Semester Project

Two-view Isometric Non-Rigid Shape-from-Motion

June 5, 2020

Author:
LONG Yuxuan

Supervisor:
Prof. Pascal Fua

Assistant:
Dr. Shaifali Parashar

Computation Science and Engineering
Ecole Polytechnique Fédérale de Lausanne

CVLab

Abstract

Non-Rigid Shape-from-Motion (NRSfM) usually requires more than two views to obtain a reasonable shape recovery. To break this limitation, we investigate the feasibility of two-view Isometric NRSfM (Iso-NRSfM). This is an ill-posed problem as there could exist almost infinite number of shapes that satisfy the same isometric deformation. For each point correspondence, we firstly solve a sextic monomial reconstruction equation based on [6], which yields multiple real roots for shape parameters (k_1, k_2). Then, the two-view Iso-NRSfM is reduced to a shape selection problem. We propose two main approaches. Locally, the desired shape parameters for each point correspondence could be selected individually by some explicit assumptions, like frontal parallelism. Globally, we propose to select the shape parameters that produce a globally smooth surface, with some designed regularization maintaining the local surface property. In the experiments, we show that both local and global approach can achieve similar reconstruction accuracy as the multiple-view Iso-NRSfM [6]. Though, the multiple-view Iso-NRSfM has better average performance, our two-view methods sometimes even outperform it. A remark is that our proposed local approach can have similar or even better performance as compared to global approach, meanwhile the time cost for the local approach is negligible. This demonstrates the two-view Iso-NRSfM can be solved in a cheap but decent manner.

Contents

1 Introduction

Shape-from-Motion or Structure-from-Motion (SfM) is an old but still popular research problem in computer vision: given multiple calibrated views (i.e. $N \geq 2$ views) taken from the same object, we want to recover the shape of the object. If the object is still (i.e. no deformation), then we have a rigid SfM (RSfM) problem, where given sufficient number of correspondences, one can find some certain recovery for both the object shape and camera poses. The problem becomes complicated once the object is deforming, known as non-rigid SfM (NRSfM). There is no definite way to solve NRSfM. Nevertheless, some prior assumptions about shape or motion can be made, e.g. low-rank shape [11], maximum depth [10] and inextensible surface [3]. In this work, we mainly follow the Isometric NRSfM (Iso-NRfSM) from [6] and assume isometric deformation and infinitesimal planarity. Empirically, the assumption of isometry can even work well under non-isometric deformation [7]. Specifically, Riemannian manifolds [6] are used to smoothly represent the 3D surface, embedded on the image space. Provided with warps, exploring the Christoffel Symbol of the manifolds gives the linear transformation between the local shape parameters (k_1, k_2) (i.e. the first order derivatives of inverse-depth) in different views. Then, the metric tensors, which embed the local geometric properties like area and lengths, are used to set up the reconstruction equation individually for each point correspondence. The reconstruction equation is consisted of two cubic equations [7] in two variables k_1, k_2, leading up to 6 real solutions for (k_1, k_2). Thus, $N \geq 3$ is required to output a single pair of (k_1, k_2), which is then used for normal integration. Indeed, $N \geq 3$ is a bottom-line for most NRSfM algorithms. This limits the application of NRSfM, since the point correspondences are not always available in many views, especially when non-sequential views are taken at large motions or different lighting conditions. As motivated by this, we will investigate the feasibility of Iso-NRSfM with $N = 2$.

As found in the experiment, for one pair of point correspondence, solving a reconstruction equation can usually give two possible shape parameters. Given n point correspondences for two views, we can roughly estimate that there are 2^n possible shapes that satisfy the isometric deformation. This means we can have almost infinite number of possible shapes. The two-view Iso-NRSfM is then a problem of properly selecting one shape out of the numerous choices, where some examples of selected shape are shown in Figure 2. In this work, we will explore both local and global method for shape selection. Locally, we can directly pick the shape parameters based on some heuristic of local geometric properties like minimum infinitesimal surface area. Globally, we can select the shape by maximizing the consistency of shape parameters, i.e. the shape parameters at one point correspondence should be similar to its neighbours. The global approach can be formulated to a Mixed-Integer Quadratic Programming (MIQP), which can be solved in cubic time complexity with proper convex relaxation. Our proposed methods have the following features:

1. Several heuristics based on some local geometric properties are proposed. Those can be used for local regularization in global approach and provide the flexibility for shape selection.

2. We empirically prove that both local and global approach can properly select

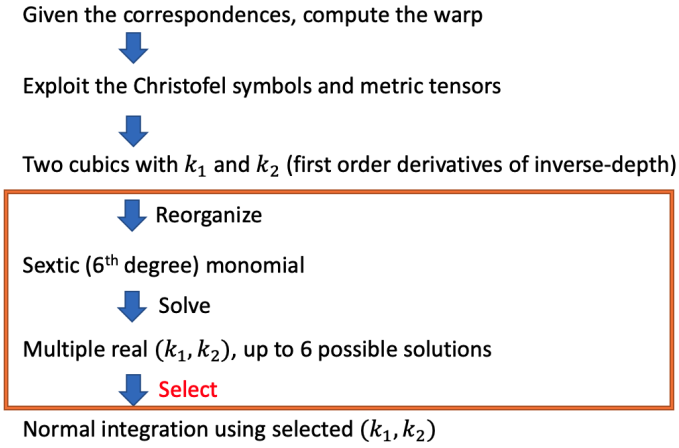
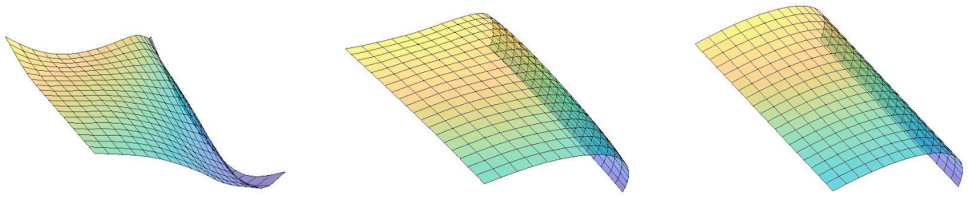


Figure 1: General flow of the two-view Iso-NRSfM. The orange box shows the main modification to the original Iso-NRSfM



(a) (b) (c)

Figure 2: Two-view NRSfM examples (a) reconstructed shape by randomly selected parameters; (b) reconstructed shape by our local approach **LN**; (c) ground truth

the shape, leading to a good two-view shape recovery which is even close to the result of multiple-view Iso-NRSfM. In other words, Iso-NRSfM can be now solvable from only $N \geq 2$ views.

3. Local approach almost does not have time consumption on shape selection, nevertheless it has similar performance as compared with global approach.

2 Review of Isometric NRSfM

2.1 Reconstruction Equations

Given the image coordinate $\mathbf{x} = \begin{pmatrix} u & v \end{pmatrix}^T$ in the image $\mathcal{I}_i$, the image embedding of the manifold $\mathcal{M}_i$ is:

$$\phi_i(\mathbf{x}) = \frac{1}{\beta_i(\mathbf{x})} \begin{pmatrix} \mathbf{x}^T & 1 \end{pmatrix}^T, \tag{1}$$

where $\beta_i(\mathbf{x})$ is the inverse depth function. The Jacobian of the embedding is computed as

$$\mathbf{J}_{\phi_i} = \frac{1}{\beta_i^2} \begin{pmatrix} \beta_i - u\beta_{i,1} & -u\beta_{i,2} \\ -v\beta_{i,1} & \beta_i - v\beta_{i,2} \\ -\beta_{i,1} & -\beta_{i,2} \end{pmatrix}. \tag{2}$$

Then the metric tensor is

$$\mathbf{g}[\phi_i(\mathbf{x})] = \mathbf{J}_{\phi_i}^T \mathbf{J}_{\phi_i} = \frac{1}{\beta_i^2} \begin{pmatrix} \epsilon k_1^2 - 2uk_1 + 1 & \epsilon k_1 k_2 - uk_2 - vk_1 \\ \epsilon k_1 k_2 - uk_2 - vk_1 & \epsilon k_2^2 - 2vk_2 + 1 \end{pmatrix}, \tag{3}$$

where $k_1 = \frac{\beta_{i,1}}{\beta_i}$, $k_2 = \frac{\beta_{i,2}}{\beta_i}$ and $\epsilon = u^2 + v^2 + 1$. Given $\mathbf{y} = \begin{pmatrix} \bar{u} & \bar{v} \end{pmatrix}^T$ in another image $\mathcal{I}_j$, the metric tensor for the embedding $\phi_j(\mathbf{y})$ can be determined similarly

$$\mathbf{g}[\phi_j(\mathbf{y})] = \frac{1}{\beta_j^2} \begin{pmatrix} \bar{E} & \bar{F} \\ \bar{F} & \bar{G} \end{pmatrix} = \frac{1}{\beta_j^2} \begin{pmatrix} \bar{\epsilon}\bar{k}_1^2 - 2\bar{u}\bar{k}_1 + 1 & \bar{\epsilon}\bar{k}_1\bar{k}_2 - \bar{u}\bar{k}_2 - \bar{v}\bar{k}_1 \\ \bar{\epsilon}\bar{k}_1\bar{k}_2 - \bar{u}\bar{k}_2 - \bar{v}\bar{k}_1 & \bar{\epsilon}\bar{k}_2^2 - 2\bar{v}\bar{k}_2 + 1 \end{pmatrix}, \tag{4}$$

where $\bar{k}_1 = \frac{\beta_{j,1}}{\beta_j}$, $\bar{k}_2 = \frac{\beta_{j,2}}{\beta_j}$ and $\bar{\epsilon} = \bar{u}^2 + \bar{v}^2 + 1$. Given the Jacobian of the warping η_{ji}, i.e. $\mathbf{J}_{\eta_{ji}} = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$, we denote

$$\begin{pmatrix} x_1 & x_2 \end{pmatrix}^T = \mathbf{J}_{\eta_{ji}}^T \begin{pmatrix} k_1 & k_2 \end{pmatrix}^T \tag{5}$$

From equating the Christoffel Symbols and the assumption of infinitesimal planarity [6], we have

$$\bar{k}_1 = x_1 + t_1, \bar{k}_2 = x_2 + t_2, \tag{6}$$

where $t_1 = -\partial_1 \mathbf{y}^2 \partial_{21}^2 \mathbf{x}^1 - \partial_2 \mathbf{y}^2 \partial_{21}^2 \mathbf{x}^2$ and $t_2 = -\partial_1 \mathbf{y}^1 \partial_{21}^2 \mathbf{x}^1 - \partial_2 \mathbf{y}^1 \partial_{21}^2 \mathbf{x}^2$. This together with (4) gives

$$\begin{aligned} \bar{E} &= \bar{\epsilon}(x_1 + t_1)^2 - 2\bar{u}(x_1 + t_1) + 1 \\ \bar{F} &= \bar{\epsilon}(x_1 + t_1)(x_2 + t_2) - \bar{u}(x_2 + t_2) - \bar{v}(x_1 + t_1) \\ \bar{G} &= \bar{\epsilon}(x_2 + t_2)^2 - 2\bar{v}(x_2 + t_2) + 1. \end{aligned}$$

