# DynamicFusion: Reconstruction and Tracking of Non-rigid Scenes in Real-Time

Richard A. Newcombe
newcombe@cs.washington.edu

Dieter Fox
fox@cs.washington.edu

Steven M. Seitz
seitz@cs.washington.edu

University of Washington, Seattle


Figure 1: Real-time reconstructions of a moving scene with DynamicFusion; both the person and the camera are moving. The initially noisy and incomplete model is progressively denoised and completed over time (left to right).

## Abstract

*We present the first dense SLAM system capable of reconstructing non-rigidly deforming scenes in real-time, by fusing together RGBD scans captured from commodity sensors. Our DynamicFusion approach reconstructs scene geometry whilst simultaneously estimating a dense volumetric 6D motion field that warps the estimated geometry into a live frame. Like KinectFusion, our system produces increasingly denoised, detailed, and complete reconstructions as more measurements are fused, and displays the updated model in real time. Because we do not require a template or other prior scene model, the approach is applicable to a wide range of moving objects and scenes.*

3D scanning traditionally involves separate capture and off-line processing phases, requiring very careful planning of the capture to make sure that every surface is covered. In practice, it's very difficult to avoid holes, requiring several iterations of capture, reconstruction, identifying holes, and recapturing missing regions to ensure a complete model. Real-time 3D reconstruction systems like KinectFusion [18, 10] represent a major advance, by providing users the ability to instantly see the reconstruction and identify regions that remain to be scanned. KinectFusion spurred a flurry of follow up research aimed at robustifying the tracking [9, 32] and expanding its spatial mapping capabilities to larger environments [22, 19, 34, 31, 9].

However, as with all traditional SLAM and dense reconstruction systems, the most basic assumption behind KinectFusion is that the observed scene is largely *static*. The core question we tackle in this paper is: *How can we generalise KinectFusion to reconstruct and track dynamic, non-rigid scenes in real-time?* To that end, we introduce DynamicFusion, an approach based on solving for a volumetric flow field that transforms the state of the scene at each time instant into a fixed, canonical frame. In the case of a moving person, for example, this transformation undoes the person's motion, warping each body configuration into the pose of the first frame. Following these warps, the scene is effectively rigid, and standard KinectFusion updates can be used to obtain a high quality, denoised reconstruction. This progressively denoised reconstruction can then be transformed back into the live frame using the inverse map; each point in the canonical frame is transformed to its location in the live frame (see Figure 1).

Defining a canonical "rigid" space for a dynamically moving scene is not straightforward. A key contribution of our work is an approach for non-rigid transformation and fusion that retains the optimality properties of volumetric scan fusion [5], developed originally for rigid scenes. The main insight is that undoing the scene motion to enable fusion of all observations into a single *fixed* frame can be achieved efficiently by computing the inverse map alone. Under this transformation, each canonical point projects along a line of sight in the live camera frame. Since the optimality arguments of [5] (developed for rigid scenes) depend only on lines of sight, we can generalize their optimality results to the non-rigid case.

Our second key contribution is to represent this volumetric warp efficiently, and compute it in real time. Indeed, even a relatively low resolution, $256^3$ deformation volume would require 100 million transformation variables to be computed at frame-rate. Our solution depends on a combination of adaptive, sparse, hierarchical volumetric basis functions, and innovative algorithmic work to ensure a real-

time solution on commodity hardware. As a result, DynamicFusion is the first system capable of real-time dense reconstruction in dynamic scenes using a single depth camera.

The remainder of this paper is structured as follows. After discussing related work, we present an overview of DynamicFusion in Section 2 and provide technical details in Section 3. We provide experimental results in Section 4 and conclude in Section 5.

## 1. Related Work

While no prior work achieves real-time, template-free, non-rigid reconstruction, there are two categories of closely related work: 1) real-time non-rigid *tracking* algorithms, and 2) *offline* dynamic reconstruction techniques.

**Real-time non-rigid template tracking.** The vast majority of non-rigid tracking research focuses on human body parts, for which specialised shape and motion templates are learnt or manually designed. The best of these demonstrate high accuracy, real-time performance capture for tracking faces [16, 3], hands [21, 20], complete bodies [27], or general articulated objects [23, 33].

Other techniques directly track and deform more general mesh models. [12] demonstrated the ability to track a statically acquired low resolution shape template and upgrade its appearance with high frequency geometric details not present in the original model. Recently, [37] demonstrated an impressive real-time version of a similar technique, using GPU accelerated optimisations. In that system, a dense surface model of the subject is captured while remaining static, yielding a template for use in their real-time tracking pipeline. This separation into template generation and tracking limits the system to objects and scenes that are completely static during the geometric reconstruction phase, precluding reconstruction of things that won't reliably hold still (e.g., children or pets).

**Offline simultaneous tracking and reconstruction of dynamic scenes.** There is a growing literature on *offline* non-rigid tracking and reconstruction techniques. Several researchers have extended ICP to enable small non-rigid deformations, e.g., [1, 2]. Practical advancements to pair-wise 3D shape and scan alignment over larger deformations make use of reduced deformable model parametrisations [14, 4]. In particular, embedded deformation graphs [25] use a sparsely sampled set of transformation basis functions that can be efficiently and densely interpolated over space. Quasi-rigid reconstruction has also been demonstrated [15, 35] and hybrid systems, making use of a known kinematic structure (e.g., a human body), are able to perform non-rigid shape denoising [36]. Other work combines non-rigid mesh template tracking and temporal denoising and completion [13], but does not obtain a single consistent representation of the scene.

