KinectDeform: Enhanced 3D Reconstruction of Non-Rigidly Deforming Objects

Hassan Afzal*, Kassem Al Ismaeil*, Djamila Aouada*, François Destelle†, Bruno Mirbach‡ and Björn Ottersten*
*Interdisciplinary Centre for Security, Reliability and Trust
University of Luxembourg, 4, rue Alphonse Weicker, L-2721, Luxembourg
Email: {hassan.afzal, kassem.alismaeil, djamila.aouada, bjorn.ottersten}@uni.lu
† Dublin City University, Insight: Centre for Data Analytics, Ireland
Email: francois.destelle@dcu.ie
‡IEE S.A., Advanced Engineering, Contern, Luxembourg
Email: bruno.mirbach@iee.lu

***Abstract*—In this work we propose *KinectDeform*, an algorithm which targets enhanced 3D reconstruction of scenes containing non-rigidly deforming objects. It provides an innovation to the existing class of algorithms which either target scenes with rigid objects only or allow for very limited non-rigid deformations or use precomputed templates to track them. *KinectDeform* combines a fast non-rigid scene tracking algorithm based on octree data representation and hierarchical voxel associations with a recursive data filtering mechanism. We analyze its performance on both real and simulated data and show improved results in terms of smoothness and feature preserving 3D reconstructions with reduced noise.**

***Keywords*—Non-rigid registration, Enhanced reconstruction, Recursive filtering.**

I. Introduction

Reconstructing real objects accurately and efficiently is one of the major goals in the field of 3D computer vision. It opens doors to various applications from object detection to environment mapping, from gesture control to security and surveillance etc. Commodity depth cameras such as recently available structured light and time-of-flight cameras, though affordable and easily accessible, acquire noisy measurements with limited resolution, and hence provide 3D representations which are only suitable for a limited number of applications. Many recent approaches try to solve the problem of attaining improved 3D reconstruction of scenes or objects from low quality raw data [1], [2]. One approach which stands out due to its performance, efficiency, and high quality results is the *KinectFusion* algorithm by Newcombe et al. [3], [4]. It either uses a slowly moving RGB-D camera or considers objects moving slowly in front of a static camera to obtain their high quality 3D reconstruction. Fig. 1 (a) shows the high-level pipeline of *KinectFusion* where a rigid-alignment of 3D data captured during sequential time-steps is followed by filtering or fusion of data accumulated over time. The key feature of *KinectFusion* is its run-time performance by using commodity graphics hardware, such that it is able to fuse and reconstruct data acquired at a rate which is as high as 30 frames per second in real-time.

KinectFusion became a cornerstone for various works which either built on it or used similar ideas, e.g., to map larger environments in one go by using a moving volume approach [5], [6], or by using octrees for memory efficient surface reconstruction [7], [8], or by using voxel hashing for even better accuracy and efficiency [9]. Kainz et al. modified the *KinectFusion* pipeline in order to incorporate multiple cameras for holistic 3D reconstruction of static objects [10]. Cerqueira et al. customized *KinectFusion* for real-time tracking and modeling of a human face [11]; whereas Sturm et al. used its components for full-body 3D reconstruction of humans [12]. Moreover, improvements were also proposed in the real-time tracking module by computing poses by directly fusing depth maps with the *truncated signed distance function* (TSDF) volume [13], or by also using visual features together with 3D information [5]–[7]. Similarly, textured 3D models were achieved by mapping visual texture information on the reconstructed 3D models [5], [6].

A downside of the techniques mentioned above is that they target environments with rigid objects. This makes tracking such objects relatively simple by merely calculating a single rigid transformation for the whole object or scene. Moving

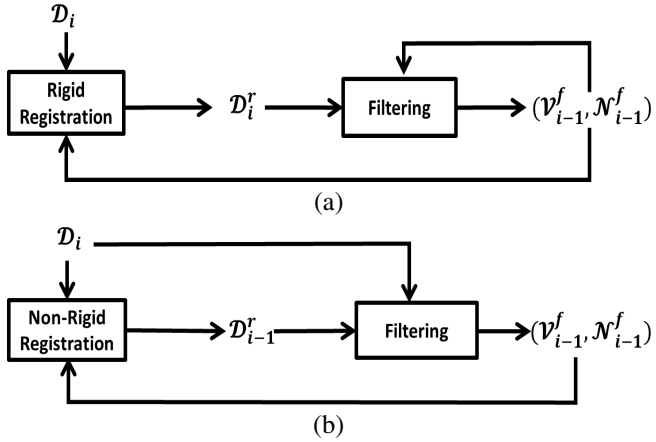


Fig. 1: High-level pipeline of: (a) *KinectFusion*, and (b) the proposed *KinectDeform*. $\mathcal{D}_i$: input depth map at time-step i, $(\mathcal{V}_{i-1}^f, \mathcal{N}_{i-1}^f)$: filtered vertex map and corresponding normal map at time-step $i-1$, $\mathcal{D}_i^r$ and $\mathcal{D}_{i-1}^r$: resulting depth maps of rigid and non-rigid registration steps correspondingly. For more details please see Section II and Section III.

objects in otherwise static scenes are considered as unstable regions, they are segmented and removed when detected [8], [14]. In the work of face modeling, the facial expressions are required to be as consistent as possible throughout the scanning period [11]. Similarly, for full-body 3D reconstruction, the person to be scanned is required to be static with small non-rigidities handled by using a rough template from the first frame [12]. For the same body scanning applications, Cui et al. on the other hand, proposed to tackle non-rigidities by using a global non-rigid alignment based on joint constraints. Their technique however cannot handle large motions, and is also not very practical for real-time applications [15]. Recently, Zöllhoefer et al. [16] have proposed what they claim to be the first 'general purpose' non-rigid 3D reconstruction system which works in real-time and produces refined 3D reconstructions. It works by first acquiring a rigid template of the object to be reconstructed. This template is then used to track non-rigidities with high flexibility.

