NRMVS: Non-Rigid Multi-View Stereo

Matthias Innmann[1,2*] Kihwan Kim[1] Jinwei Gu[1,3*] Matthias Nießner[4] Charles Loop[1]
Marc Stamminger[2] Jan Kautz[1]

[1]NVIDIA [2]University of Erlangen-Nuremberg [3]SenseTime [4]Technical University of Munich

Abstract

Scene reconstruction from unorganized RGB images is an important task in many computer vision applications. Multi-view Stereo (MVS) is a common solution in photogrammetry applications for the dense reconstruction of a static scene. The static scene assumption, however, limits the general applicability of MVS algorithms, as many day-to-day scenes undergo non-rigid motion, e.g., clothes, faces, or human bodies. In this paper, we open up a new challenging direction: dense 3D reconstruction of scenes with non-rigid changes observed from arbitrary, sparse, and wide-baseline views. We formulate the problem as a joint optimization of deformation and depth estimation, using deformation graphs as the underlying representation. We propose a new sparse 3D to 2D matching technique, together with a dense patch-match evaluation scheme to estimate deformation and depth with photometric consistency. We show that creating a dense 4D structure from a few RGB images with non-rigid changes is possible, and demonstrate that our method can be used to interpolate novel deformed scenes from various combinations of these deformation estimates derived from the sparse views.

1. Introduction

Multi-view stereo algorithms [4, 6, 13, 15, 38, 40] have played an important role in 3D reconstruction and scene understanding for applications such as augmented reality, robotics, and autonomous driving. However, if a scene contains motion, such as non-stationary rigid objects or non-rigid surface deformations, the assumption of an epipolar constraint is violated [18], causing algorithms to fail in most cases. Scenes with rigidly-moving objects have been reconstructed by segmenting foreground objects from the background, and treating these regions independently [52, 25, 50]. However, the reconstruction of scenes with surfaces undergoing deformation is still a challenging task.

[1]The authors contributed to this work when they were at NVIDIA.

(a) Canonical view (b) Updated canonical view (c) 3D point cloud for each view

Figure 1: We take a small number of unordered input images captured from wide-baseline views (top row), and reconstruct the 3D geometry of objects undergoing non-rigid deformation. We first triangulate a canonical surface from a pair of views with minimal deformation (a). Then we compute the deformation from the canonical view to the other views, and reconstruct point clouds for both original canonical views (b) and other remaining views (c)

To solve this problem for sparse point pairs, various non-rigid structure from motion (NRSfM) methods have been introduced [21]. These methods often require either dense views (video frames) [3] for the acquisition of dense correspondences with flow estimation, or prior information to constrain the problem [7]. Newcombe et al. [36] and Innmann et al. [20] recently demonstrated solutions for the 3D reconstruction of arbitrary, non-rigid, dynamic scenes using a dense stream of known metric depths captured from a commercial depth camera. However, there are many common scenarios one may encounter for which this method cannot be used, for example, when capturing scenes that contain non-rigid changes that are neither acquired as video nor captured by stereo or depth cameras, but rather come from independent single views.

In this paper, we are specifically interested in *dense 3D scene reconstruction with dynamic non-rigid deformations* acquired from images with wide spatial baselines, and

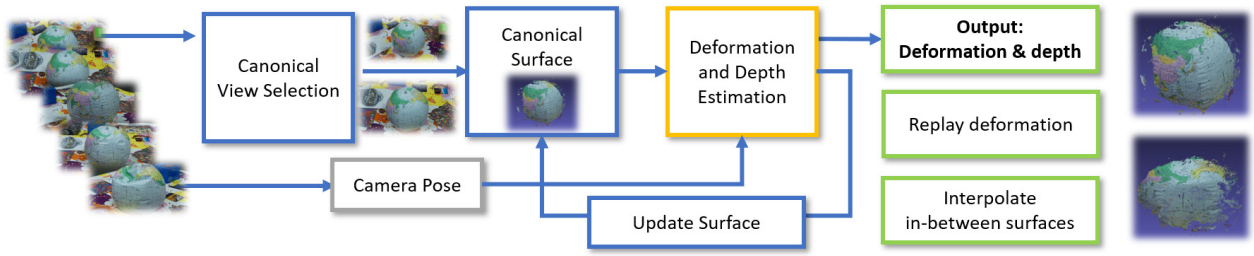


Figure 2: **Overview of our framework**: We first reconstruct the initial point cloud from the views with minimal deformation, which we call canonical views. Then, we estimate depths for the remaining views by estimating a plausible deformation from a joint optimization between depth, deformation, and dense photometric consistency. With these computed depths and deformations, we also demonstrate an interpolation deformation between input views.

sparse, unordered samples in time. This requires two solutions: (1) a method to compute the most plausible deformation from millions of potential deformations between given wide-baseline views, and (2) a dense surface reconstruction algorithm that satisfies a photometric consistency constraint between images of surfaces undergoing non-rigid changes.

In our solution, we first compute a *canonical surface* from views that have minimal non-rigid changes (Fig. 1(a)), and then estimate the deformation between the canonical pose and other views by joint optimization of depth and photometric consistency. This allows the expansion of 3D points of canonical views (Fig. 1(b)). Then, through the individual deformation fields estimated from each view to the canonical surface, we can reconstruct a dense 3D point cloud of each single view (Fig. 1(c)). A brief overview of our entire framework is described in Fig. 2.

Our contributions are as follows:

- The first non-rigid MVS pipeline that densely reconstructs dynamic 3D scenes with non-rigid changes from wide-baseline and sparse RGB views.
- A new formulation to model non-rigid motion using a deformation graph [45] and the approximation of the inverse-deformation used for the joint optimization with photometric consistency.
- Patchmatch-based [4] dense sample propagation on top of an existing MVS pipeline [38], which allows flexible implementation depending on different MVS architectures.

2. Related Work

Dynamic RGB-D Scene Reconstruction. A prior step to full dynamic scene reconstruction is dynamic template tracking of 3D surfaces. The main idea is to track a shape template over time while non-rigidly deforming its surface [8, 2, 17, 19, 30, 27, 29, 14, 53]. Jointly tracking and reconstructing a non-rigid surface is significantly more challenging. In this context, researchers have developed an impressive line of works based on RGB-D or depth-only input [51, 35, 46, 5, 9, 11, 34, 49, 31]. DynamicFusion [36] jointly optimizes a Deformation Graph [45], then fuses the deformed surface with the current depth map. Innmann et al. [20] follows up on this work by using an as-rigid-as-possible regularizer to represent deformations [43], and incorporate RGB features in addition to a dense depth tracking term. Fusion4D [10] brings these ideas a level further by incorporating a high-end RGB-D capture setup, which achieves very impressive results. More recent RGB-D non-rigid fusion frameworks include KillingFusion [41] and SobolevFusion [42], which allow for implicit topology changes using advanced regularization techniques. This line of research has made tremendous progress in the recent years; but given the difficulty of the problem, all these methods either rely on depth data or calibrated multi-camera rigs.