More closely related to our work are template-free techniques. An intriguing approach to template-free non-rigid alignment, introduced in [17] and [26], treats each non-rigid scan as a view from a 4D geometric observation and performs 4D shape reconstruction. [30, 29] reconstruct a fixed topology geometry by performing pair-wise scan alignment. [24] use a space-time solid incompressible flow prior that results in water tight reconstructions and is effective against noisy input point-cloud data. [28] introduce animation cartography that also estimates shape and a per frame deformation by developing a dense correspondence matching scheme that is seeded with sparse landmark matches. Recent work using multiple fixed kinect cameras [8] [7] demonstrates larger scale non-rigid reconstruction by densely tracking and fusing all depth map data into a novel directional distance function representation.

All of these techniques require three to four orders of magnitude more time than is available within a real-time setting.

## 2. DynamicFusion Overview

DynamicFusion decomposes a non-rigidly deforming scene into a latent geometric surface, reconstructed into a rigid canonical space $\mathbf{S} \subseteq \mathbb{R}^3$; and a per frame volumetric warp field that transforms that surface into the live frame. There are three core algorithmic components to the system that are performed in sequence on arrival of each new depth frame:

1. Estimation of the volumetric model-to-frame warp field parameters (Section 3.3)
2. Fusion of the live frame depth map into the canonical space via the estimated warp field (Section 3.2)
3. Adaptation of the warp-field structure to capture newly added geometry (Section 3.4)

Figure 2 provides an overview.

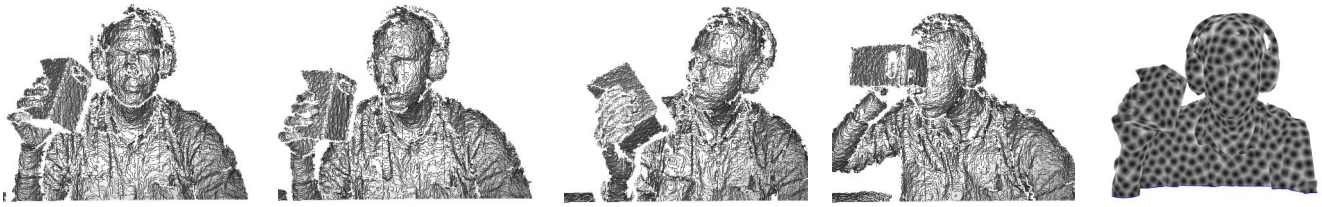
## 3. Technical Details

We will now describe the components of DynamicFusion in detail. First, we describe our dense volumetric warp-field parametrisation. This allows us to model per-frame deformations in the scene. The warp-field is the key extension over static state space representations used in traditional reconstruction and SLAM systems, and its estimation is the enabler of both non-rigid tracking and scene reconstruction.

### 3.1. Dense Non-rigid Warp Field

We represent dynamic scene motion through a volumetric warp-field, providing a per point $6D$ transformation $\mathcal{W} : \mathbf{S} \mapsto \mathbf{SE}(3)$. Whereas a dense $3D$ translation field would be sufficient to describe time varying geometry, we have found that representing the real-world transformation

SE(3): transformation matrix includes translation and rotation
6D: x,y,z angle + translation



(a) Initial Frame at $t = 0s$ (b) Raw (noisy) depth maps for frames at $t = 1s, 10s, 15s, 20s$ (c) Node Distance

(d) Canonical Model (e) Canonical model warped into its live frame (f) Model Normals

Figure 2: DynamicFusion takes an online stream of noisy depth maps (a,b) and outputs a real-time dense reconstruction of the moving scene (d,e). To achieve this, we estimate a volumetric warp (motion) field that transforms the canonical model *space* into the live frame, enabling the scene motion to be undone, and all depth maps to be densely fused into a single rigid TSDF reconstruction (d,f). Simultaneously, the structure of the warp field is constructed as a set of sparse 6D transformation nodes that are smoothly interpolated through a $k$-nearest node average in the canonical frame (c). The resulting per-frame warp field estimate enables the progressively denoised and completed scene geometry to be transformed into the live frame in real-time (e). In (e) we also visualise motion trails for a sub-sample of model vertices over the last 1 second of scene motion together with a coordinate frame showing the rigid body component of the scene motion. In (c) we render the nearest node to model surface distance where increased distance is mapped to a lighter value.

of objects with both translation and rotation results in significantly better tracking and reconstruction. For each canonical point $v_c \in \mathbf{S}$, $\mathbf{T}_{lc} = \mathcal{W}(v_c)$ transforms that point from canonical space into the live, non-rigidly deformed frame of reference.

Since we will need to estimate the warp function for each new frame, $\mathcal{W}_t$, its representation must be efficiently optimisable. One possibility is to densely sample the volume, e.g. representing a quantised $\mathbf{SE}(3)$ field at the resolution of the truncated signed distance (TSDF) geometric representation. However, a typical TSDF volume reconstruction at a relatively low resolution of $256^3$ voxels would require the solution of $6 \times 256^3$ parameters per frame, about 10 million times more than in the original KinectFusion algorithm, which only estimates a single rigid transformation. Clearly, a completely dense parametrisation of the warp function is infeasible. In reality, surfaces tend to move smoothly in space, and so we can instead use a sparse set of transformations as bases and define the dense volumetric warp function through interpolation. Due to its computational efficiency and high quality interpolation capability we use dual-quaternion blending **DQB** [11], to define our warp function:

