IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Dense isometric non-rigid shape-from-motion based on graph optimization and edge selection

Yongbo Chen[1], Liang Zhao[1], Yanhao Zhang[1], and Shoudong Huang[1]

Abstract— **In this letter, we propose a novel framework for dense isometric non-rigid shape-from-motion (Iso-NRSfM) based on graph topology and edge selection. A weighted undirected graph, of which nodes, edges, and weighted values are respectively the images, the image warps, and the number of the common features, is built. An edge selection algorithm based on maximum spanning tree and sub-modular optimization is presented to pick out the well-connected sub-graph for the warps with multiple images. Using the infinitesimal planarity assumption, the Iso-NRSfM problem is formulated as a graph optimization problem with the virtual measurements, which are based on metric tensor and Christoffel Symbol, and the variables related to the derivatives of the constructed points along the surface. The solution of this graph optimization problem directly leads to the normal field of the shape. Then, using a separable iterative optimization method, we obtain the dense point cloud with texture corresponding to the deformable shape robustly. In the experiments, the proposed method outperforms existing work in terms of constructed accuracy, especially when there exists missing/appearing (changing) data, noisy data, and outliers.**

I. INTRODUCTION

Non-rigid Shape-from-Motion (NRSfM), which is the problem of reconstructing the 3D shapes of a deforming object from a set of monocular images, plays a very important role in computer vision. For the NRSfM, the recovered 3D shapes are in the local camera coordinates, which means that the camera motion is coupled in the deformations. This point is greatly similar to the visual measurements of the deformable visual SLAM algorithms, of which the goal is to simultaneously locate a robot using a visual sensor and map its surrounding area in a partly or fully deforming environments [1], [2]. In other words, the NRSfM algorithm may be an important way to deal with the mapping part of the deformable visual SLAM problem. If we have some other tools to localize the robot poses, the NRSfM algorithms will estimate the more consistent mapping results by fusing the information of the robot poses. Hence, we consider it as a potential tool in the research of the deformable visual SLAM.

Because of the non-rigid deformation, many traditional Structure-from-Motion methods [3] for the rigid objects cannot be used to solve the NRSfM problem directly. In fact, the NRSfM problem is un-solvable without introducing other constraints, because the same images may be proposed by totally different deformations of the objects [4]. Hence, some constraints on the deformation model of the object are proposed to limit the set of solutions. The commonly used constraints include low-rank trajectory basis [5], low-rank shape basis [6], isometry [7] and so on. Due to the good approximation to many real deformable objects, like: paper, rug, flag and bag, the isometry constraints are widely used [8], [9]. Because of the lack of estimating information, the existing methods tend to be inaccurate, ill-posedness, and un-robust for the missing/appearing and noisy data. In order to overcome these challenges, this letter aims to use the well-connect graph for the image warps, which are the geometric transformation functions mapping points between 2D images of a deforming surface, and the robust graph optimization to achieve high efficiency in improving the estimation accuracy and good robustness in terms of the data changing.

A. Related work

NRSfM is still an open problem because of the complexity of deformations and less reliable information. Its usual inputs and outputs are respectively multiple calibrated images from a monocular camera and the time-varying 3D shapes of the deformed object. Popular frameworks include statistics-based methods and physics-based methods [10].

The statistics-based methods, which commonly assume a low-dimensional space of the deformed shapes using low-rank shape [11], temporal smoothness [12], or low-rank point trajectory [13], usually require a large number of images as inputs to recover all parameters corresponding to different low-rank bases. The authors in [14] propose a novel 3D shape trajectory method that solves for the deformable structure as the smooth time-trajectory of a single point in a linear shape space. In [15], the authors summarize the low-rank trajectory-based representations and the low-rank shape-based representations using a force-based representation for the NRSfM problem. This statistics-based methods can well handle small and regular objects such as human faces, but are not very suitable for the objects with very large shape spaces such as pieces of flag, which may cause the large error.

The physics-based methods, which introduce some geometry or physical constraints, like isometric manifold [7], inextensibility [4], local planarity [9] and energy minimization [16], for the deformed shapes without using the low-rank assumption, can commonly handle larger or more complex deformations. The authors [4] present a global and convex Second-Order Cone Programming (SOCP) formulation for the template-less 3D reconstruction of a deforming object with the perspective camera using a convex relaxation of isometry, called inextensibility constraint. Because many

---

*This work was not supported by any organization

[1] Y. Chen, L. Zhao, Y. Zhang, and S. Huang are with Centre for Autonomous Systems (CAS), Faculty of Engineering and Information Technology (FEIT), University of Technology Sydney, Ultimo, NSW, 2007 Australia `Yongbo.Chen@uts.edu.au`

Copyright ©2020 IEEE

physical concepts have the specific limitations for the local geometry property (like derivative) of the 3D shape, most physics-based methods are explored based on the mesh model or a discrete neighborhood-aware point cloud model [10]. Thus the physics-based methods can deal with the more complex deformations and also overcome the missing data problem. However, restricted to the accuracy of the mesh and point cloud models, the recovery accuracy performance of the physics-based methods is also limited.

B. Overview of the approach

Our framework (Fig. 1) is based on two un-directed graphs, which are respectively used in the image warps and the 3D features reconstruction. Using any cheap feature matching method, we can get a complete weighted graph, of which the nodes, the edges, and the weights are respectively the input monocular images, the image pairs based on feature matching, and the number of the common features. For this complete graph, given a fixed edge number, we seek a sub-graph to maximize its D-optimal design metric. This measurement (edge) selection problem can be solved based on Kruskal's maximum spanning tree approach [17] and the sub-modular optimization method [18]. Based on the selected well-connected sub-graph, for each two connected images, its corresponding image warp is computed using the matched features and the bicubic B-splines. For the other graph corresponding to 3D features reconstruction, using the infinitesimal planarity and isometric assumptions, we build a sparse graph optimization problem of which the variables are the directional derivatives of the common features of each image pair and the virtual measurements are their relationship obtained by metric tensor (MT) and Christoffel Symbol (CS). Its solution directly leads to the normal field of the 3D recovery surface and further generates the dense colorless point cloud. Finally, so as to texture these points, the transformation between the 3D point cloud and the 2D features of the images are computed using a novel separable iterative optimization method.

C. Paper organization and contributions

This letter presents a dense physics-based Iso-NRSfM framework for the recovery of the deformable shapes using graph optimization and edge selection. In Section II, we present some preliminaries in the NRSfM problem and an accurate image warp method based on the edge selection. A graph optimization based framework is proposed in Section III. In Section IV, we present a non-linear least squares formulation to generate the colorful dense point clouds. Simulations and experiments are presented to validate the practicality of this framework and its high-performance in Section V. The main contributions of this work are as follows:

- ◇ Novel graph optimization framework using MT and CS for obtaining the normal field of the deformed shapes.
- ◇ Efficient edge selection method using maximum spanning tree and sub-modular optimization to obtain the well-connected graph structure for robust image warps.
- ◇ Robust separable iterative optimization method for texturing the dense point cloud.

II. PRELIMINARIES AND IMAGE WARPS

A. Models Preliminaries

As shown in Fig. 2, assuming that the multiple deformation surfaces are Riemannian manifolds, using the perspective projection ($\Pi_1,\ \Pi_2,\ \cdots$), we can obtain the monocular images ($I_1,\ I_2,\ \cdots$). The inverse mapping of the perspective projection Π_i is named as the image embedding $\Phi_i:\ I_i \in \mathbb{R}^2 \to M_i \in \mathbb{R}^3$. Based on the matched features, the dense geometric transformation between the pair of images I_i and I_j is denoted by the image warps functions η_{ij} and η_{ji}. Our NRSfM goal is to compute the deformation mapping Ψ_{ij}.