Non-Rigid Structure from Motion. Since the classic structure from motion solutions tend to work well for many real world applications [48, 22, 47], many recent efforts have been devoted to computing the 4D structure of sparse points in the spatio-temporal domain, which we call non-rigid structure from motion (NRSfM) [21, 24, 37, 16]. However, most of the NRSfM methods consider the optimization of sparse points rather than a dense reconstruction, and often require video frames for dense correspondences [3] or prior information [7]. Scenes with rigidly moving objects (e.g., cars or chairs) have been reconstructed by segmenting rigidly moving regions [52, 25, 26, 50]. In our work, we focus on a new scenario of reconstructing scenes with non-rigid changes from a few images, and estimate deformations that satisfy each view.

Multi-View Stereo and Dense Reconstruction. Various MVS approaches for dense 3D scene reconstruction have been introduced in the last few decades [13, 15, 38, 40]. While many of these methods work well for static scenes, they often reject regions that are not consistent with the epipolar geometry [18], e.g., if the scene contains changing regions. Reconstruction failure can also occur if the ratio of static to non-rigid parts present in the scene is

too low [33]. A recent survey [39] on MVS shows that COLMAP [38] performs the best among state-of-the-art methods. Therefore, we adopt COLMAP's Patchmatch framework for dense photometric consistency.

3. Approach

The input to our non-rigid MVS method is a set of images of a scene taken from unique (wide-baseline) locations at different times. An overview of our method is shown in Fig. 2. We do not assume any knowledge of temporal order, i.e., the images are an unorganized collection. However, we assume there are at least two images with minimal deformation, and the scene contains sufficient background in order to measure the ratio of non-rigidity and to recover the camera poses. We discuss the details later in Sec 3.3. The output of our method is an estimate of the deformation within the scene from the canonical pose to every other view, as well as a depth map for each view. After the canonical view selection, we reconstruct an initial canonical surface that serves as a template for the optimization. Given another arbitrary input image and its camera pose, we estimate the deformation between the canonical surface and the input. Furthermore, we compute a depth map for this processed frame using a non-rigid variant of PatchMatch. Having estimated the motion and the geometry for every input image, we recompute the depth for the entire set of images to maximize the growth of the canonical surface.

3.1. Modeling Deformation in Sparse Observations

To model the non-rigid motion in our scenario, we use the well known concept of deformation graphs [45]. Each graph node represents a rigid body transform, similar to the as-rigid-as-possible deformation model [43]. These transforms are locally blended to deform nearby space.

Given a point $\mathbf{v} \in \mathbb{R}^3$, the deformed version $\hat{\mathbf{v}}$ of the point is computed as:

$$\hat{\mathbf{v}} = \sum_{i=1}^{k} w_i(\mathbf{v}) \left[\mathbf{R}_i(\mathbf{v} - \mathbf{g}_i) + \mathbf{g}_i + \mathbf{t}_i\right],$$

where $\mathbf{R}_i$ and $\mathbf{t}_i$ represent the rotation and translation of a rigid body transform about position $\mathbf{g}_i$ of the i-nearest deformation node, and k is the user-specified number of nearest neighbor nodes (we set to $k = 4$ throughout our paper). The weights w_i are defined as:

$$w_i(\mathbf{v}) = \frac{1}{\sum_{j=1}^{k} w_j(\mathbf{v})} \left(1 - \frac{\|\mathbf{v} - \mathbf{g}_i\|_2}{\|\mathbf{v} - \mathbf{g}_{k+1}\|_2}\right)^2.$$

For a complete description of deformation graphs, we refer to the original literature [45].

When projecting points between different images, we also need to invert the deformation. The exact inverse deformation can be derived given known weights:

$$\mathbf{v} = \left(\sum_{i=1}^{k} w_i(\mathbf{v})\mathbf{R}_i\right)^{-1} \left[\hat{\mathbf{v}} + \sum_{i=1}^{k} w_i(\mathbf{v}) \left[\mathbf{R}_i\mathbf{g}_i - \mathbf{g}_i - \mathbf{t}_i\right]\right]$$

However, because we do not know the weights a priori, which requires the nearest neighbor nodes and their distances, this becomes a non-linear problem. Since this computationally expensive step is necessary at many stages of our pipeline, we introduce an approximate solution:

$$\mathbf{v} \approx \left(\sum_{i=1}^{k} \hat{w}_i(\hat{\mathbf{v}})\mathbf{R}_i\right)^{-1} \left[\hat{\mathbf{v}} + \sum_{i=1}^{k} \hat{w}_i(\hat{\mathbf{v}}) \left[\mathbf{R}_i\mathbf{g}_i - \mathbf{g}_i - \mathbf{t}_i\right]\right],$$

where the weights $\hat{w}_i$ are given by

$$\hat{w}_i(\hat{\mathbf{v}}) = \frac{1}{\sum_{j=1}^{k} \hat{w}_i(\hat{\mathbf{v}})} \left(1 - \frac{\|\hat{\mathbf{v}} - (\mathbf{g}_i + \mathbf{t}_i)\|_2}{\|\hat{\mathbf{v}} - (\mathbf{g}_{k+1} + \mathbf{t}_{k+1})\|_2}\right)^2.$$

Note that our approximation can be computed directly and efficiently, without leading to any error of observable influence in our synthetic experiments.

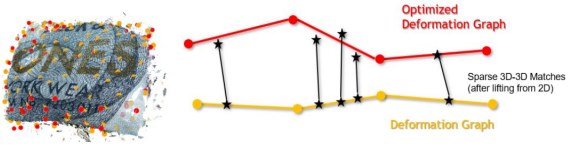


Figure 3: **Deformation nodes and correspondences:** (Left) shows the deformation nodes at t_0 (orange), and another set of nodes at t_1 (red) overlaid in the canonical view. (Right) we show the relationship between deformation nodes from two views and sparse 3D matches (after lifting) in the context of a non-rigid change. Note that we only show the sparse matching for simpler visualization while there is also a dense term for photometric consistency that drives the displacement of deformation nodes together with the sparse matches.

3.2. Non-rigid Photometric Consistency and Joint Optimization

With the deformation model in hand, we next estimate the depth of the other views by estimating deformations that are photometrically consistent with the collection images and subject to constraints on the geometry. This entire step can be interpreted as a non-rigid version of a multi-view stereo framework.