In this paper, we propose a framework which is derived from *KinectFusion* with the ability to track and reconstruct, with high accuracy, without any template or constraint on motion, rigid as well as non-rigid moving objects. Fig. 1 (b) shows the high-level pipeline of the proposed technique. Our key contributions consist of using tracking based on non-rigid registration of the result of the previous time-step to the newly acquired deformed data, followed by a recursive filtering mechanism based on the registered result and the newly acquired data. We make use of a generic tracking algorithm for non-rigid alignment which is efficient and can be easily parallelized [17]. We use both real and simulated data to validate the performance of proposed technique.

The remainder of the paper is organized as follows: Section II introduces the problem at hand and gives an overview of how *KinectFusion* tries to solve it with restrictions on object's rigidity. Section III details our proposed approach. In Section IV, we present results of experiments for quantitative and qualitative analysis of the performance of the proposed method using both simulated and real data. This is followed by a conclusion in Section V.

II. Background & Problem Formulation

Given a fixed single depth camera system with an associated camera calibration matrix $\mathbf{K}$, at each discrete time-step $i \in \mathbb{N}$, this camera acquires a depth map $\mathcal{D}_i$ which contains depth data ordered on a grid of size $(U \times V)$ with $U, V \in \mathbb{N}$. This data represents a deformable moving surface in the depth camera's field of view. It can be converted into its corresponding vertex map $\mathcal{V}_i$, where each depth value in $\mathcal{D}_i$ is associated with a vertex in $\mathcal{V}_i$ such that:

$$\begin{aligned} \mathcal{V}_i : \quad \mathbb{R}^2 \quad &\rightarrow \mathbb{R}^3 \\ \mathbf{p} \quad &\mapsto \mathcal{V}_i(\mathbf{p}) = \mathcal{D}_i(\mathbf{p})\mathbf{K}^{-1}\dot{\mathbf{p}}, \end{aligned} \tag{1}$$

where $\mathbf{p}$ represents a location on the 2D grid of $\mathcal{D}_i$ and $\mathcal{V}_i$, and $\dot{\mathbf{p}}$ represents its corresponding homogenous coorindates.
Let us consider a sequence of N acquired depth maps $\{\mathcal{D}_0, \mathcal{D}_1, \cdots, \mathcal{D}_{N-1}\}$ of the same scene deforming over time. Their corresponding vertex maps are $\{\mathcal{V}_0, \mathcal{V}_1, \ldots, \mathcal{V}_{N-1}\}$. Each vertex map $\mathcal{V}_i$ is related to the previous vertex map $\mathcal{V}_{i-1}$ via:

$$\mathcal{V}_i = h_i\left(\mathcal{V}_{i-1}\right) + \mathcal{E}_i, \tag{2}$$

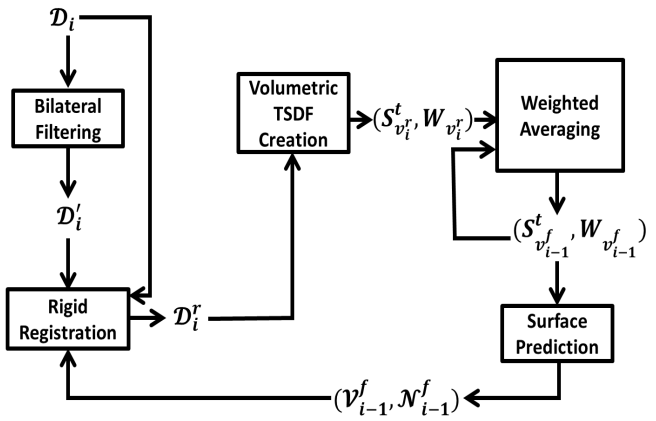


Fig. 2: Detailed pipeline of *KinectFusion*. $\mathcal{D}_i$: input depth map at time-step i, $\mathcal{D}_i^{'}$: result of bilateral filter on $\mathcal{D}_i$, $(\mathcal{V}^f_{i-1}, \mathcal{N}^f_{i-1})$: filtered vertex map and corresponding normal map at time-step $i-1$, $\mathcal{D}_i^r$: result of rigid registration of $\mathcal{D}_i^{'}$ to $\mathcal{V}^f_{i-1}$, $(S^t_{\mathcal{V}_i^r}, W_{\mathcal{V}_i^r})$ and $(S^t_{\mathcal{V}^f_{i-1}}, W_{\mathcal{V}^f_{i-1}})$: TSDF volumes corresponding to vertex maps $\mathcal{V}_i^r$ and $\mathcal{V}^f_{i-1}$ respectively. For more details please see Section II and Section III.

where $h_i(\cdot)$ is the deformation that transforms $\mathcal{V}_{i-1}$ to its consecutive vertex map $\mathcal{V}_i$. The additional term $\mathcal{E}_i$ represents the error map due to the acquisition system including camera noise, and sampling errors.
The problem at hand is therefore to attenuate $\mathcal{E}_i$ for $i > 0$, and recover an enhanced sequence $\{\mathcal{V}^f_0, \mathcal{V}^f_1, \ldots, \mathcal{V}^f_{N-1}\}$ starting from the acquisition $\{\mathcal{V}_0, \mathcal{V}_1, \ldots, \mathcal{V}_{N-1}\}$.
As a solution, a recursive filtering function $f(\cdot, \cdot)$ may be defined by sequentially fusing the current measurement $\mathcal{D}_i$ and the resulting enhanced vertex map $\mathcal{V}^f_{i-1}$ of the previous time-step such that:

$$\mathcal{V}^f_i = \begin{cases} \mathcal{V}_i & \text{for } i = 0, \\ f(\mathcal{V}^f_{i-1}, \mathcal{D}_i) & i > 0. \end{cases} \tag{3}$$