$$\mathcal{W}(x_c) \equiv SE3(\mathbf{DQB}(x_c)) \,, \tag{1}$$

where the weighted average over unit dual quaternion transformations is simply $\mathbf{DQB}(x_c) \equiv \frac{\sum_{k \in N(x_c)} \mathbf{w}_k(x_c) \hat{\mathbf{q}}_{kc}}{\left\| \sum_{k \in N(x_c)} \mathbf{w}_k(x_c) \hat{\mathbf{q}}_{kc} \right\|}$, with each unit dual-quaternion $\hat{\mathbf{q}}_{kc} \in \mathbb{R}^8$. Here, $N(x)$ are the k-nearest transformation nodes to the point $x$ and $\mathbf{w}_k : \mathbb{R}^3 \mapsto \mathbb{R}$ defines a weight that alters the radius of influence of each node and $SE3(.)$ converts from quaternions back to an $\mathbf{SE}(3)$ transformation matrix. The state of the warp-field $\mathcal{W}_t$ at time $t$ is defined by the values of a set of $n$ deformation nodes $\mathcal{N}^t_{\mathbf{warp}} = \{\mathbf{dg}_v, \mathbf{dg}_w, \mathbf{dg}_{se3}\}_t$. Each of the $i = 1..n$ nodes has a position in the canonical frame $\mathbf{dg}^i_v \in \mathbb{R}^3$, its associated transformation $\mathbf{T}_{ic} = \mathbf{dg}^i_{se3}$, and a radial basis weight $\mathbf{dg}_w$ that controls the extent of the transformation $\mathbf{w}_i(x_c) = \exp\left(-\|\mathbf{dg}^i_v - x_c\|^2 / \left(2(\mathbf{dg}^i_w)^2\right)\right)$. Each radius parameter $\mathbf{dg}^i_w$ is set to ensure the node's influence overlaps with neighbouring nodes, dependent on the sampling sparsity of nodes, which we describe in detail in section (3.4). Since the warp function defines a rigid body transformation for all supported space, both position and any associated orientation of space is transformed, e.g., the vertex $v_c$ from a surface with orientation or normal $n_c$ is transformed into the live frame as $(v_t, 1)^\top = \mathcal{W}_t(v_c)(v_c^\top, 1)^\top$ and $(n_t, 0)^\top = \mathcal{W}_t(v_c)(n_c^\top, 0)^\top$. We note that scaling of space can also be represented with this warp function, since compression and expansion of space are represented by neighbouring points moving in converging and diverging directions. Finally, we note that we can factor out any rigid body transformation common to all points in the volume, e.g., due to camera motion. We therefore introduce the explicit warped model to live camera transform, $\mathbf{T}_{lw}$, and compose this onto the volumetric warp function;

可以表示物體的膨脹和縮小

組合

our complete warp-field is then given as:

$$\mathcal{W}_t(x_c) = \mathbf{T}_{lw} SE3(\mathbf{DQB}(x_c)). \quad (2)$$

### 3.2. Dense Non-Rigid Surface Fusion

We now describe how, given the model-to-frame warp field $\mathcal{W}_t$, we update our canonical model geometry. Our reconstruction into the canonical space $\mathbf{S}$ is represented by the sampled TSDF $\mathcal{V} : \mathsf{S} \mapsto \mathbb{R}^2$ within a voxel domain $\mathsf{S} \subset \mathbb{N}^3$. The sampled function holds for each voxel $\mathbf{x} \in \mathsf{S}$ corresponding to the sampled point $x_c$, a tuple $\mathcal{V}(\mathbf{x}) \mapsto [\mathsf{v}(\mathbf{x}) \in \mathbb{R}, \mathsf{w}(\mathbf{x}) \in \mathbb{R}]^\top$ holding a weighted average of all projective TSDF values observed for that point so far $\mathsf{v}(\mathbf{x})$, together with the sum of all associated weights $\mathsf{w}(\mathbf{x})$.

We extend the projective TSDF fusion approach originally introduced by [6] to operate over non-rigidly deforming scenes. Given the live depth image $D_t$, we transform each voxel center $x_c \in \mathbf{S}$ by its estimated warp into the live frame $(x_t^\top, 1)^\top = \mathcal{W}_t(x_c)(x_c^\top, 1)^\top$, and carry through the TSDF surface fusion operation by directly projecting the warped center into the depth frame. This allows the TSDF for a point in the canonical frame to be updated by computing the projective TSDF in the deforming frame without having to resample a warped TSDF in the live frame. The projective signed distance at the warped canonical point is:

$$\mathbf{psdf}(x_c) = \left[\mathbf{K}^{-1} D_t(u_c) \left[u_c^\top, 1\right]^\top\right]_z - [x_t]_z \ , \quad (3)$$

where $u_c = \pi(\mathbf{K} x_t)$ is the pixel into which the voxel center projects. We compute distance along the optical ($z$) axis of the camera frame using the $z$ component denoted $[.]_z$. $\mathbf{K}$ is the known $3 \times 3$ camera intrinsic matrix, and $\pi$ performs perspective projection. For each voxel $\mathbf{x}$, we update the TSDF to incorporate the projective SDF observed in the warped frame using TSDF fusion:

$$\mathcal{V}(\mathbf{x})_t = \begin{cases} [\mathsf{v}'(\mathbf{x}), \mathsf{w}'(\mathbf{x})]^\top, & \text{if } \mathbf{psdf}(\mathbf{dc}(\mathbf{x})) > -\tau \\ \mathcal{V}(\mathbf{x})_{t-1}, & \text{otherwise} \end{cases} \quad (4)$$

where $\mathbf{dc}(.)$ transforms a discrete voxel point into the continuous TSDF domain. The truncation distance $\tau > 0$ and the updated TSDF value is given by the weighted averaging scheme [5], with the weight truncation introduced in [18]:

$$\begin{aligned} \mathsf{v}'(\mathbf{x}) &= \frac{\mathsf{v}(\mathbf{x})_{t-1}\mathsf{w}(\mathbf{x})_{t-1} + \min(\rho, \tau) w(\mathbf{x})}{\mathsf{w}(\mathbf{x})_{t-1} + w(\mathbf{x})} \\ \rho &= \mathbf{psdf}(\mathbf{dc}(\mathbf{x})) \\ \mathsf{w}'(\mathbf{x}) &= \min(\mathsf{w}(\mathbf{x})_{t-1} + w(\mathbf{x}), w_{max}) \ . \end{aligned} \quad (5)$$

