ has the following metric tensor:

$$\bar{\mathbf{g}} = \mathbf{J}_\eta^\top \mathbf{g} \mathbf{J}_\eta \quad \text{with} \quad \mathbf{J}_\eta = \frac{\partial \eta}{\partial \bar{\mathbf{p}}}. \tag{3}$$

The CS of the second kind, $\mathbf{\Gamma}^u$ and $\mathbf{\Gamma}^v$, are 2×2 matrix functions that measure the local rate of change in the metric tensor. They are thus defined from the first-order derivatives of $\mathbf{g}$, which involve up to second-order derivatives of φ. The CS define the surface curvature and its geodesics. They also admit a change of variable that requires the first and second derivatives of η. We refer the reader to [16] and our original method developed in [31] for the detailed expressions of the CS in tensor notation.

One important result from [31] is the invariance of the metric tensor and the CS under isometric deformations of the observed surface. Given two isometric surfaces (*i.e.*, surfaces related by an isometric mapping), defined by their embeddings φ_1 and φ_2, the metric tensors and the CS are preserved:

$$\mathbf{g}_1 = \mathbf{g}_2 \qquad \mathbf{\Gamma}_1^u = \mathbf{\Gamma}_2^u \qquad \mathbf{\Gamma}_1^u = \mathbf{\Gamma}_2^u. \tag{4}$$

The constraints in equation (4) allow us to find the NRSfM solution for a pair of views $\mathcal{I}_1, \mathcal{I}_2$ from the warp function η_{21}. For each view $i \in \{1, 2\}$ we define a surface embedding:

$$\varphi_i(\mathbf{p}_i) = \frac{1}{\beta_i(\mathbf{p}_i)} (\mathbf{p}_i \quad 1)^\top, \tag{5}$$

where $\mathbf{p}_i = (u_i, v_i)$ are normalized image coordinates, obtained by multiplying pixel coordinates by the inverse of the intrinsic matrix. The function $\beta_i(\mathbf{p})$ is the inverse depth function. Using β instead of the depth function greatly simplifies our equations. The Jacobian matrix of φ_i is expressed as:

$$\mathbf{J}_i = \frac{1}{\beta} \begin{pmatrix} 1 - u_i x_i & -v_i y_i \\ -v_i x_i & 1 - v_i y_i \\ -x_i & -y_i \end{pmatrix}, \tag{6}$$

where $(x_i, y_i) = \frac{1}{\beta_i} \left(\frac{\partial \beta_i}{\partial u_i}, \frac{\partial \beta_i}{\partial v_i} \right)$. Importantly, (x_i, y_i) are the local reconstruction unknowns for view $\mathcal{I}_i$, since they allow us to recover the surface normal as:

$$\mathbf{n} = \begin{pmatrix} x_1 & y_1 & 1 - x_1 u_1 - y_1 v_1 \end{pmatrix}^\top. \tag{7}$$

By identifying $(u_2, v_2) = \eta_{21}(u_1, v_1)$ as a change of variable between $\mathcal{I}_1$ and $\mathcal{I}_2$, and by imposing the constraints of equation (4), we find the reconstruction equations. First, the local isometric deformation constraint in terms of metric tensors is:

$$\mathbf{g}_2 = \mathbf{J}_{\eta_{21}}^\top \mathbf{g}_1 \mathbf{J}_{\eta_{21}}, \tag{8}$$

where $\mathbf{g}_i = \mathbf{J}_i^\top \mathbf{J}_i$. The expressions on both sides of equation (8) are symmetric 2×2 matrices and therefore yield 3 equations. The known and unknown quantities are $(u_1, v_1, u_2, v_2, \mathbf{J}_{\eta_{21}})$ and $(\beta_1, x_1, y_1, \beta_2, x_2, y_2)$ respectively. (β_1, β_2) are cancelled by taking ratios of the 3 equations, which leaves 2 reconstruction equations for an image pair with (x_1, y_1, x_2, y_2) as unknowns. Second, the CS between the two surfaces are preserved up to a change of variable given by η_{21}, from which [31] derived the following linear relationship between (x_1, y_1) and (x_2, y_2):

$$\begin{pmatrix} x_2 \\ y_2 \end{pmatrix} = \mathbf{J}_{\eta_{21}}^\top \begin{pmatrix} x \\ y \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \mathbf{J}_{\eta_{21}}^{-1} \begin{pmatrix} h_3 \\ h_4 \end{pmatrix}, \tag{9}$$

where $\begin{pmatrix} h_3 \\ h_4 \end{pmatrix}$ is the second column of $\mathbf{H}_{\eta_{21u}}$, the Hessian matrix of η_{21} with respect to u. Substituting equation (9) in equation (8), we arrive at 2 cubic equations with only (x_1, y_1) as unknowns, that for simplicity we refer to as (x, y) hereinafter. The cubics are:

$$\begin{aligned} A = {} & a_{30}x^3 + a_{21}x^2y + a_{12}xy^2 + a_{03}y^3 + a_{20}x^2 \\ & + a_{11}xy + a_{02}y^2 + a_{10}x + a_{01}y + a_{00}, \end{aligned} \tag{10}$$

$$\begin{aligned} B = {} & b_{30}x^3 + b_{21}x^2y + b_{12}xy^2 + b_{03}y^3 + b_{20}x^2 \\ & + b_{11}xy + b_{02}y^2 + b_{10}x + b_{01}y + b_{00}, \end{aligned} \tag{11}$$

where the coefficients a_{kl} and b_{kl} are expressed in terms of $\mathbf{p}_1$, $\mathbf{p}_2$ and:

$$\mathbf{J}_{\eta_{21}} = \begin{pmatrix} j_1 & j_3 \\ j_2 & j_4 \end{pmatrix} \qquad \mathbf{H}_{\eta_{21u}} = \begin{pmatrix} h_1 & h_3 \\ h_2 & h_4 \end{pmatrix} \qquad \mathbf{H}_{\eta_{21v}} = \begin{pmatrix} h_3 & h_5 \\ h_4 & h_6 \end{pmatrix}.$$

Appendix A shows the complete expressions of these coefficients. According to Bézout's theorem, this system has up to 9 solutions, so 3 images at least are needed to ensure a single solution. By fixing a reference image, a system of $2(M-1)$ cubics (10) and (11) can be constructed for a correspondence over $M > 2$ images with only 2 variables. By solving for (x, y), we obtain the surface normals using equation (7). The local depth is then obtained by integrating the local normals over the entire surface.

We use the system of cubics A, B in (x, y), as [31]. The system can be assembled efficiently; the remaining challenge is to solve it efficiently. [31] minimises the sum of squares of cubics for all image pairs using the generic polynomial solver [23]. In addition to being utterly computationally expensive, such a solution strategy may lead to severely degraded results due to just a single erroneous image pair, because least-squares are not statistically robust. Erroneous image pairs occur when the geometry is ill-conditioned or when the amount of erroneous correspondences is overly important. In contrast, the proposed method solves the cubics by pairs, by converting them to univariate polynomials which are easily solvable. This has a double advantage: first, it is much faster, and second, it allows us to find a statistically robust consensus amongst all cubics. The proposed method thus deals with erroneous image pairs, covering a large extent of deficiencies which may happen in practice. Concretely, the erroneous image-pairs are identified by detecting the high residuals they cause for most solutions and discarded. Ruling out such image-pairs leads to a solution that is not affected by even highly erroneous image-pairs. In summary, we propose strategies to solve the system for each image pair separately. This gives multiple solutions for a single normal and we pick the one that satisfies the majority of the image-pair constraints using a statistically robust criterion. This is a simple yet efficient way to identify and discard erroneous image-pairs, while obtaining a fast solution to the system.

3.2 Fast Solution using Substitution

Deriving the univariates. We propose a change of variable from (x, y) to $(z_1\ z_2)^\top = \mathbf{J}_{\eta_{21}}^\top (x\ y)^\top$ leading to considerably simplified cubics:

$$\begin{aligned} A' = {} & c_{21}z_1^2z_2 + c_{12}z_1z_2^2 + c_{20}z_1^2 + c_{11}z_1z_2 + c_{02}z_2^2 + c_{10}z_1 \\ & + c_{01}z_2 + c_{00}, \end{aligned} \tag{12}$$

$$B' = z_1 C + d_{03}z_2^3 + d_{02}z_2^2 + d_{01}z_2 + d_{00}, \tag{13}$$

where $C = d_{12}z_2^2 + d_{11}z_2 + d_{10}$ and:

$$\begin{aligned} & c_{21} = 2e_1(e_2t_2 - v_2) + 2e_2e_4,\ c_{12} = 2e_1(u_2 - e_2t_1) - 2e_2e_3, \\ & c_{20} = e_1e_5 - e_2(j_3^2 + j_4^2),\ c_{11} = 4(e_2t_1 - u_2)e_4 + 4(v_2 - e_2t_2)e_3, \\ & c_{02} = e_2(j_1^2 + j_2^2) - e_1e_6,\ c_{10} = 2(u_2 - e_2t_1)(j_3^2 + j_4^2) - 2e_5e_3, \\ & c_{01} = 2e_6e_4 - 2(v_2 - e_2t_2)(j_1^2 + j_2^2), \\ & c_{00} = e_5(j_1^2 + j_2^2) - e_6(j_3^2 + j_4^2), d_{00} = e_5e_7 + e_8(j_3^2 + j_4^2), \\ & d_{12} = e_2e_4 - e_1(v_2 - e_2t_2),\ d_{11} = e_1e_5 - e_2(j_3^2 + j_4^2), \\ & d_{10} = (v_2 - e_2t_2)(j_3^2 + j_4^2) - e_5e_4,\ d_{03} = e_1(u_2 - e_2t_1) - e_2e_3, \\ & d_{02} = e_2e_7 + e_1e_8 - 2(u_2 - e_2t_1)e_4 + 2(v_2 - e_2t_2)e_3, \\ & d_{01} = -e_5e_3 - 2(v_2 - e_2t_2)e_7 + (u_2 - e_2t_1)(j_3^2 + j_4^2) - 2e_8e_4, \\ & e_1 = 1 + u_1^2 + v_1^2,\ e_2 = 1 + u_2^2 + v_2^2, \\ & e_3 = j_1u_1 + j_2v_1,\ e_4 = j_3u_1 + j_4v_1,\ e_5 = 1 - 2t_2v_2 + e_2t_2^2, \\ & e_6 = 1 - 2t_1u_2 + e_2t_1^2, e_7 = j_1j_3 + j_2j_4, e_8 = t_2u_2 + t_1v_2 - e_2t_1t_2, \\ & t_1 = -(j_3h_3 + j_4h_4),\ t_2 = -(j_1h_3 + j_2h_4). \end{aligned}$$

Note that even though x, y are shared across all images, z_1, z_2 are specific to the image pair related by the warp derivatives j_1, j_2, j_3, j_4. We observe that equation (13) is linear in z_1. We first assume $C \neq 0$, substitute z_1 from equation (13) in equation (12) and rewrite A' as:

$$\begin{aligned} A'' = {} & (c_{21}z_2 + c_{20})(d_{03}z_2^3 + d_{02}z_2^2 + d_{01}z_2 + d_{00})^2 \\ & + (c_{02}z_2^2 + c_{01}z_2 + c_{00})C^2 \\ & - z_2^2(c_{12}z_2^2 + c_{11}z_2 + c_{10})(d_{03}z_2 + d_{02})C \\ & - (c_{12}z_2^2 + c_{11}z_2 + c_{10})(d_{01}z_2 + d_{00})C. \end{aligned} \tag{14}$$

On expanding, $A'' = 0$ yields a sextic (a degree 6 polynomial) in z_2. Computing the roots of a sextic is known to be simple, stable and computationally cheap [7]. We now consider $C = 0$, in which case B' in equation (13) becomes a univariate cubic in z_2, which can be easily solved. Substituting z_2 in equation (12) then yields a quadratic equation in z_1, which can also be easily solved.