Then, by the isometric mapping between the manifolds $\mathcal{M}_i$ and $\mathcal{M}_j$, the pull-back metric tensor [6] for the embedding ϕ_j is determined as:

$$\mathbf{g}[(\phi_i \circ \eta_{ji})(\mathbf{y})] = \mathbf{J}_{\eta_{ji}}^T \mathbf{g}[\phi_i] \mathbf{J}_{\eta_{ji}} = \frac{1}{\beta_i^2} \begin{pmatrix} E_p & F_p \\ F_p & G_p \end{pmatrix}, \tag{7}$$

where

$$\begin{aligned} E_p &= \epsilon x_1^2 - 2(au + bv)x_1 + (a^2 + b^2) \\ F_p &= \epsilon x_1 x_2 - (cu + dv)x_1 - (au + bv)x_2 + ac + bd \\ G_p &= \epsilon x_2^2 - 2(cu + dv)x_2 + (c^2 + d^2). \end{aligned}$$

Since $\mathbf{g}[(\phi_i \circ \eta_{ji})(\mathbf{y})] = \mathbf{g}[\phi_j(\mathbf{y})]$ by the isometric mapping, the ratios of their elements are equal:

$$\frac{E_p}{G_p} = \frac{\bar{E}}{\bar{G}}, \frac{F_p}{G_p} = \frac{\bar{F}}{\bar{G}}.$$

This gives two cubic equations in terms of only x_1 and x_2

$$E_p \bar{G} - \bar{E} G_p = 0 \tag{8}$$

$$F_p \bar{G} - \bar{F} G_p = 0. \tag{9}$$

The second cubic equation (9) is indeed linear in terms of x_1, i.e.

$$f_1 x_1 + f_2 = 0, \tag{10}$$

where $f_1(x_2)$ is a second order polynomial and $f_2(x_2)$ is a third order polynomial. $x_1 = -\frac{f_2}{f_1}$ can be substituted into (8), then multiplying f_1^2 on both sides gives an univariate 6th order polynomial equation (i.e. sextic equation).

There are at most 6 roots from the sextic equation. Some of the roots can be non-real numbers, which can be immediately excluded. However, it happens commonly with multiple real solutions, indicating many possible local shapes. In the experiments, we always have two real solutions for each reconstruction equation. This somehow lets us recall that decomposing the homography could also yield two possible surface normals and rotations [5]. We may intuitively guess that our assumption about isometry and infinitesimal planarity may be equivalent to modelling a local homography between two views. This still needs some theoretical justification.

2.2 Interpretation of the Shape Parameters

Prior to shape selection, we need to have a full understanding of k_1 and k_2. The normal of the deforming surface can be computed as the cross product of the columns of the Jacobian. So, from (2), we have the normal of the object surface

$$\begin{aligned}
\mathbf{n} &= \frac{\partial \phi_i(\mathbf{x})}{u} \times \frac{\partial \phi_i(\mathbf{x})}{v} \\
&= \frac{1}{\beta_i^2} \begin{pmatrix} 1 - uk_1 & -vk_1 & -k_1 \end{pmatrix}^T \times \begin{pmatrix} uk_2 & 1 - vk_2 & -k_2 \end{pmatrix}^T \\
&= \frac{1}{\beta_i^2} \begin{pmatrix} k_1 & k_2 & 1 - k_1 u - k_2 v \end{pmatrix}^T .
\end{aligned} \tag{11}$$

Namely,

$$\mathbf{n} \parallel \begin{pmatrix} k_1 & k_2 & 1 - k_1 u - k_2 v \end{pmatrix}^T . \tag{12}$$

This gives

$$\begin{aligned}
n_2 k_1 - n_1 k_2 &= 0 \\
n_3 k_1 - n_1 (1 - k_1 u - k_2 v) &= 0 \\
n_3 k_2 - n_2 (1 - k_1 u - k_2 v) &= 0.
\end{aligned}$$

At most time, we have $n_1 u + n_2 v + n_3 \neq 0$ unless the camera ray intersects at the smooth boundary of the object surface. Under the condition $n_1 u + n_2 v + n_3 \neq 0$, we have

$$k_1 = \frac{n_1}{n_1 u + n_2 v + n_3} \tag{13}$$

$$k_2 = \frac{n_2}{n_1 u + n_2 v + n_3}. \tag{14}$$

Let us denote

$$k_3 = 1 - k_1 u - k_2 v \tag{15}$$

Substituting (13) and (14) into (15) gives

$$k_3 = \frac{n_3}{n_1 u + n_2 v + n_3} \tag{16}$$

Those above provides significantly useful information, which we would discuss in several perspectives.

Normal Integration (13) and (14) can be used directly for normal integration of perspective camera. To explain this, we have another interpretation of k_1 and k_2

$$k_1 = \frac{\frac{\partial \beta}{\partial u}}{\beta} = \frac{\partial \log \beta}{\partial u} = -\frac{\partial \log z}{\partial u} \tag{17}$$

$$k_2 = \frac{\frac{\partial \beta}{\partial v}}{\beta} = \frac{\partial \log \beta}{\partial v} = -\frac{\partial \log z}{\partial v}, \tag{18}$$

where z is the depth such that $z = \frac{1}{\beta}$. Specifically, we can integrate the partial derivatives $\frac{\partial \log z}{\partial u} = -k_1$ and $\frac{\partial \log z}{\partial v} = -k_2$ to compute the logarithm of depth [9].

Visibility In practice, we can use Lemma 2.1 to verify the case of occlusion, by simply looking at the sign of k_3. Note that the the computed k_3 can be negative accidentally due to the numerical error and measurement error.

Lemma 2.1. *If a 3D point* $\mathbf{X}$ *with the normal* $(n_1, n_2, n_3)^T$ *on the smooth object surface (except the boundary) is visible from the camera then* $\frac{n_3}{n_1 u + n_2 v + n_3} > 0$*, where* $(u, v, 1)^T$ *is projection of* $\mathbf{X}$ *onto the image plane.*

Proof. Let the surface normal be defined as pointing outwards. Then, a visible surface must have the property that $n_3 < 0$ so that the object stands in front of the camera. There could be an infinitesimal planar surface that locally approximates the neighbourhood of $\mathbf{X}$. Suppose that planar surface has a distance d from the camera origin. With $\|\mathbf{n}\| = 1$, then

$$\mathbf{n}^T \mathbf{X} = -d$$

So we have $n_1 u + n_2 v + n_3 = \beta \mathbf{n}^T \mathbf{X} = -d\beta < 0$. So, $k_3 = \frac{n_3}{n_1 u + n_2 v + n_3} > 0$. Note that if we define the surface normal as pointing inwards, the result remains unchanged. □

Numerical Consistency Among the neighbourhood of an image point, the values of k_1 and k_2 can be numerically inconsistent, though a very smooth surface is provided. This disobeys the common intuition.

A simple example can be the reconstruction of a single plane with the special property $n_2 = 0$ $(n_1, n_3 > 0)$. For this case, we have $k_1 = \frac{n_1}{n_1 u + n_3}$ and $k_2 = 0$. When u approaches to $-\frac{-n_3}{n_1}$, $n_1 u + n_3$ tends to be small (i.e. near the surface boundary), which makes large k_1. So, a little increment in u can cause sudden decrease in magnitude of k_1. Note that approaching to the boundary of an infinite planar surface is impossible. However, this example can appear in practice when the planar surface is almost perpendicular to the image plane, i.e. frontal perpendicular, so that the shape parameters (k_1, k_2) are very sensitive with respect to (u, v).

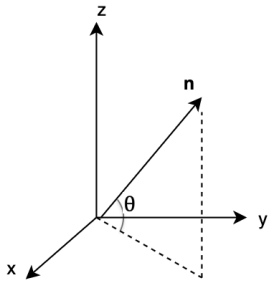


Figure 3: Visualization of normal with respect to image plane

3 Two-view NRSfM—Local Approach

Suppose there are n point correspondences in the given two views, i.e. n equations to solve. For every equation, we solve it for x_2 and extract only the real roots, then the corresponding x_1 can be computed by (10). For every (x_1, x_2), (k_1, k_2) can be determined by (5).

For the ith equation, we collect multiple (k_1, k_2) as the columns of $\mathbf{k}_i$, so that $\mathbf{k}_i \in \mathbb{R}^{2\times m_i}$. Since we know that one of the column of $\mathbf{k}_i$ should a desired local shape, now suppose $\mathbf{v}_i \in \mathbb{R}^{m_i}$ as the indicator vector (i.e. $\mathbf{v}_i \in \{0,1\}^{m_i}$), so that $\mathbf{k}_i\mathbf{v}_i$ should return the most appropriate shape parameters. The problem of shape selection is then to determine $\{\mathbf{v}_i\}_{i=1}^n$. For the local approach, we aim to determine $\mathbf{v}_i$ individually.

3.1 As-Parallel-As-Possible

We now propose a novel heuristic called as-parallel-as-possible (**APAP**), based on the frontal parallel assumption, that enables the infinitesimal planar surface to be as parallel to the image plane as possible. This heuristic could be useful at many cases, since people usually take photos of deforming objects without much tilt or slant.

Lemma 3.1. *Ideally, i.e. no measurement errors, minimizing* $(\frac{k_1}{k_3})^2+(\frac{k_2}{k_3})^2$ *is equivalent to maximizing the angle* θ *between the surface normal* $\mathbf{n}$ *and the image plane. Namely, the surface plane is most parallel to the camera plane.*

Proof. As referred to Figure 3, the angle θ can be simply computed as

$$\theta = \arctan \frac{|n_3|}{\sqrt{n_1^2+n_2^2}} = \arctan \frac{1}{\sqrt{(n_1/n_3)^2+(n_2/n_3)^2}}.$$

From (13), (14) and (16), we can notice that, since $n_3 \neq 0$, we have

$$\frac{k_1}{k_3} = \frac{n_1}{n_3}, \frac{k_1}{k_3} = \frac{n_2}{n_3}.$$

Those obviously give

$$\theta = \arctan \frac{1}{\sqrt{(k_1/k_3)^2+(k_2/k_3)^2}}. \tag{19}$$

□

From the Lemma 3.1, the local approach **APAP** should seek the minimal of $(\frac{k_1}{k_3})^2 + (\frac{k_2}{k_3})^2$, i.e.

$$(\mathbf{v}_i)_j = \delta_{j,\arg\min_q \|(\mathbf{k}_i)_{:,q}\|_2^2/(1-\mathbf{x}^T(\mathbf{k}_i)_{:,q})^2}, \tag{20}$$

where δ is the Kronecker delta function.