The *KinectFusion* algorithm proposes a practical solution for (3) for the special case where the deformation h_i is rigid, i.e., when the transformation between $\mathcal{V}_{i-1}$ and $\mathcal{V}_i$ is a single rotation and translation with 6 degrees of freedom [4]. Fig. 2 shows the detailed pipeline of the *KinectFusion* algorithm. In the first step, a bilateral filter is applied to the input depth map $\mathcal{D}_i$ resulting in a filtered map $\mathcal{D}_i^{'}$ [4], [18]. The new depth map $\mathcal{D}_i^{'}$ is then given as input to the registration module where its corresponding vertex map $\mathcal{V}_i^{'}$ is computed using (1). The normal map $\mathcal{N}_i^{'}$ is also computed for each vertex in $\mathcal{V}_i^{'}$ using vertices belonging to neighboring points. The registration step uses a multi-resolution point-plane error metric coupled with a projective data association–based variation of *Iterative Closest Point* algorithm (ICP) to estimate the camera (or conversely object) pose [4], [19]. This second step estimates the rigid deformation between $\mathcal{V}_i^{'}$ and $\mathcal{V}^f_{i-1}$ using their corresponding normal maps $\mathcal{N}_i^{'}$ and $\mathcal{N}^f_{i-1}$, respectively. This transformation is applied to $\mathcal{V}_i$ (computed from $\mathcal{D}_i$) to get $\mathcal{V}_i^r$, which is back projected using the inverse mapping of (1) in order to obtain $\mathcal{D}_i^r$. It is then fused with a global surface representation to get

an enhanced 3D surface reconstruction. We note that the reason for using $\mathcal{D}_i$ instead of $\mathcal{D}_i^{'}$ for fusion is to preserve the details which might have been lost due to bilateral filtering. For the last step of data fusion or filtering, *KinectFusion* uses a method based on SDF representation of a surface in 3D [4], [20]. An SDF $S_{\mathcal{V}_i}(.)$ corresponding to a vertex map $\mathcal{V}_i$ represents points on surface as zeros, and free spaces in front of and behind the surface as positive and negative values, respectively. These values increase as distance from the surface increases. The SDF is formally defined as:

$$S_{\mathcal{V}_i}: \quad \mathbb{R}^3 \rightarrow \mathbb{R}$$
$$\mathbf{P} \mapsto \begin{cases} d(\mathbf{P}, \mathcal{V}_i) & \mathbf{P} \text{ lies in front of } \mathcal{V}_i, \\ 0 & \mathbf{P} \in \mathcal{V}_i, \\ -d(\mathbf{P}, \mathcal{V}_i) & \mathbf{P} \text{ lies behind } \mathcal{V}_i, \end{cases}$$

where $d(.,.)$ calculates the shortest distance between a given 3D point $\mathbf{P}$ and $\mathcal{V}_i$. *KinectFusion* uses a volumetric representation of the truncated SDF (TSDF). It is called TSDF because the SDF is truncated using a limiting value of $\pm\mu$. A continuous TSDF is sampled by a volume of resolution $(Z \times Z \times Z)$ with $Z \in \mathbb{N}$, lying in the camera's reference frame. The volume consists of volumetric elements called voxels where each voxel is represented by its centroid $\mathbf{P}$. A TSDF volume corresponding to $\mathcal{V}_i^r$ is defined by two values computed for each of its voxels $\mathbf{P}$; one is the TSDF value itself $S^t_{\mathcal{V}_i^r}(\mathbf{P})$, and second is the weight $W_{\mathcal{V}_i^r}(\mathbf{P})$, using camera parameters $\mathbf{K}$, and the de-homogenization function $\pi(.)$ such that:

$$S^t_{\mathcal{V}_i^r}(\mathbf{P}) = \Psi(\|\mathbf{P}\|_2 - \|\mathcal{V}_i^r(\mathbf{q})\|_2), \tag{4}$$

where $\mathbf{q} = \lfloor \pi(\mathbf{KP}) \rfloor$, and

$$\Psi(\eta) = \begin{cases} min\{1, \frac{\eta}{\mu}\} \cdot sgn(\eta) & \text{iff } \eta \geq -\mu, \\ 0 & \text{otherwise}, \end{cases} \tag{5}$$

where μ is the truncation distance. Note that $\mathbf{q}$ represents a location on the 2D grid of $\mathcal{V}_i^r$. The weight $W_{\mathcal{V}_i^r}(\mathbf{P})$ should be proportional to the measure of similarity of pixel ray direction from $\mathbf{q}$ to $\mathbf{P}$ to local surface normal at $\mathcal{V}_i^r(\mathbf{q})$ but Newcombe et al. show that keeping the weight $W_{\mathcal{V}_i^r}(\mathbf{P}) = 1$ works well for their filtering scheme of *KinectFusion* which will be discussed next [4]. For filtering, *KinectFusion* follows a scheme of weighted average of all TSDF volumes computed for $\mathcal{V}_i^r$ resulting in one global filtered TSDF volume where each voxel in the filtered volume is represented by $S^t_{\mathcal{V}_i^f}(\mathbf{P})$ and $W_{\mathcal{V}_i^f}(\mathbf{P})$ such that:

$$S^t_{\mathcal{V}_i^f}(\mathbf{P}) = \frac{W_{\mathcal{V}_{i-1}^f}(\mathbf{P}) S^t_{\mathcal{V}_{i-1}^f}(\mathbf{P}) + W_{\mathcal{V}_i^r}(\mathbf{P}) S^t_{\mathcal{V}_i^r}(\mathbf{P})}{W_{\mathcal{V}_i^f}(\mathbf{P})}, \tag{6}$$

where

$$W_{\mathcal{V}_i^f}(\mathbf{P}) = W_{\mathcal{V}_{i-1}^f}(\mathbf{P}) + W_{\mathcal{V}_i^r}(\mathbf{P}). \tag{7}$$

It is to be noted that $W_{\mathcal{V}_i^f}(\mathbf{P})$ is reset to a default value after a fixed number of iterations. The vertex map $\mathcal{V}_i^f$ is computed from the current filtered volume for the next iteration using surface prediction via ray casting [4], [21]. The normal map $\mathcal{N}_i^f$ is also computed using the gradient of the TSDF values in the filtered volume. The final extraction of the surface or the point cloud in 3D from the filtered volume can be carried out by using zero crossings or iso-surfaces in the TSDF volume followed by linear interpolation of points.