Based on the perspective camera, its perspective projection mapping $\Pi_i : \boldsymbol{z} \to \boldsymbol{x}$ is defined as:

$$\boldsymbol{x} = (x_1, x_2)^\top = \Pi_i(\boldsymbol{z}) = (z_1/z_3, z_2/z_3)^\top, \tag{1}$$

where $\boldsymbol{z} = (z_1, z_2, z_3)^\top,\ z_3 > 0$ is a 3D point, the image embedding, which is its inverse mapping perspective projection Φ_i from 2D feature pixels $\boldsymbol{x}$ to $\boldsymbol{z}$, is defined as:

$$\boldsymbol{z} = \Phi_i \circ \Pi_i(\boldsymbol{z}) = \Phi_i(\boldsymbol{x}) = \alpha_i(x_1, x_2)^{-1}(x_1, x_2, 1)^\top, \tag{2}$$

where $\alpha_i(x_1, x_2)$ is the inverse of the depth of the feature corresponding to the i-th image. The image embedding aims to regard the depth of the i-th feature as a smooth function.

In this work, we use the infinitesimal planarity assumption [9], which means that a surface at every point $\boldsymbol{z}$ is planar. Thus, we have the plane equation $\boldsymbol{n}^\top \boldsymbol{z} + d = 0$, where $\boldsymbol{n} = (n_1, n_2, n_3)^\top$ is the normal of the small plane. By introducing the mapping (2) into this equation, we have:

$$\alpha_i(x_1, x_2) = -d^{-1}(n_1 x_1 + n_2 x_2 + n_3). \tag{3}$$

The Jacobian matrix $\boldsymbol{J}_{\Phi_i}$ of the image embedding Φ_i is:

$$\boldsymbol{J}_{\Phi_i} = \frac{1}{\alpha_i^2}\begin{bmatrix} \alpha_i - x_1\alpha_{i,1} & -x_1\alpha_{i,2} \\ -x_2\alpha_{i,1} & \alpha_i - x_2\alpha_{i,2} \\ -\alpha_{i,1} & -\alpha_{i,2} \end{bmatrix}, \tag{4}$$

where $\alpha_i(x_1, x_2)$ is written as α_i for simplification, $\alpha_{i,1} = \frac{\partial \alpha_i}{\partial x_1} = -\frac{n_1}{d}$ and $\alpha_{i,2} = \frac{\partial \alpha_i}{\partial x_2} = -\frac{n_2}{d}$. Based on this assumption, we have the second-order derivatives $\alpha_{i,11} = \alpha_{i,12} = \alpha_{i,21} = \alpha_{i,22} = 0$, where $\alpha_{i,kl}$ means $\frac{\partial^2 \alpha_i}{\partial x_k \partial x_l}$.

B. Accurate image warp based on edge selection

Our Iso-NRSfM method is a point-wise method. The different features are solved decoupling after using the image warps, which is the only step of this algorithm shown the information connection between different features. Thus, the accuracy of the image warp will decide the accuracy of the NRSfM results. In this letter, we use the image warps method based on 2D Schwarzian derivatives [19]. Under the infinitesimal planarity assumption, this method aims to find the optimal warp between two images of which the objective function is the combination of the Schwarzian penalty $\epsilon_{se}[\eta]$ and the transfer error between point correspondences $\epsilon_d[\eta]$, satisfying:

$$\min_{\eta} \epsilon_d[\eta] + \lambda \epsilon_{se}[\eta], \tag{5}$$

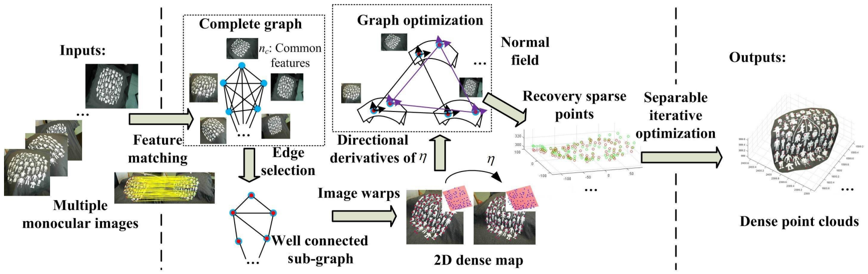


Fig. 1. Structure of our dense Iso-NRSfM method.

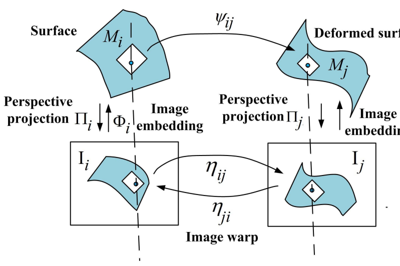


Fig. 2. Models and maps in NRSfM problem.

where λ is a hyperparameter which weighs the influence of the Schwarzian derivatives over the data term. More details are shown in [19]. Because of the non-convexity, the image warps are solved using the Levenberg-Marquardt algorithm.

The number and the distribution of the matched features will decide the accuracy of the image warps. It is hard to guarantee the geometry distribution of the features, so we consider the number of the common features only by introducing a complete undirected graph $\mathcal{G}_c = (\mathcal{V}_c,\ \mathcal{E}_c,\ w_c)$ of which the edge value is the number of the common matched features of the image pair. Then, we need to pick out a well-connected sub-graph and then compute the corresponding image warps.

The D-optimal design metric, which is to evaluate the log-determinant function of the reduced Laplacian matrix of a graph, is used to select the sub-graph. The node and edge numbers of $\mathcal{G}_c$ are $|\mathcal{G}_c| = n_c$ and $|\mathcal{E}_c| = n_c(n_c-1)/2$. So as to recover all shapes, at least $n_c - 1$ edges are required to connect all images, which is called as *spanning tree*. The spanning tree of a graph $\mathcal{T}$ is a subgraph that is a tree and covers all nodes of $\mathcal{G}_c$. We need the following definition:

Definition 2.1: Let $\mathcal{T}_\mathcal{G}$ be the set of all spanning trees of $\mathcal{G}$. The weighted number of $\mathcal{G}$, called *tree-connectivity* [20], is defined as:

$$t_w(\mathcal{G}) \triangleq \sum_{\mathcal{T}\in\mathcal{T}_\mathcal{G}} \mathbb{V}(\mathcal{T}),\ \mathbb{V}(\mathcal{T}) = \prod_{e\in\mathcal{E}(\mathcal{T})} w(e), \tag{6}$$

where $\mathbb{V}(\mathcal{T}):\ \mathcal{T}_\mathcal{G} \to \mathbb{R}_+$ is the value of a spanning tree $\mathcal{T} \in \mathcal{T}_\mathcal{G}$, $\mathcal{E}(\mathcal{T})$ represents the set of edges in $\mathcal{T}$.

A result of tree-connectivity in graph theory is as follows:

Theorem 2.1: (Weighted Matrix-Tree Theorem [18]). For a simple weighted graph $\mathcal{G} = (\mathcal{V},\ \mathcal{E},\ w)$ with $w:\mathcal{E}\to\mathbb{R}^+$, we have $t_w(\mathcal{G}) = \det(\boldsymbol{\mathcal{L}}_w^\mathcal{G})$, where $\boldsymbol{\mathcal{L}}_w^\mathcal{G}$ is the reduced weighted Laplacian matrix of $\mathcal{G}$.

Hence, for the edge selection problem, we can easily obtain the following Theorem 2.2.

Theorem 2.2: For a graph $\mathcal{G}$ with positive weights, to select the optimal spanning tree based on the D-optimal design metric is equal to select the maximum spanning tree.