3.2 Least Change of Depth

APAP may produce the flat surface, i.e. small change of depth with respect to the camera plane. Consider another case where the surface has least change of depth with respect to the image space, we can simply deduce that $\frac{\partial z}{\partial u}$ and $\frac{\partial z}{\partial v}$ should have both small magnitude. Hence, from (17) and (18), k_1 and k_2 should be small relative to the depth. Under this intuition, we propose a simple local method: we choose the shape parameters by searching the least squared l_2 norm of (k_1, k_2). This method, denoted as **LN** for simplicity, can be simply described as

$$(\mathbf{v}_i)_j = \delta_{j,\arg\min_q \|(\mathbf{k}_i)_{:,q}\|_2^2}. \tag{21}$$

Note that, smaller $k_1^2 + k_2^2$ does not necessarily imply greater smoothness of surface in the 3D space, since only the second order derivatives may directly reflect the local surface smoothness (or the curvature). Nevertheless, **LN** may be able to contract the shape. To intuitively interpret **LN**, from (13) and (14), we have:

$$k_1^2 + k_2^2 = \frac{n_1^2 + n_2^2}{(n_1 u + n_2 v + n_3)^2}.$$

Small value of $k_1^2 + k_2^2$ hence implies two possible conditions: first, $n_1^2 + n_2^2$ should be small, which indicates that the infinitesimal surface plane tends to be parallel to the camera plane, as referred to Lemma 3.1; second, $n_1 u + n_2 v + n_3$ should be large, which means the surface normal tends to be parallel with the camera ray. This interprets how the surface can have least change of depth with respect to the image space. Besides, the first condition is similar to the heuristic of **APAP**, and the second condition shows the main difference between **LN** and **APAP**. This difference is attributed to the additional division of k_3^2 in the formulation of **APAP**, so that the presence of camera ray is simply ignored by **APAP**.

3.3 Minimum Surface Area

Being simple is nice. This encourages us to propose another heuristic: minimum surface area. The surface area can be indeed measured via the metric tensor:

$$E_{\text{membrane}} = \int\int \sqrt{\det(\mathbf{g})}\, du\, dv. \tag{22}$$

Minimizing (22) resembles the natural property of the surface of a clamped soap bubble, hence it can be called membrane energy [1]. Locally near one correspondence, we can minimize the infinitesimal surface area by selecting the shape parameters with least determinant of the metric tensor. Note that we can only determine the metric tensor up to some unknown scale, as referred to (3). So we can compute

$\mathbf{G} = \beta^2\mathbf{g}$. Since $\mathbf{G}$ is indeed a function of (k_1, k_2) for some fixed image coordinates, this novel approach can be written as

$$(\mathbf{v}_i)_j = \delta_{j,\arg\min_q \det(\mathbf{G}((\mathbf{k}_i)_{:,q}))}. \tag{23}$$

We denote this local method as **MSA**, i.e. minimum surface area. This heuristic can be somehow related to **LN**, since shape with least change of depth can imply small surface area. As known for shape smoothing, the membrane energy (22) is usually replaced by its linearized approximation, leading to the Dirichlet energy [1]

$$E_{\text{Dirichlet}} = \int\int \left\|\frac{\partial\phi}{\partial u}\right\|_2^2 + \left\|\frac{\partial\phi}{\partial v}\right\|_2^2 du\, dv = \int\int \operatorname{Tr}(\mathbf{g})\, du\, dv. \tag{24}$$

From (2), we further derive that

$$\operatorname{Tr}(\mathbf{g}) = \frac{1}{\beta^2}\left(\epsilon(k_1^2 + k_2^2) + 2k_3\right). \tag{25}$$

This indicates that minimizing (24) is equivalent to seeking the minimal of $\epsilon(k_1^2 + k_2^2) + 2k_3$. This almost coincides with **LN**, where the only difference is the term k_3. However, it may be unnecessary to involve the term k_3, since

1. k_3 should be positive to enable visibility, as indicated in Lemma 2.1.
2. k_3 should be as large as possible in order to satisfy the frontal parallel heuristic, as referred to Lemma 3.1.

Once the term k_3 is removed from (25), minimizing the Dirichlet energy is now reduced to minimizing $k_1^2 + k_2^2$. In short, **LN** is an approximated but improved version of **MSA**.

3.4 Visibility Condition

From Lemma 2.1, we know that if the object surface does not have occlusion, then $k_3 > 0$. The visibility condition $k_3 > 0$ can be applied prior to the local shape selection in order to filter out some real roots. This is expected to boost the reconstruction quality.

Besides this, $(\bar{k}_1, \bar{k}_2)$ may be also used to guide the shape selection. This requires the duplication of the penalty term on the another frame. Consider the example of **LN**, to utilize the information on both views, we can now seek the minimal of $k_1^2 + k_2^2 + \bar{k}_1^2 + \bar{k}_1^2$. **MSA** and **APAP** can be also modified in this way.

4 Two-view NRSfM—Global Approach

The global approach aims to find and optimize $\{\mathbf{v}_i\}_{i=1}^n$ collectively. The smoothness of shape parameters can be simply considered as one crucial clue. The smoothness of k_1 and k_2 among the neighbourhood can be considered, but they are sometimes numerically inconsistent as discussed previously. A more direct way is to maximize the surface smoothness, which turns our attention to the surface normals.

After solving all the polynomials, k_1 and k_2 are all known and the corresponding surface normals can be simply computed. From all the alternative shape parameters at one point correspondence, we may obtain m_i possible normals.

We now only consider the surface normals on the reference view (i.e. first frame). Given multiple unit normals $\{\mathbf{n}_j^{(i)}\}_{j=1}^{m_i}$ for the ith correspondence in one image, ideally we can select one desired normal as

$$\mathbf{n}^{(i)} = \mathbf{N}_i \mathbf{v}_i. \tag{26}$$

where $\mathbf{N}_i = \begin{pmatrix} \mathbf{n}_1^{(i)} & \mathbf{n}_2^{(i)} & \dots & \mathbf{n}_{m_i}^{(i)} \end{pmatrix}$. Let $m = \sum_{i=1}^{n} m_i$. Then we stack $\{\mathbf{v}_i\}_{i=1}^{n}$ into a long vector $\mathbf{v}$, i.e.

$$\mathbf{v} = \begin{pmatrix} \mathbf{v}_1^T & \dots & \mathbf{v}_n^T \end{pmatrix}^T \in \mathbb{R}^m.$$

In this way, all choices are included in $\mathbf{v}$. We collect all normalized surface normals into three matrices $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_3 \in \mathbb{R}^{n \times m}$, which correspond to three distinct components of the surface normal. Specifically, we have

$$\mathbf{S}_k = \begin{pmatrix} (\mathbf{N}_1)_{k,:} & & & \\ & (\mathbf{N}_2)_{k,:} & & \\ & & \ddots & \\ & & & (\mathbf{N}_n)_{k,:} \end{pmatrix}, \forall k \in \{1, 2, 3\}.$$

For instance, $\mathbf{S}_1\mathbf{v}$ would give all selected first components of the surface normals. Those n selected normals can be hence expressed as

$$\begin{pmatrix} \mathbf{n}^{(1)} & \mathbf{n}^{(2)} & \dots & \mathbf{n}^{(n)} \end{pmatrix}^T = \begin{pmatrix} \mathbf{S}_1\mathbf{v} & \mathbf{S}_2\mathbf{v} & \mathbf{S}_3\mathbf{v} \end{pmatrix}. \tag{27}$$

4.1 Graph Construction

Around a point on the surface, we can find its neighbours and assume that they possess similar geometric properties, like the normals. Given two image coordinates $\mathbf{x}_i$ and $\mathbf{x}_j$, the geometric simiarlity on the ith manifold can be approximately measured by a Gaussian kernel

$$s_{ij} = \exp \frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}. \tag{28}$$

An undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ can be hence created with the edge weight defined in (28). And the edge e_{ij} is connected only if the squared distance is smaller than some threshold. Hence the adjacency matrix $\mathbf{W}$ is defined as

$$W_{ij} = \begin{cases} \exp \frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}, & \text{if } \|\mathbf{x}_i - \mathbf{x}_j\|^2 \leq t \\ 0, & \text{otherwise} \end{cases} \tag{29}$$

The threshold t can be set as scaled average of squared distance with some ratio. Given a graph signal $\mathbf{k}$, its smoothness in the graph can be measured by

$$\mathbf{k}^T \mathbf{L} \mathbf{k} = \frac{1}{2} \sum_{i,j} W_{ij} (k_i - k_j)^2, \tag{30}$$

where $\mathbf{L}$ is the unnormalized Laplacian that

$$\mathbf{L} = \mathbf{D} - \mathbf{W}. \tag{31}$$

Note $\mathbf{D}$ is the diagonal matrix with $D_{ii} = \sum_j W_{ij}$.

4.2 Energy Terms

To maximize the surface smoothness, we aim to minimize the distance between normals among the neighbourhood. Namely, an energy term is given for smoothness:

$$E_{\mathrm{s}} = \frac{1}{2n} \sum_{e_{ij} \in \mathcal{E}} W_{ij} \left\| \mathbf{n}^{(i)} - \mathbf{n}^{(j)} \right\|_2^2, \tag{32}$$

Note that (32) can be interpreted geometrically when the normals are unit vectors:

$$\left\| \mathbf{n}^{(i)} - \mathbf{n}^{(j)} \right\|_2^2 = \left\| \mathbf{n}^{(i)} \right\|_2^2 + \left\| \mathbf{n}^{(i)} \right\|_2^2 - 2\langle \mathbf{n}^{(i)}, \mathbf{n}^{(j)} \rangle = 2 - 2\cos \angle(\mathbf{n}^{(i)}, \mathbf{n}^{(j)}).$$

Therefore, minimizing (32) is equivalent to maximizing the sum of weighted cosines. (32) is indeed a quadratic function of $\mathbf{v}$, since from (27) we have

$$E_{\mathrm{s}} = \frac{1}{2n} \sum_{e_{ij} \in \mathcal{E}} W_{ij} \sum_{k=1}^{3} ((\mathbf{S}_k \mathbf{v})_i - (\mathbf{S}_k \mathbf{v})_j)^2.$$

Given the Laplacian matrix $\mathbf{L}$, from (30), we hence have

$$\begin{aligned} E_{\mathrm{s}} &= \frac{1}{2n} \sum_{k=1}^{3} \sum_{e_{ij} \in \mathcal{E}} W_{ij} ((\mathbf{S}_k \mathbf{v})_i - (\mathbf{S}_k \mathbf{v})_j)^2 \\ &= \frac{1}{n} \left((\mathbf{S}_1 \mathbf{v})^T \mathbf{L} (\mathbf{S}_1 \mathbf{v}) + (\mathbf{S}_2 \mathbf{v})^T \mathbf{L} (\mathbf{S}_2 \mathbf{v}) + (\mathbf{S}_3 \mathbf{v})^T \mathbf{L} (\mathbf{S}_3 \mathbf{v}) \right) \\ &= \frac{1}{n} \mathbf{v}^T (\mathbf{S}_1^T \mathbf{L} \mathbf{S}_1 + \mathbf{S}_2^T \mathbf{L} \mathbf{S}_2 + \mathbf{S}_3^T \mathbf{L} \mathbf{S}_3) \mathbf{v}. \end{aligned} \tag{33}$$

When the surface is very flat, e.g. a single plane, the computed normals do not vary much, so maximizing the surface smoothness may not guide the shape selection. To deal with it, we add a regularization term which follows the local approach **APAP**:

$$\begin{aligned} E_{\mathrm{apap}} &= \frac{1}{n} \sum_{i=1}^{n} \left(\frac{(\mathbf{n}^{(i)})_1}{(\mathbf{n}^{(i)})_3} \right)^2 + \left(\frac{(\mathbf{n}^{(i)})_2}{(\mathbf{n}^{(i)})_3} \right)^2 \\ &= \frac{1}{n} \left\| \mathbf{R}_1 \mathbf{v} \right\|_2^2 + \frac{1}{n} \left\| \mathbf{R}_2 \mathbf{v} \right\|_2^2, \end{aligned} \tag{34}$$

where $\mathbf{R}_1 = \mathbf{S}_1 \oslash \mathbf{S}_3$ and $\mathbf{R}_2 = \mathbf{S}_2 \oslash \mathbf{S}_3$ and $\oslash$ denotes for the element-wise non-zero division. Note that this term (34) is also quadratic and forces the surface normal to be as parallel as possible to the image plane.