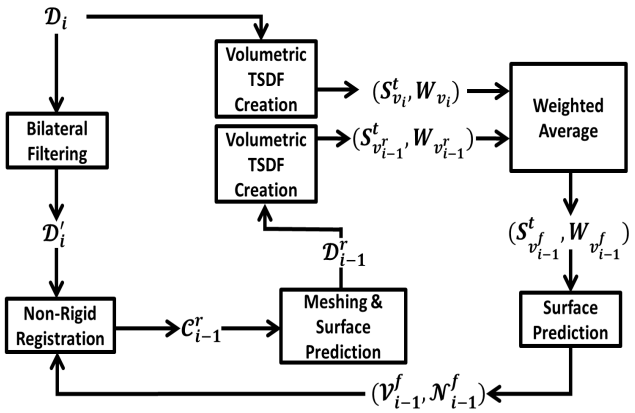


Fig. 3: Detailed pipeline of the proposed *KinectDeform*. $\mathcal{D}_i$: input depth map at time-step i, $\mathcal{D}_i^{'}$: result of bilateral filter on $\mathcal{D}_i$, $(\mathcal{V}^f_{i-1}, \mathcal{N}^f_{i-1})$: filtered vertex map and corresponding normal map at time-step $i-1$, $\mathcal{C}^r_{i-1}$: unorganized point cloud which is the result of non-rigid registration of $\mathcal{V}^f_{i-1}$ to $\mathcal{D}_i^{'}$, $\mathcal{D}^r_{i-1}$: depth map corresponding to $\mathcal{C}^r_{i-1}$, $(S^t_{\mathcal{V}_i}, W_{\mathcal{V}_i})$, $(S^t_{\mathcal{V}^r_{i-1}}, W_{\mathcal{V}^r_{i-1}})$ and $(S^t_{\mathcal{V}^f_{i-1}}, W_{\mathcal{V}^f_{i-1}})$ are TSDF volumes corresponding to vertex maps $\mathcal{V}_i$, $\mathcal{V}^r_{i-1}$ and $\mathcal{V}^f_{i-1}$ respectively. For more details please see Section II and Section III.

III. KinectDeform

We propose to modify the *KinectFusion* to achieve 3D tracking, and hence enhanced 3D reconstruction of not only rigid but also non-rigidly deforming objects as well. One of the main reasons for taking *KinectFusion* as a reference is its ease of parallelization for real-time implementation. We would like to maintain this feature in the proposed approach that we refer to as *KinectDeform* and explore it further in the future work. As depicted in the high-level descriptions of Fig. 1, *KinectDeform* modifies *KinectFusion* at two main levels; first, the registration which, from rigid, becomes non-rigid, and second, the reference frame in the filtering process changes where the newly acquired measurement is the one to act as a reference for the current state of the object and to which the resulting vertex map from the filtered TSDF from the previous iteration should be aligned and fused with. More details are provided in Fig. 3, and described in what follows.

A. Non-rigid registration

Similarly to *KinectFusion*, for an improved registration, a bilateral filter is applied to the input depth map $\mathcal{D}_i$ as a first preprocessing step. We obtain a bilateral filtered depth map $\mathcal{D}_i'$, and its corresponding vertex map $\mathcal{V}_i'$. The next step is to register the resulting vertex map of the previous iteration i.e. $\mathcal{V}^f_{i-1}$ with this new vertex map $\mathcal{V}_i'$. Conversely to other classical reconstruction methods, our pipeline captures non-rigid objects. As a consequence, this registration step aims to align two vertex maps describing the non-rigid deformation h_i in (2). This deformation is unknown but can be estimated locally by a patch-oriented method, describing the global non-rigid deformation by a set of local rigid ones. As such, we propose to apply a modified scene-flow based tracking method from [17].

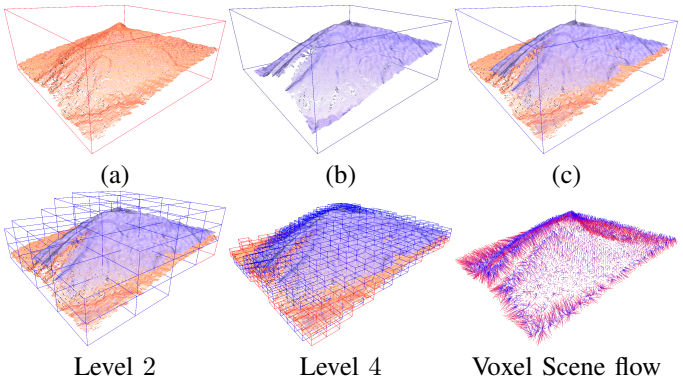


Fig. 4: Outline of the non-rigid registration algorithm used by our pipeline, from the first cloud (a) to the second one (b). As a first step, both clouds are mapped rigidly by centering their respective centroid (c). A common discrete space is then built using two separate octrees for which the root cell is the bounding box of the cloud couple. These octrees are then subdivided regularly until a fixed level S is reached. Finally, the algorithm described in [17] is used to create a voxel-to-voxel 3D scene flow, describing a global non-rigid deformation as a set of rigid ones.

As opposed to other well-known techniques [22]–[27], this algorithm offers real-time capabilities, and can handle non-rigidly deforming objects in a generic way without considering a specific motion or shape model. The proposed scene-flow tracking technique relies on several steps: the pair of vertex maps $\mathcal{V}^f_{i-1}$ and $\mathcal{V}^{'}_{i}$ are first centered by joining their respective centroids. A double voxelization step then embeds each cloud considering as a first cell the bounding box of the two point clouds, i.e., sharing the same root cell. These octrees are aimed to be subdivided in a regular way considering each cut point as the cell center. Thus the subdivision of both clouds describes the same discrete coordinate space, see Fig. 4. Then, a voxel-to-voxel scene flow is created using a local neighborhood relation among the voxels of the two octrees, several different hierarchical relations, and finally a local and computationally efficient algorithm to establish the relation from voxels of the first octree to the second one. *KinectDeform* uses the obtained voxel-to-voxel flow in order to register locally each point-based patch from $\mathcal{V}^f_{i-1}$, embedded in the first octree, to $\mathcal{V}'_i$, embedded in the second one. The result of the registration is $\mathcal{C}^r_{i-1}$, which is an unorganized 3D point cloud containing $(U \times V)$ 3D points.