Proof: Because of the monotonicity of the log function, for any tree, to maximize the D-optimality metric is same to optimize the tree connectivity based on Theorem 2.1. For a spanning tree $\mathcal{T}\in\mathcal{T}_\mathcal{G}$, we have: $t_w(\mathcal{T}) = \mathbb{V}(\mathcal{T}) = \prod_{e\in\mathcal{E}(\mathcal{T})} w(e)$. The maximum spanning tree is a spanning tree of a weighted graph $\mathcal{T}$ having maximum weights, so we have: $\max\limits_{\mathcal{T}\in\mathcal{T}_\mathcal{G}} \log(\det(\boldsymbol{\mathcal{L}}_w^\mathcal{G})) \Leftrightarrow \max\limits_{\mathcal{T}\in\mathcal{T}_\mathcal{G}} t_w(\mathcal{T}) \Leftrightarrow \max\limits_{\mathcal{T}\in\mathcal{T}_\mathcal{G}} \mathbb{V}(\mathcal{T})$. ■

Algorithm 1: Edge selection method

```
Input: The complete graph $\mathcal{G}_c = (\mathcal{V}_c,\ \mathcal{E}_c,\ w_c)$, the number
    of the edges $N = n_c - 1 + k$
Output: The optimal sub-graph $\mathcal{G}_{opt}$ with $N$ edges
$\mathcal{T}_{opt} = Kruskal\_MST(\mathcal{G}_c)$; //Using the Kruskal's algorithm
    to select the optimal spanning tree.
$\mathcal{G}_{opt} \leftarrow \mathcal{T}_{opt}$;
if $k == 0$ then
else if $k > 0$ then
    $\mathcal{E}_s \leftarrow \mathcal{E}(\mathcal{G}_c/\mathcal{G}_{opt})$; $\boldsymbol{\mathcal{L}}_w^b \leftarrow \boldsymbol{\mathcal{L}}_w^{\mathcal{T}_{opt}}$;
    for $l = 1:1:k$ do
        //Greedy-based method.
        $e_{opt} = f_{1-ESP}(\mathcal{E}_s, \boldsymbol{\mathcal{L}}_w^b)$; //Use Algorithm 2.
        $\mathcal{G}_{opt} \leftarrow \mathcal{G}_{opt} \cup \{e_{opt}\}$, $\mathcal{E}_s \leftarrow \mathcal{E}_s/\{e_{opt}\}$, update $\boldsymbol{\mathcal{L}}_w^b$;
    end
end
return $\mathcal{G}_{opt}$
```

Hence, if the edge number is $n_c - 1$, we can pick out the maximum spanning tree using the Kruskal's maximum spanning tree algorithm. If the edge number is allowed to be larger than $n_c - 1$, we can build a stronger sub-graph $\mathcal{G}_s$ for the image warps. Based on the maximum spanning tree, this new problem becomes the famous k edge selection problem ($k-$ESP) [20], of which the aim is to find k additional edges to a given graph that leads to the highest increase in tree connectivity of the new pose graph. The k-ESP problem is:

$$\max_{\mathcal{E}_k\in\mathcal{E}(\mathcal{G}_c/\mathcal{T}_{opt})} t_w(\mathcal{T}_{opt}\cup\mathcal{E}_k),\ s.t.\ |\mathcal{E}_k| = k. \tag{7}$$

where $\mathcal{E}(\mathcal{G}_c/\mathcal{T}_{opt})$ is a set of new candidate edges which belongs to the complete graph $\mathcal{G}_c$, but does not belong to the maximum spanning tree $\mathcal{T}_{opt}$.

Theorem 2.3: ([20]) The k-ESP problem is an instance of a monotone sub-modular function maximization problem subject to a cardinality constraint.

For the sub-modular maximization problem with a cardinality constraint, the greedy-based method is widely used with some performance guarantee. The k-ESP problem is divided into k new 1-ESP sub-problems. As an example, we explain the operations in the first 1-ESP sub-problem. Using this exhaustive method, the optimal edge for 1-ESP can be found by exhaustively computing the tree connectivity for every candidate edge $e_s\in\mathcal{E}_s$, leading to a new graph with nodes $\mathcal{V}$ and edges $\mathcal{T}_{opt}\cup\{e_s\}$. For computational efficiency, the 1-ESP is written as:

$$\max_{e_s\in\mathcal{E}_s}\ f(e_s) = \log(\det(\boldsymbol{\mathcal{L}}_w^{\mathcal{T}_{opt}} + \boldsymbol{a}_s w(e_s)\boldsymbol{a}_s^\top)), \tag{8}$$

Algorithm 2: 1-ESP f_{1-ESP}

```
Input: The candidate edge set $\mathcal{E}_s$ and the weigted Laplacian matrix $\mathcal{L}_w^b$ for the base graph
Output: The optimal edge $e_s^*$
$p_{order} = AMDP(\mathcal{L}_w^b)$; //Column approximate minimum degree permutation
$H^b = Cholesky(\mathcal{L}_w^b(p_{order}, p_{order}))$;
$v \leftarrow cs\_etree(H^b, 'col')$; //Return the elimination tree of $H^{b\top} H^b$ for rank-1 update $CholeskyUpdate$
$f^* \leftarrow 0$
for all $e_s$ in $\mathcal{E}_s$ do
    Generate $a_s$ based on $e_s$, $\bar{a}_s \leftarrow w(e_s)^{\frac{1}{2}} a_s$;
    $\bar{H}^b = CholeskyUpdate(H^b, \bar{a}_s, v)$;
    $f \leftarrow 2\sum_i \log(\bar{H}^b)_{i,i}$;
    if $f > f^*$ then
        $f^* \leftarrow f$, $e_s^* \leftarrow e_s$;
    end
end
return $e_s^*$
```

where $\boldsymbol{a}_s$ is a column in the incidence matrix corresponding to new added edge, and $w(e_s)$ is the edge weight corresponding to the edge e_s. $f(e_s)$ can be calculated by a rank-1 update to the reduced weighted Laplacian matrix setting $\bar{\boldsymbol{a}}_s = \boldsymbol{a}_s w(e_s)^{\frac{1}{2}}$, $\mathcal{L}_w^{\mathcal{T}_{opt}} = \boldsymbol{H}^{\mathcal{T}_{opt}} \boldsymbol{H}^{\mathcal{T}_{opt}\top}$.[1]

Overall, our edge selection method including the maximum spanning tree and the k-ESP problem is summarized in Algorithm 1. For the obtained well-connected sub-graph, based on its edges, we can compute the corresponding image wraps from the small index image to the large index image.

III. ISO-NRSFM ALGORITHM

A. Virtual measurements based on metric tensor and CS

For a smooth mapping, we can define its metric tensor and Christoffel Symbols, which are two properties in the Riemannian manifolds. They are introduced in the NRSfM problem by [8] to describe the mappings in Fig. 2.

