Patch-based Non-rigid 3D Reconstruction from a Single Depth Stream

Carmel Kozlov Miroslava Slavcheva Slobodan Ilic
Technische Universität München
Siemens Corporate Technology

Abstract

We propose an approach for 3D reconstruction and tracking of dynamic surfaces using a single depth sensor, without any prior knowledge of the scene. It is robust to rapid inter-frame motions due to the probabilistic expectation-maximization non-rigid registration framework. Our pipeline subdivides each input depth image into non-rigidly connected surface patches, and deforms it towards the canonical pose by estimating a rigid transformation for each patch. The combination of a data term imposing similarity between model and data, and a regularizer enforcing as-rigid-as-possible motion of neighboring patches ensures that we can handle large deformations, while coping with sensor noise. We employ a surfel-based fusion technique, which lets us circumvent the repeated conversion between mesh and signed distance field representations which are used by related techniques. Furthermore, a robust keyframe-based scheme allows us to keep track of correspondences throughout the entire sequence. Through a variety of qualitative and quantitative experiments, we demonstrate resistance to larger motion and achieving lower reconstruction errors than related approaches.

1. Introduction

Recently, techniques for non-rigid scene reconstruction have seen increased attention due to advancements in the virtual and augmented reality domains [38]. Considerable efforts have been devoted in both academia and industry to develop robust dynamic reconstruction and tracking algorithms using specialized mutli-view systems [8, 34, 5, 19, 10]. However, the rise of commodity RGB-D sensors has inspired a new wave of development, as their portability allows capture outside of the studio setting, which is more easily accessible to the general user. Despite significant progress [23, 11, 16, 27, 15], most methods are limited to very contrived motions, leaving accurate dynamic scene capture a challenging open problem. As objects deform freely and the acquisition process is unreliable, including noisy data and occlusions, this problem involves a large number of parameters and is highly underconstrained.

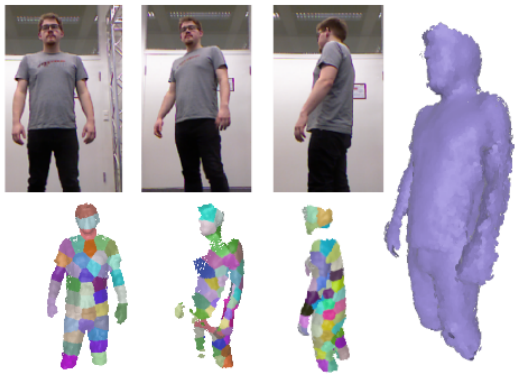

Figure 1. Given a single depth stream, we divide each frame into surface patches and deform it towards a canonical pose using a non-rigid deformation framework. Despite noisy data and occlusions we are able to obtain a 3D model of the deforming object, shown on the right hand side. The RGB images here are used for visualization purposes only.

The first method to achieve single-stream dynamic reconstruction was DynamicFusion [23], which devised a way to simultaneously track and reconstruct a deforming surface in real time. However, this method failed to reconstruct tangential motion and suffered from drift. More recently, VolumeDeform [16] incorporated the use of color SIFT features to enhance the handling of tangential motion. However, this method was still limited to slow inter-frame motions. Recent works such as the approach of Dou *et al*. [11] showed impressive results, yet they were computationally intensive, rendering them impractical to the average user.

In this paper, we aim for a balanced solution that is computationally reasonable, handles drift and fast inter-frame motions, and does not require prior knowledge of the scene. Our key idea is using rigidity constraints between surface patches to enable non-rigid deformation under fast inter-frame motion. In particular, we split the depth data into patches and back-project to a 3D point cloud, as illustrated in Figure 1. These patches impose rigidity constraints on their neighbors, such that they deform within a

locally rigid neighborhood. Each depth frame is deformed towards a canonical pose under a probabilistic expectation-maximization variant of non-rigid iterative closest point (ICP) [1]. Patch based representations have been explored for multi-view reconstruction [4, 3], but partial views, high occlusions and noisy data make their use challenging for the single-view setting. To enable the use of patch-based representation and to handle lengthy sequences we leverage the idea of keyframes [10, 6]. To obtain the final 3D reconstruction, we fuse all the models obtained from our keyframes, using the correspondence information tracked through the entire sequence.

To summarize, the contributions of our paper are:

- We incorporate patch-rigidity constraints for single stream depth-based 3D reconstruction, which allows our method to work for arbitrary shapes and to be robust to fast motions.
- Our method uses a surfel-based representation for both the deformation estimation as well as the fusion framework in a single-stream depth setting.
- We demonstrate that our method has the ability to cope with larger movements compared to previous works, while achieving lower reconstruction errors.

2. Related Work

While reconstructing a static scene entails only estimating the 6 degrees-of-freedom camera pose, dynamic scenes pose a significantly more challenging task as each point may have a different motion [13]. In this section we first discuss approaches that address the dynamic reconstruction problem but require a multi-view setting or are conditioned on prior knowledge of the scene. Next, we review state-of-the-art work on single-stream *dynamic* reconstruction, the exact setting we are tackling. Finally, we discuss approaches employing a patch-based representation, from which we draw inspiration for our proposed methodology.

Multi-view Reconstruction: Mutli-view setups decrease the underconstrained nature of this problem, as multiple cameras provide more data, and reduce noise and occlusions significantly. Early methods in this domain [8, 34, 4] required many hours of processing, as well as the installation and calibration of many high-quality cameras making them non-practical for everyday use. While more recent methods [19, 10] performed significantly better, they still required a great amount of processing time and the studio setting severely limited their application. Recently, Fusion4D [10] and Motion2Fusion [9] have demonstrated compelling results, but both required multiple depth cameras, each with a dedicated GPU. Such specialized setups are not available to the general public. In contrast to these approaches, our method focuses on the single-stream case which vastly reduces computational time. Additionally, the use of portable commodity hardware allows anyone to use our approach and enables reconstructions outside the studio setting.