B. TSDF volume creation and fusion

To create a TSDF volume using the approach explained in Section II from the information in $\mathcal{C}^r_{i-1}$, an organized point cloud or depth map needs to be extracted from it. An idea would be to simply back project points in $\mathcal{C}^r_{i-1}$ to the image plane using the camera matrix $\mathbf{K}$. This would result in several points in $\mathcal{C}^r_{i-1}$ being projected to the same pixel location in the image plane to which only one depth value is to be assigned. Hence, a lot of valuable information would be lost. To get a more accurate representation of $\mathcal{C}^r_{i-1}$ with respect to the camera, we perform surface reconstruction based on *Delaunay triangulation* [28]. The resulting mesh, is used for generating the depth map $\mathcal{D}^r_{i-1}$ by simulating a noise-free camera with the same pose and camera matrix $\mathbf{K}$ as the real camera used for acquiring the initial raw data and by performing ray-tracing [29]. Next step is to use the resultant depth map $\mathcal{D}^r_{i-1}$ and input depth map $\mathcal{D}_i$ to fuse them to get a filtered and enhanced reconstruction of the object at time i. Here again we use $\mathcal{D}_i$ for fusion and filtering instead of $\mathcal{D}^{'}_{i}$ to avoid loss of important information due to bilateral filtering. For data fusion and filtering we also use the volumetric TSDF for surface representation as done by *KinectFusion* [4], [20]. The reason for choosing this representation scheme over other similar non-parametric representations is ease of surface extraction and parallelization of volumetric TSDF computation and fusion [4]. As mentioned in the begining of Section III, for handling non-rigid deformations we cannot keep a globally consistent surface representation as reference and keep fusing newly acquired information to it. Instead we create TSDF volumes for both $\mathcal{D}^r_{i-1}$ and $\mathcal{D}_i$ using their corresponding $\mathcal{V}^r_{i-1}$ and $\mathcal{V}_i$ using (4) and (5) to get $S^t_{\mathcal{V}^r_{i-1}}$ and $S^t_{\mathcal{V}_i}$, respectively. We propose to modify the weighting scheme of *KinecFusion* in order to take the following factors into account. On one hand $\mathcal{V}^r_{i-1}$, which is the deformed version of $\mathcal{V}^f_{i-1}$, brings valuable information due to temporal filtering and also improved registration due to it being aligned to the filtered version of $\mathcal{V}_i$. On the other hand we also have to take into account errors during registration and also loss of some details in $\mathcal{V}^{'}_{i}$ caused by bilateral filtering which in turn might cause loss of some details in $\mathcal{V}^r_{i-1}$. Similarly we should also consider the sensor or acquisition noise introduced in each acquisition $\mathcal{V}_i$. Therefore, to reflect these factors the weights $W_{\mathcal{V}_i}$ and $W_{\mathcal{V}^r_{i-1}}$ are initialized and updated as follows:

$$W_{\mathcal{V}_i}(\mathbf{P}) = \mathcal{N}_{\sigma_c}(\varepsilon_{n_i}), \tag{8}$$

and

$$W_{\mathcal{V}^r_{i-1}}(\mathbf{P}) = \begin{cases} \mathcal{N}_{\sigma_c}(\varepsilon_{t_{i-1}}) & \text{iff } i = 1, \\ \mathcal{N}_{\sigma_p}(\varepsilon_{t_{i-1}}) & \text{otherwise}, \end{cases} \tag{9}$$

where $\mathcal{N}_\sigma(x) = exp(-x^2\sigma^{-2})$, and ε_{n_i} is a global estimate of sensor noise in the current acquisition $\mathcal{D}_i$ and $\varepsilon_{t_{i-1}}$ is the root-mean-square error (RMSE) based on point-wise Euclidean distances between $\mathcal{V}_i$ and $\mathcal{V}^r_{i-1}$:

$$\varepsilon_{t_{i-1}} = \sqrt{\frac{1}{M}\left(\sum_{\mathbf{p}=1}^{M} \|\mathcal{V}_i(\mathbf{p}) - \mathcal{V}^r_{i-1}(\mathbf{p})\|^2\right)}, \tag{10}$$

where $M = (U \times V)$, and $\varepsilon_{t_{i-1}}$ is an estimate of the registration error and details lost during bilateral filtering, meshing and back projection in $\mathcal{V}^r_{i-1}$ with respect to $\mathcal{V}_i$ assuming that bilateral filtering removes the sensor noise from $\mathcal{V}^{'}_{i}$ and hence from $\mathcal{V}^r_{i-1}$. The parameters σ_c and σ_p are chosen empirically for now, taking into account the factors mentioned above by giving a higher weight to the temporally filtered deformed data compared to the raw input with increasing time. The two newly created volumes are fused by following (6) to get the filtered TSDF volume $S^t_{\mathcal{V}^f_i}$ which is used to extract the vertex map $\mathcal{V}^f_i$ and the normal map $\mathcal{N}^f_i$ for the next iteration using the same method as *KinectFusion*.

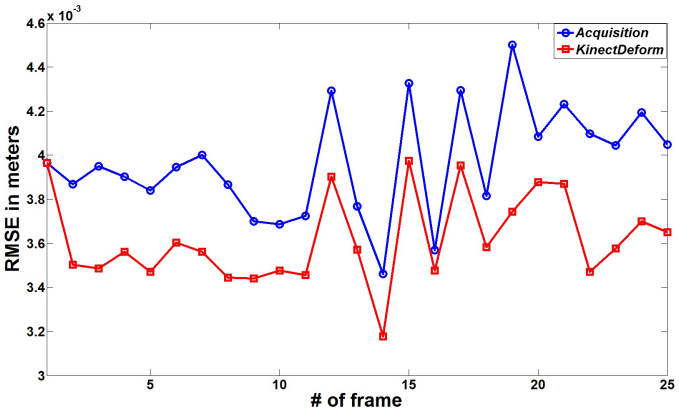


Fig. 5: RMSE of raw and filtered data with ground truth for simulated "cloth" dataset