Canonical View Selection From the set of input images, we select two views with a minimal amount of deformation. We run COLMAP's implementation of PatchMatch [38] to acquire an initial temple model of the canonical pose. Based on this template, we compute the deformation graph by distributing a user-specified number of nodes on the surface.

To this end, we start with all points of the point cloud as initial nodes. We iterate over all nodes, and for each node remove all its neighbors within a given radius. The process is repeated with a radius that is increased by 10%, until we have reached the desired number of nodes. In our experiments, we found that 100 to 200 nodes are sufficient to faithfully reconstruct the motion. Fig. 3(left) shows an example of the node distribution.

Correspondence Association For sparse global correspondences, we detect SIFT keypoints [32] in each image and match descriptors for every pair of images to compute a set of feature tracks $\{\mathbf{u}_i\}$. A *feature track* represents the same 3D point and is computed by connecting each keypoint with each of its matches. We reject inconsistent tracks, i.e., if there is a path from a keypoint $\mathbf{u}_i^{(j)}$ in image i to a different keypoint $\mathbf{u}_i^{(k)}$ with $j \neq k$ in the same image.

We lift keypoints $\mathbf{u}_i$ to 3D points $\mathbf{x}_i$, if there is a depth value in at least one processed view, compute its coordinates in the canonical pose $\mathbf{D}_i^{-1}(\mathbf{x}_i)$ and apply the current estimate of our deformation field $\mathbf{D}_j$ for frame j to these points. To establish a sparse 3D-3D correspondence $(\mathbf{D}_i^{-1}(\mathbf{x}_i), \mathbf{x}_j)$ between the canonical pose and the current frame j for the correspondence set S, we project $\mathbf{D}_j(\mathbf{D}_i^{-1}(\mathbf{x}_i))$ to the ray of the 2D keypoint $\mathbf{u}_j$ (see Fig. 4). To mitigate ambiguities and to constrain the problem, we also aim for dense photometric consistency across views. Thus, for each point of the template of the canonical pose, we also add a photometric consistency constraint with a mask $C_i \in \{0, 1\}$.

Deformation and Depth Estimation In our main iteration (see also Algorithm 1), we estimate the deformation $\hat{\mathbf{D}}$ between the canonical pose and the currently selected view by minimizing the joint optimization problem:

$$E = w_{\text{sparse}} E_{\text{sparse}} + w_{\text{dense}} E_{\text{dense}} + w_{\text{reg}} E_{\text{reg}} \tag{1}$$

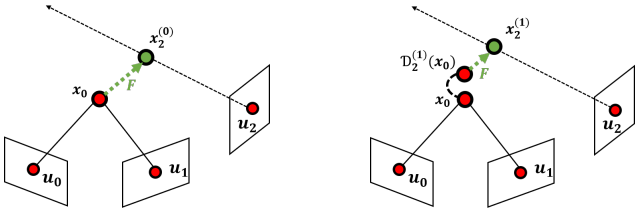
$$E_{\text{sparse}} = \sum_{(i,j)\in S} \|\hat{\mathbf{D}}(\mathbf{x}_i) - \mathbf{x}_j\|_2^2$$

$$E_{\text{dense}} = \sum_r \sum_s \sum_i C_i \cdot (1 - \rho_{r,s}(\hat{\mathbf{D}}(\mathbf{x}_i), \hat{\mathbf{D}}(\mathbf{n}_i), \mathbf{x}_i, \mathbf{n}_i))^2$$

$$E_{\text{reg}} = \sum_{j=1}^{m} \sum_{k\in N(j)} \|\mathbf{R}_j(\mathbf{g}_k - \mathbf{g}_j) + \mathbf{g}_j + \mathbf{t}_j - (\mathbf{g}_k + \mathbf{t}_k)\|_2^2$$

To measure photometric consistency $\rho_{r,s}$ between a reference image r, i.e. the canonical pose, and a source view s, we use the bilaterally weighted adaption of normalized cross-correlation (NCC) as defined by Schoenberger et al. [38]. Throughout our pipeline, we employ COLMAP's default settings, i.e. a window of size 11×11. The regularizer E_{reg} as defined in [45] ensures a smooth deformation result. To ensure non-local convergence, we solve the problem in a coarse-to-fine manner using an image pyramid with 3 levels in total.

Both the sparse and dense matches are subject to outliers. In the sparse case, these outliers manifest as incorrect keypoint matches across images. For the dense part, outliers mainly occur due to occlusions, either because of the camera pose or because of the observed deformation.



(a) Iteration 1 (b) Iteration 2

Figure 4: **Sparse correspondence association**: In iteration i, we transform the 3D point $\mathbf{x}_0$ according to the previous estimate of the deformation $\mathbf{D}_2^{(i-1)}$ and project $\mathbf{D}_2^{(i-1)}(\mathbf{x}_0)$ onto the ray defined by $\mathbf{u}_2$. The projection is used to define a force F pulling the point towards the ray.

To reject outliers in both cases, we reject correspondences with the highest residuals calculated from the result of the non-linear solution. We re-run the optimization until a user-specified maximum error is satisfied. This rejection is run in a 2-step process. First, we only solve for the deformation considering the sparse 3D-3D matches. Second, we fix the retained 3D-3D matches and solve the joint optimization problem, discarding only dense correspondences, resulting in a consistency map $C_i \in \{0, 1\}$.

We iterate this process (starting with the correspondence association) until we reach convergence. In our experiments, we found that 3 to 5 iterations suffice to ensure a converged state.

To estimate the depth for the currently processed view, we then run a modified, non-rigid variant of COLMAP's PatchMatch [38]. Instead of simple homography warping, we apply the deformation to the point and its normal.

3.3. Implementation Details

In this section, we provide more details on our implementation of the NRMVS framework (Fig. 2). Algorithm 1 shows the overall method, introduced in Sec. 3.1, and Sec. 3.2.

Given input RGB images, we first pre-process the input. To estimate the camera pose for the images, we use the SfM implementation of Agisoft Photoscan [1]. Our tests showed accurate results for scenes containing at least 60% static background. A recent study [33] shows that 60∼90% of static regions in a scene results in less than 0.02 degree RPE [44] error for standard pose estimation techniques (see more discussion in the appendix C.3.Given the camera pose, we triangulate sparse SIFT matches [32], i.e., we compute the 3D position of the associated point by minimizing the reprojection error. We consider matches with a reprojection

error of less than 1 pixel to be successfully reconstructed (static inliers). The ratio of static inliers to the number of total matches is a simple yet effective indication of the non-rigidity in the scene. We pick the image pair with the highest ratio to indicate the minimum amount of deformation and use these as the canonical views to bootstrap our method.