Reconstruction using Prior Knowledge: Some methodologies employ the use of prior knowledge to constrain the problem. Zollhöfer *et al*. [37] deformed a rigidly acquired template to each new depth frame. Both the work of Yu *et al*. [35] and Liu *et al*. [21] showed promising results using a single RGB-D stream, but first required generating a template of the object in a static pose. Templates allow subject-specific reconstruction only, and may not be straightforward to acquire, especially for some subjects such as animals and children. Alternative frameworks employed different prior knowledge on the class of reconstructed objects, such as parametric human shape [2], hand [30] and face models [31] or skeletons [14], which prevented the capture of arbitrary surfaces. Other systems that are less technically demanding combine prior knowledge and a multi-view system. This can be achieved by obtaining an input mesh from a multi-vew system and then deforming a template to it [33, 4]. In contrast, our approach can work on arbitrary shapes and does not require prior knowledge of the scene.

Single-stream Dynamic Reconstruction: The rise of single-stream non-rigid capture started with the seminal DynamicFusion method [23], which was able to incrementally track and reconstruct a non-rigid scene in real-time using dense depth-based correspondences. However, their method was prone to drift, unable to handle tangential motions, and limited to highly contrived and slow motions. DynamicFusion inspired a line of follow-up work [16, 15], but all techniques have only demonstrated results on relatively slow and controlled movements. VolumeDeform [16] added color SIFT features to better handle tangential movement, yet it was still unable to handle fast inter-frame motions. In comparison, our method offers a unique way to handle fast movements through a patch-based deformation framework. KillingFusion [27] proposed a solution handling larger motion and topological changes, using a deformation field operating purely over a signed distance field (SDF). While it performed favorably, it lacked data association, severely limiting the range of possible applications such as character animation. In contrast, our method stays in one data representation which allows for correspondence tracking throughout the entire sequence used for reconstruction.

Recently, Guo *et al*. [15] incorporated surface albedo constraints in order to capture surfaces of uniform geometry. BodyFusion [36] focused on reconstructing humans by adding motion priors and solved for the skeleton and graph-node deformations simultaneously. However, these two methods suffered from drift and performed poorly under

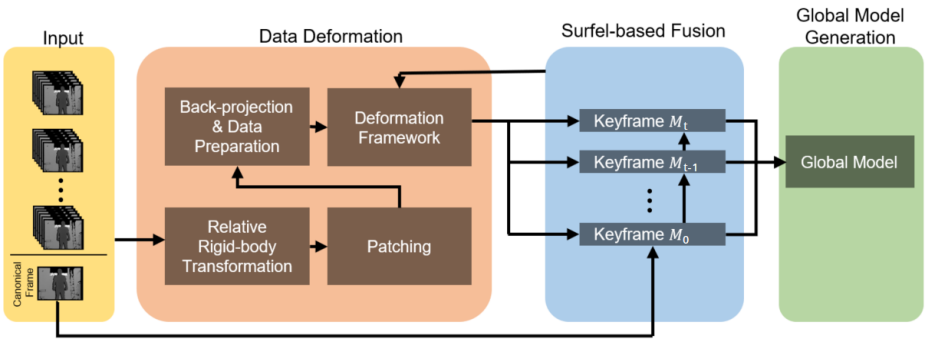

Figure 2. Our proposed pipeline. The reconstruction process is incremental; each frame is patched and back-projected. We divide the sequence into keyframes, in each we build a local incremental model. We obtain the final reconstruction using a correspondence-based global fusion approach.

fast inter-frame motions. To combat drift, Dou *et al*. [11] employed non-rigid bundle adjustment to automatically detect loop closures. They demonstrated high quality results, but required hours of processing and they were still unable to fully handle rapid movements. We aim to handle drift by the use of a keyframe-based approach which eliminates the use of bundle-adjustment based techniques and speeds up the reconstruction pipeline.

Patch-based Representations: Patch-based representations have been proposed for the problem of non-rigid structure from motion [25, 7] . The work of Varol *et al*. [32] calculated homographies between corresponding planar patches from a short image sequence. This enabled reconstruction of textured deformable surfaces. Similarly, Agapito *et al*. [12] reconstructed strongly deforming objects, such as a waving flag, by dividing the surface into overlapping patches and reconstructing each of these inidvidually to obtain a final 3D model. Recent works in non-rigid structure from motion (NRSfM) such as the work of Ji *et al*. [18] have shown more accurate results than previous state-of-the-art methods for non-rigid shape reconstruction. The patch-based method of Cagniart *et al*. [3] has been particularly successful in the multi-view tracking setting, as it splits the surface into non-rigidly connected rigid patches, which can withstand noise and occlusions. We also adopt a patch-based representation for our approach, but we couple it with a keyframe-based framework to address problems of the single-view sensor setting, i.e., partial views, noisy data and occlusions.

3. Proposed Method

Our framework reconstructs a 3D model of a dynamic object, as illustrated in Figure 2. The method keeps track of correspondences throughout the sequence allowing for fast inter-frame motions. The most crucial aspect of our pipeline is the use of a patch-based surface representation, enabling us to deform incoming frames non-rigidly onto a growing model. The reconstruction process is incremental; each frame is patched (§3.1) and deformed onto a growing model using a deformation framework (§3.2). We use a surfel-based representation for the data, which enables us to fuse new geometry onto the model easily (§3.3).