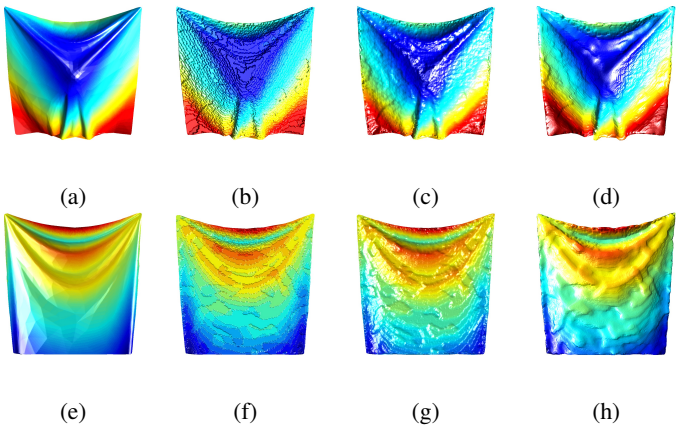


Fig. 6: "Cloth" dataset. **Top row:** Frame 5 (a) Ground truth, (b) raw data, (c) result of *KinectDeform*, (d) result of *KinectDeform* after deblurring. **Bottom row:** Frame 20 (e) Ground truth, (f) raw data, (g) result of *KinectDeform*, (h) result of *KinectDeform* after deblurring.

IV. Experiments & Results

To analyze the performance of *KinectDeform* both quantitatively and qualitatively, we test it on both simulated and real non-rigidly deforming depth sequences. For quantitative analysis, we use two different data sources, the first one is the simulated deforming "cloth" dataset acquired using the *ArcSim* simulator [30], [31]. The second one is the high quality"facial" dataset which was provided courtesy of the research group of *Graphics, Vision & Video of the Max-Planck-Institute for Informatics* [32].
In order to create *Kinect* acquired raw data, we simulate a realistic acquisition of the "cloth" sequence using *Blensor* by placing the camera at a distance of $1.8m$ [29]. We have used a sequence of 25 frames from this dataset. This noisy data is then filtered in *KinectDeform* with $\sigma_c = 0.0185m$ and $0.00225m \leq \sigma_p \leq 0.00655m$. From *Blensor* we can get an estimate of the sensor noise ε_n. The simulated noisy data and results of *KinectDeform* are compared with the ground truth data to compute RMSE based on Euclidean distances with nearest neighbors using *CloudCompare* [33]. The quantitative and qualitative improvements due to *KinectDeform* are shown in Figure 5. For qualitative evaluation we compare the reconstructions of frames 5 and 15 obtained using *KinectDeform* with ground truth and raw acquisitions as shown in Fig. 6. Fig. 6 (d) and 6 (h) show the results of applying a deblurring filter on the results of *KinectDeform* to remove remaining artifacts and get more refined reconstructions [34]. Results show significant improvements in the 3D reconstructions as a result of *KinectDeform* both qualitatively and quantitatively.

For the "facial" dataset we use a sequence of 21 frames, simulate a laser scanner in *V-REP* with object placed at $0.5m$ away from the camera [35] and add depth noise to the acquisitions based on Laplacian distribution with 0 mean and standard deviation of $0.00025m$. The standard deviation parameters chosen for the weighting scheme of *KinectDeform* are $\sigma_c = 0.0004m$ and $0.0004m \geq \sigma_p \leq 0.000425m$. The results are shown in Fig. 7 and Fig. 8. Though similar improvements in 3D reconstructions can be seen in this case as well, an important factor apparent here is the effect of temporal filtering due to which the error decreases gradually as shown in the Fig. 7.

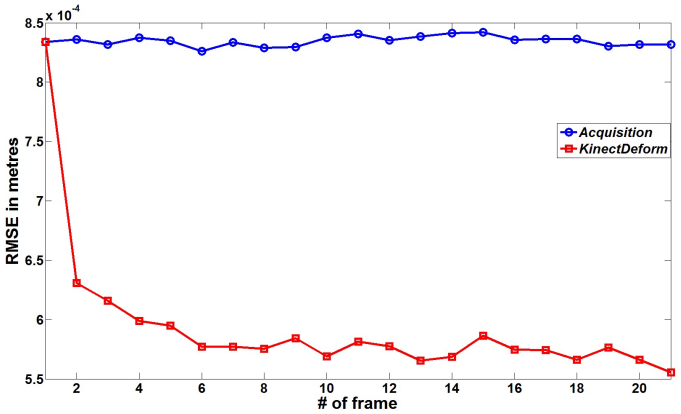
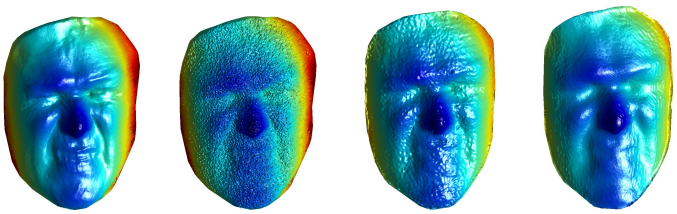
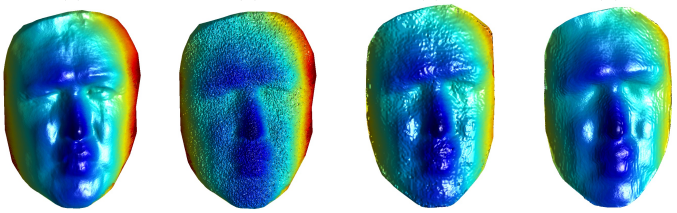


Fig. 7: RMSE of raw and filtered data with ground truth for "facial" dataset

To explain this difference in the temporal effect of filtering between two sequences, a closer look at the deformations introduced in both sequences is required. Fig. 9 (a) and Fig. 9 (b) show a large amount of deformation between frame 10 and frame 15 of the "cloth" sequence. Large deformations break the temporal effect of filtering because of factors such as self occlusions and by significantly changing geometry of the incoming reference frame thus reducing the value of important details brought by the result of previous iterations. That is why when the rate of deformation is small as in the sequence of "facial" dataset as shown in Fig. 9 (c) and Fig. 9 (d) the effect of temporal filtering is clearly visible as shown in Fig. 7.
We also tested *KinectDeform* on real data captured by *Asus Xtion Pro Live* camera using a plain cloth being waved in front of it. In this case we tested the empirical weighting scheme similar to *KinectFusion* in which the weight of reference is increased by 1 after every iteration until a threshold is reached. *KinectDeform* was run over 25 frames from this dataset and results for frames 10, 15 and 20 are shown in Fig. 10. It shows that even using this empirical weighting scheme results in smoother surfaces with well preserved details.