Under the large motion of the object, **APAP** is no longer a good heuristic, since the surface can have large tilting angle. **LN** could be considered here to compensate

the weakness of **APAP**, since **LN** may help to contract the shape as opposed to flattening. We then propose another local energy term

$$E_{\text{ln}} = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{k}_i \mathbf{v}_i\|_2^2. \tag{35}$$

In practice, the reconstructed surface can be 'twisted', as sometimes the computed normals may point away from the camera. Namely, the reconstructed surface has occlusion problem. From Lemma 2.1, we then create a new energy term that forces the surface to be visible:

$$\begin{aligned} E_{\text{visb}} &= \sum_{i=1}^{n} \min\left((\mathbf{n}^{(i)})_3, 0\right)^2 \\ &= \|\min\left(\mathbf{S}_3, 0\right)\mathbf{v}\|_2^2, \end{aligned} \tag{36}$$

which only penalizes on the negative third component of the selected normals. The energy terms mentioned above only relate to the reference frame. By using the computed $(\bar{k}_1, \bar{k}_2)$ on the second view, we can create the energy terms $\bar{E}_{\text{s}}$, $\bar{E}_{\text{apap}}$, $\bar{E}_{\text{ln}}$ and $\bar{E}_{\text{visb}}$ which force the smoothness, local parallelism and visibility of the second surface, to further optimize $\mathbf{v}$.

4.3 MIQP

In this global approach, an important constraint is to force $\mathbf{v}$ to be always binary so that $\mathbf{v}$ can directly indicate the selected shape parameters. It can be observed that

$$\mathbf{v}_i \in \{0,1\}^{m_i}, \|\mathbf{v}_i\|_0 = 1 \iff \mathbf{1}^T\mathbf{v}_i = 1, \mathbf{v}_i \succeq 0, \|\mathbf{v}_i\|_0 = 1 \tag{37}$$

Since $\mathbf{v}$ stacks all $\mathbf{v}_i$, we apply a similar constraint on $\mathbf{v}$: $\mathbf{B}\mathbf{v} = \mathbf{1}_{n\times 1}, \mathbf{v} \succeq 0$ and $\|\mathbf{v}\|_0 = n$, where

$$\mathbf{B} = \begin{pmatrix} \mathbf{1}_{1\times m_1} & & & \\ & \mathbf{1}_{1\times m_2} & & \\ & & \ddots & \\ & & & \mathbf{1}_{1\times m_n} \end{pmatrix}.$$

These above give a problem of mixed-integer quadratic programming (MIQP):

$$\begin{aligned} \min_{\mathbf{v}\in\mathbb{R}^m} \quad & \frac{1}{2}(E_{\text{s}} + \bar{E}_{\text{s}}) + \frac{c_1}{2}(E_{\text{apap}} + \bar{E}_{\text{apap}}) + \frac{c_2}{2}(E_{\text{ln}} + \bar{E}_{\text{ln}}) + \frac{c_3}{2}(E_{\text{visb}} + \bar{E}_{\text{visb}}) \\ \text{s.t.} \quad & \mathbf{B}\mathbf{v} = \mathbf{1}_{n\times 1}, \mathbf{v} \succeq \mathbf{0}, \|\mathbf{v}\|_0 = n, \end{aligned} \tag{38}$$

where c_1, c_2, c_3 are some positive weights.

4.4 Simple Convex Relaxation

MIQP is a decision problem and proven to be NP-complete [4]. Therefore, it would be very expensive to solve MIQP without any relaxation. A simple way to relax

(38) is to drop the non-convex constraint (i.e. sparsity constraint), then the problem becomes a standard quadratic programming

$$\begin{aligned} \min_{\mathbf{v}\in\mathbb{R}^m} \quad & \frac{1}{2}\mathbf{v}^T\mathbf{A}\mathbf{v} \\ \text{s.t.} \quad & \mathbf{B}\mathbf{v} = \mathbf{1}_{n\times 1}, \mathbf{v} \succeq 0, \end{aligned} \tag{39}$$

where $\mathbf{A} \in \mathbb{R}^{m\times m}$ summarizes the sum of the quadratic energy terms. Due to the inequality constraint, (38) does not have a closed-form solution. A common way to deal with the inequality constraint is to find the active set that yields a minimal but feasible solution, which usually brings some computational cost. A cheaper option is to directly ignore the inequality constraint in (39), then we can solve a KKT equation

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0}_{n\times n} \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mu \end{pmatrix} = \begin{pmatrix} \mathbf{0}_{m\times 1} \\ \mathbf{1}_{n\times 1} \end{pmatrix}, \tag{40}$$

where $\mu \in \mathbb{R}^n$ is the dual variable for the equality constraint. This KKT equation can be solved explicitly:

$$\mathbf{v}^\star = \mathbf{A}^{-1}\mathbf{B}^T(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T)^{-1}\mathbf{1}_{n\times 1}. \tag{41}$$

Finally, we could convert $\mathbf{v}^\star$ into the binary indicator by piece-wise finding the maximum value in $\mathbf{v}_i^\star$. Under this simple convex relaxation, we denote (41) as the **QP** solver to the global approach.

4.5 A Better Relaxation

The removal of l_0 constraint may lead to sub-optimal solution. Particularly, we should notice that the energy terms are formulated based on assuming $\mathbf{v}$ being binary. [12] proposes an approximation method for l_0 norm that $\|\mathbf{v}\|_0 \approx \|\mathbf{v}\|_1 - \|\mathbf{v}\|_2$. Inspired by this, the problem (38) is further relaxed as

$$\begin{aligned} \min_{\mathbf{v}\in\mathbb{R}^m} \quad & \frac{1}{2}\mathbf{v}^T\mathbf{A}\mathbf{v} + \lambda(\|\mathbf{v}\|_1 - \frac{s}{2}\|\mathbf{v}\|_2^2) \\ \text{s.t.} \quad & \mathbf{B}\mathbf{v} = \mathbf{1}_{n\times 1}, \end{aligned} \tag{42}$$

where in practice we set $s \in (0, 1)$ in order to constrain $\mathbf{v}$ between 0 and 1.

We can notice that the inequality constraint has been removed. This can reduce significant computation time. Indeed, this new formulation is well supported by the Lemma 4.1.

Lemma 4.1. *Suppose there exists a set of global minimizers $\mathcal{P}^\star$ for problem (38). Let $\mathbf{v}^\star = \mathbf{v}^\star(\lambda)$ be the global minimizer of the problem (42), dependent on λ. Then, those two problems are related as*

$$\lim_{\lambda\to+\infty} \mathbf{v}^\star(\lambda) \ \in \mathcal{P}^\star. \tag{43}$$

Proof. $\mathbf{v}^\star$ is composed of $\mathbf{v}_i^\star$, then $\mathbf{B}\mathbf{v}^\star = \mathbf{1}$ implies that $\mathbf{1}^T\mathbf{v}_i^\star = 1$ for all i. As $\lambda \to +\infty$, $\|\mathbf{v}_i^\star\|_1$ is minimized, i.e. $\|\mathbf{v}_i^\star\|_1 = 1$, which implies that $\mathbf{1} \succeq \mathbf{v}_i^\star \succeq \mathbf{0}$. At the same time, $\|\mathbf{v}_i^\star\|_2^2$ is maximized, so that $\mathbf{v}_i^\star$ must be binary. This indicates that, as $\lambda \to +\infty$, $\mathbf{v}^\star$ is forced to stay in the feasible set of the problem (38). Note that $\mathbf{v}^\star$ should also minimize the energy terms when $\mathbf{v}^\star$ satisfies all constraints in (38) as $\lambda \to +\infty$. This simply means that (43) holds to be true. □

In the experiments, large λ has been found leading unstable results since the term $-\lambda\frac{s}{2}\|\mathbf{v}\|_2^2$ brings some non-convexity. As $\mathbf{A}$ is usually strictly positive definite, we can set a small positive λ so that (42) remains convex. In the experiments, setting $\lambda = 10$ can already give a decent approximation to the original MIQP problem (38). The problem (42) involves the $l1$ norm in the objective, so we employ scaled ADMM [2] to tackle with this non-smooth term. By introducing a new variable $\mathbf{w}$, the problem (42) is rewritten as

$$\begin{aligned} \min_{\mathbf{v}\in\mathcal{V},\mathbf{w}\in\mathbb{R}^m} \quad & \frac{1}{2}\mathbf{v}^T(\mathbf{A} - s\lambda\mathbf{I})\mathbf{v} + \lambda\|\mathbf{w}\|_1 \\ \text{s.t.} \quad & \mathbf{w} = \mathbf{v}, \end{aligned} \tag{44}$$

where $\mathcal{V} = \{\mathbf{v} \in \mathbb{R}^m \mid \mathbf{B}\mathbf{v} = \mathbf{1}_{n\times 1}\}$. Hence, the augmented Lagrangian is

$$L_\tau(\mathbf{v}, \mathbf{w}, \mu) = \frac{1}{2}\mathbf{v}^T(\mathbf{A} - s\lambda\mathbf{I})\mathbf{v} + \lambda\|\mathbf{w}\|_1 + \frac{\tau}{2}\|\mathbf{w} - \mathbf{v} + \mu\|_2^2, \tag{45}$$

where μ is the dual variable. ADMM updates are then given as

$$\mathbf{v}^{(k+1)} = \arg\min_{\mathbf{v}\in\mathcal{V}} L_\tau(\mathbf{v}, \mathbf{w}^{(k)}, \mu^{(k)}) \tag{46}$$

$$\mathbf{w}^{(k+1)} = \arg\min_{\mathbf{w}\in\mathbb{R}^m} L_\tau(\mathbf{v}^{(k+1)}, \mathbf{w}, \mu^{(k)}) \tag{47}$$

$$\mu^{(k+1)} = \mu^{(k)} + \mathbf{w}^{(k+1)} - \mathbf{v}^{(k+1)}, \tag{48}$$

and we simply set $\mu^{(0)} = \mathbf{0}$, and $\mathbf{w}^{(0)}$ as the solution from the local approach. For the update in $\mathbf{v}$, it can be solved simply via the KKT equation

$$\begin{pmatrix} \mathbf{A} + (\tau - s\lambda)\mathbf{I} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0}_{n\times n} \end{pmatrix} \begin{pmatrix} \mathbf{v}^{(k+1)} \\ \mathbf{q} \end{pmatrix} = \begin{pmatrix} \tau(\mathbf{w}^{(k)} + \mu^{(k)}) \\ \mathbf{1}_{n\times 1} \end{pmatrix}, \tag{49}$$

where $\mathbf{q} \in \mathbb{R}^n$ is the dual variable for the equality constraint. From this equation, we notice that τ should be set such that $\tau > s\lambda$, so that the linear system is warranted to be invertible and numerically stable. Solving (49) analytically gives