The metric tensor of the mapping Φ_i is defined as:

$$\boldsymbol{g}[\Phi_i] \triangleq \boldsymbol{J}_{\Phi_i}^\top \boldsymbol{J}_{\Phi_i}. \tag{9}$$

Let $k_1 = \frac{\alpha_{i,1}}{\alpha_i}$ and $k_2 = \frac{\alpha_{i,2}}{\alpha_i}$, introducing the Jacobian matrix of the perspective projection (4) into the definition of the metric tensor, for the mapping Φ_i, we have:

$$\begin{aligned}
&\boldsymbol{g}[\Phi_i] = \boldsymbol{J}_{\Phi_i}^\top \boldsymbol{J}_{\Phi_i} = \begin{bmatrix} \boldsymbol{g}_{11}[\Phi_j] & \boldsymbol{g}_{12}[\Phi_i] \\ \boldsymbol{g}_{12}[\Phi_i] & \boldsymbol{g}_{22}[\Phi_i] \end{bmatrix}, \\
&\boldsymbol{g}_{11}[\Phi_i] = \alpha_i^{-2}(k_1^2 + (k_1 x_1 - 1)^2 + (k_1 x_2)^2), \\
&\boldsymbol{g}_{12}[\Phi_i] = \alpha_i^{-2}(k_1 k_2 (1 + x_1^2 + x_2^2) - k_2 x_1 - k_1 x_2), \\
&\boldsymbol{g}_{22}[\Phi_i] = \alpha_i^{-2}(k_2^2 + (k_2 x_2 - 1)^2 + (k_2 x_1)^2).
\end{aligned} \tag{10}$$

The (m,n)-th element $\boldsymbol{\Gamma}_{mn}^p[\Phi_i]$ of two 2×2 CS metrics ($p = 1, 2$) of the mapping Φ_i, which represents the change of the metric tensors, is defined as:

$$\begin{aligned}
\boldsymbol{\Gamma}_{mn}^p[\Phi_i] \triangleq &\sum_{p=1}^{2}\sum_{m=1}^{2}\sum_{n=1}^{2}\sum_{l=1}^{2} \frac{1}{2}\boldsymbol{g}^{pl}[\Phi_i] \left(\boldsymbol{g}_{lm,n}[\Phi_i]\right. \\
&\left. + \boldsymbol{g}_{ln,m}[\Phi_i] - \boldsymbol{g}_{mn,l}[\Phi_i]\right),
\end{aligned} \tag{11}$$

where $\boldsymbol{g}^{pl}[\Phi_i]$ is the (p,l)-th element of the 2×2 inverse matrix of MT $\boldsymbol{g}[\Phi_i]$; $\boldsymbol{g}_{mn,l}[\Phi_i]$ is the partial derivative of the (m,n)-th element of MT $\boldsymbol{g}[\Phi_i]$ along the l-th direction.

Based on the infinitesimal planarity assumption, the CS matrices of the mapping Φ_i can be re-written as:

$$\boldsymbol{\Gamma}^1[\Phi_i] = \begin{bmatrix} -2k_1 & -k_2 \\ -k_2 & 0 \end{bmatrix}, \ \boldsymbol{\Gamma}^2[\Phi_i] = \begin{bmatrix} 0 & -k_1 \\ -k_1 & -2k_2 \end{bmatrix}. \tag{12}$$

In Fig. 2, for the image mappings Φ_i and Φ_j, we can define their metric tensors and CS: $\boldsymbol{g}[\Phi_i]$, $\boldsymbol{g}[\Phi_j]$, $\boldsymbol{\Gamma}^1[\Phi_i]$, $\boldsymbol{\Gamma}^1[\Phi_j]$, $\boldsymbol{\Gamma}^2[\Phi_i]$, and $\boldsymbol{\Gamma}^2[\Phi_j]$ using (10) and (12). Naturally, we have the mapping Φ_j satisfying: $\Phi_j = \Psi_{ij} \circ \Phi_i \circ \eta_{ji}$ and the Jacobian matrices following: $\boldsymbol{J}_{\Phi_j} = \boldsymbol{J}_{\Psi_{ij}} \boldsymbol{J}_{\Phi_i} \boldsymbol{J}_{\eta_{ji}}$, so its MT is: $\boldsymbol{g}[\Phi_j] = \boldsymbol{J}_{\Phi_j}^\top \boldsymbol{J}_{\Phi_j} = \boldsymbol{J}_{\eta_{ji}}^\top \boldsymbol{J}_{\Phi_i}^\top \boldsymbol{J}_{\Psi_{ij}}^\top \boldsymbol{J}_{\Psi_{ij}} \boldsymbol{J}_{\Phi_i} \boldsymbol{J}_{\eta_{ji}}$.

Based on the isometric assumption on the mapping Ψ_{ij} from i-th deformable shape M_i to j-th deformable shape M_j ($\boldsymbol{J}_{\Psi_{ij}}^\top \boldsymbol{J}_{\Psi_{ij}} = \boldsymbol{I}_{3\times 3}$), we have the following theorem:

Theorem 3.1: Let Ψ_{ij} be an isometric mapping, then $\boldsymbol{g}_{mn}[\Phi_j] = \boldsymbol{g}_{mn}[\Phi_i \circ \eta_{ji}]$ and $\boldsymbol{\Gamma}_{mn}^p[\Phi_j] = \boldsymbol{\Gamma}_{mn}^p[\Phi_i \circ \eta_{ji}]$ with $(i,j) \in [1,N] \times [1,N]$ [7].

Let's further consider the composite function $\Phi_i \circ \eta_{ji}$, based on [8], its MT and CS can be written by:

$$\begin{aligned}
&\boldsymbol{g}_{st}[\Phi_i \circ \eta_{ji}] = \sum_{m=1}^{2}\sum_{n=1}^{2} \frac{\partial x_m}{\partial y_s}\frac{\partial x_n}{\partial y_t}\boldsymbol{g}_{mn}[\Phi_i], \\
&\boldsymbol{\Gamma}_{st}^p[\Phi_i \circ \eta_{ji}] = \sum_{p=1}^{2}\sum_{m=1}^{2}\sum_{n=1}^{2}\sum_{l=1}^{2} (\frac{\partial x_m}{\partial y_s}\frac{\partial x_n}{\partial y_t}\boldsymbol{\Gamma}_{mn}^p[\Phi_i] \\
&\cdot \frac{\partial y_q}{\partial x_p} + \frac{\partial y_q}{\partial x_l}\frac{\partial^2 x_l}{\partial y_s y_t}),
\end{aligned} \tag{13}$$

Combining Theorem 3.1 and the equation (12) related to the CS, we can get the first two virtual measurements $f_1(k_1, k_2, \bar{k}_1, \bar{k}_2) = 0$ and $f_2(k_1, k_2, \bar{k}_1, \bar{k}_2) = 0$ about the matched feature corresponding to the i-th and j-th images:

$$\begin{aligned}
&\frac{\partial x_1}{\partial y_1}k_1 + \frac{\partial x_2}{\partial y_1}k_2 - \frac{\partial y_2}{\partial x_1}\frac{\partial^2 x_1}{\partial y_1 y_2} - \frac{\partial y_2}{\partial x_2}\frac{\partial^2 x_2}{\partial y_1 y_2} = \bar{k}_1, \\
&\frac{\partial x_1}{\partial y_2}k_1 + \frac{\partial x_2}{\partial y_2}k_2 - \frac{\partial y_1}{\partial x_1}\frac{\partial^2 x_1}{\partial y_1 y_2} - \frac{\partial y_1}{\partial x_2}\frac{\partial^2 x_2}{\partial y_1 y_2} = \bar{k}_2,
\end{aligned} \tag{14}$$

where $\bar{k}_1 = \frac{\alpha_{j,1}}{\alpha_j}$ and $\bar{k}_2 = \frac{\alpha_{j,2}}{\alpha_j}$ are corresponding to the j-th image, which are the linear combinations of k_1 and k_2, $\frac{\partial x_m}{\partial y_n}$, $\frac{\partial y_m}{\partial x_n}$ and $\frac{\partial^2 x_m}{\partial y_1 y_2}$ are obtained based on the derivatives of the image warps and its inverse mapping [10].