Solving. We have two possible cases depending on the nullity of C. Numerically, an explicit choice is not desirable. Hence we solve both cases and collect the solutions in a potential solution set. Selecting the optimal solution is done by using extra images, as described in the next paragraph. We now analyse the solvability of z_1, z_2 in each case. The original shape variables x, y are then found as $(x, y)^\top = \mathbf{J}_{\eta_{21}}^{-\top}(z_1, z_2)^\top$.

In the case $C = 0$, z_2 is found from a univariate cubic, and hence always has at least one real solution. Using z_2 thus obtained,

z_1 is found by solving a quadratic. Theoretically this does not guarantee a real solution for z_1.

In the case $C \neq 0$, the sextic A'' factors into a quadratic and a quartic as:

$$A'' = D^2(s_1 z_2 - s_5) + D(s_{13} z_2^2 - s_{12} z_2 - s_{11})^2 + D(s_{13} z_2^2 - s_{12} z_2 - s_{11})(s_{10} z_2^2 + s_9 z_2 + s_8) \quad (15)$$

where $D = s_{16} z_2^2 + s_{15} z_2 + s_{14}$ and:

$s_1 = 2e_2e_{11} - 2e_1e_9,\ s_5 = e_2e_{14} - e_1e_5, s_8 = 2e_3e_5 - 2e_{14}e_{10},$
$s_9 = 4e_{10}e_{11} - 4e_3e_9,\ s_{10} = 2e_2e_3 - 2e_1e_{10}, s_{11} = e_4e_5 - 2e_{14}e_9,$
$s_{12} = e_{14}e_2 - e_1e_5 + 2e_9e_{12},\ s_{13} = e_2e_{11} - e_1e_9 + e_2e_{12},$
$s_{14} = e_{13}e_5 - e_{14}e_6,\ s_{15} = 2e_{11}e_6 - e_9e_{13},\ s_{16} = e_2e_{13} - e_1e_6,$
$s_2 = s_{10}s_{11} + s_9s_{12} - s_{13}s_8,\ s_3 = s_8s_{12} + s_9s_{11},$
$s_4 = s_9s_{13} - s_{10}s_{12},\ s_6 = s_{12}^2 - 2s_{11}s_{13},\ s_7 = s_{15}^2 + 2s_{14}s_{16},$
$e_9 = v_2 - e_2t_2,\ e_{10} = u_2 - e_2t_1,\ e_{11} = j_2u_1 + j_4v_1,$
$e_{12} = j_2u_1 - j_3u_1,\ e_{13} = j_1^2 + j_2^2,\ e_{14} = j_3^2 + j_4^2.$

The discriminant of D is always non-negative. Hence, at least one real solution is always achievable for z_2. This holds even in the limit case of an infinitesimal local motion, as backed up by our experiments. Further, z_1 can be obtained by back-substituting in the linear equation (13). The system has 6 possible solutions. In most cases, we find only one or two real solutions.

3.3 Fast Solution using Resultants

Deriving the univariates. We use the theory of resultants [5], [12], which states that the resultant of two equations that bear at least one common root evaluates to zero. Given two univariate polynomials of degrees n and m respectively, $p = \sum_{f=0}^{n} p_f z^f$ and $q = \sum_{f=0}^{m} q_f z^f$, the resultant is defined as the determinant of the Sylvester matrix, an $(n+m) \times (n+m)$ matrix formed using the coefficients of p and q. For instance, with $(n, m) = (2, 1)$ the Sylvester matrix is:

$$S = \begin{pmatrix} p_2 & p_1 & p_0 \\ q_1 & q_0 & 0 \\ 0 & q_1 & q_0 \end{pmatrix}. \quad (16)$$

In our case, the Sylvester matrix S of the cubics A and B constructed to eliminate variable y is:

$$S = \begin{pmatrix} a_{03} & \sigma_5 & \sigma_3 & \sigma_1 & 0 & 0 \\ 0 & a_{03} & \sigma_5 & \sigma_3 & \sigma_1 & 0 \\ 0 & 0 & a_{03} & \sigma_5 & \sigma_3 & \sigma_1 \\ b_{03} & \sigma_6 & \sigma_4 & \sigma_2 & 0 & 0 \\ 0 & b_{03} & \sigma_6 & \sigma_4 & \sigma_2 & 0 \\ 0 & 0 & b_{03} & \sigma_6 & \sigma_4 & \sigma_2 \end{pmatrix}, \text{ where:} \quad (17)$$

$\sigma_1 = a_{30}x^3 + a_{20}x^2 + a_{10}x + a_{00}, \sigma_4 = b_{21}x^2 + b_{11}x + b_{01},$
$\sigma_2 = b_{30}x^3 + b_{20}x^2 + b_{10}x + b_{00},\ \sigma_5 = a_{02} + a_{12}x,$
$\sigma_3 = a_{21}x^2 + a_{11}x + a_{01},\ \sigma_6 = b_{02} + b_{12}x.$

We form $\det(S) = 0$ symbolically, which gives a univariate polynomial P in x of degree 9.

Solving. At runtime, the coefficients of P are computed from the coefficients of the bivariate cubics A and B, x is obtained by solving $P = 0$ and y is obtained by back-substituting x in $A = 0$ or $B = 0$. The degree 9 guarantees at least one real solution to x. On backsubstituting x in either A or B yields cubics in y which also guarantees at least one real solution to y. In practice, only 1 or 3 solutions for x and y are real. The solvability of the cubics A and B depends on the structure of the Sylvester matrix. Some coefficients of A and B may be less significant than others in some cases, which may lead to a rank-deficient Sylvester matrix (17) and thus to a wrong solution. It is therefore fundamental that we handle all possible cases of rank-deficiency of S to ensure that a reliable solution is computed. Table 1 shows the Sylvester matrix for all possible combinations of degrees of A and B. It also reports the degree of P and whether it can be symbolically factored into lower degree polynomials. In all cases, P either has an odd degree or factors into odd degree polynomials, which ensures the existence of one real root. Instead of computing the resultants online for each correspondence, which would takes around 3-4 seconds, we use the pre-computed analytical resultants for each of the possible 9 cases. Thus, for each correspondence, we look up for the relevant resultant based on the coefficients of A and B. It then only takes about $1ms$ to form the resultant. In practice, we choose the case at hand using the following steps:

1) Compute the medians λ_a and λ_b of the absolute values of the coefficients a_{kl} and b_{kl}, $k, l \in [0, 3]$ respectively.
2) Discard the coefficients a_{kl} and b_{kl} whose absolute value falls below $\mathrm{th}_a = 0.01\lambda_a$ and $\mathrm{th}_b = 0.01\lambda_b$ respectively.

Equation (10)	Equation (11)	Sylvester matrix	Degree	Factors
Quadratic	Cubic	$\begin{pmatrix} \sigma_5 & \sigma_3 & \sigma_1 & 0 & 0 \\ 0 & \sigma_5 & \sigma_3 & \sigma_1 & 0 \\ 0 & 0 & \sigma_5 & \sigma_3 & \sigma_1 \\ b_{03} & \sigma_6 & \sigma_4 & \sigma_2 & 0 \\ 0 & b_{03} & \sigma_6 & \sigma_4 & \sigma_2 \end{pmatrix}$	9	-
Linear	Cubic	$\begin{pmatrix} \sigma_3 & \sigma_1 & 0 & 0 \\ 0 & \sigma_3 & \sigma_1 & 0 \\ 0 & 0 & \sigma_3 & \sigma_1 \\ b_{03} & \sigma_6 & \sigma_4 & \sigma_2 \end{pmatrix}$	9	-
Cubic	Quadratic	$\begin{pmatrix} a_{03} & \sigma_5 & \sigma_3 & \sigma_1 & 0 \\ 0 & a_{03} & \sigma_5 & \sigma_3 & \sigma_1 \\ \sigma_6 & \sigma_4 & \sigma_2 & 0 & 0 \\ 0 & \sigma_6 & \sigma_4 & \sigma_2 & 0 \\ 0 & 0 & \sigma_6 & \sigma_4 & \sigma_2 \end{pmatrix}$	9	-
Cubic	Linear	$\begin{pmatrix} a_{03} & \sigma_5 & \sigma_3 & \sigma_1 \\ \sigma_4 & \sigma_2 & 0 & 0 \\ 0 & \sigma_4 & \sigma_2 & 0 \\ 0 & 0 & \sigma_4 & \sigma_2 \end{pmatrix}$	9	-
Quadratic	Quadratic	$\begin{pmatrix} \sigma_5 & \sigma_3 & \sigma_1 & 0 \\ 0 & \sigma_5 & \sigma_3 & \sigma_1 \\ \sigma_6 & \sigma_4 & \sigma_2 & 0 \\ 0 & \sigma_6 & \sigma_4 & \sigma_2 \end{pmatrix}$	8	5,3
Linear	Quadratic	$\begin{pmatrix} \sigma_3 & \sigma_1 & 0 \\ 0 & \sigma_3 & \sigma_1 \\ \sigma_6 & \sigma_4 & \sigma_2 \end{pmatrix}$	7	-
Quadratic	Linear	$\begin{pmatrix} \sigma_5 & \sigma_3 & \sigma_1 \\ \sigma_4 & \sigma_2 & 0 \\ 0 & \sigma_4 & \sigma_2 \end{pmatrix}$	7	-
Linear	Linear	$\begin{pmatrix} \sigma_3 & \sigma_1 \\ \sigma_4 & \sigma_2 \end{pmatrix}$	5	-

TABLE 1: The Sylvester matrix for the degenerate cases of cubics A and B. The degree of the determinant of this matrix along with the degree of its possible factors are reported.