Unlike the static fusion scenario where the weight $w(\mathbf{x})$ encodes the uncertainty of the depth value observed at the projected pixel in the depth frame, we also account for uncertainty associated with the warp function at $x_c$. In the case of the single rigid transformation in original TSDF fusion, we are certain that observed surface regions, free space, and unobserved regions transform equivalently. In our non-rigid case, the further away the point $x_c$ is from an already mapped and observable surface region, the less certain we can be about its transformation. We use the average distance from $x_c$ to its $k$-nearest deformation nodes as a proxy for this increase in uncertainty and scale: $w(\mathbf{x}) \propto \frac{1}{k} \sum_{i \in N(x_c)} \|\mathbf{dg}_w - x_c\|_2$. We note that our non-rigid fusion generalises the static reconstruction case used in KinectFusion, replacing the single (rigid) model-to-camera transform with a per voxel warp that transforms the associated space into the live (non-rigid) frame (see Figure 3). This technique greatly simplifies the non-rigid reconstruction process over methods where all frames are *explicitly* warped into a canonical frame. Furthermore, given a correct warp field, then, since all TSDF updates are computed using distances in the camera frame, the non-rigid projective TSDF fusion approach maintains the optimality guarantees for surface reconstruction from noisy observations originally proved for the static reconstruction case in [6].

### 3.3. Estimating the Warp-field State $\mathcal{W}_t$

We estimate the current values of the transformations $\mathbf{dg}_{se3}$ in $\mathcal{W}_t$ given a newly observed depth map $D_t$ and the current reconstruction $\mathcal{V}$ by constructing an energy function that is minimised by our desired parameters:

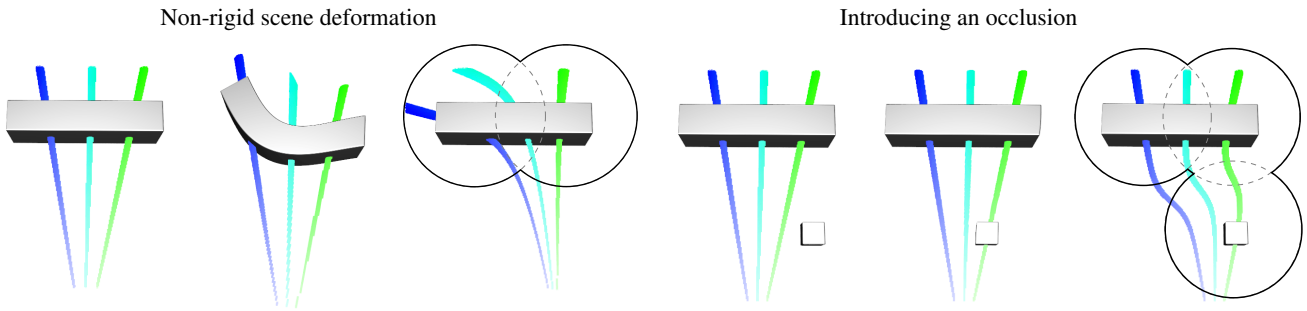
$$E(\mathcal{W}_t, \mathcal{V}, D_t, \mathcal{E}) = \mathbf{Data}(\mathcal{W}_t, \mathcal{V}, D_t) + \lambda \mathbf{Reg}(\mathcal{W}_t, \mathcal{E}) \ . \quad (6)$$

Our data term consists of a dense model-to-frame ICP cost $\mathbf{Data}(\mathcal{W}_t, \mathcal{V}, D_t)$ which is coupled with a regularisation term $\mathbf{Reg}(\mathcal{W}_t, \mathcal{E})$ that penalises non-smooth motion fields, and ensures as-rigid-as-possible deformation between transformation nodes connected by the edge set $\mathcal{E}$. The coupling of a data-term formed from linearly blended transformations with a rigid-as-possible graph based regularisation is a form of the embedded deformation graph model introduced in [25]. The regularisation parameter $\lambda$ enables a trade-off between relaxing rigidity over the field when given high quality data, and ensuring a smooth consistent deformation of non or noisily observed regions of space. We defined these terms in the next subsections.

#### 3.3.1 Dense Non-Rigid ICP Data-term

Our aim is to estimate all non-rigid transformation parameters $\mathbf{T}_{ic}$ and $\mathbf{T}_{lw}$ that warp the canonical volume into the live frame. We achieve this by performing a dense non-rigid alignment of the current surface reconstruction, extracted from the canonical volume's zero level set, into the live frame's depth map.

**Surface Prediction and Data-Association:** The current zero level set of the TSDF $\mathcal{V}$ is extracted by marching cubes and stored as a polygon mesh with point-normal pairs in the canonical frame: $\hat{\mathcal{V}}_c \equiv \{V_c, N_c\}$. We non-rigidly transform this mesh into the live frame using the current warp field $\mathcal{W}_t$ resulting in the warped point-normals $\hat{\mathcal{V}}_w$.




(a) Live frame $t = 0$ (b) Live Frame $t = 1$ (c) Canonical $\mapsto$ Live (d) Live frame $t = 0$ (e) Live Frame $t = 1$ (f) Canonical $\mapsto$ Live

Figure 3: An illustration of how each point in the canonical frame maps, through a correct warp field, onto a ray in the live camera frame when observing a deforming scene. In (a) the first view of a dynamic scene is observed. In the corresponding canonical frame, the warp is initialized to the identity transform and the three rays shown in the live frame also map as straight lines in the canonical frame. As the scene deforms in the live frame (b), the warp function transforms each point from the canonical and into the corresponding live frame location, causing the corresponding rays to bend (c). Note that this warp can be achieved with two $6D$ deformation nodes (shown as circles), where the left node applies a clockwise twist. In (d) we show a new scene that includes a cube that is about to occlude the bar. In the live frame (e), as the cube occludes a portion of the bar, the points in the canonical frame (f) are warped to correctly pass through the cube.