Let $\boldsymbol{G}_{mn} = \boldsymbol{g}_{mn}[\Phi_i]\alpha_i^2$ and $\bar{\boldsymbol{G}}_{mn} = \boldsymbol{g}_{mn}[\Phi_j]\alpha_j^2$, combining Theorem 3.1 and the equation (10) related to the metric tensor, then we can get the other measurements $f_3(k_1, k_2, \bar{k}_1, \bar{k}_2) = 0$ and $f_4(k_1, k_2, \bar{k}_1, \bar{k}_2) = 0$ [7]:

$$\begin{aligned}
&\sum_{m=1}^{2}\sum_{n=1}^{2}(\frac{\partial x_m}{\partial y_1}\frac{\partial x_n}{\partial y_1}\boldsymbol{G}_{mn}\bar{\boldsymbol{G}}_{12} - \frac{\partial x_m}{\partial y_1}\frac{\partial x_n}{\partial y_2}\boldsymbol{G}_{mn}\bar{\boldsymbol{G}}_{11}) = 0, \\
&\sum_{m=1}^{2}\sum_{n=1}^{2}(\frac{\partial x_m}{\partial y_1}\frac{\partial x_n}{\partial y_1}\boldsymbol{G}_{mn}\bar{\boldsymbol{G}}_{22} - \frac{\partial x_m}{\partial y_2}\frac{\partial x_n}{\partial y_2}\boldsymbol{G}_{mn}\bar{\boldsymbol{G}}_{11}) = 0.
\end{aligned} \tag{15}$$

Finally, we can get the above four measurements functions (14) and (15) for every image pair of which the coefficient has computed the image warp in Section II.

---

[1] 'SuiteSparse' library [21] provides efficient subroutines for sparse rank-1 Cholesky update including cs_etree and $CholeskyUpdate$ in Algorithm 2.

B. Point-wise solution using graph optimization

For i-th feature in the j-th shape, we can define two variables $k_1^{(i,j)}$ and $k_2^{(i,j)}$ similar to k_1 and k_2. Based on Section II-B, given the edge number N, we can generate a well-connected graph $\mathcal{G}_{opt}$ using edge selection. Assuming that all features are observed from all images, the NRSfM problem aims to compute N_p 3D points from N_m monocular images, which means that, using the coefficients from the image warps, this problem can be written as an optimization problem of which the variable dimension is $2N_pN_m$. For e-th edge $(i_e,\ j_e) \in \mathcal{G}_{opt}$, considering i-th feature in i_e-th and j_e-th shapes, we can also write four virtual measurements based on (14) and (15). Finally, as shown in Fig. 3, different from [8], we formulate a non-linear least squares problem:

$$\min \sum_{(i_e,j_e)\in\mathcal{G}_{opt}} \sum_{i=1}^{N_p} \omega(e) \sum_{j=1}^{4} \|\bar{f}_j(i_e,j_e,i)\|^2, \tag{16}$$

where $\bar{f}_j(i_e,j_e,i) = f_j(k_1^{(i,i_e)}, k_2^{(i,i_e)}, k_1^{(i,j_e)}, k_2^{(i,j_e)})$. Exploiting its sparseness, we can easily solve it using the trust-region-reflective algorithm with the given sparse Jacobian matrix in seconds when $2N_pN_m$ is limited in 10^5.

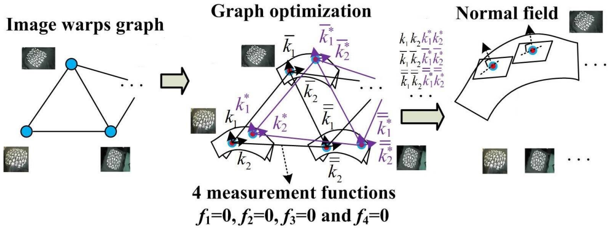


Fig. 3. NRSfM solution based on graph optimization.

After computing all parameters $k_1^{(i,j)}$ and $k_2^{(i,j)}$, introducing them in the Jacobian matrix $\boldsymbol{J}_{\Phi_i}$ of the image embedding (4), we obtain the normals by normalizing the cross-product of two columns of the Jacobian matrix and recover the surfaces by integrating the normal fields using [9].

IV. DENSE COLORED POINT CLOUD GENERATION

In order to texture the generated dense point clouds, their corresponding pixels on the image are found. After applying the translation, scale adjustment and perspective projection for the 3D recovered points, we seek to minimize the distances between the transformed 2D features and the 2D features on the images. For one shape, we have:

$$\min_{s,\boldsymbol{T}} \ f_d(s,\boldsymbol{T}) = \sum_{i=1}^{N_p} \|s f_{per}(\boldsymbol{p}_i^r - \boldsymbol{T}) - \boldsymbol{p}_i^f\|^2, \tag{17}$$

where s means the scale, $\boldsymbol{T}$ is the translations to transform the point cloud, $f_{per}(\star)$ is the perspective projection function shown in (1), $\boldsymbol{p}_i^r$ means the i-th 3D recovered point, $\boldsymbol{p}_i^f$ is its corresponding 2D detected features on the image.

Because of the non-convexity and the different scales of s and $\boldsymbol{T}$, an algorithm for solving problem (17) can be easily trapped into a local minimum. We use the separable way to solve it. Given a rough initialization[2], the problem is considered as the single-variable-only minimization problem. With the given Jacobian matrix and limited iterations, we can robustly and quickly compute its solution using Algorithm 3. Finally, all points are transformed to the 2D image coordinate and their colors are found by the nearest pixels.

Algorithm 3: Solution of the optimization problem (17)

```
Input: A rough initialization for $s_0$ and $\boldsymbol{T}_0$
Output: The optimized solution $s^*$ and $\boldsymbol{T}^*$
$s_k \leftarrow s_0,\ \boldsymbol{T}_k \leftarrow \boldsymbol{T}_0$;
while $\|s_{k+1} - s_k\| + \|\boldsymbol{T}_{k+1} - \boldsymbol{T}_k\| < \kappa$ do
    $s_{range} \leftarrow [c_{lb}s_k,\ c_{ub}s_k],\ 0 < c_{lb} < 1,\ c_{ub} > 1$
    $s_{k+1} = \min_{s_k \in s_{range}} f_d(s_k, \boldsymbol{T}_k)$;
    $\boldsymbol{T}_{k+1} = \min_{\boldsymbol{T}_k} f_d(s_k, \boldsymbol{T}_k)$;
end
$s^* \leftarrow s_{k+1},\ \boldsymbol{T}^* \leftarrow \boldsymbol{T}_{k+1}$;
return $s^*$, $\boldsymbol{T}^*$
```

V. SIMULATIONS AND EXPERIMENTS

In this section, we implement simulations and experiments with synthetic and real datasets using MATLAB on a Dell E5570 laptop with an Intel Core i5-6500 3.20 GHz processor. We compare our method with two other state-of-the-art NRSfM methods **iso** [7] and **infP** [8], which were shown to have better performances than other six algorithms [4], [5], [9], [22], [23], [24], by the shape error (mean difference between obtained and ground truth normals in degrees) and % 3D error[3], often used in the NRSfM literature [7].

A. Simulation with synthetic data

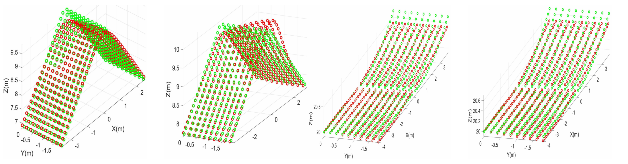
In this part, we simulate two synthetic datasets with 10 deforming scenes isometrically using the perspective projection model. The first dataset is to simulate the V-shape folding paper with multiple features connected by a small circular arc surface. The second one is to simulate the random scenes of the J-shape deforming paper, which is obtained by connecting a plane and a circular arc surface with small curvature. The deformations include many different folding angles. The reconstruct results for four shapes are shown in Fig. 4. Based on the whole sequential images, the comparison results for each image using three methods are shown in Fig. 5 [4].

---

[2] The rough initialization of the scale s_0 is based on the ratio between the maximal distance of the 2D features and the maximal horizontal distance of the 3D recovered points. The initial value of $\boldsymbol{T}_0$ is zero vector.