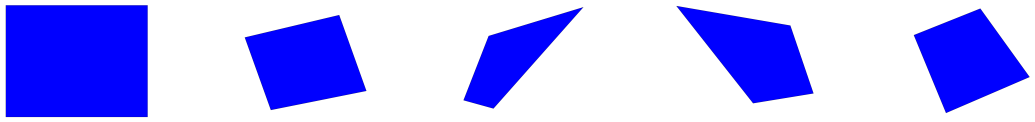
$$\mathbf{v}^{(k+1)} = \tau\mathbf{C}(\mathbf{w}^{(k)} + \mu^{(k)}) + \mathbf{b}, \tag{50}$$

where

$$\mathbf{C} = \tilde{\mathbf{A}}^{-1} - \tilde{\mathbf{A}}^{-1}\mathbf{B}^T(\mathbf{B}\tilde{\mathbf{A}}^{-1}\mathbf{B}^T)^{-1}\mathbf{B}\tilde{\mathbf{A}}^{-1}, \tag{51}$$

$$\mathbf{b} = \tilde{\mathbf{A}}^{-1}\mathbf{B}^T(\mathbf{B}\tilde{\mathbf{A}}^{-1}\mathbf{B}^T)^{-1}\mathbf{1}_{n\times 1}, \tag{52}$$

$$\tilde{\mathbf{A}} = \mathbf{A} + (\tau - s\lambda)\mathbf{I}. \tag{53}$$



(a) View 1 (b) View 2 (c) View 3 (d) View 4 (e) View 5

Figure 4: Example of synthetic images in **Plane1**

The update in $\mathbf{w}$ is obtained simply by $l1$ norm proximal operator:

$$\mathbf{w}^{(k+1)} = \text{sign}\left(\mathbf{v}^{(k+1)} - \mu^{(k)}\right) \circ \max\left(|\mathbf{v}^{(k+1)} - \mu^{(k)}| - \frac{\lambda}{\tau}, 0\right). \tag{54}$$

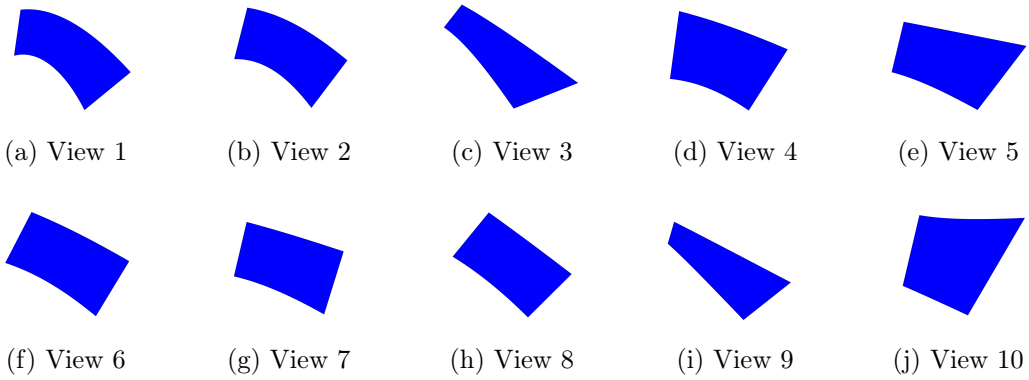
It can be noticed that the computation cost is dominated by the update of $\mathbf{v}$. However, updating $\mathbf{v}$ does not need to solve linear system at every iteration. To be efficient, $\mathbf{C}$ and $\mathbf{b}$ can be computed in advance, then all the subsequent updates in $\mathbf{v}$ are simply matrix-vector multiplications. Note that, from (51) and (52), computing $\mathbf{C}$ and $\mathbf{b}$ only needs to invert two matrices, i.e. $\tilde{\mathbf{A}}$ and $\mathbf{B}\tilde{\mathbf{A}}^{-1}\mathbf{B}^T$. If the total number of iterations is much less than n, we can conclude that the time complexity of this proposed global approach is $\mathcal{O}(n^3)$. In the experiments, running more than 50 iterations can yield a very good solution such that the output $\mathbf{v}$ is nearly binary, so the conversion of $\mathbf{v}$ into binary indicator has negligible influence on the optimality. As the computation is mainly dominated only by two matrix inversions, the global shape selection can be implemented in real-time performance.

5 Experiments

Note that, in some plots showing the reconstructed 3D shape, the red points always represent our reconstruction and green points always represent the ground truth.

5.1 Synthetic Data

Plane A single plane is simulated to generate the synthetic data, i.e. the image points and the related warping. This is to justify if our algorithm works for the simplest shape, since the plane is undoubtedly isometric under any motion. We here create two datasets, which are based on different creation of the reference view. The first one contains 21 frames, where the plane in the first frame (i.e. reference view) has the normal $(0, 0, 1)^T$. The second dataset has also 21 frames, but the plane in the first frame has the normal $(0.8, 0, 0.2)^T$. Motion of the plane with respect to the reference frame is applied with some random rotation and translation, while keeping the plane being always in front of the camera. Some examples of synthetic images are shown in Figure 4. The warping data is directly obtained from the known homography (i.e. from rotation and translation). The synthetic data about the plane is hence noiseless, which lets us observe if the algorithm can have errors for such ideal data. For simplicity, we call the first plane dataset as **Plane1**, and second as **Plane2**.



(a) View 1 (b) View 2 (c) View 3 (d) View 4 (e) View 5

(f) View 6 (g) View 7 (h) View 8 (i) View 9 (j) View 10

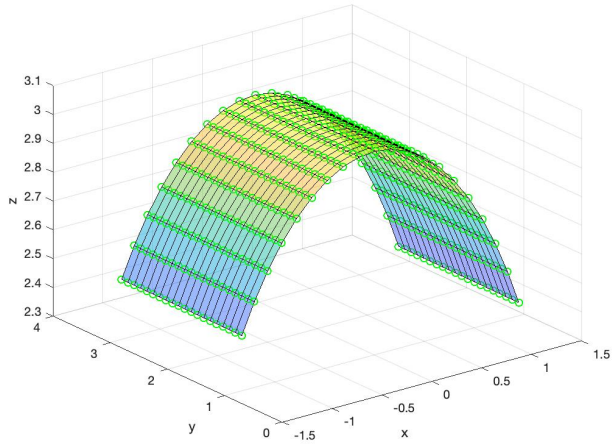
Figure 5: Synthetic images in **Cylinder3**

Cylinder Synthetic data is also created for cylinders with varying radius from 2 to 11. This lets us examine the quality of reconstruction under non-isometric deformation and also the varying curvature. The cylinder in each frame is also applied with some random rotation and translation. We create three datasets with different magnitude of random rotation, from small to large, implying increasing reconstruction difficulty, where some examples are shown in Figure 6. For simplicity, we denote them as **Cylinder1**, **Cylinder2**, **Cylinder3**. There are 10 frames generated for each dataset, with the reference frame having the smallest radius of the cylinder. Some examples of synthetic images in **Cylinder3** are drawn in Figure 5. The warping is computed by the Schwarp algorithm [8]. The ground truth normals (not the actual normals) used for comparison are obtained from the estimated Jacobians.

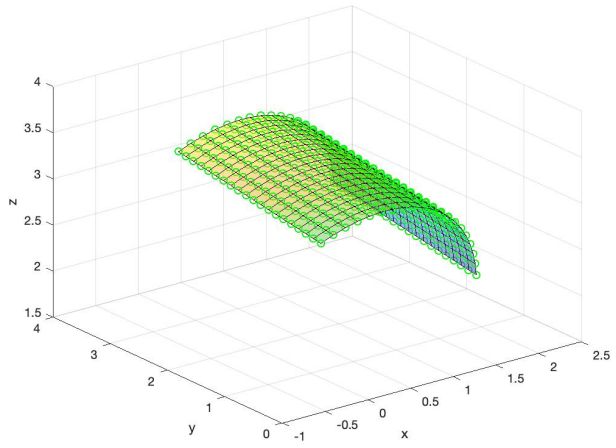
5.2 Local Approach on Synthetic Data

There are three local methods proposed: **LN** (i.e. (21)), **APAP** (i.e. (20)) and **MSA** (i.e. (23)). The visibility condition $k_3 > 0$ can be also applied on those local approaches. Besides, the local shape selection can be done by referring to two views simultaneously. For convenience of notation, when visibility condition is used, we append $+$**visb** to the method's name; when two views are both used, we append $+$ at the end. For each method, we measure the average error of the computed normals among all the pairs of views in a synthetic dataset, as shown in Table 1 and Table 2. Note for each pair of views, the first view is always chosen to be the fixed reference frame. Also, the error of normals on one view is obtained by averaging the angle difference with respect to the ground truth normal.