We obtain an initial estimate for data-association (correspondence) between the model geometry and the live frame by rendering the warped surface $\hat{\mathcal{V}}_w$ into the live frame shaded with canonical frame vertex positions using a rasterizing rendering pipeline. This results in a prediction of the canonical frame's geometry that is currently predicted to be visible in the live frame: $\mathcal{P}(\hat{\mathcal{V}}_c)$. We store this prediction as a pair of images $\{\mathbf{v}, \mathbf{n}\} : \Omega \mapsto \mathcal{P}(\hat{\mathcal{V}}_c)$, where $\Omega$ is the pixel domain of the predicted images, storing the rendered canonical frame vertices and normals.

Given optimal transformation parameters for the current time frame, the predicted-to-be-visible geometry should transform close, modulo observation noise, to the live surface $\mathbf{vl} : \Omega \mapsto \mathbb{R}^3$, formed by back projection of the depth image $[\mathbf{vl}(u)^\top, 1]^\top = \mathbf{K}^{-1} D_t(u) [u^\top, 1]^\top$. This can be quantified by a per pixel dense model-to-frame point-plane error, which we compute under the robust Tukey penalty function $\psi_{\mathbf{data}}$, summed over the predicted image domain $\Omega$:

$$\mathbf{Data}(\mathcal{W}, \mathcal{V}, D_t) \equiv \sum_{u \in \Omega} \psi_{\mathbf{data}} \left( \hat{\mathbf{n}}_u^\top \left( \hat{\mathbf{v}}_u - \mathbf{vl}_{\tilde{u}} \right) \right) \quad (7)$$

augment. The transformed model vertex $\mathbf{v}(u)$ is simply $\tilde{\mathbf{T}}^u = \mathcal{W}(\mathbf{v}(u))$, producing the current canonical to live frame point-normal predictions $\hat{\mathbf{v}}_u = \tilde{\mathbf{T}}^u \mathbf{v}(u)$ and $\hat{\mathbf{n}}_u = \tilde{\mathbf{T}}^u \mathbf{n}(u)$, and data-association of that model point-normal is made with a live frame point-normal through perspective projection into the pixel $\tilde{u} = \pi(\mathbf{K}\hat{\mathbf{v}}_u)$.

We note that, ignoring the negligible cost of rendering the geometry $\hat{\mathcal{V}}_w$, the ability to extract, predict, and perform projective data association with the *currently visible* canonical geometry leads to a data-term evaluation that has a computational complexity with an upper bound in the number of pixels in the observation image. Furthermore, each data-term summand depends only on a subset of the $n$ transformations when computing $\mathcal{W}$, and the region over which each node has a numerically significant impact on the error function is compact. In practice, the result is a computational cost similar to a single rigid body dense projective point-plane data-term evaluation (as used in KinectFusion).

### 3.3.2 Warp-field Regularization

It is crucial for our non-rigid TSDF fusion technique to estimate a deformation not only of currently visible surfaces, but over all space within S. This enables reconstruction of new regions of the scene surface that are about to come into view. However, nodes affecting canonical space within which no currently observed surface resides will have no associated data term. In any case, noise, missing data and insufficient geometric texture in the live frame – an analogue to the aperture problem in optical-flow – will result in optimisation of the transform parameters being ill-posed. How should we constrain the motion of non-observed geometry? Whilst the fully correct motion depends on object dynamics and, where applicable, the subject's volition, we make use of a simpler model of unobserved geometry: that it deforms in a piece-wise smooth way.

We use a deformation graph based regularization defined between transformation nodes, where an edge in the graph between nodes $i$ and $j$ adds a rigid-as-possible regularisation term to the total error being minimized, under the discontinuity preserving Huber penalty $\psi_{\mathbf{reg}}$. The total regularisation term sums over all pair-wise connected nodes:

$$\mathbf{Reg}(\mathcal{W}, \mathcal{E}) \equiv \sum_{i=0}^{n} \sum_{j \in \mathcal{E}(i)} \alpha_{ij} \psi_{\mathbf{reg}} \left( \mathbf{T}_{ic} \mathbf{dg}_v^j - \mathbf{T}_{jc} \mathbf{dg}_v^j \right) , \quad (8)$$

where $\mathcal{E}$ defines the regularisation graph topology, and $\alpha_{ij}$ defines the weight associated with the edge, which we set to $\alpha_{ij} = \max(\mathbf{dg}_w^i, \mathbf{dg}_w^j)$.

**Hierarchical Deformation Tree:** In original applications of the embedded deformation graph [25] approach to non-rigid tracking, $\mathcal{E}$ is defined as the $k-$nearest neighbours of each node or all nodes within a specified radius. We find that either of these edge sets work well in practice, but have further found that by constructing a hierarchical deformation graph with no explicit edge connectivity between siblings, both stability of the deformation field increases while computational costs of minimising the total energy function decreases. Given the current set of deformation nodes $\mathcal{N}_{\mathbf{warp}}$, we construct a hierarchy of regularisation nodes $\mathcal{N}_{\mathbf{reg}} = \{\mathbf{r}_v, \mathbf{r}_{se3}, \mathbf{r}_w\}$ (construction of the hierarchy is described in section 3.4). Importantly, we do not use $\mathcal{N}_{\mathbf{reg}}$ within the warp function $\mathcal{W}$; they are used to induce longer range regularisation across the warp function with reduced computational complexity. Each level of the regularisation node hierarchy also defines the node positions, transforms, and support weights. Our regularisation graph topology is then simply formed by adding edges from each node of the hierarchy (starting in $\mathcal{N}_{warp}$) to its $k-$nearest nodes in the next coarser level. Since the latent surface reconstruction will grow within the canonical frame (until completely observed), we need to continuously update the deformation nodes and the regularisation graph, potentially at frame-rate, which we describe in Section (3.4).