The idea of patch-based representation has been used before in multi-view reconstruction and tracking [4]. The key to enabling this for single-stream 3D reconstruction is the incorporation of keyframes into our approach (§3.4). This is accomplished by dividing the sequence of depth images into keyframes, which hold F consecutive frames, in each we build a local incremental model. Finally, we fuse all the models to obtain a global model of the deforming object (§3.5). As we use a surfel-based representation, we provide a quick overview of our visualization approach (§3.6).

3.1. Surface Patching

Our patching approach constructs roughly equally sized patches distributed uniformly on the object surface. The patching is computed directly on each incoming depth frame, as depicted in Figure 3, allowing us to propagate the patch information for each point when projected to 3D. We employ a superpixel approach for the patch generation, based on the mask Simple Linear Iterative Clustering (maskSLIC) [17] algorithm. In certain situations the mask of the object is complex, containing holes or disparate features, which makes it possible to generate disconnected patches. These detached patches may contain regions on the surface object, which deform non-rigidly to different locations. We show an example of detached patches in Figure 3. Finally, we ensure that detached patches are split, such that the deformation framework is not adversely affected by such patches.

3.2. Patch-based Deformation Framework

We employ a patch-based surface rigidity deformation framework, inspired from [4], to compute the non-rigid deformations of each frame with respect to the canonical pose.

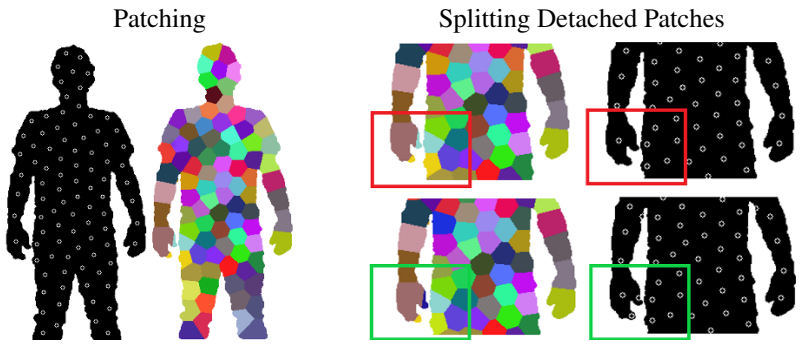

Figure 3. The patching, depicted on the left, is accomplished by placing seeds and clustering all pixels to a nearest seed location. Patches which are detached due to disparate features, as illustrated on the right hand side, are split to ensure they can deform non-rigid to their respective locations. On the top row we show the output from the patching framework before applying the last step of splitting the detached patches. On the bottom row we show the additional patches that are created to split the detached patches.

The surface of an object $\mathcal{X} = (\mathbf{X})$ is represented in 3D coordinates, where $\mathbf{X} = \{x_v\}_{v=1}^{V} \subset \mathbb{R}^3$ are the positions of all its points. The task of the deformation framework is to register the points of the incoming frame $\mathcal{X}$ to the canonical pose $\mathcal{Y}$. With each incoming depth frame, we deform the frame onto the reference model and integrate the new geometry.

Patch Parameterization. Once the surface patching is completed on a frame, the depth data is back-projected into a point cloud. A rigid transformation with respect to the world coordinates is associated with each patch P_k, parameterized by the position of the patch center $\mathbf{c}_k$ and a rotation matrix $\mathbf{R}_k$. The rigid transformation for each point x_k can be computed as follows:

$$x_k = \mathbf{R}_k(\mathbf{x} - \mathbf{c}_k) + \mathbf{c}_k, \tag{1}$$

The parameters representing each patch $\Theta = \{\theta_k\}_{k=1:K}$ are combined into a vector $\Theta = \{\mathbf{R}_k, \mathbf{c}_k\}_{k=1:N_p}$, describing the entire surface. N_p describes the number of patches on the surface. The framework computes one rigid transformation per patch, while the motion of each vertex is computed using linear blending from the patch center. A gaussian weighting function, α_k is employed in the blending, and the position for a point x_k is computed using the patch and its direct neighbors $k \cup \mathcal{N}_k$:

$$x = \sum_{s \in k \cup \mathcal{N}_k} \alpha_s \mathbf{x}_s. \tag{2}$$

Deformation Framework. Given an observed point cloud $\mathcal{X} = (\mathbf{X})$ where $\mathcal{Y} = (\mathbf{Y})$ is the canonical pose, we register $\mathbf{X}$ to $\mathbf{Y}$, estimating $\hat{\Theta}$ such that $\mathbf{X}(\hat{\Theta})$ resembles $\mathbf{Y}$ as closely as possible. This is accomplished in a two-part process. First, each point in $\mathcal{Y}$ is associated with a point in $\mathcal{X}$ to build a correspondence set $\mathcal{C}$. Next, $\hat{\Theta}$ is estimated by minimizing an energy E which describes the discrepancy between each pair association from the first step $\mathcal{C} : \hat{\Theta} = \arg\min_{\Theta} E(\Theta; \mathcal{C})$.

Assuming that $\mathbf{Y}$ and $\mathbf{X}(\Theta)$ lie near to each other, the correspondence set $\mathcal{C}$ is built with a nearest-neighbor search. A deformation parameter Θ is initialized and $\mathbf{X}(\Theta)$ is transformed accordingly. We then evaluate a new correspondence set $\mathcal{C}$ for $\mathbf{X}(\Theta)$, and the process repeats until convergence. This iterative approach is accomplished with a variant of non-rigid ICP [1].