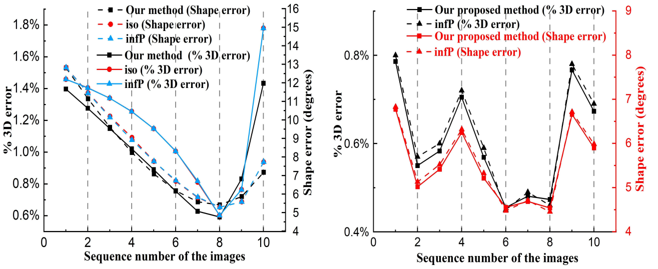
[3] The % 3D error of the i-th shape is defined as: % 3D error$_i$ $= \frac{\text{mean}_i \sqrt{\text{mean}^{\text{W}}(\|\boldsymbol{p}^r(i) - \boldsymbol{p}^{GT}(i)\|^2)}}{\max^{\text{W}}(\max_i^{\text{W}}(\boldsymbol{p}^{GT}(i)) - \min_i^{\text{W}}(\boldsymbol{p}^{GT}(i)))}$, where $\boldsymbol{p}^r(i) \in \mathbb{R}^{3\times N_p}$ and $\boldsymbol{p}^{GT}(i) \in \mathbb{R}^{3\times N_p}$ are the obtained coordinate and the ground truth of the recovery points in the i-th shape after using the scale and transformation adjustions. $\|\|^2$ is performed for every compponent similar to $.^2$ in MATLAB. mean$_i(\star) \in \mathbb{R}$ denotes the average function of all points in the i-th shape, mean$^{\text{W}}(\star) \in \mathbb{R}^{N_p}$ denotes the average function of the different components for every point, max$^{\text{W}}(\star) \in \mathbb{R}$ is the maximal element in a vector, $\max_i^{\text{W}}(\star) \in \mathbb{R}^3$ and $\min_i^{\text{W}}(\star) \in \mathbb{R}^3$ are respectively the maximal and minimal values of all points in different components. The physical meanings of the numerator and the denominator of this definition are respectively the average error and the shape scale in different components.

[4] In all our simulations, we find that the **infP** method and **iso** method show the simular performances, so their result curves commonly stack together.



(a) Shape 6 (b) Shape 9 (c) Shape 6 (d) Shape 9

Fig. 4. NRSfM ground truth (green) data and recovery results (red). First dataset: (a) and (b); Second dataset: (c) and (b).



(a) First dataset (b) Second dataset

Fig. 5. Comparison result of the shape error (degrees) and % 3D error$_i$. For the first dataset, our proposed method shows better performance than the others except the 8-th and 9-th images. For the second one, we find that the **iso** method fails to deal with this dataset and cannot obtain the recovery results. We also find that our method has some advantages compared with the **infP** in terms of the shape and % 3D errors.

We present the comparison results with the changing data and the additional graph edges. Table I shows the mean %3D error (in %) and the mean shape error (in °) obtained on two synthetic datasets (S1 is the first dataset and S2 is the second one) when 0-50% point data are missing from every image, including the first image. When a point is missing in the first image and is not missing in others, it is considered as an appearing point[5]. In Table I, **ours** and **+3 edges** respectively mean our method with $n_c - 1 = 9$ and $n_c - 1 + 3 = 12$ edges for the image warps. The result shows that our methods are better than the other one in terms of the data changing. The reasons of the good robustness of our method are as follows.

The **infP** method aims to get the image warps between the first image and the other images, then derive a system of two quartics in two variables for each image pair and finally solve the sum-of-squares of these polynomials of which the variables are only corresponding to the first image. Its structure is the star-like graph with the first image as the graph center. It is not based on the graph optimization. We improve the **infP** method on the following points.

- ◇ Balance the importances of the MT and the CS: The **infP** method introduces the CS constraints into the MT constraints to get the minimization of the sum-of-squares, which means that the CS constraints will be certainly satisfied and the MT are not. In our method, we consider them as the measurements, minimize them equally, and introduce the graph optimization idea.
- ◇ Strong graph structure: By using the graph optimization, every two images are possible to make connection which leads to the introducing of the edge selection method. Meanwhile, the features in every image will be fully used instead of only using the sub-set of the features of the first images. We have the good graph structure selected by the edge selection to deal with the missing and appearing instead of the star-like shape.

TABLE I

COMPARISON USING MISSING AND APPEARING DATA

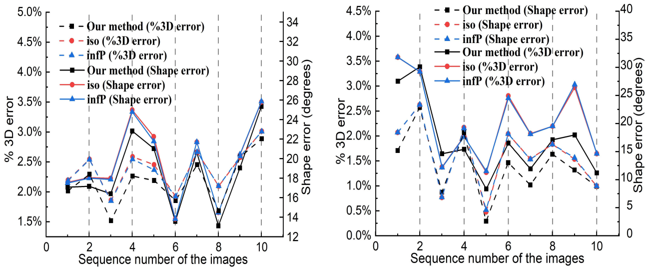
	Missing	**infP**	**ours**	**+3 edges**
S1	0%	1.16%-8.24°	1.00%-7.90°	**1.00%-7.90°**
	10%	1.34%-9.15°	**0.85%-6.73°**	0.91%-7.17°
	20%	1.71%-10.77°	1.02%-7.67°	**0.97%-7.54°**
	30%	2.25%-13.31°	1.14%-8.44°	**1.10%-8.39°**
	40%	2.74%-15.85°	1.36%-9.33°	**1.31%-8.99°**
	50%	3.41%-19.14°	**1.17%-7.91°**	1.25%-8.93°
S2	0%	0.61%-5.55°	0.60%-5.49°	**0.60%-5.49°**
	10%	0.60%-**5.36°**	0.61%-5.50°	**0.60**%-5.49°
	20%	0.58%-5.36°	0.56%-**5.32°**	**0.55**%-5.39°
	30%	0.67%-**6.08°**	0.67%-6.15°	**0.66**%-6.10°
	40%	0.65%-5.41°	0.62%-**5.30°**	**0.61**%-5.42°
	50%	0.71%-6.10°	0.68%-5.88°	**0.66%-5.79°**

When there are no missing data, the weights of all edges of the complete graph $\mathcal{G}_c$ are equal. Based on Theorem 2.2, the tree-connectivities of all spanning trees are the same. Hence, using our edge selection method with 9 edges (Kruskal's algorithm), the default sub-graph using MATLAB is the star-like graph with the first image as the center, which is the same as the **infP** and **iso** methods. For the 0% missing data, we find that balancing the importance of the MT and CS can help us to get better performance. For the missing data cases, commonly, introducing 1-3 additional edges, our method can obtain better results, because of the edge selection method.

B. *Experiments with real data*

1) Comparison of the NRSfM results: In this part, we conducted experiments with Hulk, T-shirt, Flag, Rug, and KinectPaper datasets to compare with other two methods.

a) T-shirt and Hulk datasets: The T-shirt and Hulk datasets are two public datasets [4] with 10 different images respectively involving the isometric deformable cloth and paper. Based on the full datasets and all features, the shape error and the % 3D error, compared with the others, are respectively shown in Fig. 6a (T-shirt) and Fig. 6b (Hulk).



(a) T-shirt dataset (b) Hulk dataset

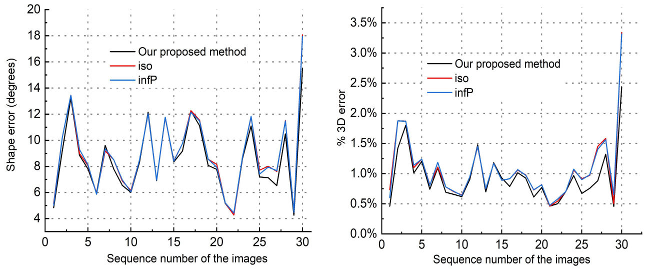
Fig. 6. Comparison using T-shirt and Hulk datasets. Our proposed method shows better performance than the other two methods in most images of the T-shirt and Hulk datasets in terms of the shape error and the % 3D error.

b) Flag dataset: The Flag dataset [25] is a semi-synthetic data with real objects and a virtual camera using

---

[5] The **infP** and **iso** methods cannot deal with the appearing points case, so we delete the features, which are not detected by the first image for these two methods, in the recovery shape and the error computation. Our method can easily deal with the appearing points, hence, these points are remained. In fact, if we conisder the errors of these deleting points for the **infP** and **iso** methods, our advantage shown in Table I and II will be larger.