#### 3.3.3 Efficient Optimization

Estimation of all transformation parameters, $\mathbf{T}_{lw}$, $\mathbf{dg}_{se3}$ and $\mathbf{r}_{se3}$, is performed by minimising the total energy $E$ (Eq. 6). We minimise $E$ through Gauss-Newton non-linear optimisation, which requires iteratively re-linearising $E$ around the currently estimated deformation node parameters and forming and solving the normal equations $\mathbf{J}^\top\mathbf{J}\hat{\mathbf{x}} = \mathbf{J}^\top\mathbf{e}$. We formulate compositional updates $\hat{\mathbf{x}}$ through the exponential map with a per-node twist $\xi_i \in se(3)$, requiring 6 variables per node transform, and perform linearisation around $\xi_i = 0$. It is crucial that our solver is efficient, since a warp field of a deforming scene (for example a person gesticulating) may require several hundred deformation nodes, corresponding to many thousands of parameters requiring solution at frame rate. Recently, [37] demonstrated a real-time solution to a related non-rigid tracking optimization using a GPU accelerated pre-conditioned conjugate gradient descent solver. We take a different approach, using instead a direct sparse Cholesky factorization of each linearised system. We note that direct solvers resolve low-frequency residuals very effectively, which is critical to ensuring minimal drift during reconstruction.

The main computational complexity in minimising $E$ involves constructing and factorizing the Gauss-Newton approximation of the Hessian: $\mathbf{J}^\top\mathbf{J} = \mathbf{J}_\mathbf{d}^\top\mathbf{J}_\mathbf{d} + \lambda\mathbf{J}_\mathbf{r}^\top\mathbf{J}_\mathbf{r}$. First, we note that the $k$-nearest node field induces non-zero blocks in the data term component $\mathbf{J}_\mathbf{d}^\top\mathbf{J}_\mathbf{d}$ for each pair of nodes currently involved in deforming $\mathcal{V}$ into an observable region of the live frame. Building the full linear system on the GPU currently hinders real-time performance due to requirements on global GPU memory read and writes. Fortunately, since the associated weight of each deformation node reduces to a very small value outside of $3\mathbf{dg}_w$, any data term there can be safely ignored. Approximating further, we compute only the block diagonal terms for $\mathbf{J}_\mathbf{d}^\top\mathbf{J}_\mathbf{d}$, as if the effect of each node on the warp function were independent, resulting in a computational cost of building the structure similar to a single rigid body transformation for the frame. This technique is reasonable since, after solution of the linearised system, the current canonical model surface is re-warped into the live frame, resulting in a form of time-lagged linearisation of the objective.

A second optimization efficiency comes in the form of the sparse linear system that our hierarchical regularisation term induces. We construct the regularisation Hessian approximation $\mathbf{J_r}^\top\mathbf{J_r}$ using the linearisation of the virtual deformation node parameters, laying out the parameter blocks with a coarse to fine ordering. The resulting complete system matrix has a block arrow-head form which is efficiently factorized with a block-Cholesky decomposition.

Prior to non-rigid optimisation, given a new frame, we first estimate the factorised transformation $\mathbf{T}_{lw}$ using the dense ICP introduced in KinectFusion. This resolves the relative rigid body transformation, i.e. due to camera motion and improves data-association for the non-rigid solver. To that end, we re-render the predicted surface geometry $\hat{\mathcal{V}}$ and perform 2 or 3 iterations of the dense non-rigid optimization. Finally, we factorise out any resulting rigid body transformation $\tilde{\mathbf{T}}$ common across all deformation nodes and update $\mathbf{T}_{lw} \leftarrow \tilde{\mathbf{T}}\mathbf{T}_{lw}$.

It is important that computation of $\mathcal{W}$ is fast, requiring evaluation many millions of times per frame, all residing on the GPU. We therefore pre-compute, for each updated set of deformation node positions, a discretisation of the $k-$nearest node field required in dual-quaternion blending (**DQB**) with the same resolution of our volumetric TSDF: $\mathcal{I} : \mathsf{S} \mapsto \mathbb{N}^k$. Due to the sparsity of the deformation nodes relative to the sampling density of $\mathsf{S}$, this is a very fine approximation and is efficiently updated on the GPU whenever the set of nodes is updated.

### 3.4. Extending the Warp-field

In the preceding subsections we defined how the canonical space can be deformed through $\mathcal{W}$ (Section 3.1), introduced the optimisation required to estimate warp-field state through time (3.3), and showed how, given an estimated warp field, we can incrementally update the canonical surface geometry (3.2). As this model grows, so must the support of the warp function. In this subsection we de-