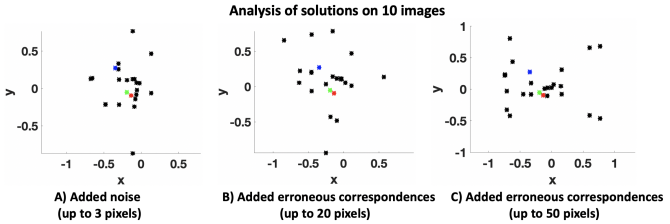


Fig. 3: Solution from [31] and our methods for three scenarios.

3.4 Algorithm

Given a set of point correspondences over M images and the η_{ji} warps between a fixed reference frame i and the other images $j \in [1, M], j \neq i$, we reconstruct the normals and depth. The normals are found independently for each correspondence:

1) *Initialize* x, y.
   a) Randomly select a few images (10% of M or 10, whichever is maximum), including the reference image i.
   b) Solve x, y for each image pair i, j using the substitution or resultant method. With the substitution method, obtain z_2 by solving equations (15) and (13) and z_1 by substituting the real solutions for z_2 in equation (13). Set x, y to $\mathbf{J}_{\eta_{21}}^{-\top}(z_1, z_2)^\top$. With the resultant method, obtain a univariate equation in x by expressing the resultant of A and B in terms of y according to table 1. Using x, obtain y by solving $A = 0$ or $B = 0$.
   c) From the x, y thus obtained, compute the sum of absolute values of A and B over the M images. Pick the solution that gives the least residual. Flag the images with a residual larger than 10 times the median over the entire image set.
2) *Refine* x, y. Refine x, y by minimising the sum of squares of A and B over the set of non-flagged images using Levenberg-Marquardt.
3) *Transfer of normals.* x, y thus obtained represent local depth derivatives in image i. According to [31], the ones on image j is obtained from (x, y), $\mathbf{J}_{\eta_{ji}}$ and $\mathbf{H}_{\eta_{ji}}$. The normal at each surface is finally obtained from the local depth derivatives and image coordinates. For example, for a point u, v in image i having x, y as local depth derivatives, the normal is given by $(x, y, 1 - ux - vy)^\top$.

Finally, as in [31], we obtain depth by integrating the normals for each image, giving M surfaces.

3.5 Comparison of [31] and our method

We illustrate the potential problems caused by the solution in [31], and how these are fixed by our method, in figure 3. The solutions to (x, y) from 10 image-pairs are shown as black dots. Ideally, all the solutions should be identical but due to noise and correspondence errors, they are different for each image-pair. The red dot represents the true solution, the blue dot represents the solution obtained by [31] and the green dot represents our solution. Figure 3A demonstrates that under good registration conditions (when the image-pair registrations are all reliable), most of the solutions for all image-pairs are close to the ground truth with only slight perturbations due to noise. In this example, we added a gaussian noise of standard deviation of 3 pixels to ground truth point correspondences to simulate good -but noisy- registration conditions. The solutions are computed for 10 image-pairs on a single correspondence but we see more than 10 solutions in the plot as the pairwise solution to the cubics yields more than one solution (typically 2 or 3 solutions). While [31] finds a unique solution by jointly minimizing the sum-of-squares of cubics over all image-pairs, which is not robust, our strategy solves for each image-pair separately and picks a statistically robust consensus. We can observe that, even in this easy case, our solution lies closer to ground truth than [31]'s. In contrast to figure 3A, figures 3B and 3C were constructed for cases where the image-pair registrations have high errors (namely, optical drift caused by matching low textured image regions or using sparse registration techniques such as SIFT, which are prone to create erroneous correspondences) or the object moves very fast (which causes motion blur). Concretely, we created erroneous correspondences by adding random errors of up to 20 and 50 pixels respectively on 20% of the point correspondences. With such errors, one can see how the solution space diversifies, which strongly affects the computation of the solution from [31], but not the proposed one.

4 Proposed Robust NRSfM Pipeline

The three main steps of our pipeline are given in the introduction. The inputs are correspondences for N images which do not need to be visible in all images. We split the images into subsets of $M \leq N$ images. We choose $M = 7$ for wide-baseline data. For short-baseline videos, we choose $M = 35$. This is because the relative motion between two images is small in videos and a larger number of images are required to obtain well-formed constraints. However in this case, we uniformly sample the subset to pick only 7 images to be treated as reference (considering all images with small relative motion would be computationally irrelevant). However, this number can be arbitrarily adapted. For the rest of the paper, $M = 7$ means 7 images each serving as a reference for wide-baseline data and 35 images with 7 uniformly sampled references for short-baseline data.

4.1 Step *(i)*: Robust Optical Flow Derivatives

The ability to reconstruct a normal field correspondence-wise at step *(ii)* requires that the correspondences are augmented with the optical flow derivatives at first and second orders. Our approach is to interpolate the correspondences by computing a Schwarp, as it was shown to provide an extremely reliable estimate of these derivatives [34]. We estimate a total of $M(M-1)$ Schwarps, for all pairs of images within each image subset. The existing Schwarp computation method [34] is not robust. However, image points have a local influence on the Schwarp's behavior, thanks to its high flexibility. We thus propose to estimate the Schwarp with [34] and flag as outliers those image points whose predicted position is significantly different from the measured position, indicating local inconsistency. We then iterate these two steps until convergence, similarly to [33]. We assume that the noise on the true image points follows a Gaussian distribution with an unknown standard deviation σ. We use the Median Absolute Deviation (MAD) to compute an estimate $\hat{\sigma} = 1.4826\,\mathrm{MAD}$ of σ at each iteration. An image point is then flagged as inlier and stored in a temporary index set $\mathcal{L}$ if its position discrepancy is lower than a threshold $3\hat{\sigma}$, allowing us to ensure that 99.7% of the true image points are kept on average. We write the M-image subset as $\mathcal{I} = [1, M]$ and the set of s points on image $t \in \mathcal{I}$ as $\{\mathbf{p}_t^1, \ldots, \mathbf{p}_t^s\}$. Our algorithm for an ordered image pair $(t, u) \in \mathcal{I}$ is initialized with $\mathcal{L} = [1, s]$:

1) Compute Schwarp η_{tu} from correspondences in $\mathcal{L}$.

2) Predict discrepancy $d_u^j = \left\| \eta_{tu}(\mathbf{p}_t^j) - \mathbf{p}_u^j \right\|_1$ for $j \in [1, s]$ and $\hat{\sigma} = 1.4826\,\mathrm{med}_{j\in[1,s]}(d_u^j)$.
3) Flag inliers as $\mathcal{L} = \{j \in [1, s] \,|\, d_u^j < 3\hat{\sigma}\}$ and estimate $\hat{\sigma}' = 1.4826\,\mathrm{med}_{j\in\mathcal{L}}(d_u^j)$.
4) Compute Schwarp η_{tu} from correspondences in $\mathcal{L}$.
5) If $|\hat{\sigma} - \hat{\sigma}'| < \delta$ return else loop to 2).

The inliers are flagged during estimation to ensure robustness but all points are kept for the next step. Convergence is assessed from the evolution of MAD, which we threshold using δ, chosen as 0.1% of the diagonal of image u. Note that all parameters in this step are fixed and found automatically.

4.2 Step *(ii)*: Robust Correspondence-wise Normal Reconstruction using Multiple References

We assume that we have a *base* correspondence-wise normal reconstruction method, which is one of the methods given in §3. Given a correspondence for three images or more from step *(i)*, this base method estimates the surface normal at all unknown 3D points of this correspondence. The base method is not robust but serves as a key building block of our robust method described directly below. We consider one correspondence at a time. The robustness principle we use is to gradually remove the images which show maximal inconsistency with the others, until we settle on a consistent image subset for the given correspondence. Concretely, we start with the complete $\mathcal{I} = [1, M]$. We thus have that initially $\texttt{size}(\mathcal{I}) = M$ and eventually $\texttt{size}(\mathcal{I}) \leq M$, indicating the inlier image points. Importantly, the base method requires one to fix a reference image. We write V_k^t the normal for image $k \in \mathcal{I}$ reconstructed using $t \in \mathcal{I}$ as reference image. For a consistent image set $\mathcal{I}$, V_k^t should be relatively independent of t. Therefore, the consistency of V_k^t for the different reference images suggests that it was reliably estimated and thus the absence of correspondence errors. We measure the inconsistency $S_{t,u}$ between two reference images with indices $t, u \in \mathcal{I}$ using the angle between the normal estimates they produce for all images as:

$$S_{t,u} = \mathrm{angle}(V_k^t, V_k^u). \tag{18}$$

The inconsistency U_t of an individual reference t is then given by marginalizing u in $S_{t,u}$ using the median, giving $U_t = \mathrm{median}_{u\in\mathcal{I},u\neq t} S_{t,u}$. Finally, the overall inconsistency G of $\mathcal{I}$ is given by the minimum of U_t as $G = \min_{t\in\mathcal{I}} U_t$. A small value indicates that all images were consistent and they all gave a similar reconstruction. The consistency of an image set is decided using a threshold ε on G. Our algorithm starts with $\mathcal{I} = [1, M]$ and iterates as:

1) Reconstruct V_k^t for $k, t \in \mathcal{I}$ using a base method in §3. Find $U_t = \mathrm{median}_{u\in\mathcal{I},u\neq t} S_{t,u}$ and $G = \min_{t\in\mathcal{I}} U_t$.
2) If $G < \varepsilon$, set $\hat{V}_k$ for $k \in \mathcal{I}$ to the most consistent estimate $\hat{V}_k = V_k^t$ with $t = \arg\min_{t\in\mathcal{I}} U_t$ and return. If $\texttt{size}(\mathcal{I}) < 5$, set $\mathcal{I} = \emptyset$ and return. Drop $t = \arg\max_{t\in\mathcal{I}} U_t$ from $\mathcal{I}$, loop to 1).

The outputs are $\mathcal{I} \subset [1, M]$, indicating the inlier image points, and $\hat{V}_k$ with $k \in \mathcal{I}$, giving the reconstructed normals. In this step, only ε needs to be set. We choose $\varepsilon = 5$ degrees.