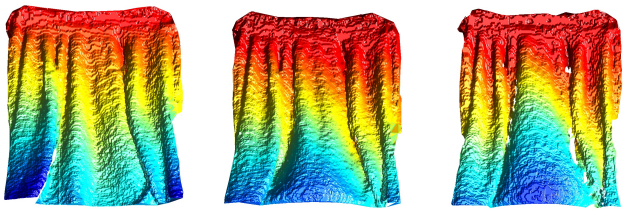


(a) (b) (c) (d)

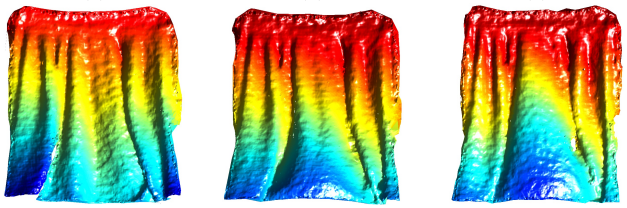


(e) (f) (g) (h)

Fig. 8: "Facial" dataset. **Top row:** Frame 5 (a) Ground truth, (b) raw data, (c) result of *KinectDeform*, (d) result of *KinectDeform* after deblurring. **Bottom row:** Frame 15 (e) Ground truth, (f) raw data, (g) result of *KinectDeform*, (h) result of *KinectDeform* after deblurring.

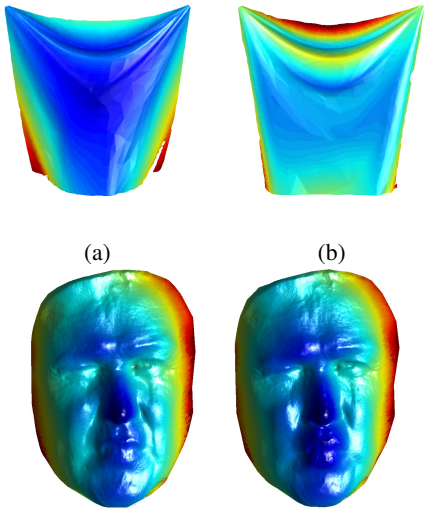


(a) (b) (c)



(d) (e) (f)

Fig. 10: Real moving cloth dataset. **Top row:** Raw acquisitions for (a) frame 10, (b) frame 15, (c) frame 20. **Bottom row:** Results of *KinectDeform* for (d) frame 10, (e) frame 15, (f) frame 20.



(a) (b) (c) (d)

Fig. 9: **Top row:** "Cloth" dataset, deformation between (a) frame 10 and (b) frame 15. **Bottom row:** "Facial" dataset, deformation between (c) frame 10 and (d) frame 15.

V. Conclusion

We have presented *KinectDeform*, a novel method for enhanced 3D reconstruction based on tracking of dynamic non-rigid objects. It has two main components, first is the use of an efficient and effective pair-wise non-rigid tracking which allows for tracking of non-rigid objects without any constraints and without using a template. Second is the use of a recursive filtering mechanism derived from *KinectFusion* but with a change in the reference being used and a weighting scheme which takes into account different sources of noise present in the input data. We have carried out both quantitative and qualitative evaluation of our method and we show that this algorithm is successfully able to filter noisy depth data to give smoother and feature preserving reconstructions over time. *KinectDeform* has been designed keeping in mind its planned extension to a completely automated real-time system which should enable us to analyze its performance over longer sequences constituting hundreds of data frames. It should also enable us to study further the domain of filtering based on non-rigid tracking for data acquired from consumer depth cameras which constitutes our future work.

Acknowledgment

This work was supported by the National Research Fund (FNR), Luxembourg, under the CORE project C11/BM/1204105/FAVE/Ottersten.

References

[1] O. Mac Aodha, N. D. Campbell, A. Nair, and G. J. Brostow, "Patch Based Synthesis for Single Depth Image Super-Resolution," in *ECCV (3)*, 2012, pp. 71–84.

[2] S. Schuon, C. Theobalt, J. Davis, and S. Thrun, "LidarBoost: Depth Superresolution for ToF 3D Shape Scanning," *In Proc. of IEEE CVPR 2009*, 2009.

[3] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, ser. UIST '11. New York, NY, USA: ACM, 2011, pp. 559–568. [Online]. Available: http://doi.acm.org/10.1145/2047196.2047270

[4] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-Time Dense Surface Mapping and Tracking," in *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, ser. ISMAR '11. Washington, DC,

USA: IEEE Computer Society, 2011, pp. 127–136. [Online]. Available: http://dx.doi.org/10.1109/ISMAR.2011.6092378

[5] H. Roth and M. Vona, "Moving Volume KinectFusion," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2012, pp. 112.1–112.11.

[6] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially Extended KinectFusion," in *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, Sydney, Australia, Jul 2012.

[7] F. Steinbrcker, C. Kerl, and D. Cremers, "Large-Scale Multi-resolution Surface Reconstruction from RGB-D Sequences," in *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ser. ICCV '13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 3264–3271. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2013.405

[8] M. Zeng, F. Zhao, J. Zheng, and X. Liu, "Octree-based fusion for realtime 3D reconstruction," *Graphical Models*, vol. 75, no. 3, pp. 126 – 136, 2013, computational Visual Media Conference 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1524070312000768

[9] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3D Reconstruction at Scale using Voxel Hashing," *ACM Transactions on Graphics (TOG)*, 2013.

[10] B. Kainz, S. Hauswiesner, G. Reitmayr, M. Steinberger, R. Grasset, L. Gruber, E. Veas, D. Kalkofen, H. Seichter, and D. Schmalstieg, "OmniKinect: Real-Time Dense Volumetric Data Acquisition and Applications," in *Proceedings of the 18th ACM symposium on Virtual reality software and technology*, ser. VRST '12. New York, NY, USA: ACM, 2012, pp. 25–32. [Online]. Available: http://doi.acm.org/10.1145/2407336.2407342

[11] C. Mrcio, A. L. Apolinario Jr., and A. C. S. Souza, "KinectFusion for Faces: Real-Time 3D Face Tracking and Modeling Using a Kinect Camera for a Markerless AR System," *SBC Journal on 3D Interactive Systems*, vol. 4, pp. 2–7, 2013.