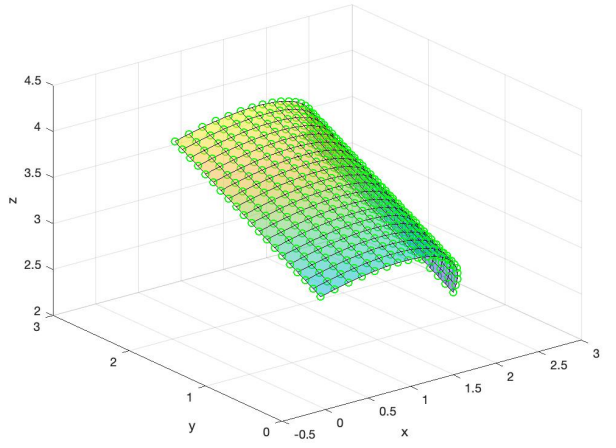
Comments on Plane From Table 1 and 2, it can be seen that the local approach works almost perfectly for **Plane1**. Particularly, **LN** and **APAP** work out a perfect reconstruction on the reference frame, even without the help of the visibility condition or other tricks. An example of the reconstructed reference frame is shown in Figure 7. This good performance on **Plane1** can be attributed to the fact $k_1, k_2 = 0$ on the first frame, so that **LN** and **APAP** can always choose the best shape parameters. However, it can be noticed that the error on the second view is always



(a) First frame of **Cylinder1**



(b) First frame of **Cylinder2**



(c) First frame of **Cylinder3**

Figure 6: Examples of the synthetic shapes for the cylinder

	Plane1	Plane2	Cylinder1	Cylinder2	Cylinder3
LN	**0**	7.216	7.691	8.420	**8.069**
LN+visb	**0**	7.216	7.579	8.404	8.073
LN+visb+	**0**	7.253	7.581	8.447	8.147
APAP	**0**	**2.168**	7.792	8.608	11.739
APAP+visb	**0**	**2.168**	**7.558**	**8.149**	9.589
APAP+visb+	0.217	2.770	**7.558**	8.219	8.373
MSA	0.139	7.830	7.764	8.494	8.436
MSA+visb	0.139	7.830	7.642	8.446	8.443
MSA+visb+	0.139	7.830	7.642	8.446	8.443

Table 1: Average error (in degrees) of normals on the first view among all pairs of views in the synthetic dataset

	Plane1	Plane2	Cylinder1	Cylinder2	Cylinder3
LN	**1.016**	9.897	7.467	9.241	11.057
LN+visb	**1.016**	9.897	7.337	9.218	**10.988**
LN+visb+	**1.016**	9.925	7.338	9.216	11.006
APAP	**1.016**	9.628	7.154	11.262	22.128
APAP+visb	**1.016**	9.628	**7.077**	**8.933**	11.590
APAP+visb+	1.677	**8.670**	**7.077**	9.065	11.161
MSA	1.343	10.340	7.655	9.217	11.139
MSA+visb	1.343	10.340	7.469	9.169	11.067
MSA+visb+	1.343	10.340	7.469	9.169	11.067

Table 2: Average error (in degrees) of normals on the second view among all pairs of views in the synthetic dataset

non-zero. Indeed, this is a consequence of error propagation, since the output normals in our algorithm are re-computed using the integrated shape (for smoothing purpose). Once we compare the ground truth with the 'unprocessed' normals, it can be found they are identical at some cases. This is not surprising since **Plane1** holds perfect isometric property and planar surface.

When both views in a pair are used for shape selection, we can notice that **APAP+visb+** has slightly degraded the performance in **Plane1** as shown in Table 1 and 2. This can be explained by the fact that the second view is not always that parallel to the camera plane, so the as-parallel-as-possible heuristic does not work well on second view. This phenomenon appears commonly in some other datasets or some other approaches, since there is always a trade-off between the reconstruction of the first and second view. The trade-off here means the fact that any heuristic of our approach always suits better for only one of the views. Note **MSA** is a special case since there is no trade-off on it. This can be observed from Table 1 and 2, that considering both views has no effect on the results. It is due to a direct fact that the determinant of the metric tensor on the second view is just a scaled one

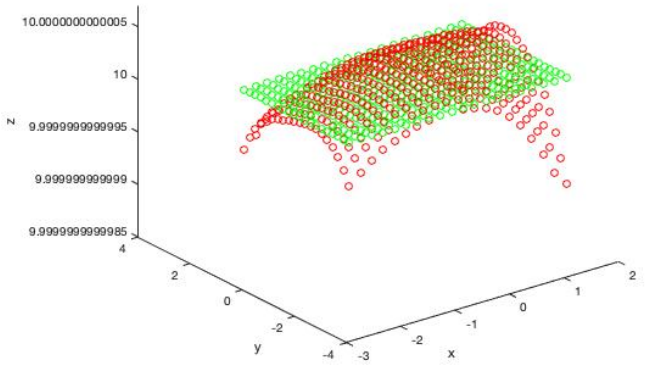


Figure 7: Reconstruction example in the first frame of **Plane1** by **APAP** (red dots represent the reconstruction, green dots the ground truth)

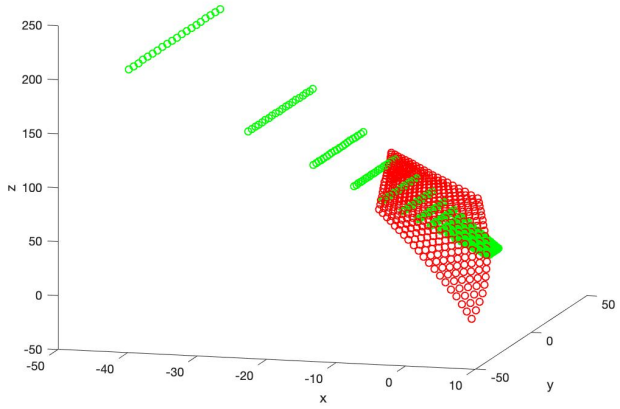


Figure 8: Reconstruction example in the first frame of **Plane2** by **APAP**

on the first view. In **Plane2**, **APAP** and its children methods work best among all local approaches. It can be noticed that the results in **Plane2** are much worse than those in **Plane1**. An example of reconstruction by **APAP** is shown in Figure 8, and it can be seen that the reconstruction deviates very much to the ground truth. The poor reconstruction is mainly a consequence of ill-posed plane, since the plane in the first frame is already almost perpendicular to the camera plane. This means the sudden change of depth, i.e. large magnitude of k_1 and k_2, which are particularly not favoured by **LN** and **APAP**. Though the error of normals is not large, the reconstruction for **Plane2** can be extremely sensitive to errors. If we find a shape that acts opposite to **APAP**, i.e. as perpendicular as possible, the result becomes interesting, where an example is shown in Figure 9. In this example, the result seems more reasonable than before. However, the error of the estimated normals is much greater than before, increased from 2.168 to 7.02 degrees for the reference frame. This disobeys our intuition, since when using the opposite heuristic

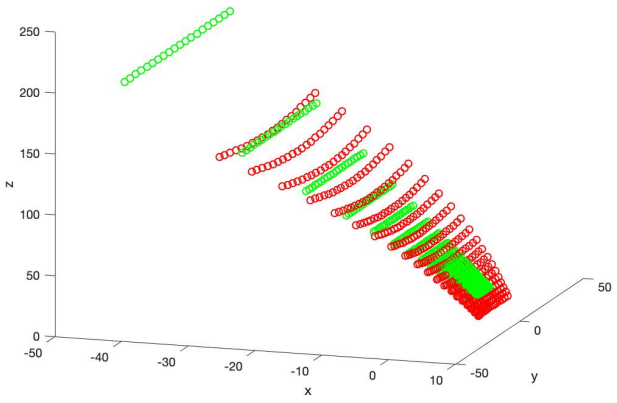
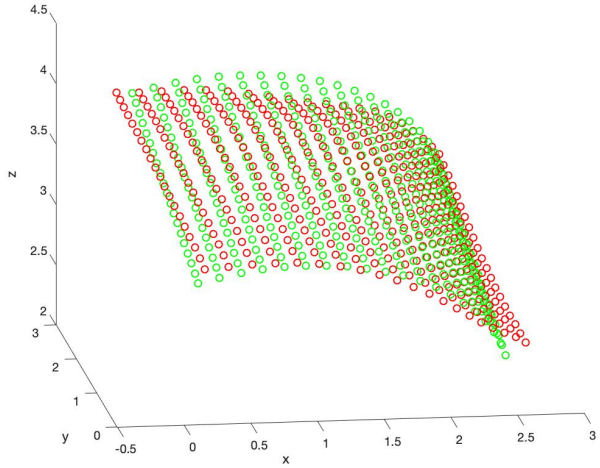


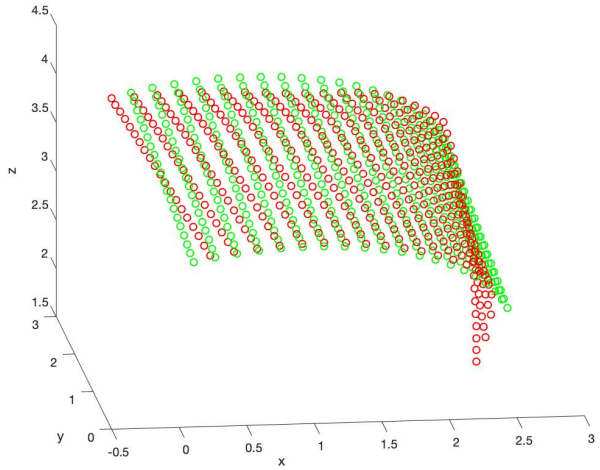
Figure 9: Reconstruction example in the first frame of **Plane2** by opposite choice of **APAP**

of **APAP**, the computed normals directly from the reconstruction equations are sometimes identical to the ground truth. Indeed, this is again the consequence of error propagation, as discussed previously. The problem is that the normals that are almost perpendicular to the camera plane are not numerically stable for normal integration. So, the re-computed normals are much worse than before. Empirically, this implies that **APAP** is numerically favoured, so that the re-computation of the normals would somehow compensate for the erroneously selected shape parameters. This intuitively explains why **APAP** and its chidren methods work best in **Plane2**.