Two important aspects of our main iteration are described in more detail: Our method to filter sparse correspondences (line 16 in Algorithm 1) is given in Algorithm 2. The hierarchical optimization algorithm (line 17 in Algorithm 1) including filtering for dense correspondences is given in Algorithm 3.

The joint optimization in our framework is a computationally expensive task. The deformation estimation, which strongly dominates the overall run-time, is CPU intensive, while the depth computation runs on the GPU. Specifically, for the face example shown in Fig. 1 (6 images with 100 deformation nodes) the computation time needed is approximately six hours (Intel i7-6700 3.4 GHz, NVIDIA GTX 980Ti). More details about the computational expense will be discussed in the appendix C.2.

Algorithm 1: Non-rigid multi-view stereo

Data: RGB input images $\{\mathbf{I}_k\}$
Result: Deformations $\{\mathbf{D}_k\}$, depth $\{d_k\}$

1 $P := \{1, \dots, k\}, Q := \emptyset$;
2 $\{\mathbf{C}_k\}$ = PhotoScanEstimateCameraPoses();
3 (i, j) = selectCanonicalViews();
4 $(d_i^{(0)}, \mathbf{n}_i^{(0)}, d_j^{(0)}, \mathbf{n}_j^{(0)})$ = ColmapPatchMatch($\mathbf{I}_i, \mathbf{I}_j$);
5 $\mathbf{D}_i^{(0)} = \mathbf{D}_j^{(0)}$ = initDeformationGraph($d_i^{(0)}, d_j^{(0)}$);
6 $\{\mathbf{u}_k\}$ = computeFeatureTracks();
7 $Q := Q \cup \{i, j\}$;
8 **while** $Q \neq P$ **do**
9 l = nextImage($P \setminus Q$);
10 $\{\mathbf{x}_k\}$ = liftKeyPointsTo3D($\{\mathbf{u}_k\}_{k \in Q}$) ;
11 $\{\mathbf{x}_i\} = \mathbf{D}_k^{-1}(\{\mathbf{x}_k\})$;
12 $\mathbf{D}_l^{(1)} = \mathbf{Id}$;
13 **for** $m = 1$ **to** N **do**
14 $\{\hat{\mathbf{x}}_l^{(m)}\} = \mathbf{D}_l^{(m)}(\{\mathbf{x}_i\})$;
15 $\{\mathbf{x}_l^{(m)}\}$ = projToRays($\{\hat{\mathbf{x}}_l^{(m)}\}, \{\mathbf{u}_l\}$) ;
16 $\{(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_l^{(m)})\}$ = filter($\mathbf{D}_l^{(m)}, \{(\mathbf{x}_i, \mathbf{x}_l^{(m)})\}$);
17 $\mathbf{D}_l^{(m+1)}$ = solve($\mathbf{D}_l^{(m)}, \{(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_l^{(m)})\}$, $\mathbf{I}_i, d_i^{(0)}, \mathbf{n}_i^{(0)}, \mathbf{I}_j, d_j^{(0)}, \mathbf{n}_j^{(0)}, \mathbf{I}_l$)
18 $\mathbf{D}_l = \mathbf{D}_l^{(m+1)}$;
19 $Q := Q \cup \{l\}$;
20 $(d_l^{(0)}, \mathbf{n}_l^{(0)})$ = NRPatchMatch($\{\mathbf{I}_k, \mathbf{D}_k\}_{k \in Q}$);
21 $\{(d_k, \mathbf{n}_k)\}_{k \in Q}$ = NRPatchMatch($\{\mathbf{I}_k, \mathbf{D}_k\}_{k \in Q}$);

Algorithm 2: Filtering of sparse correspondences

Data: Threshold d_{max}, Ratio $\tau \in (0, 1)$
1 **Function** `filter` $(\mathbf{D}_l, \{(\mathbf{x}_i, \mathbf{x}_l)\})$ **:**
2 **while** *true* **do**
3 $\mathbf{D}_l^*$ = solve($\mathbf{D}_l, \{(\mathbf{x}_i, \mathbf{x}_l)\}$);
4 $\{r_k\} = \{\|\mathbf{D}_l^*(\mathbf{x}_i) - \mathbf{x}_l\|_2\}$;
5 $e_{\text{max}} = \max\{r_k\}$;
6 **if** $e_{max} < d_{max}$ **then**
7 break;
8 $d_{\text{cut}} := \max\{d_{\text{max}}, \tau \cdot e_{\text{max}}\}$;
9 $\{(\mathbf{x}_i, \mathbf{x}_l)\} := \{(\mathbf{x}_i, \mathbf{x}_l) : r_k < d_{\text{cut}}\}$;
10 **return** $\{(\mathbf{x}_i, \mathbf{x}_l)\}$;

Algorithm 3: Solving the joint problem

Data: Threshold ρ_{max}, Ratio $\tau \in (0, 1)$
1 **Function**
`solve` $(\mathbf{D}_l, \{(\mathbf{x}_i, \mathbf{x}_l)\}, \mathbf{I}_i, d_i, \mathbf{n}_i, \mathbf{I}_j, d_j, \mathbf{n}_j, \mathbf{I}_l)$ **:**
2 $\hat{\mathbf{D}}_l = \mathbf{D}_l$;
3 **for** $m = 1$ **to** *levels* **do**
4 $\rho_{\text{cut}} := \tau \cdot (1 - \text{NCC}_{\text{min}}) = \tau \cdot 2$;
5 $C_p := 1 \quad \forall p$;
6 **while** *true* **do**
7 $\mathbf{D}_l^*$ = solveEq1($\hat{\mathbf{D}}_l$) ;
8 $\{r_p\} = \{C_p \cdot (1 - \rho(\mathbf{D}_l^*(\mathbf{x}_p), \mathbf{D}_l^*(\mathbf{n}_p), \mathbf{x}_p, \mathbf{n}_p))\}$;
9 $e_{\text{max}} = \max\{r_p\}$;
10 **if** $e_{max} < \rho_{max}$ **then**
11 $\hat{\mathbf{D}}_l := \mathbf{D}_l^*$;
12 break;
13 **if** $m = levels$ **then**
14 $\hat{\mathbf{D}}_l := \mathbf{D}_l^*$;
15 $C_p := 0 \quad \forall p : r_p > \rho_{\text{cut}}$;
16 $\rho_{\text{cut}} := \max\{\rho_{\text{max}}, \tau \cdot \rho_{\text{cut}}\}$
17 **return** $\hat{\mathbf{D}}_l$

4. Evaluation

For existing non-rigid structure from motion methods, different types of datasets are used to evaluate sparse points [21, 7], and dense video frames with small baseline (including actual camera view variation) [3]. Since our problem formulation is intended for dense reconstruction of scenes with sufficient variation in both camera view and deformation, there are only few examples applicable to our scenario [31, 50, 20]. Unfortunately, these datasets are either commercial and not available [31], or only exhibit rigid changes [50]. Few depth-based approaches share the input

Table 1: **Evaluation for ground truth data:** (a) using COLMAP, i.e., assuming a static scene, (b) applying our dense photometric optimization on top of an implementation of non-rigid ICP (NRICP), and (c) using different variants of our algorithm. S denotes *sparse*, D denotes *dense*, photometric objective. N equals the number of iterations for sparse correspondence association (see paper for more details). We compute the mean relative error (MRE) for all reconstructed values as well as the overall completeness. The last row (w/o filter) shows the MRE, with disabled rejection of outlier depth values, i.e., a completeness of 100 %.