Energy Definition. We utilize an energy formulation combining a data term E_{data}, which enforces the deformed cloud to be aligned with the reference model, and a regularization term E_r, which imposes as-rigid-as-possible constrains between neighboring patches. Given a correspondence set $\mathcal{C}$, where each patch i in the input cloud is associated with a point p on the target, we formulate the data term as:

$$E_{data}(\Theta; \mathcal{C}) = \sum_{(i,p \in C)} w_{i,p} \| y_i - x_p(\Theta) \|_2^2, \tag{3}$$

where each correspondence pair is associated with a weight $w_{i,p}$. Given two neighboring patches k and $l \in \mathcal{N}_k$, the rigidity energy enforces the predictions $\mathbf{x}_k(v)$ and $\mathbf{x}_l(v)$ of a vertex point v to be consistent:

$$E_r(\Theta) = \sum_{k=1}^{K} \sum_{l \in \mathcal{N}_k} \sum_{v \in P_k \cup P_l} w_{kl}(v) \| x_k(v) - x_l(v) \|^2, \tag{4}$$

where the weights $w_{kl}(v)$ encode a property of uniform stiffness. These are computed and normalized proportionally to the sum of the blending weights from Equation 2 for each patch and its neighbors. Our final energy is defined as:

$$\arg\min_{\Theta} E(\Theta) = \arg\min_{\Theta} E_{data}(\mathbf{X}(\Theta)) + \lambda_r E_r(\Theta). \tag{5}$$

EM-ICP. To minimize the above energy we model the problem in a Bayesian context and employ Expectation-Maximization (EM) for MAP estimation [4]. This formulation is robust to outliers in the canonical model, reducing

the overall error accumulation as well as reducing drift. Additionally, it is robust to noise and missing data in incoming depth frames.

3.3. Surfel-based Fusion

Our surfel representation is based on [20], and the surfels of a point cloud are represented as a set of points $\bar{P}_k$, each associated with a position $v_k \in \mathbb{R}^3$, a normal $n_k \in \mathbb{R}^3$, a confidence $c_k \in \mathbb{R}$, and a timestamp $t_k \in \mathbb{R}$. The deformation framework produces a deformed point cloud with associated normals of an incoming depth frame with respect to its keyframe model. The deformed point cloud must be integrated into this model. As we use a surfel-based representation, we do not need to switch to a different data representation (e.g., DynamicFusion [23] requires constant conversions between a point cloud representation and an SDF). When averaging points, we use correspondences within a pre-defined radius δ_{near}. If the correspondence of the point from the deformed point cloud and the model are within this radius, we average them together with the same averaging scheme as in [20]. Any point from the deformed point cloud which does not have a correspondence and is not outside of a radius δ_{far}, is added to the model as new geometry. Points which are outside of the bounding radius δ_{far} are discarded, as they are most likely outliers or artifacts from sensor noise. To further eliminate noise, we remove points with a low confidence ($c_k < c_{stable}$) after a time interval t_{max}. We choose $c_{stable} = 10$ and compute the confidence in the same manner as in [20].

3.4. Keyframes

We propose a *Keyframes*-based scheme similar to [10], outlined in Figure 2, which enables us to handle lengthy sequences such as a human turning. This approach consists of dividing the entire sequence into keyframes, holding F frames each, and using correspondences-based fusion to obtain the final reconstruction. Following the initialization of the first keyframe, we process all the frames incrementally. We use the initial frame in the first keyframe to initialize the model. We perform surface-patching on each subsequent frame and deform it non-rigidly using the EM-ICP deformation framework onto the local keyframe model. New geometry is fused into the model using surfel-based fusion (§3.3). At the end of each keyframe, the last output from the deformation framework initializes the model for the next keyframe, which helps to combat drift and allows us to establish correspondences throughout the entire sequence. This is accomplished due to our surfel-based representation. The pipeline generates a model for each keyframe, all of which are fused at the end of the sequence to form a final global model.

Each depth frame is surface patched (§3.1) and back-projected into a 3D point cloud, stored using a surfel-based representation. Consistent with prior work [27], we estimate the rigid motion of the camera employing a fully volumetric approach [28], enabling our method to work on sequences with free camera motion. We obtain the deformed point cloud after warping the incoming frame through the EM-ICP framework with respect to the current keyframe model. We fuse this output (§3.3) into the current keyframe model. If a frame is the last one in a keyframe, the deformed point cloud from the deformation framework for that frame is fused into the keyframe model and is additionally used to initialize the next keyframe model. To obtain the final global model, all keyframes are fused together (§3.5) using a global correspondence-based approach. This fusion approach is particularly advantageous for our deformation framework and allows us to track correspondences throughout the entire sequence.

3.5. Global Correspondence-based Fusion

To obtain the final reconstruction all keyframe models must be fused together. Contrary to [10], we can accomplish this without switching data representations by fusing the keyframes together using keyframe-correspondences. We track correspondences between frames and propagate these between each keyframe; when the last deformation of a keyframe is obtained, it is used to initialize the next keyframe model and the correspondences are propagated as well. Other keyframe-based methods, such as [36, 9], cannot track correspondences and must instantiate a new volume for each new keyframe without any associations between them.

The global correspondence-based fusion is done as the last step of the proposed method. Starting at the last model generated, the corresponding surfels between the last keyframe model and its predecessor are fused, while non-corresponding surfels are added as new geometry of the global model. This process is repeated, fusing all preceding models onto the growing global model. Furthermore, the global correspondence-based fusion enables us to obtain accurate results for lengthy sequences while reducing drift and eliminating the need for complicated constraints such as bundle-adjustment. The correspondences found between each keyframe are averaged with the same surfel-based scheme as outlined in §3.3. For all new points without correspondences we add them to the global model as new geometry.