[12] J. Sturm, E. Bylow, F. Kahl, and D. Cremers, "CopyMe3D: Scanning and printing persons in 3D," in *German Conference on Pattern Recognition (GCPR)*, Saarbrücken, Germany, September 2013.

[13] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers, "Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions," in *Robotics: Science and Systems Conference (RSS)*, June 2013.

[14] 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*, June 2013, pp. 1–8.

[15] Y. Cui, W. Chang, T. Nll, and D. Stricker, "KinectAvatar: Fully Automatic Body Capture Using a single Kinect," in *ACCV Workshop on Color Depth fusion in computer*. ACCV, 2012.

[16] 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 Transactions on Graphics (TOG)*, 2014.

[17] F. Destelle, C. Roudet, M. Neveu, and A. Dipanda, "Towards a real-time tracking of dense point-sampled geometry," *International Conference on Image Processing*, pp. 381–384, 2012.

[18] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," in *Proceedings of the Sixth International Conference on Computer Vision*, ser. ICCV '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 839–. [Online]. Available: http://dl.acm.org/citation.cfm?id=938978.939190

[19] S. Rusinkiewicz, O. A. Hall-Holt, and M. Levoy, "Real-time 3D model acquisition," in *SIGGRAPH*, 2002, pp. 438–446.

[20] B. Curless and M. Levoy, "A Volumetric Method for Building Complex Models from Range Images," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '96. New York, NY, USA: ACM, 1996, pp. 303–312. [Online]. Available: http://doi.acm.org/10.1145/237170.237269

[21] S. Parker, P. Shirley, Y. Livnat, C. Hansen, and P.-P. Sloan, "Interactive Ray Tracing for Isosurface Rendering," in *Proceedings of the Conference on Visualization '98*, ser. VIS '98. Los Alamitos, CA, USA: IEEE Computer Society Press, 1998, pp. 233–238. [Online]. Available: http://dl.acm.org/citation.cfm?id=288216.288266

[22] T. Basha, Y. Moses, and N. Kiryati, "Multi-view Scene Flow Estimation: A View Centered Variational Approach," in *International Journal of Computer Vision*, 2011, pp. 1–16.

[23] J. Cech, J. Sanchez-Riera, and R. P. Horaud, "Scene Flow Estimation by Growing Correspondence Seeds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, CO, 2011. [Online]. Available: http://perception.inrialpes.fr/Publications/2011/CSH11

[24] S. Hadfield and R. Bowden, "Kinecting the dots: Particle Based Scene Flow From Depth Sensors," in *Proceedings, International Conference on Computer Vision*, Barcelona, Spain, 6-13 Nov 2011, pp. 2290 – 2295. [Online]. Available: http://personal.ee.surrey.ac.uk/Personal/S.Hadfield/papers/Kinecting%20the%20dots%20Particle%20Based%20Scene%20Flow%20From%20Depth%20Sensors.pdf

[25] H. Li, B. Adams, L. J. Guibas, and M. Pauly, "Robust Single-view Geometry and Motion Reconstruction," in *ACM SIGGRAPH Asia 2009 Papers*, ser. SIGGRAPH Asia '09. New York, NY, USA: ACM, 2009, pp. 175:1–175:10. [Online]. Available: http://doi.acm.org/10.1145/1661412.1618521

[26] H. Li, R. W. Sumner, and M. Pauly, "Global Correspondence Optimization for Non-Rigid Registration of Depth Scans," *Computer Graphics Forum (Proc. SGP'08)*, vol. 27, no. 5, July 2008.

[27] Zeng, Ming and Zheng, Jiaxiang and Cheng, Xuan and Liu, Xinguo, "Templateless quasi-rigid shape modeling with implicit loop-closure." in *CVPR*. IEEE, 2013, pp. 145–152. [Online]. Available: http://dblp.uni-trier.de/db/conf/cvpr/cvpr2013.html#ZengZCL13

[28] F. Cazals and J. Giesen, "Delaunay triangulation based surface reconstruction: Ideas and algorithms," in *Effective Computational Geometry for Curves and Surfaces*. Springer, 2006, pp. 231–273.

[29] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, "BlenSor: Blender Sensor Simulation Toolbox Advances in Visual Computing," ser. Lecture Notes in Computer Science, G. Bebis, R. Boyle, B. Parvin, D. Koracin, S. Wang, K. Kyungnam, B. Benes, K. Moreland, C. Borst, S. DiVerdi, C. Yi-Jen, and J. Ming, Eds. Berlin, Heidelberg: Springer Berlin / Heidelberg, 2011, vol. 6939, ch. 20, pp. 199–208. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-24031-7_20

[30] R. Narain, T. Pfaff, and J. F. O'Brien, "Folding and crumpling adaptive sheets," *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 51:1–8, Jul. 2013, proceedings of ACM SIGGRAPH 2013, Anaheim. [Online]. Available: http://graphics.berkeley.edu/papers/Narain-FCA-2013-07/

[31] R. Narain, A. Samii, and J. F. O'Brien, "Adaptive Anisotropic Remeshing for Cloth Simulation," *ACM Transactions on Graphics*, vol. 31, no. 6, pp. 147:1–10, Nov. 2012, proceedings of ACM SIGGRAPH Asia 2012, Singapore. [Online]. Available: http://graphics.berkeley.edu/papers/Narain-AAR-2012-11/

[32] L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, and C. Theobalt, "Lightweight Binocular Facial Performance Capture Under Uncontrolled Lighting," *ACM Trans. Graph.*

[33] "CloudCompare." [Online]. Available: http://www.cloudcompare.org/

[34] K. A. Ismaeil, D. Aouada, B. Mirbach, and B. E. Ottersten, "Depth Super-Resolution by Enhanced Shift and Add," in *Computer Analysis of Images and Patterns - 15th International Conference, CAIP 2013, York, UK, August 27-29, 2013, Proceedings, Part II*, 2013, pp. 100–107.

[35] "V-REP." [Online]. Available: http://www.coppeliarobotics.com/