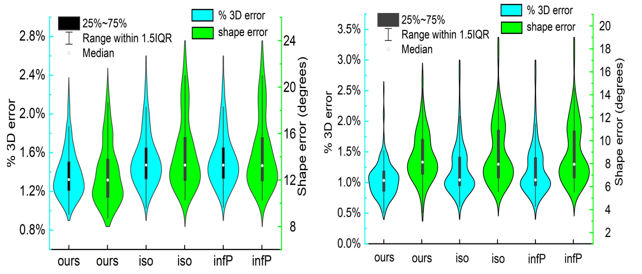
perspective projection. We use 30 images from the whole 450 frames to test the performance of our method and the others based on the full features. The comparison results of the shape error and the % 3D error are shown in Fig. 7.



(a) Shape error (b) % 3D error

Fig. 7. Comparison using Flag dataset: (a) shape error; (b) % 3D error. In this dataset, the mean % 3D errors of all images are respectively 0.9382% (**ours**), 1.0725% (**infP**), and 1.0698% (**iso**). The average shape errors are 8.5279° (**ours**), 8.9091° (**infP**), and 8.9210° (**iso**). Our method shows better performance than the other two methods for these datasets.

c) Rug dataset: The Rug dataset [7] is a public data with 159 images and 300 features showing the deformed rug. Because of the low frame-rate of the recorded sequences, the given correspondences are not very accurate and contain outliers. The comparison results using the full datasets and three methods are shown in Fig. 8a.



(a) Rug dataset (b) KinectPaper dataset

Fig. 8. Comparison of two datasets including shape error and % 3D error. In the Rug dataset, the mean % 3D errors of all images are respectively 1.3782% (**ours**), 1.5286% (**infP**), and 1.5318% (**iso**). The average shape errors are 12.5522° (**ours**), 14.2981° (**infP**), and 14.3206° (**iso**). In the KinectPaper dataset, the mean % 3D errors of all images are respectively 1.05% (**ours**), 1.24% (**infP**), and 1.21% (**iso**). The average shape errors are 8.57° (**ours**), 8.64° (**infP**), and 8.92° (**iso**). Our method shows best performance in all methods for these datasets.

d) KinectPaper dataset: The KinectPaper dataset is a video sequence of 191 frames and 1500 points of a isometrically deformed paper [8]. It is noted that this sequence contains outliers. Based on the full datasets and all features, using three methods, we present the statistical results of the shape error and the % 3D error for every image in Fig. 8b.

2) Changing data: We also present the comparison results with changing data for the T-shirt and Hulk datasets. Table II shows the mean % 3D error and shape error obtained on two datasets when 0-50% features are missing at every image.

The result shows that our method has the most robust performance among the three methods in terms of the data changing. We also find that too many additional edges (more than 10 new edges) may not greatly improve the result. The main reason is that only using the common feature number may select the poor image warps, introduce the bad measurements, and finally get the even low-accurate results.

3) Noisy data: For the T-shirt and Hulk datasets, we also present the comparison results with randomly generated noisy data. The presented results in Fig. 9 are the mean values of 10 times testings. From Fig. 9, we find that every method degrades when noise varying between 1-20 pixels is added. Our method shows good tolerance to noise and gives best performance. Even though the **infP** and **iso** methods transform the problem into two variables optimization corresponding to the first image improving the result robustness of the first image, for the rest images, they directly use the linear relationship shown in (14), which means that they sacrifice the performance of the rest images. Our method balances all information in all images using a strong graph optimization, which helps us to achieve the overall robust performance.

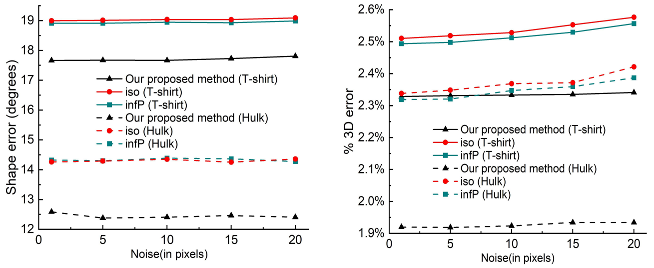


Fig. 9. Comparison for the noisy data case using T-shirt and Hulk datasets.

4) Computational complexity: In this part, we analysis the computational complexity of our method and present some numerical results. With no missing features, for a NRSfM problem with N_m images and N_p features on each image, the computational complexities of Kruskal's algorithm, k-ESP, and graph optimization are respectively $O(N_m^2 \log N_m)$, $O(kN_m^4)$, and $O(m_i(N_m N_p)^3)$.[6] Commonly, the NRSfM problem satisfies $N_m \ll N_p$ and k is small, so the computational complexity of our method is $O(m_i(N_m N_p)^3)$. The computational complexities of the **infP** and **iso** methods are both $O(N_p O(1)))$, where $O(1)$ means the computational complexity of the small-scale moment SDP programming. In theory, our method has the much higher computational cost. However, because of the highly-sparse Hessian matrix, for the real problems, our method shows the similar computational time, when N_m is small, compare with others. The computational time grows when the scale of the datasets grows. The numerical results are in Table III [7].

---

[6] m_i is the number of iteration in the trust-region-reflective method. It is noted that the computational complexity of the graph optimization is for the worst case with a dense matrix and does not consider the sparsity of the Hessian matrix. In fact, based on the measurements (14) and (15), their Jacobian matrices are highly sparse with only 3 or 4 non-zero elements in every row. Hence, the exact computational complexity of the graph optimization will be much for smaller than $O(m_i(N_m N_p)^3)$.

[7] We consider the feature matching and image warp as the front-end, so we do not include their computation time in Table III. Meanwhile, all running time results of our method are based on all-zero initialization, which causes the large convergence iteration. For the large datasets, like Rug and KinectPaper, if we use a few images (like first 4 images) to build the small graph optimization problem, solve it, and use the linear relationship (14) to predict $\bar{k}_1$ and $\bar{k}_2$ of the other images as initialization of the full graph optimization, the convergence iteration and the running time will be greatly reduced. For example, by this way, the running time of the KinectPaper will be 239.21s.

TABLE II

COMPARISON USING MISSING AND APPEARING DATA

	Missing	**infP**	**iso**	**ours**	**+1 edges**	**+2 edges**	**+3 edges**
T-shirt	0%	2.49%-18.91°	2.51%-18.99°	2.33%-17.66°	2.34%-17.64°	**2.29%-17.40°**	2.30%-17.53°
	10%	2.60%-20.65°	2.61%-20.66°	**2.29%-17.24°**	2.34%-17.58°	2.32%-17.92°	2.34%-18.23°
	20%	2.82%-19.49°	2.82%-19.44°	2.58%-18.20°	2.52%-18.13°	**2.49%-18.12°**	2.50%-18.23°
	30%	2.92%-20.44°	2.92%-20.46°	**2.40%-18.57°**	2.43%-18.84°	2.44%-19.00°	2.43%-19.00°
	40%	2.99%-21.25°	2.99%-21.30°	2.64%-**17.87°**	2.68%-18.29°	2.69%-18.50°	**2.61%**-17.97°
	50%	3.24%-20.85°	3.22%-20.78°	2.74%-**19.41°**	2.70%-19.41°	2.66%-19.86°	**2.63%**-19.91°
Hulk	0%	2.32%-14.32°	2.32%-14.27°	1.92%-12.38°	**1.92%-12.38°**	2.03%-12.49°	2.03%-12.98°
	10%	2.50%-14.33°	2.50%-14.27°	1.68%-11.00°	1.60%-10.69°	1.62%-10.79°	**1.58%-10.48°**
	20%	2.67%-14.35°	2.67%-14.28°	1.88%-11.80°	1.80%-11.60°	1.72%-11.45°	**1.72%-11.32°**
	30%	2.93%-14.39°	2.94%-14.40°	2.08%-12.77°	2.08%-12.93°	2.02%-12.58°	**1.97%-12.28°**
	40%	2.99%-14.98°	2.99%-14.99°	2.16%-12.44°	**2.08%-12.11°**	2.10%-12.32°	2.14%-12.42°
	50%	3.10%-16.68°	3.11%-16.66°	2.44%-15.26°	2.34%-14.87°	2.33%-14.82°	**2.31%-14.63°**