3.6. Visualization

The surfel-based representation consists of point primitives containing no connectivity [24]. As such, we employ a surface splatting approach similar to [39] for visualization. This is an advantage compared to other approaches in non-rigid monocular reconstruction, which switch data representations to an SDF, where visualization is extracted

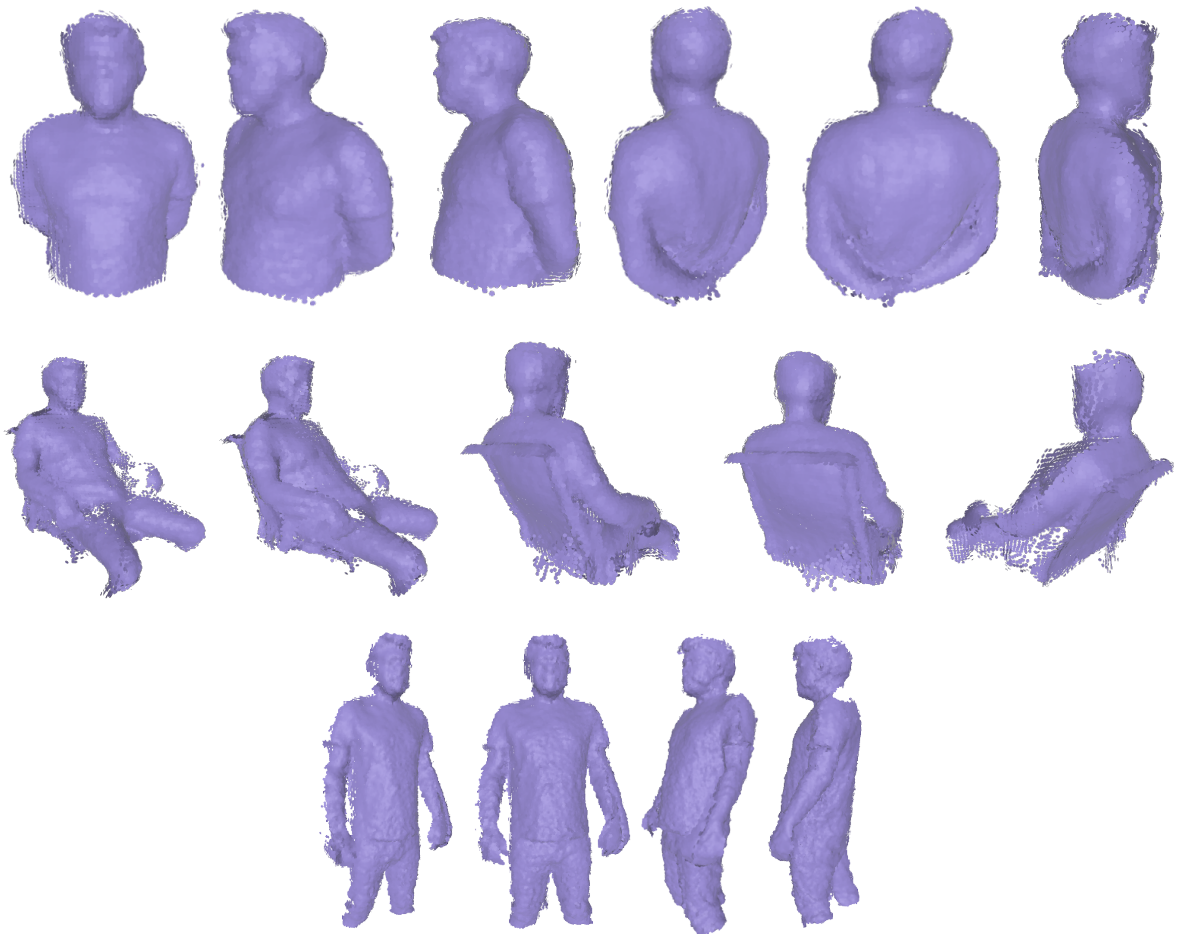

Figure 4. Patch-based non-rigid reconstruction results on several non-rigid sequences. Each sequence contains natural dynamic movements of a human such as swaying from side to side, leg movements, and head tilting. Our proposed method achieves full loop reconstructions without using any explicit loop closure detection methods.

through ray tracing or marching cubes [22]. In contrast, our approach allows us to stay in one representation and is less prone to data loss through data representation switching.

4. Evaluation

In this section we present extensive qualitative and quantitative evaluation of our proposed framework, using both publicly available datasets as well as our own acquisitions. As we employ a surfel-based representation, the rendering is completed using surface splatting, as described in § 3.6. Our entire pipeline is implemented on commodity hardware. Running on a 2.70GHz i7 CPU, the EM-ICP deformation framework takes 3 seconds per frame on average. Additionally, the final step (§3.5) takes on average 2 seconds for the entire sequence. This is significantly faster than the non-rigid bundle adjustment scheme of Dou *et al*. [11], which is the only other related single-stream technique executed on a CPU, taking 30 seconds per frame for pre-processing, together with another 6 hours for joint optimization. It may be possible to accelerate the EM-ICP framework using a GPU implementation, but due to the specific nature of our formulation, this is beyond the scope of this paper.

4.1. Qualitative Evaluation

Our Sequences: The experiments on the new sequences captured by us highlight our ability to recover reconstructions of full loops around deformable objects; something not typically demonstrated in comparable approaches. Our method is capable of capturing their geometry with high fidelity by only employing the patch-based rigidity constraints, without explicitly accounting for drift or non-rigid loop closure. Figure 4 depicts three of our own sequences, where a man deforms dynamically while turning. Our framework is able to successfully complete the reconstruction despite motions such as head tilts, back-and-forth swaying, and leg and arm movements.