4.3 Step *(iii)*: Robust Isometry Consistent Filtering

Isometric deformations preserve geodesic distances. Since geodesic distances are expensive to compute (they cannot be computed in closed-form for a general surface) and sensitive to noise, we approximate them with Euclidean distances, which is a fair approximation on short distances and for surfaces with mild local curvature, as shown in previous work [10], [37], [44], [45].It is highly likely that erroneously reconstructed points obtained from step *(ii)* do not comply with isometry and can be disregarded as outliers. We thus measure the consistency of inter-point distances across the reconstructed point clouds to detect outliers. However, step *(ii)* produces up-to-scale point clouds. In order to compare distances, we thus need to first recover the relative scales. We use a Nearest-Neighbor Graph (NNG) as in [10], using the rationale that if two points are close in 3D they must also be close in the images. We store the indices of the r nearest neighbors of each correspondence of index $k \in [1, s]$ in a vector R_k of size r, where s is the number of correspondences. We set $r = 20$. This step is also devoid of parameters to be set manually.

4.3.1 Robust relative scale estimation

Without loss of generality, we set the relative scale for image 1 as $\alpha_1 = 1$. We then estimate α_i, $i \in [2, M]$. We first define distance vectors P_j^i of length r computed from the reconstructed shape $i \in [2, M]$ for the neighborhood of each correspondence $j \in [1, s]$: $P_j^i(l)$ contains the 3D Euclidean distance between the points of indices j and $l \in R_j$. Each vector P_j^i contributes r equations towards scale estimation. A robust estimate for α_i is then the median of distance ratios, given by:

$$\alpha_i = \mathrm{med}_{j\in[1,s],l\in[1,r]} \frac{P_j^i(l)}{P_j^1(l)}, \tag{19}$$

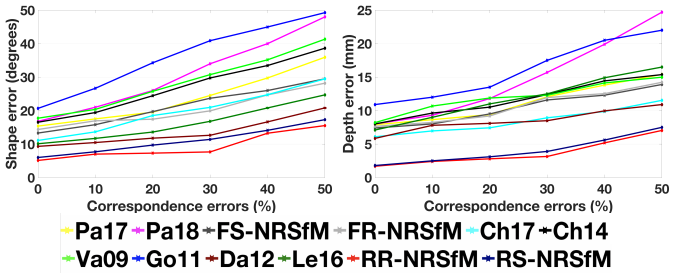
where we only run over the indices of the correspondences visible in both images 1 and i. If image 1 does not contain sufficient correspondences with all other images, we use multiple reference images and propagate the estimated scales. We then rescale the reconstructed shapes using α_i.

4.3.2 Robust isometry-based inlier/outlier classification

We rescale the distance vectors P_j^i as $Q_j^i = \alpha_i P_j^i$. In the absence of correspondence errors, these should be identical over $i \in [1, M]$, due to the temporal consistency of isometry. We thus compute an isometry consistency measure for each point as the proportion of neighbors and images for which the distance variation is lower than a threshold, which we chose as 10% of the average distances in the neighborhood. This tolerance is important to model noise and the fact that the geodesic distance was approximated by the Euclidean distance, which underestimates it. If the computed proportion is more than 50%, then the point is compatible with over half of its neighbors from over half of the images where they are visible. This suggests than the point should be classified as an inlier and as an outlier otherwise.

5 Experimental Results

Datasets and methods. We used one synthetic and four real existing datasets showing various objects deforming isometrically. We introduce a video of a use case example depicting a realistic scenario of two objects, a paper and a cushion, recorded using Kinect. We also introduce a footage downloaded from Youtube showing a spotted eagle-ray (an endangered species of fish). We denote the fast local normal estimator of §3.2 as **FS-NRSfM** and the one of §3.3 as **FR-NRSfM**. It is the baseline reconstruction method for our robust pipeline that has three steps, as described in §4. Step *(i)* in §4.1 identifies potential mismatches and is denoted



Erroneous data (in %) TNR	Error (in pixels) 1-25	26-50	51-75	76-100	ROC analysis TPR	FNR	TNR	FPR
10	52	63	66	95	96	4	95	5
20	48	75	78	94	96	4	94	6
30	39	77	83	92	95	5	92	8
40	27	82	88	89	94	6	89	11
50	21	86	92	88	92	8	88	12

Missing data	M=5 Es	M=5 Ed	M=7 Es	M=7 Ed	M=10 Es	M=10 Ed
0%	13.5	6.6	13.1	6.5	12.8	6.5
50%	17.3	7.1	13.7	6.8	13.2	6.5

Fig. 4: The *Cylinder* dataset. (top) Shape and depth errors (400 points, 7 images, 1 pixel noise, correspondence errors between $0-50\%$ at 100 pixels). (middle) Analysis of **RR-NRSfM**: TNR shown across perturbation amplitudes and ROC analysis in the error range of $76-100$ pixels. (bottom) Shape (Es) and depth (Ed) error for **FR-NRSfM** with various sizes of M with/without the influence of missing data. The mean computation time for M as 5, 7 and 10 are 2.5s, 3s and 5s respectively. The performance becomes stable for $M \geq 7$.

as **MAD**. Step *(ii)* in §4.2 performing reconstruction from multiple reference views using **FS-NRSfM** is denoted as **MVS** and using **FR-NRSfM** as **MVR**. Step *(iii)* in §4.3 identifies 3D points as inliers and outliers based on the isometric constraint and is denoted as **OR**. Our proposed robust NRSfM method is denoted by **RS-NRSfM**, which is **MAD+MVS+OR** or **RR-NRSfM**, which is **MAD+MVR+OR**. We also show results of all possible combinations of these steps in order to quantify their importance. We compare our methods with the state-of-the-art methods designed for short-baseline data, **Go11** [21], **Da12** [13] and **Le16** [26], wide-baseline data, **Ch17** [10], **Ch14** [9], **Va09** [44] and **Pa17** [31], which deals with both short and wide-baseline data. We also compare with **Pa18** [29], which solves NRSfM with uncalibrated cameras on both short and wide-baseline data. In order to keep the comparison fair we however use the known intrinsics in **Pa18**.

RS-NRSfM and **RR-NRSfM** are the only methods that perform inlier-outlier classification. We identify the true correspondences as positives P and the false ones as negatives N. TP and TN represent the image points which are correctly classified as true and false correspondences respectively. Similarly, FP and FN represent the image points which are incorrectly classified as true and false correspondences respectively. We compute the true positive rate TPR = TP/P, true negative rate TNR = TN/N, false positive rate FPR = 1-TNR and false negative rate FNR = 1-TPR.

Measured Errors. We measure the depth error (RMSE between reconstructed and ground truth 3D points in mm) and the shape error (RMSE between reconstructed and ground truth normals in degrees). The depth error is position dependent and reflects the extrinsic quality of the reconstruction. It depends on the object size. A depth error lower than 5% of the object size indicates a reasonable reconstruction. However, a flawed reconstruction may also yield a low depth error by lying in the vicinity of the ground truth. The shape error thus complements the depth error by being position independent and reflecting the intrinsic quality of the reconstruction. A reasonable reconstruction is obtained for a shape error lower than 20 degrees. Overall, a reconstruction is successful if depth and shape errors are below 5% of the object size and 20 degrees respectively.

5.1 Synthetic Dataset

We used the *Cylinder* dataset [31]. We generated 1920×1080 images of a cylindrical surface of length 20 cm deforming isometrically. We compared the methods on $N = 7$ images with 400 correspondences. We introduced a Gaussian noise of 1 pixel standard deviation and between $0-50\%$ correspondence errors by applying a uniformly distributed perturbation with 100 pixels standard deviation. The results are shown in figure 4 (top). All methods are sensitive to correspondence errors and degrade when the number of correspondence errors increases. Our robust methods **RS-NRSfM** and **RR-NRSfM** perform consistently well, even with correspondence errors: their shape error is consistently lower than 15 degrees and their depth error lower than 10 mm, which represents 5% of the object size. Note that although [31] is based on the same system of equations than ours, it finds a non-robust solution and uses only one of the images as reference. In contrast, our robust method solves for each image-pair separately and uses several reference image candidates, picking a solution as the best consensus to a set of coherent image-pairs. This contributes to a better solution overall as, even in the absence of mismatches, the optical flow computation contains errors due to lack of features in some areas of the object or challenging imaging conditions, such as motion blur. We discuss this in more details in the upcoming section. **RR-NRSfM** performs slightly better than **RS-NRSfM** as its underlying local normal estimator **FR-NRSfM** performs slightly better than **FS-NRSfM**. Our local normal estimators **FR-NRSfM** and **FS-NRSfM** show a similar performance, slightly better than **Pa17**. However, they all break at about 20% correspondence errors. **Pa18** shows a slightly worse performance than **Pa17** in the range of low correspondences errors but it breaks much earlier than **Pa17**, at about only 10%. This shows that our robust pipeline brings a substantial improvement compared with the local estimators **FS-NRSfM**, **FR-NRSfM**, **Pa18** and **Pa17**. **Ch17** performs slightly better than **FR-NRSfM** and **FS-NRSfM** but significantly worse than **RS-NRSfM** and **RR-NRSfM**. **Da12** and **Le16** show a good performance and tolerance to correspondence errors up to about 30%. **Go11** has a poor performance, even in the absence of correspondence errors. **Ch14** and **Va09** have a marginally good performance and degrade quickly, not tolerating any correspondence errors. Overall, we observe that our robust pipelines **RR-NRSfM** and **RS-NRSfM** bring strong benefits.