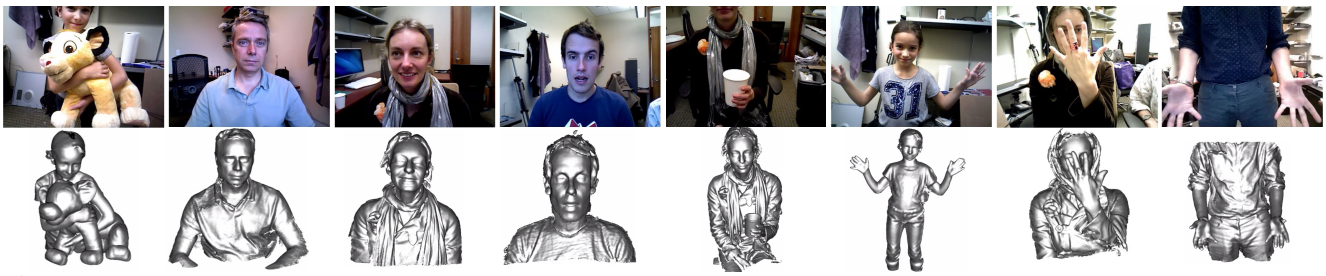


Figure 4: The first frames and final canonical models from DynamicFusion results shown in our accompanying video, available on our project website: http://grail.cs.washington.edu/projects/dynamicfusion.

scribe our approach to extending the warp-field parametrisation to ensure deformations are represented, over both the newly emerging surface geometry and soon to be observed space. This consists of incrementally updating the deformation graph nodes $\mathcal{N}_{\mathbf{warp}}$, and then recomputing a new hierarchical edge topology $\mathcal{E}$ that expands the regularisation to include the new nodes.

**Inserting New Deformation Nodes into $\mathcal{N}_{\mathbf{warp}}$:** After performing a non-rigid TSDF fusion step, we extract the surface estimate in the canonical frame as the polygon mesh $\hat{\mathcal{V}}_c$. Given the current set of nodes $\mathcal{N}_{\mathbf{warp}}$, we compute the extent to which the current warp function *covers* the extracted geometry. This simply entails computing the normalised distance from each vertex $v_c \in \hat{\mathcal{V}}_c$ to its supporting nodes. An unsupported surface vertex is detected when the distance $\min_{k \in N(x_c)} \left( \frac{\|\mathbf{dg}_v^k - v_c\|}{\mathbf{dg}_w^k} \right) \geq 1$. The set of all unsupported vertices is then spatially sub-sampled using a simple radius search averaging to reduce the vertices to a set of new node positions $\tilde{\mathbf{dg}}_v$ that are at least $\epsilon$ distance apart. We note that $\epsilon$ is an important parameter in DynamicFusion; while the regularisation parameter $\lambda$ enforces global deformation smoothness, $\epsilon$ defines the effective resolution of the motion field. Each new node center $\mathbf{dg}_v^* \in \tilde{\mathbf{dg}}_v$ requires an initialisation of its current transformation, which is obtained directly through **DQB** with the current warp $\mathbf{dg}_{se3}^* \leftarrow \mathcal{W}_t(\mathbf{dg}_v^*)$. Finally, we update the current set of deformation nodes to correspond to the current time $\mathcal{N}_{\mathbf{warp}}^t = \mathcal{N}_{\mathbf{warp}}^{t-1} \cup \{\tilde{\mathbf{dg}}_v, \tilde{\mathbf{dg}}_{se3}, \tilde{\mathbf{dg}}_w\}$. For each new node to be inserted, we perform an efficient GPU based update to the pre-computed $k$ nearest node field $\mathcal{I}$.

**Updating the Regularisation Graph $\mathcal{E}$:** Given the newly updated set of deformation nodes, we construct an $L \geq 1$ level regularisation graph node hierarchy, where the $l = 0$ level nodes will simply be $\mathcal{N}_{\mathbf{warp}}$. We compute the next $l = 1$ level of regularisation nodes by running the radius search based sub-sampling on the warp field nodes $\mathbf{dg}_v$ to an increased decimation radius of $\epsilon\beta^l$, where $\beta > 1$, and again compute the initial node transforms through **DQB** with the now updated $\mathcal{W}_t$. We repeat this to compute the remaining levels of the hierarchy. A completely new set of regularisation edges $\mathcal{E}$ is then constructed, starting with edges from $l = 0$ (i.e. $\mathcal{N}_{\mathbf{warp}}$) to the nodes in $\mathcal{N}_{\mathbf{reg}}$ at $l = 1$. Edges are added for each node in the finer level to its $k$-nearest neighbours in the coarser level.

## 4. Results

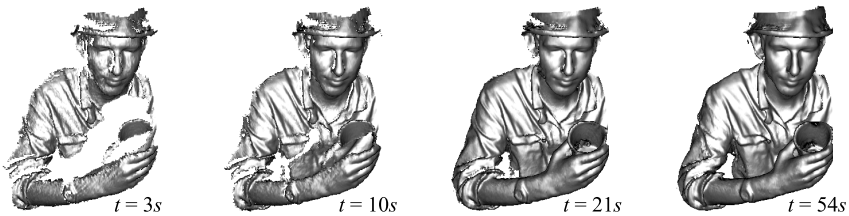
We demonstrate the system with a range of deforming scenes captured directly from DynamicFusion in Figure (5), and throughout the paper. In Figure (4) we show the first frames and reconstructions for further results in our accompanying video. These examples highlight the ability of DynamicFusion to (1) continuously track across large motion during reconstruction, (2) fill in initially occluded parts of the scene, and (3) generate consistent geometry despite many loop closures occurring during the capture process.

**Parameters:** results presented were obtained live from the system with optimisation parameters $\lambda = 200$, $\psi_{\mathbf{data}} = 0.01$, $\psi_{\mathbf{reg}} = 0.0001$; $L = 4$ levels in the regularisation hierarchy with $\beta = 4$ and a decimation density of $\epsilon = 25mm$ for all reconstructions except Figure (1) where $\epsilon = 15mm$. We have found that the system works reliably across a range of dynamic scenes and settings of parameters. We urge readers to view the associated video and supplementary material for additional details of the reconstruction process and to fully appreciate the live capabilities of the system.