Large Inter-frame Motions: Next we demonstrate the ability of our framework to recover rapid motions in chal-

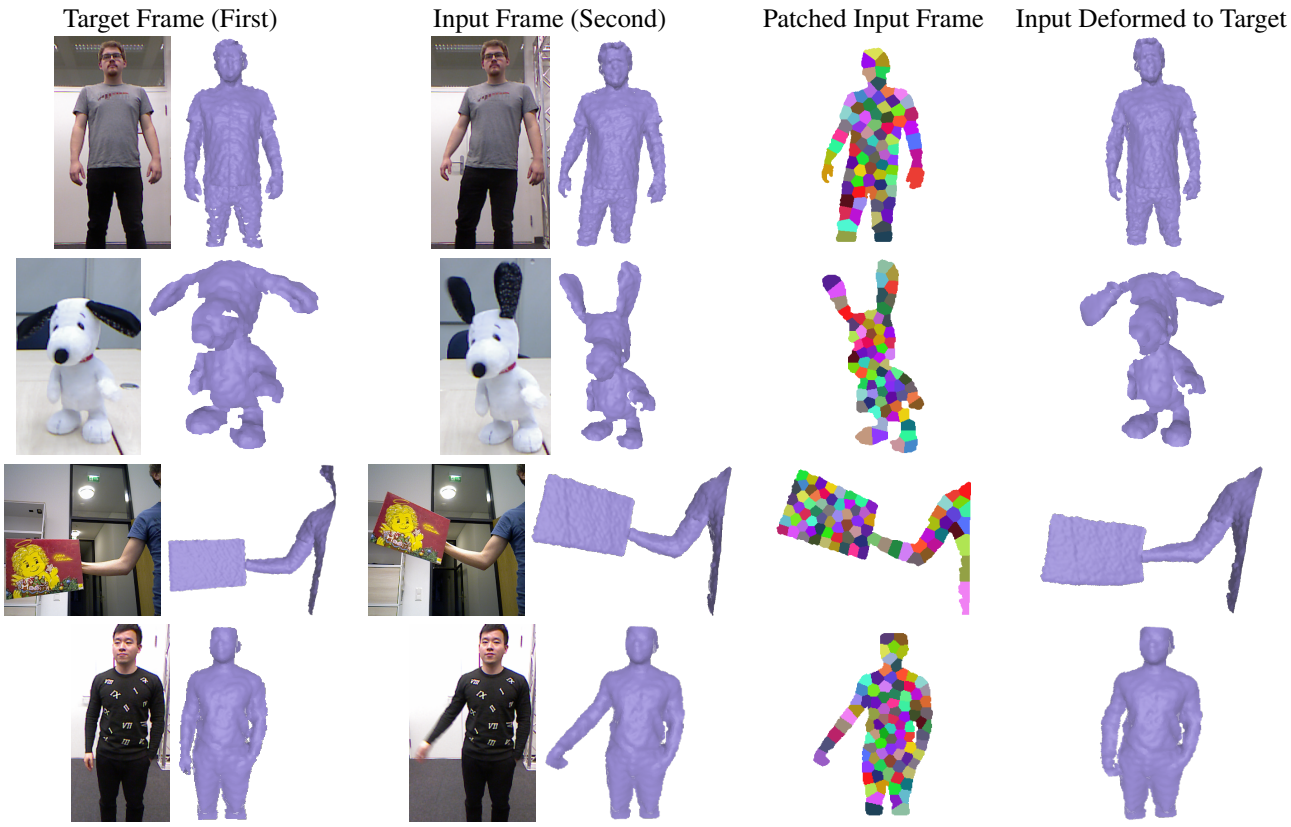

Figure 5. Ability of our proposed patch-based deformation framework to handle large motions. We manage to successfully warp the input frame towards the target frame despite the large pose difference between them.

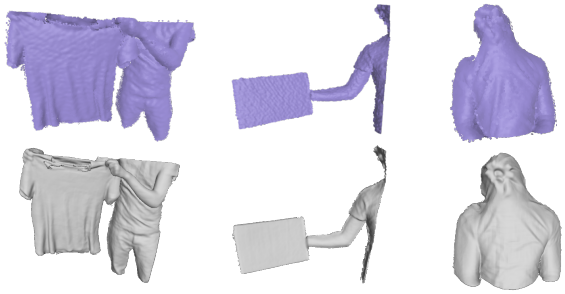

Figure 6. Comparison to the non-rigid reconstruction method of VolumeDeform [16]: our approach, illustrated in the top row, achieves similar quality, without switching between data representations.

lenging two frame examples, where the deforming object has undergone a large movement. In these two frame sequences, we use the first frame as a target frame and we deform the second frame to match the target frame. This is illustrated in Figure 5, where we show the target frame in the first column, and the input frame in the second. We also provide the patched input frame in the third column. Finally, we render the output of the deformed input frame onto the target frame, illustrated in column four. In all cases our approach manages to suitably warp the input towards the canonical model despite their large pose difference, demonstrating our method's ability to deform well despite the large movements.

Public Sequences: To further demonstrate the generality of our framework, we evaluate on data from the VolumeDeform paper [16]. Figure 6 depicts the final canonical reconstruction for several sequences. Our method achieves results of similar quality. Moreover, we do not need to switch between data representations, unlike VolumeDeform which stores the scene geometry in an SDF, while estimating correspondences via mesh rendering. As the SDF resolution may significantly reduce the spatial extents that can be reconstructed, while a marching cubes rendering to extract a mesh can be very computationally expensive, we identify the use of a single data representation as one of our main contributions. Furthermore, we test on full loop sequences from other authors, namely the *Andrew-Chair* sequence from Dou *et al*. [11]. Our result is shown in Figure 7, which once again demonstrates the ability of our framework to reconstruct subjects after long loopy sequences without explicit loop closure detection.