Local normal estimators: Pa17, Pa18, FS-NRSfM and FR-NRSfM. Pa17 estimates local normal by minimizing the sum of squares of bivariate cubics using an computationally expensive solver. **FS-NRSfM** simplifies this problem by reducing the bivariate cubics into sextic univariate using substitution. It is also able to identify and remove the images that do not yield good constraints. **Pa18** and **FR-NRSfM** both use resultants to

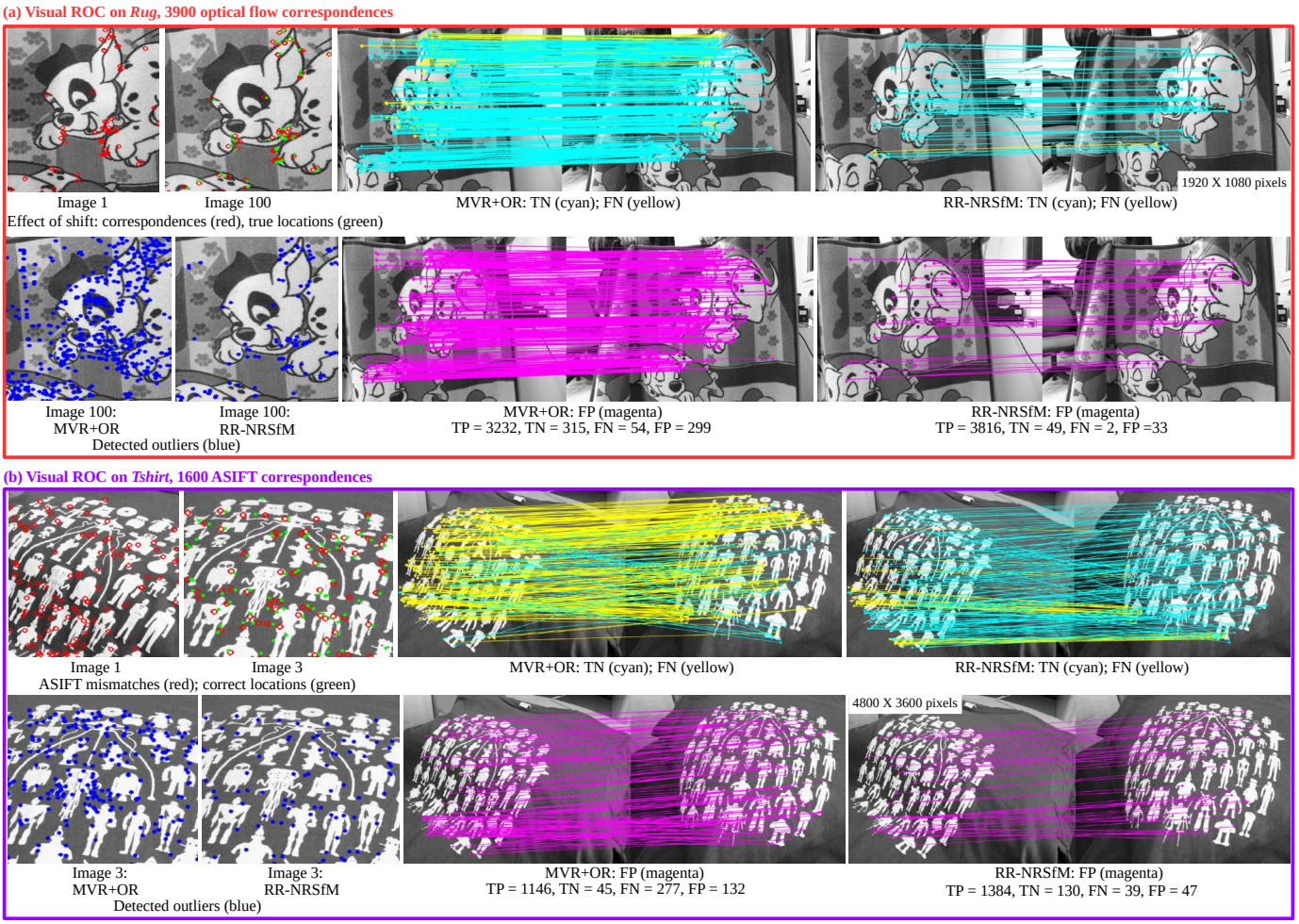


Fig. 5: Visual ROC for *Rug* and *Tshirt*. **RR-NRSfM** performs **MAD+MVR+OR**. The use of **MAD** and **OR** drastically reduces the falsely classified 3D points. In *Rug*, **OR** identifies less points when used with **MAD** while in *Tshirt*, it identifies more incorrect 3D points.

segregate variables. **Pa18** uses resultants to isolate focal length from two other variables, which represent the local shape, and finds an optimal solution from the univariate constraints. It then uses this solution to obtain univariate polynomials for the local shape variables which are solved by minimizing the respective sum-of-squares. Even when used with a known focal length, **Pa18** solves degree 18 polynomials, making it sensitive to noise and correspondence errors. In contrast, **FR-NRSfM** uses resultants to solve for local shape. It pre-computes them analytically and finds a solution to univariate polynomials of degree 9. Like **FS-NRSfM**, it is also capable of identifying and removing images that yield outlying constraints.

Analysis of RS-NRSfM and RR-NRSfM. We used the same synthetic dataset to analyze the ability of our robust methods to detect correspondence errors. They perform very much alike in terms of detecting outliers. We measured TNR for a varying correspondence error rate and magnitude of the point perturbation causing the errors. We report the TNR for **RR-NRSfM** in figure 4 (middle). The statistics for **RS-NRSfM** are similar and we therefore do not show them. As expected, we observe that correspondence errors caused by a large perturbation (76-100 pixels) are very well detected. TNR naturally degrades with decreasing perturbation magnitude. This confirms the intuition that a large error is easier to detect than a small error, which can more easily be identified as noise by a deformation model. ROC analysis shows very successful results, as TPR and TNR are mostly 90% or higher. The correspondences errors larger than 25 pixels (spanning up to 50% data) can be detected by our proposed method with a very high accuracy, above 80%. However, in case of 60% errors, the performance sharply degrades to 53% and reduces to 20% for the data with 80% errors.

Parameter tuning. Importantly, all parameters were fixed to a single value for all experiments.We use Schwarps to obtain first and second order derivatives, which requires the schwarzian parameter to be tuned. We have fixed it to 1e-3, as suggested by [31] and it does not need to be changed. Both **FS-NRSfM** and **FR-NRSfM** solve univariate equations to initialize the system. In order to ensure a good initial solution, we should solve these univariates multiple times and pick the solution which is most suitable. We discard image pairs whose residual (on cubics A, B) is 10 times higher than the median over the entire image-set. This threshold is approximately twice the optimal threshold for the gaussian noise distribution and was empirically chosen.We found that a good initialization can be achieved by solving with $10 - 15$ images. An important parameter is M, the number of images in a subset on which our robust pipeline operates. Our methods **FS-**

NRSfM and **FR-NRSfM** can reconstruct from only 3 images. Therefore, M should be 3 at minimum but the solution with just 3 images can be highly affected by noise. In addition, the possibility of missing data should also be taken into account. Therefore, we fix $M = 7$. We also made experiments with $M = 5$ and $M = 10$. The computation time is 2.5s, 3s and 5s for the progressive values of M. With only 5 images, the reconstruction quality is similar to 7 images, with a lower computation time. However in case of missing data which can be as high as 50% in realistic scenarios, using only 5 images does not ensure constraints on enough points and the reconstruction accuracy drops significantly, see figure 4 (bottom). With 7 images, the performance stabilizes with a small overhead in computation time. With 10 images, the accuracy is slightly improved but the computational overhead is not worth the increase in performance.

Our robust NRSfM pipeline has 3 steps described in §4.1, §4.2 and §4.3 and each of them has parameters that need to be tuned. Step *(i)* computes optical flow derivatives and has two important parameters: σ and δ. σ decides how many points are to be discarded at each step and it is automatically chosen using a statistical criterion. δ decides when the convergence is achieved so that we do not need to remove points any further. It is image size dependent and we thus fix it to 0.1% of the image diagonal. Step *(ii)* requires ε. Ideally the multiple normals reconstructed using the different reference images should be close but it is not the case in realistic scenarios. Therefore, we allow the closeness threshold $\varepsilon = 5$ degrees. This means that we consider an image-set as genuine if the median of the angles between reconstructed normals with different reference images is 5 degrees. In our experiments, we found that this is a generous limit to allow the reconstruction from image correspondences contaminated with Gaussian noise of up to 10 pixels. Step *(iii)* requires r, which decides the neighborhood size for computing the isometric consistency. Since we deal with semi-dense data, we need r to be small enough for the Euclidean approximation of geodesic distances to be valid. Also, we need to consider enough points to have a reliable estimate. We therefore set $r = 20$ neighbors.

Computation time. We used a desktop with an i5 CPU and 8GB RAM. Our base normal estimators **FS-NRSfM** and **FR-NRSfM** take about 7-10 ms to solve the reconstruction equations while **Pa17** takes about 1.5 s for the same data. Recall that the reconstruction equations for one correspondence depend on two variables for any number of images and are very fast to assemble (forming there coefficients takes much less than 10 ms). **FS-NRSfM** and **FR-NRSfM** are thus more than 100 orders of magnitude faster than **Pa17**. **Pa18** takes about 200 s to compute the resultant, 80 s to obtain the focal length and 20 ms to find the local shape. The MATLAB implementation of our robust pipeline **RS-NRSfM** and **RR-NRSfM** runs in about $100 - 120$ s for 10 images without any code optimization and can presumably be made way faster in a parallelized implementation, as each subset of images and each correspondence could be treated independently at some level. The computation time for other methods is comparable to **RS-NRSfM** and **RR-NRSfM** for a small number of images. However, most of them have a cubic complexity while **RS-NRSfM** and **RR-NRSfM** has a quadratic complexity, giving them a huge advantage for large sets of images.

5.2 Real Datasets

***Without* Correspondence Errors: *Paper* [43], *manual-Tshirt* and *manual-Hulk* [9].** The *Paper* dataset has 190 short-baseline images of a 35 cm wide deforming paper with 1500 error-free SIFT correspondences. The 20 cm long *Tshirt* and 30 cm long *Hulk* datasets consist of 10 and 21 wide-baseline images with 85 and 122 error-free manually-set correspondences. It may seem surprising that we test our robust method on data without correspondence errors. This however represents an important sanity check. Non-robust methods use a strong prior, which is that all the correspondences are correct. They thus just solve for the reconstruction. In contrast, robust methods do not use this prior and also classify the correspondences as inliers and outliers. Therefore, for data without correspondence errors, robust methods may underperform non-robust methods. A very important question is to which extent this happens to our robust methods and how close the detected outlier rate is to 0%.

Figure 6 (a) shows the results. Since these datasets are error-free, all methods perform well except **Go11**, **Da12** and **Le16**. They are designed for short-baseline data only, therefore they fail on *manual-Tshirt* and *manual-Hulk*. **Pa17** shows a good performance. We could not manage to run **Va09** and **Ch14** on *Paper* because of the large number of images. **Ch14** shows a decent performance on *manual-Tshirt* but fails on *manual-Hulk*. **Va09** does not perform well on these datasets. **Ch17** shows the best performance. It works extremely well for *manual-Tshirt* and *manual-Hulk* as it requires wide-baseline data with very high perspective images which are the characteristics of these datasets. **Pa17** performs well, similarly to **Ch17**. **FS-NRSfM** and **FR-NRSfM** improve the results and perform even closer to **Ch17**, **FR-NRSfM** being slightly better than **FS-NRSfM**. **Pa18** performs worse than **Pa17**.