			Ours				
	COLMAP [38]	NRICP [28]	S ($N = 1$)	S ($N = 10$)	D	S ($N = 1$) + D	S ($N = 10$) + D
Completeness	68.74 %	99.30 %	97.24 %	97.71 %	96.41 %	98.76 %	**98.99** %
MRE	2.11 %	0.53 %	1.48 %	1.50 %	2.37 %	1.12 %	**1.11** %
MRE w/o filter	6.78 %	0.74 %	2.16 %	2.05 %	3.32 %	1.63 %	**1.34** %

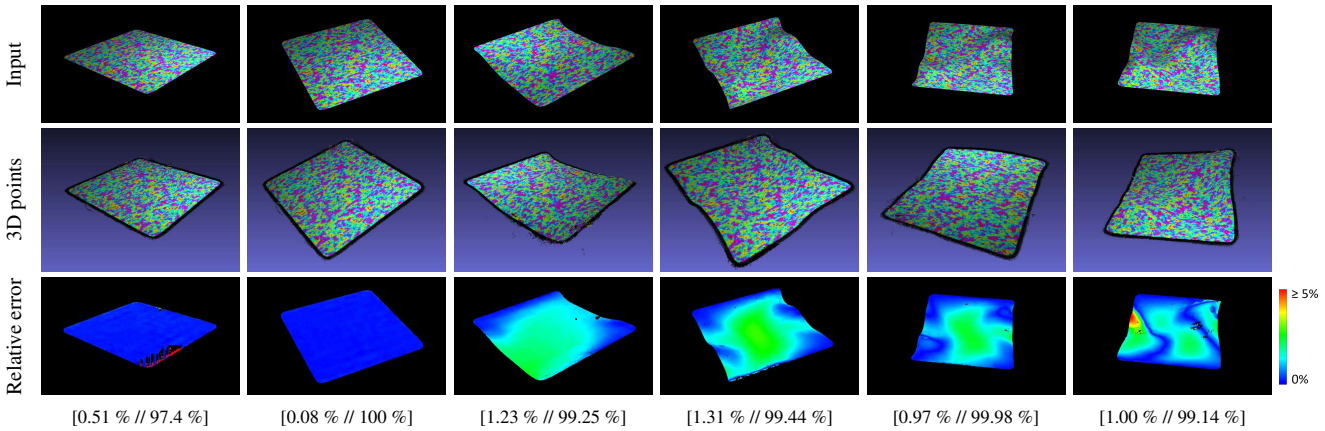


Figure 5: **Quantitative evaluation with synthetic data:** We created images of a deforming surface with 10 different views. For the evaluation, we randomly chose six examples from the set and reconstructed the surface. The first row shows the input images. The first two columns show the chosen canonical views. The results of the reconstructed surface (with the point cloud propagated to each view) are shown in the second row. In the third row, we visualize the relative depth error compared to the ground truth. We also show the mean relative depth error value (%) and the completeness (%). The overall quantitative evaluation including a comparison to other baselines are shown in Table 1.

RGB as well [20], but the quality of the images is not sufficient for our method (i.e., severe motion blur, low resolution (VGA) that does not provide sufficient detail for capturing non-rigid changes). Thus, we created both synthetic data and captured real-world examples for the evaluation. To quantitatively evaluate how our method can accurately capture a plausible deformation and reconstruct each scene undergoing non-rigid changes, we rendered several synthetic scenes with non-rigid deformations as shown in the first row of Fig. 5. We also captured several real-world scenes containing deforming surfaces from different views at different times. Some examples (face, rubber globe, cloth and paper) appear in Fig. 6, and several more viewpoints are contained in the supplementary video[1].

[1] https://youtu.be/et_DFEWeZ-4

4.1. Quantitative Evaluation with Synthetic Data

First, we evaluate the actual depth errors of the reconstructed depth of each time frame (i.e., propagated/refined to a specific frame), and of the final refined depth of the *canonical view*. Because we propose the challenging new problem of reconstructing non-rigid dynamic scenes from a small set of images, it is not easy to find other baseline methods. Thus, we conduct the evaluation with an existing MVS method, COLMAP [38], as a lower bound, and use as an upper bound a non-rigid ICP method similar to Li et al. [28] based on the ground truth depth. The non-rigid ICP using the point-to-plane error metric serves as a geometric initialization. We refine the deformation using our dense photometric alignment (see Sec. 3.2).

To compare the influence of our proposed objectives for deformation estimation, i.e. sparse 3D-3D correspondences and dense non-rigid photometric consistency, we evaluate

Figure 6: **Qualitative evaluation with real data**: In each row, the first two columns show the views used to create each canonical surface. The first column of each result row (even row) shows the original canonical surface. The remaining views from the second column of each result row shows the propagated version of reconstructed surfaces for each view.