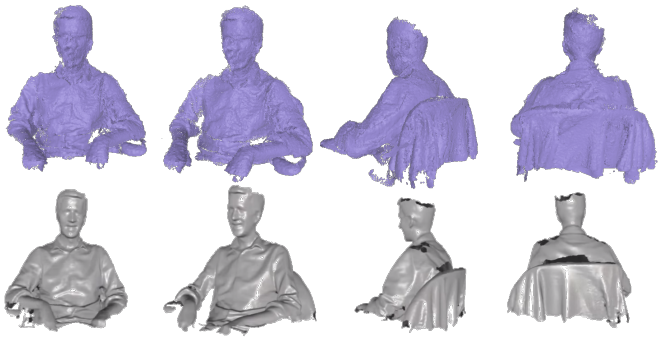

Figure 7. Comparison to the non-rigid bundle adjustment method of Dou *et al*. [11]. Our method, illustrated in the top row, is able to reconstruct the dynamic sequence without employing any loop closure methodologies, preserving features such as garment folds. We manage to accurately reconstruct small features (which may appear noisy due to the surfel-based visualization, e.g the nose.)

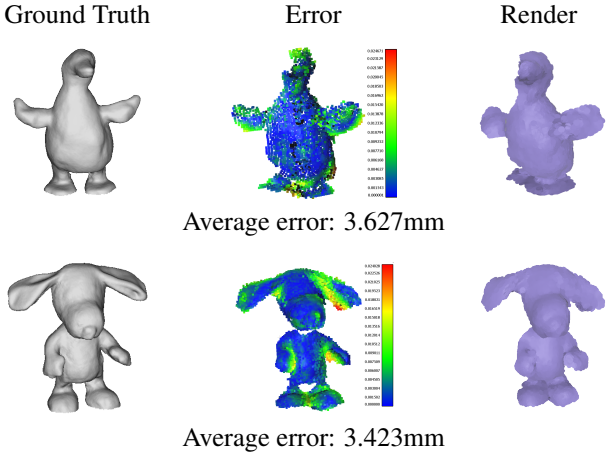

Figure 8. Non-rigid reconstruction of the *Duck* and *Snoopy* sequences from the KillingFusion dataset [27]. Our method reconstructs the sequences with a smaller error margin.

4.2. Quantitative Evaluation

Quantitative evaluation for dynamic 3D reconstruction is difficult, as ground truth data is typically not available. Nevertheless, to assess the geometric accuracy of our framework we test it on publicly available datasets for which quantitative evaluation is possible. First, we evaluate on the Deformable 3D Reconstruction Dataset from KillingFusion [27], as their approach also targets fast motions. We selected the *Duck* and *Snoopy* sequences, as they are the only two that have the ground truth models available. Our final reconstructions are displayed in Figure 8. We achieve smaller geometric error than the KillingFusion method for both sequences. Our average error for the *Duck* was 3.657mm compared to 3.896mm of KillingFusion, and the average error for *Snoopy* was 3.423mm compared to 3.543mm of KillingFusion.

Lastly, we ran our framework against the BodyFusion [36] Vicon markers. For the *YT* sequence we report an average global error of 2.3cm, using 20 frames in each keyframe and 70 patches. This compares favourably to the average error of 4.4cm of DynamicFusion [23] and 3.7cm of VolumeDeform [16]. BodyFusion [36] reports a lower error of 2.2cm, but they require skeleton-based priors to reduce the ambiguities of the deformation parameters, while our method does not involve the use of priors. Therefore, our patch-based method manages to track correspondences with high accuracy under fast motions. This is an advantage of our approach compared to other keyframe-based techniques [10, 9] which instantiate a new volume for every keyframe and lose data association between keyframes, and compared to purely SDF-based approaches like KillingFusion [27] which completely lacks correspondence information.

5. Conclusions and Future Work

We have presented a novel non-rigid 3D reconstruction scheme that robustly captures rapid inter-frame motions from a single depth stream relying on patch-based rigidity constraints in a probabilistic deformation framework. A variety of qualitative experiments has demonstrated these capabilities of our method, while quantitative comparisons have shown that we achieve lower reconstruction and tracking errors than state-of-the-art techniques [27]. Furthermore, our surfel-based fusion allows us to stay within one representation, while other approaches require constant conversion between meshes and SDFs, which both impedes their speed and limits the volume of space that they can reconstruct. As the current trend in related methods is the use of specialized hardware systems, we believe that the proposed lighter-weight solution, together with the use of a single representation, will offer a different avenue for further research that will make the capture of dynamic 3D scenes more accessible to the general public.

In future work, we plan to make use of the available RGB images as well, and incorporate additional constraints, such as SIFT features [16] or surface albedo terms [15], so that drift can be further reduced. Additionally, explicit loop closure detection can aid in capturing longer sequences, but it is a very challenging problem in a non subject-specific setting [26]. Finally, as the computational bottleneck of our approach is the EM-ICP computation, we will investigate how to adapt existing ways of parallelizing it on the GPU [29] for our particular variation.

References

[1] P. J. Besl and N. D. McKay. Method for registration of 3-D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.

[2] F. Bogo, M. J. Black, M. Loper, and J. Romero. Detailed full-body reconstructions of moving people from monocular RGB-D sequences. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[3] C. Cagniart, E. Boyer, and S. Ilic. Free-form mesh tracking: A patch-based approach. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[4] C. Cagniart, E. Boyer, and S. Ilic. Probabilistic deformable surface tracking from multiple videos. In *European Conference on Computer Vision (ECCV)*, pages 326–339. Springer, 2010.

[5] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (TOG)*, 34(4), 2015.

[6] A. Collet, M. Chuang, P. Sweeney, D. Gillett, D. Evseev, D. Calabrese, H. Hoppe, A. Kirk, and S. Sullivan. High-quality streamable free-viewpoint video. *ACM Transactions on Graphics (TOG)*, 34(4):69, 2015.

[7] T. Collins and A. Bartoli. Locally affine and planar deformable surface reconstruction from video. In *International Workshop on Vision, Modeling and Visualization*, pages 339–346, 2010.

[8] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H. Seidel, and S. Thrun. Performance capture from sparse multi-view video. *ACM Transactions on Graphics (TOG)*, 27(3), 2008.