TABLE III

COMPARISON OF THE RUNNING TIME

	T-shirt	Hulk	Flag	Rug	KinectPaper
N_m/N_p	10/85	10/122	30/250	159/300	191/1500
infP (s)	8.85	8.88	22.40	58.90	327.79
iso (s)	18.36	25.62	57.67	127.47	613.32
ours (s)	7.68	8.30	16.15	410.56	1362.95

5) Dense reconstruct results: In this section, we generate and texture the dense point cloud based on the recovery shape. The partial dense NRSfM results for the T-shirt and Hulk dataset using Algorithm 3 are shown in Fig. 10.

(a) T-shirt dataset (b) Hulk dataset

Fig. 10. Dense NRSfM results of three images in two datasets (the upper figures are the 3D point cloud with the red 3D features and the lower figures are the corresponding monocular images). The compuational time of the whole datasets is 7.21s (T-shirt) and 9.16s (Hulk). The results show that our proposed dense reconstruct method is efficient.

VI. CONCLUSIONS

This letter presents a theoretical framework for solving the dense Iso-NRSfM problem with highly-accurate and robust performance. Based on the powerful sub-graph for the image warps and the sparse graph optimization, we can robustly deal with the changing and noisy data. We test our method on both synthetic and real datasets with different baseline viewpoints and deformations. The results show that our method has better performance than the other methods.

REFERENCES

[1] J. Lamarca, S. Parashar, A. Bartoli, and J. M. M. Montiel, “DefSLAM: Tracking and mapping of deforming scenes from monocular sequences,” arXiv preprint arXiv:1908.08918, 2019.

[2] J. Song, J. Wang, L. Zhao, S. Huang, and G. Dissanayake, “Misslam: Real-time large-scale dense deformable slam system in minimal invasive surgery based on heterogeneous computing,” *IEEE Robot. and Autom. Lett.*, vol. 3, no. 1, pp. 4068?4075, 2018.

[3] R. I. Hartley and A. Zisserman, “Multiple view geometry in computer vision,” Cambridge University Press, 2000.

[4] A. Chhatkuli, D. Pizarro, T. Collins, and A. Bartoli, “Inextensible non-rigid structure-from-motion by second-order cone programming,” *IEEE PAMI*, vol. 40, no. 10, pp. 2428-2441, 2017.

[5] P. F. Gotardo, and A. M. Martinez, “Kernel non-rigid structure from motion,” *IEEE Int. Conf. on Computer Vision*, 2011, pp. 802-809.

[6] Y. Dai, H. Li, and M. He, “A simple prior-free method for non-rigid structure-from-motion factorization,” *Int. J. of Comput. Vision*, vol. 107, no. 2, pp. 101-122, 2014.

[7] S. Parashar, D. Pizarro, and A. Bartoli, “Isometric non-rigid shape-from-motion with riemannian geometry solved in linear time,” *IEEE PAMI*, vol. 40, no. 10, pp. 2442-2454, 2017.

[8] S. Parashar, D. Pizarro, and A. Bartoli, “Isometric non-rigid shape-from-motion in linear time,” *IEEE Conf. on Comp. Vision and Pattern Recogn. (CVPR)*, 2016, pp. 4679-4687.

[9] A. Chhatkuli, D. Pizarro, and A. Bartoli, “Non-rigid shapefrom-motion for isometric surfaces using infinitesimal planarity,” *British Machine Vision Conf. (BMVC)*, 2014, pp. 1-12.

[10] S. Parashar, D. Pizarro, and A. Bartoli, “Local deformable 3D reconstruction with Cartan’s connections,” *IEEE Trans. PAMI*, 2019.

[11] R. Hartley and R. Vidal, “Perspective nonrigid shape and motion recovery,” *European Conf. on Comp. Vision*, 2008, pp. 276-289.

[12] L. Torresani, A. Hertzmann, and C. Bregler, “Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors,” *IEEE Trans. PAMI*, vol. 30, no. 5, pp. 878-892, 2008.

[13] I. Akhter, Y. Sheikh, S. Khan, and T. Kanade, “Nonrigid structure from motion in trajectory space,” *Adv. in Neural Inf. Process. Syst. (NIPS)*, 2009, pp. 41-48.

[14] P. F. U. Gotardo and A. M. Martinez, “Computing smooth time trajectories for camera and deformable shape in structure from motion with occlusion,” *IEEE PAMI*, vol. 33, no. 10, pp. 2051-2065, 2011.

[15] A. Agudo and F. M. Noguer, “Force-based representation for non-rigid shape and elastic model estimation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 9, pp. 2137-2150, 2017.

[16] B. Willimon, S. Hickson, I. Walker, and S. Birchfield, “An energy minimization approach to 3D non-rigid deformable surface estimation using RGBD data,” *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2012, pp. 2711-2717.

[17] A. Simon, J. C. Pret, and A.P. Johnson, “A fast algorithm for bottom-up document layout analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 3, pp. 273-277, 1997.

[18] Y. Chen, K. M. B. Lee, C. Yoo, and R. Fitch, “Broadcast your weaknesses: cooperative active pose-graph SLAM for multiple robots,” *IEEE Robot. and Autom. Lett.*, vol. 5, no 2, pp. 2200-2007, 2020.

[19] D. Pizarro, R. Khan, and A. Bartoli, “Schwarps: Locally projective image warps based on 2D schwarzian derivatives,” *Int. J. of Comput. Vision*, vol. 119, no. 2, pp. 93-109, 2016.

[20] K. Khosoussi, M. Giamou, G.S. Sukhatme, S. Huang, G. Dissanayake, and J.P. How, “Reliable graph topologies for SLAM,” *Int. J. Robot. Res.*, vol. 38, no. 2-3, pp. 260-298, 2019.

[21] A. Davis, “SUITESPARSE: A suite of sparse matrix software,” available at http://faculty.cse.tamu.edu/davis/suitesparse.html.

[22] A. Varol, M. Salzmann, E. Tola, and P. Fua, “Template-free monocular reconstruction of deformable surfaces,” *IEEE Conf. on Comp. Vision and Pattern Recogn. (CVPR)*, 2009, pp. 1811-1818.

[23] J. Taylor, A. D. Jepson, and K. N. Kutulakos, “Non-rigid structure from locally-rigid motion,” *IEEE Conf. on Comp. Vision and Pattern Recogn. (CVPR)*, 2010, pp. 2761-2768.

[24] S. Vicente and L. Agapito, “Soft inextensibility constraints for template-free non-rigid reconstruction,” *European Conf. on Comp. Vision (ECCV)*, 2012, pp. 426-440.

[25] R. White, K. Crane, and D. A. Forsyth, “Capturing and animating occluded cloth,” ACM *Transactions on Graphics*, vol. 26, no. 3, pp. 34, 2007.