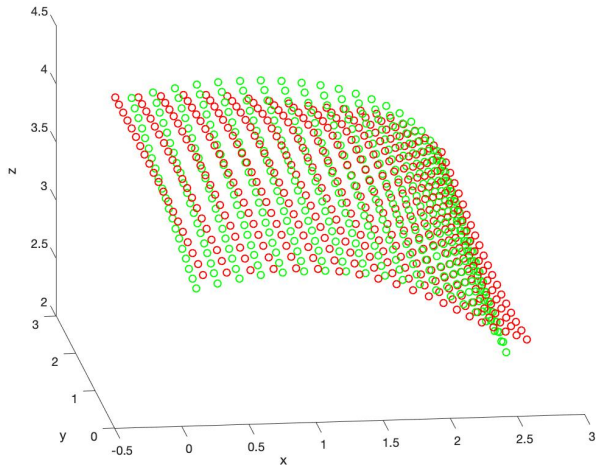
Comments on Cylinder From the previous discussion, the reconstruction is more numerically stable on the surface that tends to be parallel to the camera plane. This empirical statement is well reflected in the experiments of cylinder datasets, since the overall average error of normals tends to be smaller in **Cylinder1**, as observed in last three columns of Table 1 and 2. From these tables, it can be also observed that **LN**, **MSA** and their children methods have a relatively good performance. This can be attributed to the property of **LN** and **MSA**, that they tend to 'contract' the shape in terms of either least change of depth or minimum surface area. Note that if a shape has small change of depth, the surface should intuitively have small surface area, so **LN** may imply a similar condition as **MSA**. In fact, this similarity can be implicitly observed in Table 1 and 2. In Figure 10, the reconstruction results of the seventh pair of views (i.e. first frame and eighth frame) by our three local approaches are visualized, where here we target at the first frame of **Cylinder3**. On one hand, from this example, we can see that the reconstructed surface by **LN** is very similar to that of **MSA**. On the other hand, from Figure 10(b), it can be found that **APAP** in this example has an obvious reconstruction error, such that the surface is 'twisted' at the one corner. This shows a potential issue of **APAP**, that **APAP** may not be suitable for object with large motion, since seeking the most parallel surface is no longer a good heuristic. This weakness of **APAP**



(a) result by **LN**



(b) result by **APAP**



(c) result by **MSA**

Figure 10: Reconstructed cylinder in the first frame of **Cylinder3**

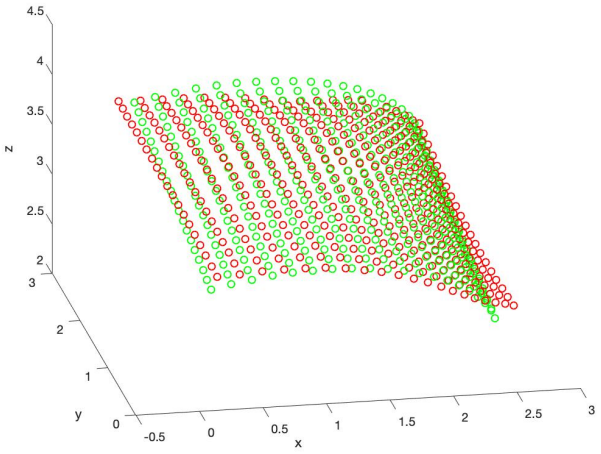


Figure 11: Reconstruction example in the eighth frame of **Cylinder3** by **APAP+visb**

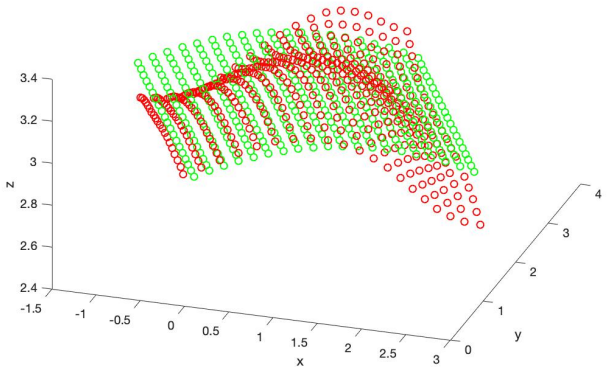


Figure 12: Reconstruction example of eighth frame in **Cylinder3** by **LN**

can be also reflected in Table 1 and 2, that the average error of **APAP** appears to be highest in **Cylinder3**. However, the visibility condition can be applied to reduce some obviously erroneous reconstruction. From the tables, the error is apparently reduced for **APAP+visb**, as compared with **APAP**. In fact, making the surface visible can at least eliminate some cases of 'twisted surface'. An example is shown in Figure 11, where the visibility condition has been applied, and we can compare this result with Figure 10(b). In Figure 10, we can see that **LN** and **MSA** have a good reconstruction of cylinder in the first frame, at least there is no large deviation from the ground truth. On the second view in the reconstruction pair, nevertheless, the result may not be that promising as in the reference view. An example by **LN** is shown in Figure 12. This may indicate the potential limitation of the solutions obtained from the reconstruction equation, since we have the assumption about isometric mapping and infinitesimal planar surface (i.e. second derivative of depth is assumed to be zero). Particularly, the presence of curvature is ignored in our

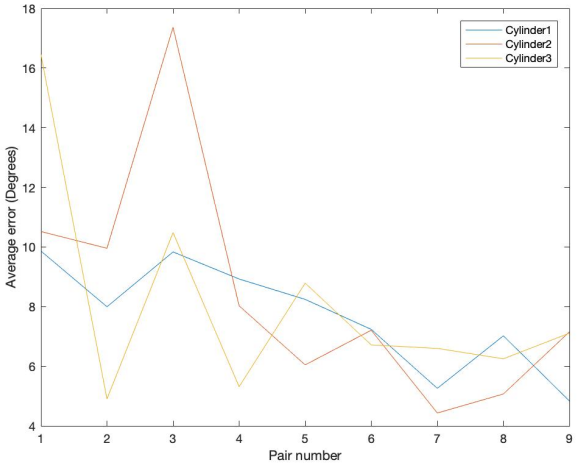


Figure 13: Average reconstruction error of first frame by **LN** in all pairs of views

current NRSfM algorithm. Intuitively, the less the curvature (i.e. inverse of radius), the better the reconstruction. This hypothesis can be empirically verified in Figure 13. Specifically, we can see that there is a trend that the average reconstruction error of the first frame decreases as we move forward the pair. Note as the pair number increases, the radius of the cylinder in the second view of the pair becomes larger (hence smaller curvature), meanwhile the first view is always fixed and has the smallest radius of cylinder.

5.3 Global Approach on Synthetic Data

Global approach aims to take all the advantages of the local approach while maintaining the global consistency. In the experiment of global approach, we set $\sigma = 0.2$, ratio $= 0.1$ for graph construction. The solver of the global approach can be either chosen as **ADMM** (i.e. (46), (47) and (48)) or **QP** (i.e. (41)). If we choose **ADMM** as the solver, it means a more refined approximation of the original MIQP. For **ADMM**, we set the total number of iterations as 50, and $\tau = 10$, $s = 0.2$, $\lambda = 10$, and $\mathbf{w}^{(0)}$ as the solution from **APAP**+**visb**.

The parameters c_1, c_2, c_3 in the objective function can directly influence the two-view reconstruction, since they manage the weight for the local regularization. A positive c_3 is crucial as it forces the surface to be visible, hence preventing the surface from being 'twisted'. The rest parameters c_1, c_2 can be set regarding the previous discussion about local approach. For the surface that tends to be aligned slightly parallel to the camera plane, then a large c_1 may be preferred; for the curved surface with some motion, **LN** may be a preferred local approach, so that c_2 should be set to be large at this case. Automatic tuning of these parameters remains as a unsolved problem. In this experiment, we settle down to the same set of parameters for both **ADMM** and **QP** solvers: $c_1 = 1$, $c_2 = 0.1$ and $c_3 = 1$. Here c_2 is deliberately set to be small, since we prefer to the local regularization that resembles the local approach **APAP**+**visb**+.

Several local approaches which have obtained minimal error previously are chosen to be compared with our global approach, as shown in Table 3 and 4. Though

the global approach has similar performance as local approach, it sometimes makes some slight improvement on two-view reconstruction. For the two solvers of global approach, they have similar performance. This empirically grants the convex relaxation from MIQP to a simple QP, where the sparsity constraint on $\mathbf{v}$ could be even removed without loss of performance.

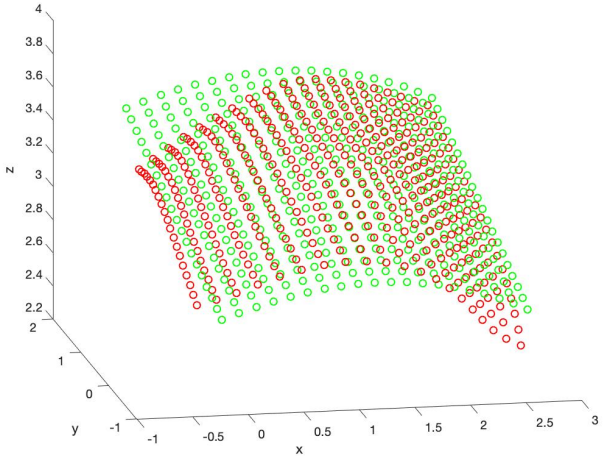
	Plane1	Plane2	Cylinder1	Cylinder2	Cylinder3
LN	**0**	7.216	7.691	8.420	8.069
LN+visb	**0**	7.216	7.579	8.404	8.073
APAP	**0**	**2.168**	7.792	8.608	11.739
APAP+visb	**0**	**2.168**	7.558	8.149	9.589
APAP+visb+	0.217	2.770	7.558	8.219	8.373
QP	0.006	3.505	**7.475**	8.164	**7.983**
ADMM	**0**	3.455	**7.475**	**8.093**	8.362

Table 3: Average error (in degrees) of normals on the first view among all pairs of views in the synthetic dataset

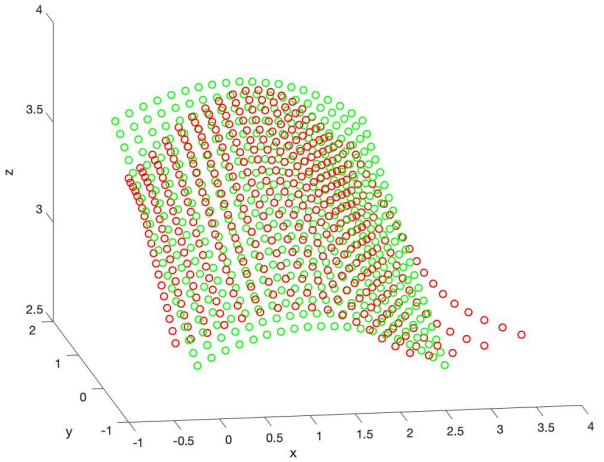
	Plane1	Plane2	Cylinder1	Cylinder2	Cylinder3
LN	1.016	9.897	7.467	9.241	11.057
LN+visb	1.016	9.897	7.337	9.218	**10.988**
APAP	1.016	9.628	7.154	11.262	22.128
APAP+visb	1.016	9.628	7.077	**8.933**	11.590
APAP+visb+	1.677	8.670	7.077	9.065	11.161
QP	**0.667**	6.349	**7.019**	9.087	11.571
ADMM	1.016	**6.252**	**7.019**	9.048	11.500

Table 4: Average error (in degrees) of normals on the second view among all pairs of views in the synthetic dataset

There is no perfect parameter setup in our current global approach, since a trade-off always exists between some local properties like parallelism and curvature. An example is shown in Figure 14, where **LN** has obviously better reconstruction than the global approach solved by **ADMM**. As seen from the 'flat' boundary in Figure 14(b), this is a consequence of our preference on the local parallelism as the regularization for the global approach. The performance of global approach would degrade if local shape regularization is removed. Such performance degradation is found particularly obvious for **Cylinder3**. Figure 15 shows an example by global approach without any regularization, which can be compared with Figure 14(b). We could see that smooth surface normals could somehow exaggeratedly stretch the shape, leading to an erroneous result like Figure 15, so here local regularization can come to help contract the shape, e.g. Figure 14(b). This reflects both the drawback of global approach and advantage of the local approach.