[9] M. Dou, P. Davidson, S. R. Fanello, S. Khamis, A. Kowdle, C. Rhemann, V. Tankovich, and S. Izadi. Motion2Fusion: Real-time volumetric performance capture. In *ACM Transactions on Graphics (TOG)*, 2017.

[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 Transactions on Graphics (TOG)*, 35(4):114, 2016.

[11] M. Dou, J. Taylor, H. Fuchs, A. Fitzgibbon, and S. Izadi. 3D scanning deformable objects with a single RGBD sensor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 493–501, 2015.

[12] J. Fayad, L. Agapito, and A. Del Bue. Piecewise quadratic reconstruction of non-rigid surfaces from monocular sequences. In *European Conference on Computer Vision (ECCV)*, pages 297–310. Springer, 2010.

[13] K. Fujiwara, K. Nishino, J. Takamatsu, B. Zheng, and K. Ikeuchi. Locally rigid globally non-rigid surface registration. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1527–1534. IEEE, 2011.

[14] J. Gall, C. Stoll, E. de Aguiar, C. Theobalt, B. Rosenhahn, and H. P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[15] K. Guo, F. Xu, T. Yu, X. Liu, Q. Dai, and Y. Liu. Real-time geometry, albedo and motion reconstruction using a single RGB-D camera. *ACM Transactions on Graphics (TOG)*, 2017.

[16] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger. VolumeDeform: Real-time volumetric non-rigid reconstruction. In *European Conference on Computer Vision (ECCV)*, pages 362–379. Springer, 2016.

[17] B. Irving, I. A. Popescu, R. Bates, P. D. Allen, A. L. Gomes, P. Kannan, P. Kinchesh, S. Gilchrist, V. Kersemans, S. Smart, et al. maskSLIC: regional superpixel generation with application to local pathology characterisation in medical images. *arXiv preprint arXiv:1606.09518*, 2016.

[18] P. Ji, H. Li, Y. Dai, and I. D. Reid. " maximizing rigidity" revisited: A convex programming approach for generic 3d shape reconstruction from multiple perspective views. In *ICCV*, pages 929–937, 2017.

[19] H. Joo, H. Liu, L. Tan, L. Gui, B. Nabbe, I. Matthews, T. Kanade, S. Nobuhara, and Y. Sheikh. Panoptic studio: A massively multiview system for social motion capture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3334–3342, 2015.

[20] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *3D Vision-3DV 2013, 2013 International Conference on*, pages 1–8. IEEE, 2013.

[21] Q. Liu-Yin, R. Yu, L. Agapito, A. Fitzgibbon, and C. Russell. Better together: Joint reasoning for non-rigid 3d reconstruction with specularities and shading. *arXiv preprint arXiv:1708.01654*, 2017.

[22] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, pages 163–169, 1987.

[23] R. A. Newcombe, D. Fox, and S. M. Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 343–352, 2015.

[24] H. Pfister, M. Zwicker, J. Van Baar, and M. Gross. Surfels: Surface elements as rendering primitives. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 335–342. ACM Press/Addison-Wesley Publishing Co., 2000.

[25] C. Russell, J. Fayad, and L. Agapito. Energy based multiple model fitting for non-rigid structure from motion. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3009–3016. IEEE, 2011.

[26] T. Schmidt, R. Newcombe, and D. Fox. Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, 2(2):420–427, 2017.

[27] M. Slavcheva, M. Baust, D. Cremers, and S. Ilic. Killingfusion: Non-rigid 3D reconstruction without correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 3, page 7, 2017.

[28] M. Slavcheva, W. Kehl, N. Navab, and S. Ilic. Sdf-2-Sdf: Highly accurate 3D object reconstruction. In *European Conference on Computer Vision (ECCV)*, pages 680–696. Springer, 2016.

[29] T. Tamaki, M. Abe, B. Raytchev, and K. Kaneda. Softassign and EM-ICP on GPU. In *First International Conference on Networking and Computing*, 2010.

[30] D. J. Tan, T. Cashman, J. Taylor, A. Fitzgibbon, D. Tarlow, S. Khamis, S. Izadi, and J. Shotton. Fits like a glove: Rapid

and reliable hand shape personalization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[31] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2Face: Real-time face capture and reenactment of RGB videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[32] A. Varol, M. Salzmann, E. Tola, and P. Fua. Template-free monocular reconstruction of deformable surfaces. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1811–1818. IEEE, 2009.

[33] D. Vlasic, I. Baran, W. Matusik, and J. Popović. Articulated mesh animation from multi-view silhouettes. *ACM Transactions on Graphics (TOG)*, 27(3), 2008.

[34] D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz, and W. Matusik. Dynamic shape capture using multi-view photometric stereo. *ACM Transactions on Graphics (TOG)*, 28(5), 2009.

[35] R. Yu, C. Russell, N. D. Campbell, and L. Agapito. Direct, dense, and deformable: Template-based non-rigid 3d reconstruction from rgb video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 918–926, 2015.

[36] T. Yu, K. Guo, F. Xu, Y. Dong, Z. Su, J. Zhao, J. Li, Q. Dai, and Y. Liu. BodyFusion: Real-time capture of human motion and surface geometry using a single depth camera. In *The IEEE International Conference on Computer Vision (ICCV)*. ACM, October 2017.

[37] M. Zollhöfer, M. Nießner, S. Izadi, C. Rhemann, 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 Transactions on Graphics (TOG)*, 33(4), 2014.

[38] M. Zollhöfer, A. G. Patrick Stotko, C. Theobalt, M. Nießner, R. Klein, and A. Kolb. State of the art on 3D reconstruction with RGB-D cameras. 2018.

[39] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross. Surface splatting. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 371–378. ACM, 2001.