The three steps of our robust pipeline, **MAD**, **MVS/MVR** and **OR** eliminate points from input. This is why we see that the performance of **FS-NRSfM** and **FR-NRSfM** is either slightly degraded or only insignificantly improved when combined with **MAD** and **MV**. However, **OR** always improves the results as it removes 3D points from the reconstruction. It removes almost 1% 3D points when used without **MAD**. On the other hand, when used with **MAD**, it slightly improves the results while removing only 0.1% of the 3D points on the entire data. Hence, **MAD** prevents **OR** from removing a large number of 3D points. Overall, **RS-NRSfM** and **RR-NRSfM** both pass the sanity check of error-free correspondences: they obtain the best performance or are very close to the best performing non-robust method, while ruling out only 0.1% of the data as outliers.

***With* Correspondence Errors: *Rug* [31], *ASIFT-Tshirt* and *ASIFT-Hulk*.** The *Rug* dataset has 159 short-baseline images of a 1 m wide deforming carpet with 3900 correspondences obtained from optical flow [19], containing many correspondence errors of small amplitude caused by drift. We defined the ground truth for the correspondences manually and found that at least one-third of the images that appear towards the end of this video are affected by the drift. Many of these affected images contain up to 70% of correspondences affected by an average drift of 20 pixels. *ASIFT-Tshirt* and *ASIFT-Hulk* are formed of the same images as the original datasets [9] but with correspondences obtained with ASIFT [28]. These correspondences may contain large errors because of the weakly discriminative local texture, see figure 5 (b). Also, there is approximately 95% missing data in these correspondences.

Rug is impacted by optical drift. Figure 6 (a) shows that **Go11** has large errors. **Da12** and **Le16** perform slightly better than **Go11**. We could not manage to run **Va09** and **Ch14** because of the large number of images. **Ch17** and **Pa17** show a similar performance.

	Without correspondence errors						*With* correspondence errors					
	Paper		*manual- Tshirt*		*manual- Hulk*		*Rug*		*ASIFT- Tshirt*		*ASIFT- Hulk*	
	Ed	*Es*	*Ed*	*Es*	*Ed*	*Es*	*Ed*	*Es*	*Ed*	*Es*	*Ed*	*Es*
Go11	28.8	18.7	91.2	60.4	86.7	56.0	99.8	21.4	38.4	59.0	33.3	38.8
Da12	22.1	20.8	25.3	25.4	21.6	29.2	84.3	15.7	15.1	28.2	42.2	42.2
Le16	21.3	21.6	21.1	18.9	17.4	24.2	81.2	16.9	18.7	35.0	14.2	27.3
Ch14	-	-	12.6	13.4	19.4	32.0	-	-	17.8	38.0	21.9	27.4
Va09	-	-	14.6	27.7	18.3	27.8	-	-	XX	XX	XX	XX
Ch17	5.4	6.8	5.5	11.5	5.3	14.0	63.4	17.5	XX	XX	XX	XX
Pa17	7.2	8.6	8.1	17.2	4.8	17.4	54.4	18.2	24.3	34.0	19.8	28.4
Pa18	10.2	11.2	8.9	22.8	6.8	19.4	68.3	24.1	31.2	43.4	36.4	42.9
FS-NRSfM	5.3	8.0	7.1	17.2	4.2	13.9	43.4	17.5	24.0	30.4	12.9	24.5
RS-NRSfM	4.6	9.6	6.8	14.5	4.3	13.0	31.5	15.4	4.3	12.3	4.9	14.4
FR-NRSfM	5.2	7.1	7.1	14.9	4.3	13.6	44.5	17.8	24.0	30.4	12.9	24.5
RR-NRSfM	4.7	9.7	6.7	11.9	4.1	13.1	31.2	15.1	4.2	11.2	4.7	12.4

a) Summary of compared methods

	With correspondence errors					
	Rug		*ASIFT- Tshirt*		*ASIFT- Hulk*	
	Ed	*Es*	*Ed*	*Es*	*Ed*	*Es*
FR-NRSfM	44.5	17.8	24.0	30.4	12.9	24.5
FR-NRSfM+OR	42.7	15.2	9.5	20.8	10.1	18.7
MAD+FR-NRSfM	46.4	18.4	13.1	25.8	10.8	21.9
MAD+FR-NRSfM+OR	40.2	17.8	6.3	20.1	7.1	20.1
MVR	41.3	15.5	4.5	13.9	6.9	13.8
MVR+OR	39.8	14.0	4.3	13.5	6.2	12.9
MAD+MVR	38.3	15.9	4.3	13.7	4.9	13.6
RR-NRSfM	31.2	15.1	4.2	11.2	4.7	12.4

b) Ablation study

		TPR	FNR	TNR	FPR
ASIFT-Tshirt	**FR-NRSfM+OR**	81	19	25	75
	MAD+FR-NRSfM+OR	96	4	73	27
	MVR+OR	83	17	27	73
	RR-NRSfM	97	3	75	25
ASIFT-Hulk	**FR-NRSfM+OR**	80	20	27	73
	MAD+FR-NRSfM+OR	97	3	77	23
	MVR+OR	82	18	31	69
	RR-NRSfM	98	2	79	21
Rug	**FR-NRSfM+OR**	98	2	48	52
	MAD+FR-NRSfM+OR	99	1	60	40
	MVR+OR	98	2	51	49
	RR-NRSfM	99	1	63	37

c) ROC analysis

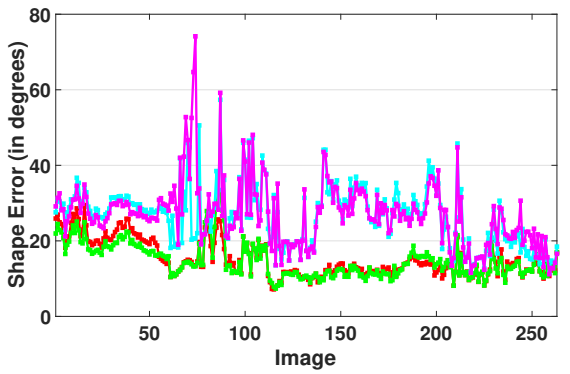
Fig. 6: (a) Summary of methods. For each dataset, the depth (Ed) and shape (Es) errors are reported. '–' represents the methods that did not complete in 24 hours. 'XX' represents the methods which failed due to missing data. (b) Ablation study and (c) ROC analysis of our robust pipeline using **FR-NRSfM** as a base method.

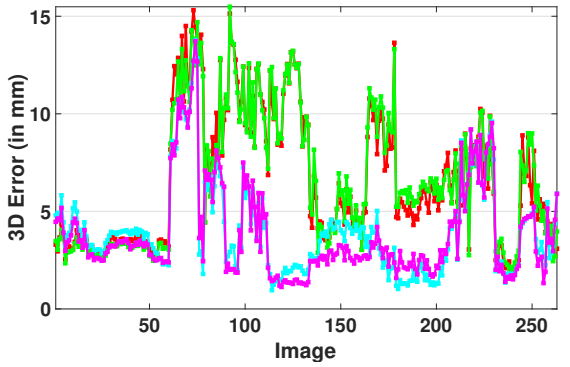
Pa18 performs worse than **Pa17**. It is very close to **Go11**. For *ASIFT-Tshirt* and *ASIFT-Hulk*, the performance of all state-of-the-art methods degrade significantly. **Ch17**, which showed the best performance with manual correspondences, cannot handle such large amount of missing data. It does not draw enough constraints and thus fails to perform reconstruction. **Va09** also cannot cope with such a large amount of missing data and fails. **Pa17**, **Pa18** and **Ch17** do not perform well. **FS-NRSfM** and **FR-NRSfM** also do not show a good performance, being only slightly better than **Pa17** and **Ch17**.

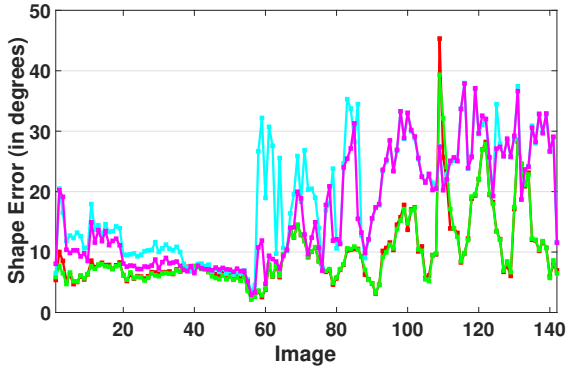
In *Rug*, **MVS** and **MVR** perform better than their baselines, **FS-NRSfM** and **FR-NRSfM**. Both methods perform slightly better when combined with **MAD** and **OR**. In *ASIFT-Tshirt* and *ASIFT-Hulk*, the performances of **FS-NRSfM** and **FR-NRSfM** improve when combined with **MAD** and **OR**, however it still remains either unsatisfactory or only marginally satisfactory. **MVS** and **MVR** shows significantly better results, which are further improved with the use of **MAD** and **OR**. While the use of **OR** alone on these three datasets sacrifices almost 15% 3D points on the entire dataset, the use of **OR** along with **MAD** shows similar or slightly better results and removes fewer than 0.1% 3D points. Figure 6 (b) shows the ablation study and 6 (c) shows the ROC analysis of our robust pipeline with **FR-NRSfM** as the base method. The results are very similar with **FS-NRSfM** and therefore, we did not report them. Figure 5 shows the distribution of mismatches on one of the images in *Rug* and *ASIFT-Tshirt* and the distribution of the detected outliers with and without **MAD**. In *Rug*, most of false correspondences lie in the low error range of 10-20 pixels. Figure 5 (a) shows that without **MAD**, **OR** does remove a larger number of false 3D points but at the cost of losing a large number of 3D points which is not preferable. However in *ASIFT-Tshirt*, correspondence errors are much larger. Figure 5 (b) shows that **MAD** correctly classifies a larger number of false 3D points and drastically reduces the falsely classified 3D points. Overall, we observe that **RR-NRSfM** and **RS-NRSfM** significantly improve the results.