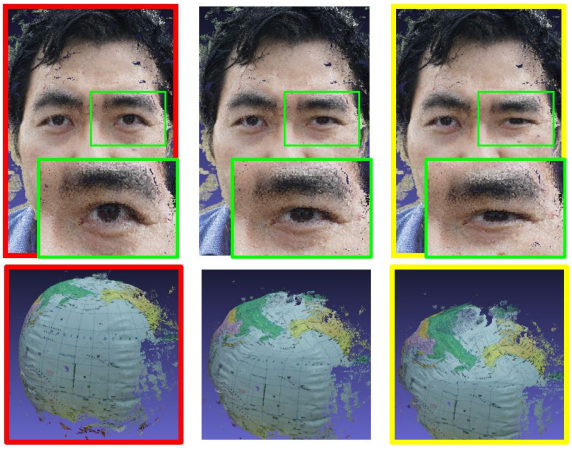
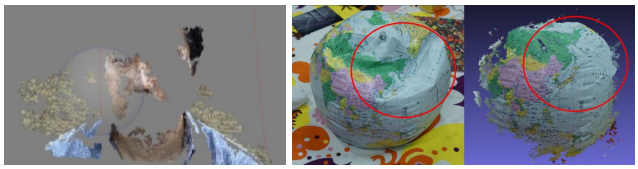


Figure 7: **Dynamic 3D scene interpolation with in-between deformations:** We interpolate a point cloud between two reconstructed views from their depth and deformation. We show two key-frames, source and target, denoted as red and yellow frames respectively, and then demonstrate the interpolated *intermediate point cloud* in the middle column. For the top row, zoomed in-set images of the eye region show how the deformation is applied to the intermediate point cloud. More interpolated frames and 4D animations created from our deformation estimation are shown in the supplementary video with various views[1].

our algorithm with different settings. The relative performance of these variants can be viewed as an ablation study. We perform evaluation on the following variants: 1) considering only the sparse correspondence association using different numbers of iterations (see Sec. 3.2), 2) considering only the dense photometric alignment, and 3) the combination of sparse and dense. The results of the quantitative evaluation can be found in Table 1. As can be seen, all methods/variants obtain a mean relative error $< 2.4\%$, overall resulting in faithfully reconstructed geometry. Our joint optimization algorithm considerably improves the reconstruction result both in terms of accuracy (by a factor of 1.9) and completeness (by 30 pp, a factor of 1.4). Additionally, we compute the mean relative depth error (MRE) without rejecting outliers; i.e., resulting in depth images with a completeness of 100%.

4.2. Qualitative Evaluation with Real Data

Fig. 6 shows results of our non-rigid 3D reconstruction. For each pair of rows, we show six input images and the corresponding deformed 3D point clouds. Note that the *deformed* surfaces belong to the collection of 3D reconstructed points propagated by the computed deformations using the other views as described in Sec. 3.2. The point cloud of each first column of Fig. 6 shows the first canonical surface (triangulated points from two views with minimal deformation). For evaluation purposes, we visualize each reconstructed scene from a similar viewpoint as one of the canonical views. More viewpoints of the reconstructed 3D results can be found in the supplementary video[1].



(a) Bad canonical views selection (b) Ambiguity along view direction

Figure 8: **Failure cases:** (a) shows the result of canonical surface reconstruction from two views that are incorrectly selected (large deformation between two views: images in first and third column in top row of Fig. 6). While the camera pose is successfully computed, since there are large portions of non-rigid changes happening in the upper part of face and near the mouth, there are many holes on the face, which is not the best case if we choose this pair. (b) shows a failure case when deformation (red circles) occurs along the view direction, which causes the ambiguity.

4.3. Dynamic Scene Interpolation

Since we estimate deformations between each view and the canonical surface, once all deformation pairs have been created, we can easily interpolate the non-rigid structure. To blend between the deformations, we compute interpolated deformation graphs by blending the rigid body transform at each node using dual-quaternions [23].

In Fig. 7, we show interpolated results from reconstructed scenes of the face example and the globe example shown in Fig. 6. The scene deformations used for this interpolation (like key-frames) are framed in as red and yellow. Note that even though the estimated deformation is defined between each view and the canonical pose, any combination of deformation interpolation is possible. More examples and interpolated structures from various viewpoints can be found in the supplementary video[1].

5. Conclusion and Discussion

We propose a challenging new research problem for dense 3D reconstruction of scenes containing deforming surfaces from sparse, wide-baseline RGB images. As a solution, we present a joint optimization technique that optimizes over depth, appearance, and the deformation field in order to model these non-rigid scene changes. We show that an MVS solution for non-rigid change is possible, and that the estimated deformation field can be used to interpolate motion in-between views.

It is also important to point out the limitations of our ap-

proach (Fig. 8). We first assume that there is at least one pair of images that has minimal deformation for the initial canonical model. This can be interpreted as the first step used by many SLAM or 3D reconstruction algorithms for the initial triangulation. Fig. 8(a) shows an example of a canonical surface created from two views that contain too much deformation, only leading to a partial triangulation. Fig. 8(b) shows an example where the deformation occurs mostly along the view direction. While we successfully estimate the deformation and reconstruct a similar example shown in Fig. 6, depending on the view this can cause an erroneous estimation of the deformation. On the other hand, we believe that recent advances in deep learning-based approaches to estimate depth from single RGB input [12] or learning local rigidity [33] for rigid/non-rigid classification can play a key role for both the initialization and further mitigation of these ambiguities.

References

[1] Agisoft. PhotoScan: MVS software, 2000–2004. 4, 11

[2] B. Allain, J.-S. Franco, and E. Boyer. An efficient volumetric framework for shape tracking. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[3] M. D. Ansari, V. Golyanik, and D. Stricker. Scalable dense monocular surface reconstruction. *Intl. Conf. on 3D Vision*, 2017. 1, 2, 5

[4] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo - stereo matching with slanted support windows. In *British Machine Vision Conf. (BMVC)*, pages 14.1–14.11, 2011. 1, 2

[5] M. Bojsen-Hansen, H. Li, and C. Wojtan. Tracking surfaces with evolving topology. *ACM Trans. on Graphics (TOG)*, 31(4):53, 2012. 2

[6] N. D. F. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *European Conf. on Computer Vision (ECCV)*, pages 766–779, 2008. 1

[7] Y. Dai, H. Deng, and M. He. Dense non-rigid structure-from-motion made easy - A spatial-temporal smoothness based solution. *CoRR*, abs/1706.08629, 2017. 1, 2, 5

[8] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM Trans. on Graphics (SIGGRAPH)*, 27:1–10, 2008. 2

[9] M. Dou, H. Fuchs, and J.-M. Frahm. Scanning and tracking dynamic objects with commodity depth cameras. In *IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*, pages 99–106, 2013. 2

[10] M. Dou, S. Khamis, Y. Degtyarev, P. Davidson, S. R. Fanello, A. Kowdle, S. O. Escolano, C. Rhemann, D. Kim, J. Taylor, et al. Fusion4d: Real-time performance capture of challenging scenes. *ACM Trans. on Graphics (TOG)*, 35(4):114, 2016. 2

[11] M. Dou, J. Taylor, H. Fuchs, A. Fitzgibbon, and S. Izadi. 3D scanning deformable objects with a single RGBD sensor. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 2

[12] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 9

[13] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007. 1, 2

[14] J. Gall, B. Rosenhahn, and H. P. Seidel. Drift-free tracking of rigid and articulated objects. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008. 2

[15] S. Galliani, K. Lasinger, and K. Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Intl. Conf. on Computer Vision (ICCV)*, 2015. 1, 2