(a) result by **LN**



(b) result by global approach

Figure 14: Reconstructed cylinder in the fourth frame of **Cylinder3**

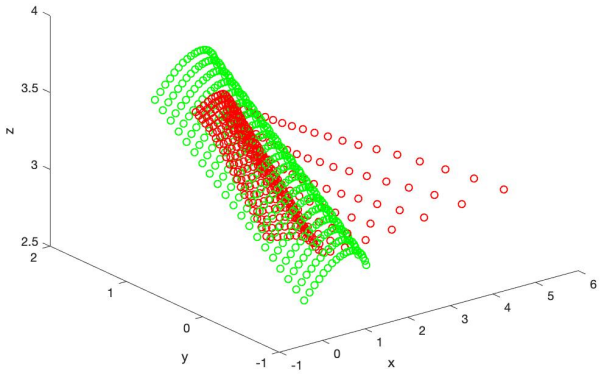


Figure 15: Reconstructed cylinder in the fourth frame of **Cylinder**, by ADMM solver without any regularization

(a) View 1 (b) View 2 (c) View 3 (d) View 4 (e) View 5

Figure 16: Example of 2D images in **T-shirt**

5.4 Two-view NRSfM on Real Data

We use the same parameter setup to do the experiments on real dataset **Kinect Paper** and **T-shirt**. Some 2D frames of **T-shirt** are shown in Figure 16. The results are shown in Table 5 and 6. As we can see from the tables, all local approaches have better performance on reconstructing the first frame of **Kinect Paper** than global approach. Apart from that, the global approach has better average performance than the local approach, especially on **T-shirt**.

5.5 Comparison with Multiple-view NRSfM

The solution in the reconstruction equation can be also selected by using the information from multiple views. First, at one correspondence of the first frame, each shape parameter (k_1, k_2) solved from the sextic equation can be linearly transformed into $(\bar{k}_1, \bar{k}_2)$ of some other frame, by using the warping. Second, for every other frame, we substitute the transformed shape parameters into the corresponding cubics, i.e. (8) and (9), hence the sum of squared errors can be computed. For each real solution, a row of sum of squared errors from all the reconstruction equations can be hence obtained, we compute the median among them. Finally, we choose the shape parameters with the smallest median. The reason for computing the median is to reduce the influence of noisy errors. For convenience, we call this shape selection method as least median (**LM**).

The comparison of our two-view methods with multiple-view methods is shown

	Kinect Paper	T-shirt
LN	8.457	20.048
LN+visb	8.457	20.047
LN+visb+	8.444	20.124
APAP	8.441	19.241
APAP+visb	8.441	19.251
APAP+visb+	**8.404**	19.552
MSA	8.554	20.471
MSA+visb	8.554	20.470
MSA+visb+	8.554	20.470
QP	8.780	**18.076**
ADMM	8.568	18.941

Table 5: Average error (in degrees) of normals on the first view among all pairs of views in the real dataset

	Kinect Paper	T-shirt
LN	6.163	16.086
LN+visb	6.163	16.084
LN+visb+	6.189	16.107
APAP	6.110	15.672
APAP+visb	6.111	15.676
APAP+visb+	6.178	15.743
MSA	6.233	16.237
MSA+visb	6.233	16.235
MSA+visb+	6.233	16.235
QP	5.973	15.684
ADMM	**5.862**	**15.575**

Table 6: Average error (in degrees) of normals on the second view among all pairs of views in the real dataset

	MSA+visb	LN+visb	APAP+visb	ADMM	LM	infP
Kinect Paper	8.554	8.457	8.441	8.568	8.505	**7.052**
T-shirt	20.470	20.047	19.251	18.941	19.874	**17.551**
Plane1	0.139	**0**	**0**	**0**	**0**	**0**
Plane2	7.830	7.216	**2.168**	3.455	6.277	12.662
Cylinder1	7.642	7.579	7.558	7.475	7.743	**6.198**
Cylinder2	8.446	8.404	8.149	8.093	8.421	**5.616**
Cylinder3	8.443	8.073	9.589	8.362	8.572	**7.107**

Table 7: Average error (in degrees) of normals on the first frame

	MSA+visb	LN+visb	APAP+visb	ADMM	LM	infP
Kinect Paper	6.233	6.163	6.111	**5.862**	6.163	8.322
T-shirt	16.235	16.084	15.676	**15.575**	15.952	15.964
Plane1	1.343	**1.016**	**1.016**	**1.016**	**1.016**	**1.016**
Plane2	10.340	9.897	9.628	**6.252**	14.714	18.793
Cylinder1	7.469	7.337	7.077	7.019	7.626	**5.494**
Cylinder2	9.169	9.218	8.993	9.048	9.207	**6.863**
Cylinder3	11.067	10.988	11.590	11.500	11.185	**10.639**

Table 8: Average error (in degrees) of normals on all the frames except first one

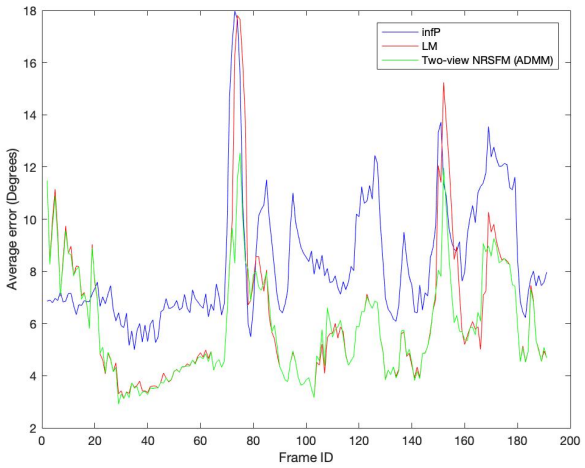


Figure 17: Average reconstruction error in all frames (except first one) of **Kinect Paper**

in Table 7 and 8, where only several representative two-methods are included. Note that, for **infP** (the original isometric NRSfM), each frame has unique reconstruction, so that the error on the first frame is no longer averaged through all the view pairs. From Table 7, it can be seen that **infP** has best reconstruction result on the first frame, except the one in **Plane2**. This is reasonable since **infP** optimizes the shape parameters among all the frames. Note that **Plane2** is a special dataset, as discussed before, the errors can even become greater if the surface is better reconstructed. Also, we should notice that, the error (on the first frame) in the two-view methods is measured by averaging the error through all the view pairs, so sometimes the error on reconstructing the first frame can be lower than the averaged error. We can even expect that the error by the two-view method can be even lower than the error by **infP**.

In Table 8, for the all the frames except first one, it can be seen that the global approach by **ADMM** has best reconstruction results in **Kinect Paper**, **T-shirt**, **Plane1** and **Plane2**. On the other hand, **infP** performs best in the other datasets of cylinder. This somehow demonstrates that two-view methods may not be good at reconstructing the surfaces with high curvature and large motion. Despite this, our proposed two-view methods perform better than **LM** at most time, especially for

the global approach by **ADMM**. Furthermore, our two-view methods have shown the potential to outperform **infP**. An example is shown in Figure 17, where the yellow curve (**ADMM**) is below the blue curve (**infP**) in most frames of **Kinect Paper**.

5.6 Timing Performance

The experiments are run in a dual-core 2.6 GHz Mac, with no parallel implementation. Assume we are given 400 point correspondences between different frames. Computing the Schwarps [8] for a pair of images takes averagely 24 seconds. Solving all the reconstruction equations for one pair of views can cost around 0.017 seconds. The normal integration costs about 0.7 seconds per pair of views. For the shape selection, local approach has negligible cost, which is about 0.002 seconds per pair of views. For the global approach, it takes 0.15 seconds for **ADMM** and 0.094 seconds for **QP**. We can see that the time cost is mainly dominated by computing warps and normal integration.

It can be noticed that our two-view method takes negligible time when solving the reconstruction equation, since a monomial can be very efficiently solved. The multiple-view Iso-NRSfM needs to solve sum of squared cubics, so it takes much more time to solve the reconstruction equation. In experiment, multiple-view Iso-NRSfM takes over 2 seconds per frame for solving the reconstruction equation.

6 Conclusion and Future Works

We show that it is somehow feasible to solve Iso-NRSfM given only $N = 2$ views. The key of two-view Iso-NRSfM is to select a proper shape out of numerous alternatives. The shape can be selected both locally and globally. The proposed methods are all based on explicit geometric properties, such as frontal parallelism and surface area. The local approach could be applied with negligible time cost, meanwhile it has similar performance with the global approach. In the experiments, it could be even found that our two-view method could sometimes outperform than the multiple-view method. For this, we could consider incorporating the two-view method into the multiple-view Iso-NRSfM, in order to further improve the multiple-view method. For instance, some local shape regularization from our local approach could be added into the multiple-view Iso-NRSfM.

Besides, choosing different reference frame may influence the shape recovery. It is hence worth investigating some method on choosing the reference frame. This could be also important for multiple-view Iso-NRSfM.

References

[1] Mario Botsch, Leif Kobbelt, Mark Pauly, Pierre Alliez, and Bruno Lévy. *Polygon mesh processing*. CRC press, 2010.

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

[3] Ajad Chhatkuli, Daniel Pizarro, Toby Collins, and Adrien Bartoli. Inextensible non-rigid structure-from-motion by second-order cone programming. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2428–2441, 2017.

[4] Alberto Del Pia, Santanu S Dey, and Marco Molinaro. Mixed-integer quadratic programming is in np. *Mathematical Programming*, 162(1-2):225–240, 2017.

[5] Ezio Malis and Manuel Vargas. Deeper understanding of the homography decomposition for vision-based control. 2007.

[6] Shaifali Parashar, Daniel Pizarro, and Adrien Bartoli. Isometric non-rigid shape-from-motion in linear time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4679–4687, 2016.

[7] Shaifali Parashar, Daniel Pizarro, and Adrien Bartoli. Isometric non-rigid shape-from-motion with riemannian geometry solved in linear time. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2442–2454, 2017.

[8] Daniel Pizarro, Rahat Khan, and Adrien Bartoli. Schwarps: Locally projective image warps based on 2d schwarzian derivatives. *International Journal of Computer Vision*, 119(2):93–109, 2016.

[9] Yvain Quéau, Jean-Denis Durou, and Jean-François Aujol. Normal integration: a survey. *Journal of Mathematical Imaging and Vision*, 60(4):576–593, 2018.

[10] Mathieu Salzmann and Pascal Fua. Reconstructing sharply folding surfaces: A convex formulation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1054–1061. IEEE, 2009.

[11] Lorenzo Torresani, Aaron Hertzmann, and Chris 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.

[12] Penghang Yin, Yifei Lou, Qi He, and Jack Xin. Minimization of 1-2 for compressed sensing. *SIAM Journal on Scientific Computing*, 37(1):A536–A563, 2015.