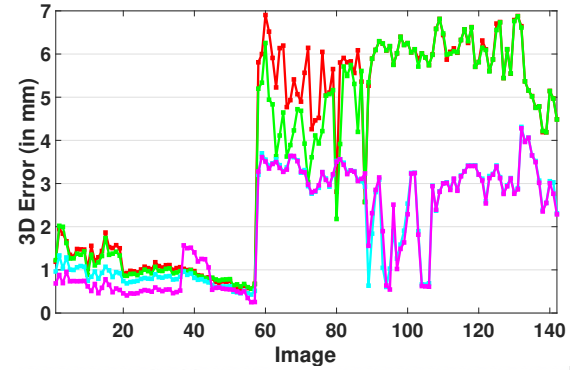
To summarize, our robust methods perform significantly better than other methods in datasets with errors, whereas they are very close to the best performing method on datasets without errors. We found that each step in our robust NRSfM pipeline is important. **MVR/MVS** allows us to reconstruct from the best ordered set of images while **OR** combined with **MAD** achieves an intelligent detection of outliers which is key to achieve robustness. Both **RS-NRSfM** and **RR-NRSfM** show a similar performance. **RS-NRSfM** is slightly computationally cheaper than **RR-NRSfM** as it does not have to compute resultant to create the univariate polynomial. On the other hand, **RR-NRSfM** is slightly more accurate as it prunes insignificant coefficients of the equations in order to compute the resultants which makes the equations well conditioned.

NRSfM Use-case. We introduce a video of two isometrically deforming objects, a $40cm$ long cushion and a $20cm$ paper sheet, depicting a realistic interaction of objects in the real world as shown in figure 7. The ground truth is obtained using Kinect. The user picks the two objects, one by one, and deforms them in front of the camera. The user provides as input an object bounding box









	Cushion				Paper			
	FR	**FS**	**RR**	**RS**	**FR**	**FS**	**RR**	**RS**
Es	26.6	27.0	15.1	14.7	18.5	16.5	9.8	9.8
Ed	6.7	6.7	4.0	3.8	3.9	3.6	2.0	2.0

Fig. 7: Shape and depth errors for the cushion (top) and the paper (bottom) in the NRSfM use-case. The table summarises the mean shape errors (Es) and mean depth errors (Ed) for **FR-NRSFM** (**FR**), **FS-NRSFM** (**FS**), **RR-NRSFM** (**RR**) and **RS-NRSFM** (**RS**) over the entire data.

in one frame from which we start tracking it using ASIFT. While the user is picking up and dropping an object, they may suffer from occlusions and motion blur. There are frames in which none or both of the objects are visible. At some point, the user keeps the object still which introduces degeneracies. These conditions are avoided in the usual datasets used to evaluate NRSfM. However, they are extremely important to deal with as capturing a deforming object and avoiding these conditions is unrealistic and impractical. **Go11**, **Da12**, **Le16**, **Ch17**, **Ch14**, **Va09**, **Pa18** and **Pa17** failed. Figure 7 shows that **FR-NRSfM** and **FS-NRSfM** did not perform well for both objects. **RR-NRSfM** and **RS-NRSfM** performed better than **FR-NRSfM** and **FS-NRSfM**, with **RR-NRSfM** being slightly better than **RS-NRSfM**. The spikes in the graphs of figure 7 for both objects arise due to the object remaining still, occlusions and motion blur as the user picks up or puts down the objects. Our robust methods **RR-NRSfM** and **RS-NRSfM** successfully reconstruct all frames without failing even if the viewing conditions are not favorable and re-establish accuracy once the object is viewed in better conditions. Figure 8 shows the reconstruction of the Cushion and Paper using **RR-NRSfM** on the use-case sequence. The images are uniformly sampled images from the entire sequence.

Eagle-ray Footage. We downloaded a low resolution 360×480 video from Youtube. We computed the camera intrinsics using Structure-from-Motion [1] with points in the background and correspondences on the deformable eagle-ray using optical flow [38]. **Go11**, **Da12**, **Le16**, **Ch17**, **Ch14**, **Va09**, **Pa18** and **Pa17** failed. **FR-NRSfM** and **FS-NRSfM** did not perform well as most correspondences are short-ranged. **RR-NRSfM** and **RS-NRSfM** perform similarly well, some of the reconstructions using **RR-NRSfM** are shown in figure 9.

Discussion. The first step in our robust NRSfM pipeline is **MAD** which identifies the potential mismatches between image correspondences. [42] is one of the existing methods that performs an outlier-rejection on image correspondences. We replaced **MAD** with this method in our robust NRSfM pipeline and reconstructed the ASIFT-Tshirt and ASIFT-Hulk datasets. In doing so, the mean depth error on these datasets dropped slightly from 8.3 mm and 7.5 mm to 6.7 mm and 6.2 mm respectively. However, [42] takes around 500 ms per image to identify mismatches whereas **MAD** does so in only 150-200 ms. Therefore we did not use [42] in further experiments.

6 Conclusions

We have presented the first entirely statistically robust NRSfM pipeline for near-isometric objects. It computes the local optical flow at each correspondence in the images, reconstructs the 3D point clouds up-to-scale and performs isometry consistent rescaling. It identifies inliers and outliers and thus deals with correspondence errors effectively. We have also introduced two new fast correspondence-wise solutions to NRSfM that estimate the normals locally and guarantee a real solution. Experimental results on synthetic and real datasets showed that our robust methods outperform the existing ones in both accuracy and robustness. They are thus the first truly robust NRSfM methods, in the sense of their ability to reject false correspondences. With them, we were able to drop the manual error-free correspondences of a public, wide-baseline dataset, and to use instead automatically established correspondences using an off-the-shelf ASIFT implementation. This is a significant step forward in making NRSfM able to deal

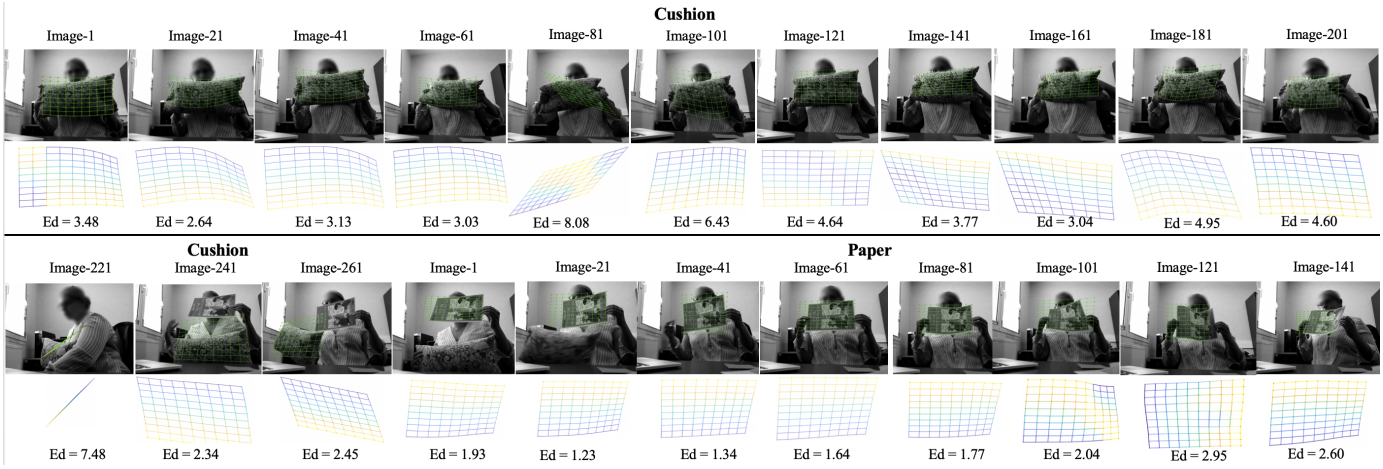


Fig. 8: Reconstruction of Cushion and Paper from the uniformly sampled NRSfM use-case sequence using **RR-NRSfM**. A grid computed from reconstructed points is shown to display the geometry of the 3D object. The color of the grid points represent the relative depth of the grid points, blue being closest to the camera and yellow being the farthest. Ed represents the mean depth error measured in mm.

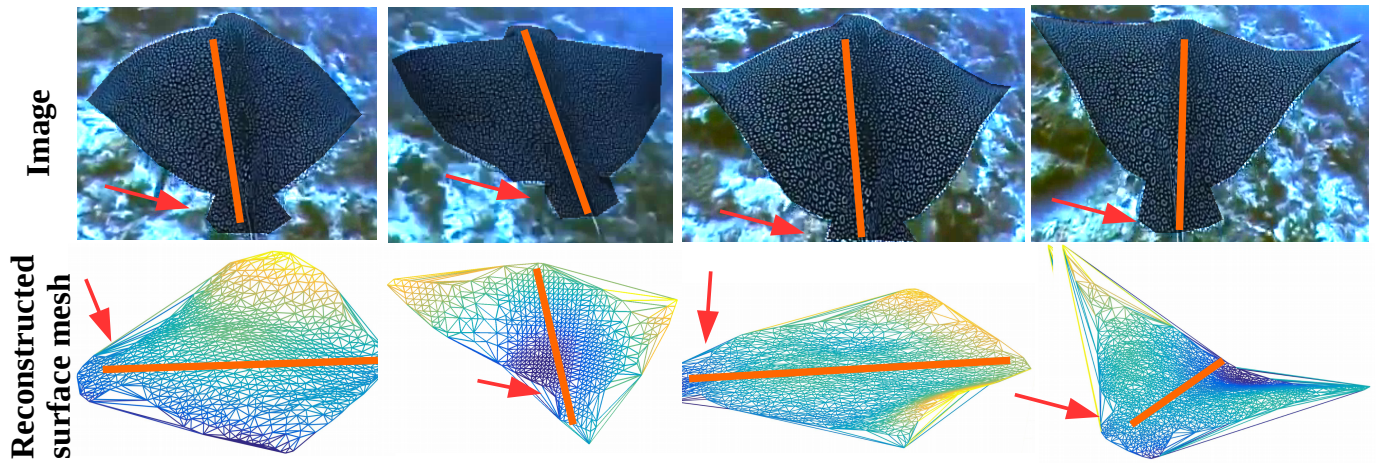


Fig. 9: The eagle-ray footage with **RR-NRSfM** reconstruction. The red arrow indicates the tail and the orange line indicates the spine of the eagle-ray.

with realistic datasets in full autonomy. In future work, we plan to explore the application of our method incrementally in realtime.

REFERENCES

[1] Agisoft Photoscan 1.0.4, 2014.

[2] Brunet, F. and Gay-Bellile, V. and Bartoli, A. and Navab, N. and Malgouyres R. Feature-driven direct non-rigid image registration. *International Journal of Computer Vision*, 93(1):33–52, 2011.

[3] A. Agudo, F. Moreno-Noguer, B. Calvo, and J. 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 *NIPS*, 2009.

[5] A. G. Akritas. Sylvester's forgotten form of the resultant. *Fibonacci Quarterly*, 31:325–332, 1993.

[6] A. Bartoli, Y. Gérard, F. Chadebecq, T. Collins, and D. Pizarro. Shape-from-template. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2099–2118, 2015.