[16] R. Garg, A. Roussos, and L. Agapito. Dense variational reconstruction of non-rigid surfaces from monocular video. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2

[17] K. Guo, F. Xu, Y. Wang, Y. Liu, and Q. Dai. Robust non-rigid motion tracking and surface reconstruction using L0 regularization. *Intl. Conf. on Computer Vision (ICCV)*, 2015. 2

[18] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. 1, 2

[19] C. Hernández, G. Vogiatzis, G. J. Brostow, B. Stenger, and R. Cipolla. Non-rigid photometric stereo with colored lights. In *Intl. Conf. on Computer Vision (ICCV)*, 2007. 2

[20] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger. VolumeDeform: Real-time volumetric non-rigid reconstruction. In *European Conf. on Computer Vision (ECCV)*, 2016. 1, 2, 5, 6

[21] S. H. N. Jensen, A. Del Bue, M. E. B. Doest, and H. Aanæs. A benchmark and evaluation of non-rigid structure from motion. *Arxiv*, abs/1801.08388, 2018. 1, 2, 5

[22] T. Kanade and D. D. Morris. Factorization methods for structure from motion. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 356(1740):1153–1173, 1998. 2

[23] L. Kavan, S. Collins, J. Žára, and C. O'Sullivan. Skinning with dual quaternions. In *Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 39–46. ACM, 2007. 8

[24] S. Kumar, Y. Dai, and H. Li. Monocular dense 3D reconstruction of a complex dynamic scene from two perspective frames. *Intl. Conf. on Computer Vision (ICCV)*, 2017. 2

[25] A. Kundu, K. M. Krishna, and C. V. Jawahar. Real-time multibody visual SLAM with a smoothly moving monocular camera. In *Intl. Conf. on Computer Vision (ICCV)*, 2011. 1, 2

[26] Ľ. Ladický, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. Torr. Joint optimisation for object class segmentation and dense stereo reconstruction. In *British Machine Vision Conf. (BMVC)*, 2010. 2

[27] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. *ACM Trans. on Graphics (TOG)*, 28(5):175, 2009. 2

[28] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust single-view geometry and motion reconstruction. In *ACM Trans. on Graphics (SIGGRAPH Asia)*, pages 175:1–175:10, 2009. 6

[29] H. Li, L. Luo, D. Vlasic, P. Peers, J. Popović, M. Pauly, and S. Rusinkiewicz. Temporally coherent completion of dynamic shapes. *ACM Trans. on Graphics (TOG)*, 31(1):2, 2012. 2

[30] H. Li, R. W. Sumner, and M. Pauly. Global correspondence optimization for non-rigid registration of depth scans. In *Computer Graphics Forum*, volume 27, pages 1421–1430, 2008. 2

[31] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev. 3d self-portraits. *ACM Trans. on Graphics (TOG)*, 32(6):187:1–187:9, Nov. 2013. 2, 5

[32] D. G. Lowe. Object recognition from local scale-invariant features. In *Intl. Conf. on Computer Vision (ICCV)*, 1999. 4, 12

[33] Z. Lv, K. Kim, A. Troccoli, D. Sun, J. Rehg, and J. Kautz. Learning rigidity in dynamic scenes with a moving camera for 3d motion field estimation. In *European Conf. on Computer Vision (ECCV)*, 2018. 3, 4, 9

[34] C. Malleson, M. Klaudiny, J. Y. Guillemaut, and A. Hilton. Structured representation of non-rigid surfaces from single view 3D point tracks. In *Intl. Conf. on 3D Vision*, volume 1, pages 625–632, 2014. 2

[35] N. J. Mitra, S. Flöry, M. Ovsjanikov, N. Gelfand, L. J. Guibas, and H. Pottmann. Dynamic geometry registration. In *Symposium on Geometry Processing (SGP)*, pages 173–182, 2007. 2

[36] R. A. Newcombe, D. Fox, and S. M. Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 1, 2

[37] V. Rabaud and S. Belongie. Re-thinking non-rigid structure from motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008. 2

[38] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 1, 2, 3, 4, 6, 11, 12

[39] T. Schöps, J. L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3, 12

[40] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006. 1, 2

[41] M. Slavcheva, M. Baust, D. Cremers, and S. Ilic. Killingfusion: Non-rigid 3d reconstruction without correspondences. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[42] M. Slavcheva, M. Baust, and S. Ilic. Sobolevfusion: 3d reconstruction of scenes undergoing free non-rigid motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[43] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing (SGP)*, volume 4, pages 109–116, 2007. 2, 3

[44] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2012. 4

[45] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. *ACM Trans. on Graphics (TOG)*, 26(3):80, 2007. 2, 3, 4

[46] A. Tevs, A. Berner, M. Wand, I. Ihrke, M. Bokeloh, J. Kerber, and H.-P. Seidel. Animation cartography—intrinsic reconstruction of shape and motion. *ACM Trans. on Graphics (TOG)*, 31(2):12, 2012. 2

[47] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *Intl. J. of Computer Vision*, 9(2):137–154, 1992. 2

[48] B. Triggs. Factorization methods for projective structure and motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1996. 2

[49] R. Wang, L. Wei, E. Vouga, Q. Huang, D. Ceylan, G. Medioni, and H. Li. Capturing dynamic textured surfaces of moving targets. In *European Conf. on Computer Vision (ECCV)*, 2016. 2

[50] T. Y. Wang, P. Kohli, and N. J. Mitra. Dynamic SfM: Detecting Scene Changes from Image Pairs. *Computer Graphics Forum*, 2015. 1, 2, 5

[51] M. Zeng, J. Zheng, X. Cheng, and X. Liu. Templateless quasi-rigid shape modeling with implicit loop-closure. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 2

[52] G. Zhang, J. Jia, and H. Bao. Simultaneous multi-body stereo and segmentation. In *Intl. Conf. on Computer Vision (ICCV)*, 2011. 1, 2

[53] M. Zollhöfer, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, C. Loop, C. Theobalt, and M. Stamminger. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Trans. on Graphics (TOG)*, 33(4), 2014. 2

Appendices

We give an overview of the mathematical symbols used in our algorithm (Sec. A). In Sec B, we enumerate the parameters that we used for the optimization and Patchmatch steps shown in Sec 3.2. In Sec. C, we show additional implementation details as well as extra experiments for our preprocessing steps (Sec. C.3, Sec. C.4), and show how our optimization effects the photometric consistency. In Sec. C.2, we discuss the runtime of our approach. Finally, in Sec. D, we provide extra experimental results in both qualitative and quantitative manners.

We demonstrate various 4D animation examples created from few images by using our NRMVS framework in the attached video[2].