[7] S. Basu, R. Pollack, and M. Roy. *Algorithms in Real Algebraic Geometry*, volume 10. Springer, 2016.

[8] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *CVPR*, 2000.

[9] A. Chhatkuli, D. Pizarro, and A. Bartoli. Non-rigid shape-from-motion for isometric surfaces using infinitesimal planarity. In *BMVC*, 2014.

[10] A. Chhatkuli, D. Pizarro, T. Collins, and A. Bartoli. Inextensible Non-Rigid Structure-from-Motion by Second-Order Cone Programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2428 – 2441, 2018.

[11] T. Collins and A. Bartoli. Realtime Shape-from-Template: System and Applications. In *ISMAR*, 2015.

[12] D. A. Cox, J. Little, and D. O'shea. *Using algebraic geometry*, volume 185. Springer Science & Business Media, 2006.

[13] 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.

[14] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[15] A. Del Bue, F. Smeraldi, and L. Agapito. Non-rigid structure from motion using non- parametric tracking and non-linear optimization. In *CVPR*, 2004.

[16] M. P. Do Carmo. *Differential geometry of curves and surfaces: revised and updated second edition*. Courier Dover Publications, 2016.

[17] J. Fayad, A. Del Bue, L. Agapito, and P. M. Aguiar. Non-rigid structure from motion using quadratic deformation models. In *BMVC*, 2009.

[18] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Computer Vision, Graphics and Image Processing*, 24(6):381–395, 1981.

[19] R. Garg, A. Roussos, and L. Agapito. A variational approach to video registration with subspace constraints. *International Journal of Computer*

Vision, 104(3):286–314, 2013.
[20] V. Golyanik, T. Fetzer, and D. Stricker. Accurate 3d reconstruction of dynamic scenes from monocular image sequences with severe occlusions. In *WACV*, 2017.
[21] P. Gotardo and A. Martinez. Kernel non-rigid structure from motion. In *ICCV*, 2011.
[22] R. Hartley and R. Vidal. Perspective nonrigid shape and motion recovery. In *ECCV*. 2008.
[23] D. Henrion and J. B. Lasserre. GloptiPoly: Global optimization over polynomials with MATLAB and SeDuMi. *ACM Transactions on Mathematical Software*, 29(2):165–194, 2003.
[24] 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 *ICCV*, 2017.
[25] Z. Kukelova, M. Bujnak, and T. Pajdla. Polynomial Eigenvalue Solutions to Minimal Problems in Computer Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:1381–1393, 2012.
[26] M. Lee, J. Cho, and S. Oh. Consensus of non-rigid reconstructions. In *CVPR*, 2016.
[27] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
[28] J.-M. Morel and G. Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM journal on imaging sciences*, 2(2):438–469, 2009.
[29] S. Parashar, A. Bartoli, and D. Pizarro. Self-calibrating isometric non-rigid structure-from-motion. In *ECCV*, 2018.
[30] S. Parashar, D. Pizarro, and A. Bartoli. Isometric non-rigid shape-from-motion in linear time. In *CVPR*, 2016.
[31] 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, 2018.
[32] S. Parashar, D. Pizarro, and A. Bartoli. Local deformable 3d reconstruction with cartan's connections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.
[33] J. Pilet, V. Lepetit, and P. Fua. Fast non-rigid surface detection, registration and realistic augmentation. *International Journal of Computer Vision*, 76(2):109–122, February 2008.
[34] D. Pizarro, R. Khan, and A. Bartoli. Schwarps: Locally projective image warps based on 2D schwarzian derivatives. *International Journal of Computer Vision*, 119(2):93–109, 2016.
[35] T. Probst, D. Pani Paudel, A. Chhatkuli, and L. Van Gool. Incremental non-rigid structure-from-motion with unknown focal length. In *ECCV*, 2018.
[36] C. Russell, R. Yu, and L. Agapito. Video pop-up: Monocular 3D reconstruction of dynamic scenes. In *ECCV*. 2014.
[37] M. Salzmann and P. Fua. Linear local models for monocular reconstruction of deformable surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):931–944, 2011.
[38] N. Sundaram, T. Brox, and K. Keutzer. Dense point trajectories by GPU-accelerated large displacement optical flow. In *ECCV*, 2010.
[39] J. Taylor, A. D. Jepson, and K. N. Kutulakos. Non-rigid structure from locally-rigid motion. In *CVPR*, 2010.
[40] D. Tien Ngo, S. Park, A. Jorstad, A. Crivellaro, C. D. Yoo, and P. Fua. Dense image registration and deformable surface reconstruction in presence of occlusions and minimal texture. In *ICCV*, 2015.
[41] 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.
[42] Q.-H. Tran, T.-J. Chin, G. Carneiro, M. S. Brown, and D. Suter. In defence of ransac for outlier rejection in deformable registration. In *ECCV*, 2012.
[43] A. Varol, M. Salzmann, P. Fua, and R. Urtasun. A constrained latent variable model. In *CVPR*, 2012.
[44] A. Varol, M. Salzmann, E. Tola, and P. Fua. Template-free monocular reconstruction of deformable surfaces. In *CVPR*, 2009.
[45] S. Vicente and L. Agapito. Soft inextensibility constraints for template-free non-rigid reconstruction. In *ECCV*, 2012.

Shaifali Parashar received her PhD degree in Computer Vision from Université Clermont Auvergne. She is currently a Post-doc researcher in CVLAB, EPFL. Her research interest are non-rigid 3D reconstruction, deformable SLAM and 3D shape registration.

Daniel Pizarro has been an Associate Professor at the Universidad de Alcalá (Spain) since 2012. He is a member of the GEINTRA group and an invited member of EnCoV group. His research interests include image registration, deformable reconstruction and their applications to minimally invasive surgeries.

Adrien Bartoli has been a Professor of Computer Science at Université Clermont Auvergne since fall 2009. He leads the EnCoV research group at Institut Pascal (CNRS, UCA, CHU de Clermont-Ferrand). His research interests include image registration and Shape-from-X for rigid and non-rigid environments, with applications to computer-aided endoscopy.

APPENDIX A
COEFFICIENTS OF THE RECONSTRUCTION EQUATIONS

The coefficients of the two cubic reconstruction equations (10) and (11) are:

$$
\begin{aligned}
&a_{30}=2e_1j_1{j_3}^2e_{10}-2e_1j_1^2j_3e_9+2e_2j_1j_3v_1e_{16},\\
&a_{21}=2e_1j_3(j_1j_4+e_{15})e_{10}-2e_1j_1(e_{15}+j_2j_3)e_9\\
&\quad+2e_{16}e_2(v_1e_{15}-j_1j_3u_1),\\
&a_{12}=2e_1j_4(e_{15}+j_2j_3)e_{10}-2e_1j_2(j_1j_4+e_{16})e_9\\
&\quad+2e_2e_{16}(j_2j_4v_1-u_1e_{15}),\\
&a_{03}=2e_1j_2{j_4}^2e_{10}-2e_1j_2^2j_4e_9-2e_2j_2j_4u_1e_{16},\\
&a_{20}=4j_1j_3e_3e_9-4j_1j_3e_4e_{10}+j_1^2e_1e_5-j_3^2e_1e_6-e_2e_{15}e_{16},\\
&a_{11}=4(e_3e_{15}e_9-e_4e_{16}e_{10})+2e_1(j_1j_2e_5-j_3j_4e_6)\\
&\quad+2e_2(j_1j_3-j_2j_4)e_{16},\\
&a_{02}=4j_2j_4e_3e_9-4j_2j_4e_4e_{10}+e_1j_2^2e_5+e_1j_4^2e_6+e_2e_{15}e_{16},\\
&a_{10}=2j_1e_{14}e_{10}-2j_3e_{13}e_9-2j_1e_3e_5+2j_3e_4e_6,\\
&a_{01}=2j_2e_{14}e_{10}-2j_4e_{13}e_9-2j_2e_3e_5+2j_4e_4e_6,\\
&a_{00}=e_{13}e_5-e_{14}e_6,\ b_{30}=e_1j_3^3e_{10}-j_1j_3^2e_1e_9+j_3^2e_2v_1e_{16},\\
&b_{21}=3j_3^2j_4e_1e_{10}-j_3e_1(j_1j_4+e_{15})e_9+j_3e_2(2j_4v_1-j_3u_1)e_{16},\\
&b_{12}=3j_3j_4^2e_1e_{10}-j_4e_1(e_{15}+j_2j_3)e_9+j_4e_2(j_4v_1-2j_3u_1)e_{16},\\
&b_{03}=e_1j_4^3e_{10}-j_2j_4^2e_1e_9+j_4^2e_2u_1e_{16},\\
&b_{20}=j_1j_3e_1e_5+2j_3^2e_3e_9-2j_3^2e_4e_{10}+j_3^2e_1e_8-j_3j_4e_2e_{16},\\
&b_{11}=e_{15}e_1e_5+2j_3j_4(2e_3e_9-2j_4v_1e_{10}+e_1e_8)+e_{16}e_2(j_3^2-j_4^2),\\
&b_{02}=j_2j_4e_1e_5+2j_4^2e_3e_9-2j_4^2e_4e_{10}+j_4^2e_1e_8+j_3j_4e_2e_{16},\\
&b_{10}=j_3e_{14}e_{10}-(j_3e_7-j_4e_{16})e_9-(j_1e_4+j_3e_3)e_5-2j_3e_4e_8,\\
&b_{01}=j_4e_{14}e_{10}-(j_3e_{16}-j_4e_7)e_9-(j_4e_3+j_2e_4)e_5-2j_4e_4e_8,\\
&b_{00}=e_{14}e_8+e_7e_5,\ e_1=1+u_1^2+v_1^2,\ e_2=1+u_2^2+v_2^2,\\
&e_3=j_1u_1+j_2v_1,\ e_4=j_3u_1+j_4v_1,\ e_5=1-2t_2v_2+e_2t_2^2,\\
&e_6=1-2t_1u_2+e_2t_1^2, e_7=j_1j_3+j_2j_4, e_8=t_2u_2+t_1v_2-e_2t_1t_2,\\
&e_9=v_2-e_2t_2,\ e_{10}=u_2-e_2t_1,\ e_{11}=j_2u_1+j_4v_1,\\
&e_{12}=j_2u_1-j_3u_1,\ e_{13}=j_1^2+j_2^2,\ e_{14}=j_3^2+j_4^2,\ e_{15}=j_1j_4+j_2j_3,\\
&e_{16}=j_1j_4-j_2j_3,\ t_1=-(j_3h_3+j_4h_4),\ t_2=-(j_1h_3+j_2h_4).
\end{aligned}
$$