A. List of Mathematical Symbols

Symbol	Description
$\mathbf{D}_i$	deformation from canonical pose to image i
$\mathbf{v}, \mathbf{x}$	point in $\mathbb{R}^3$
$\mathbf{n}$	normal vector in $\mathbb{R}^3$
$\mathbf{u}_i$	SIFT keypoint in image i
C_i	consistency mask $\in \{0, 1\}$ for point $\mathbf{x}_i$
$\rho_{r,s}$	weighted NCC value for images r and s [38]
$\mathbf{I}_i$	greyscale image i
d_i	depthmap for image i

B. Parameter choices

Parameter	Value
w_{sparse}	1000
w_{dense}	0.01
w_{reg}	10
$d_{\max}$	0.1 cm . . . 0.5 cm
$\rho_{\max}$	0.9
τ	0.9

$d_{\max}$ is chosen depending on the scale of the scene geometry; e.g., we choose 0.1 cm for the face example and 0.5 cm for the globe example. In case of the synthetic ground truth data, we use $d_{\max} = 0.01$, with the rendered plane having a size of 6.

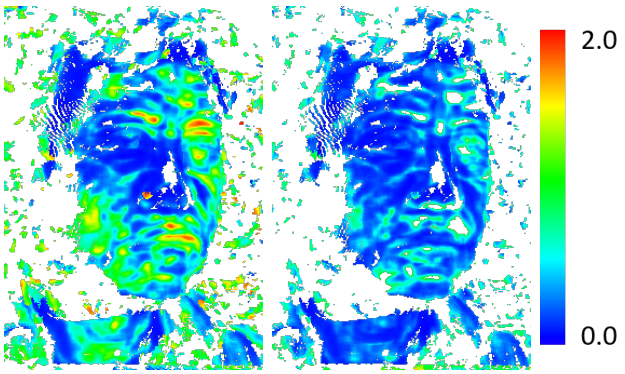
The parameter `filter_min_num_consistent` in the implementation of COLMAP's PatchMatch [38] as well as our non-rigid PatchMatch is set to 1 (default 2). Besides that, we use COLMAP's default parameters throughout our pipeline.

[2] https://youtu.be/et_DFEWeZ-4

C. Approach

C.1. Deformation Estimation

In Fig. 9, we show an example of the photometric consistency before and after the estimation of the non-rigid deformation. As shown, the photometric error gets reduced and some inconsistent regions (not satisfying a user-specified threshold $\rho_{\max}$) get masked out by the consistency map C_i.



(a) Initial state (b) After optimization

Figure 9: Photoconsistency cost including consistency mask $C_i \cdot (1 - \text{NCC})$ between Fig. 11(a) and (c). Masked out pixels are transparent in (b).

C.2. Performance

In Table 2, we report the run-time for the face example (Fig. 11) with 100 deformation nodes, depending on the number of iterations N used for the sparse correspondence association.

Step	Time ($N = 1$)	Time ($N = 5$)
Total	154 min	422 min
Filter	0.2%	0.4%
Optimize	92.3%	96.9%
Depth	7.5%	2.7%

Table 2: Computation time (in minutes) needed for different steps to process the example in Fig. 11 depending on the number of iterations N (see main paper for more details): Filtering of sparse correspondences, Joint hierarchical optimization and depth estimation; file I/O not included.

C.3. Camera Pose Estimation

In Fig. 10, we show an example result for the estimated camera poses using Agisoft PhotoScan [1]. As can be seen, the camera poses for the input images have been successfully recovered.

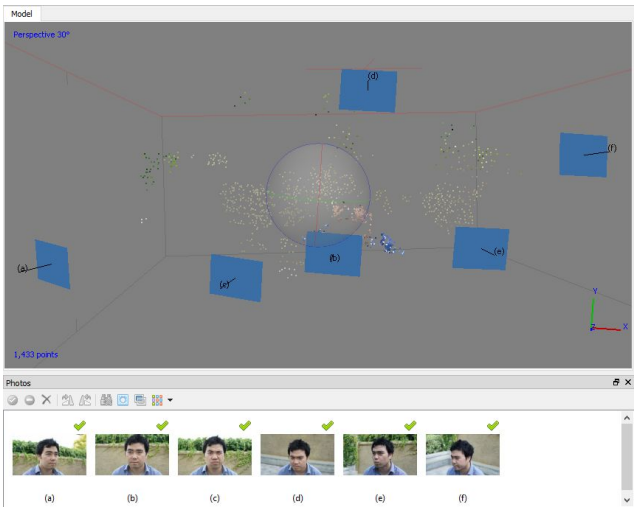


Figure 10: Screenshot of Agisoft PhotoScan

	(b)	(c)	(d)	(e)	(f)
(a)	**10.3%**	0.42%	0.00%	0.00%	0.83%
(b)		0.17%	0.24%	1.14%	0.00%
(c)			0.00%	0.00%	0.65%
(d)				0.28%	0.88%
(e)					0.26%

Table 3: Confusion matrix for static inlier ratio for all 2-view combinations (see Fig. 11).

C.4. Canonical View Selection

To pick the canonical views, we analyze how many matches result in a faithful static reconstruction. I.e., we triangulate each match (after doing the ratio test with $r = 0.7$ [32]) and reject those with a reprojection error above 1 pixel. As can be seen in Table 3, the image pair (a)-(b) dominates the ratio of static inliers. Therefore, our algorithm chooses these views to reconstruct the initial canonical surface.

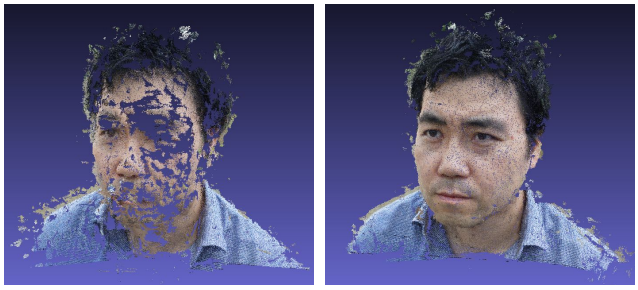
(a) (b) (c)

(d) (e) (f)

Figure 11: Input images for the face example

D. Additional Results

In Fig. 12, we show an example result comparing our algorithm with a state-of-the-art MVS approach that performs best in a recent survey [39]. As can be seen, the geometry of the deforming region can not be reconstructed successfully, if the method assumes static geometry.



(a) COLMAP (b) Ours

Figure 12: Comparison with COLMAP [38]: a state-of-the-art MVS algorithm for static scenes fails to reconstruct images undergoing non-rigid motion (Fig. 11).