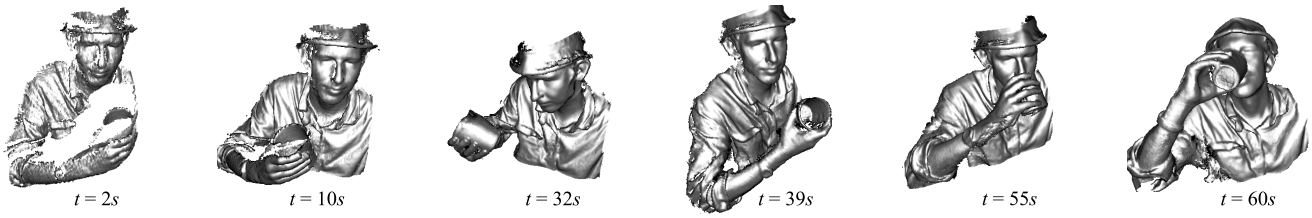
**Limitations and Discussion:** While DynamicFusion can easily handle closing topological surfaces (see the hands in 5b), it is currently limited in its ability to achieve dynamic reconstruction of scenes that quickly move from a closed to open topology (for example starting a reconstruction with closed hands and then opening).

More generally, failures common to real-time differential tracking can cause unrecoverable model corruption or result in loop closure failures. Large inter-frame motions, or motion of occluded regions, will also lead to an inaccurate surface prediction that prevents projective data-association in later frames.

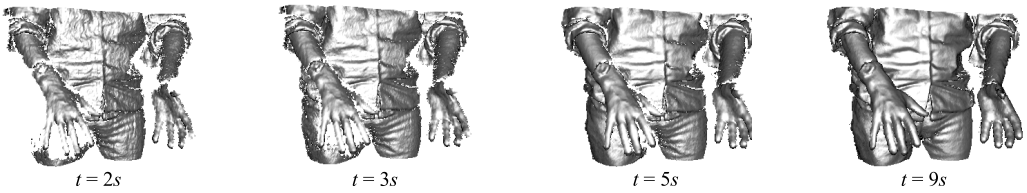
Stability of the warp field and subsequent reconstruction is achieved by the combined qualities of the as-rigid-as-possible regularisation, warp field parametrisation, and the use of a dense data-term. However, we have observed limits on this stability when attempting reconstruction of highly dynamic scenes. For example, reducing the regularisation weight and increasing the density of nodes enables



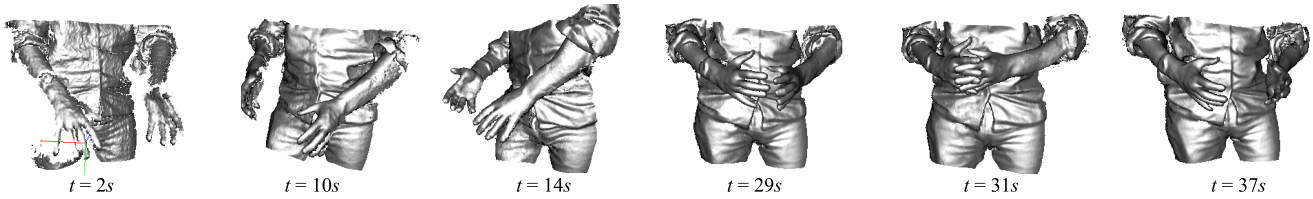

Canonical Model for "drinking from a cup"




(a) Canonical model warped into the live frame for "drinking from a cup"




Canonical Model for "Crossing fingers"




(b) Canonical model warped into the live frame for "crossing fingers"

Figure 5: Real-time non-rigid reconstructions for two deforming scenes. Upper rows of (a) and (b) show the canonical models as they evolve over time, lower rows show the corresponding warped geometries tracking the scene. In (a) complete models of the arm and the cup are obtained. Note the system's ability to deal with large motion and add surfaces not visible in the initial scene, such as the bottom of the cup and the back side of the arm. In (b) we show full body motions including clasping of the hands where we note that the model stays consistent throughout the interaction.

tracking scenes with more fluid deformations than shown in the results, but the long term stability can degrade and tracking will fail when the observed data term is not able to constrain the optimisation sufficiently. Finally, we show reconstruction results that are at the current limit of what can be obtained in real-time with DynamicFusion, and there are two limitations for scaling further. As in KinectFusion, volumetric TSDF memory limits geometric extent; but there are many solutions to this in the literature. More challenging is the estimation of a growing warp field. As the size and complexity of the scene increases, proportionally more is occluded from the camera, and the problem of predicting the motion of occluded areas becomes much more challenging. This is a subject of our ongoing research.

## 5. Conclusions

In this paper we introduced DynamicFusion, the first real-time dense *dynamic scene* reconstruction system, removing the static scene assumption pervasive across real-time 3D reconstruction and SLAM systems. We achieved this by generalising the volumetric TSDF fusion technique to the non-rigid case, as well as developing an efficient approach to estimate a volumetric 6D warp field in real-time. DynamicFusion obtains reconstructions of objects whilst they deform and provides dense correspondence across time. We believe that DynamicFusion, like KinectFusion, will open up a number of interesting applications of real-time 3D scanning and SLAM systems in dynamic environments.

## Acknowledgements

This work was funded in part by Google, the Intel Science and Technology Center for Pervasive Computing (ISTC-PC) and by ONR grant N00014-13-1-0720. We would like to thank Daniel Canelhas, Steven Lovegrove and Tanner Schmidt for many useful conversations and insights during the development of this work.

## References

[1] B. Brown and S. Rusinkiewicz. Non-Rigid Range-Scan Alignment Using Thin-Plate Splines. In *Symposium on 3D Data Processing, Visualization, and Transmission*, Sept. 2004. 2

[2] B. J. Brown and S. Rusinkiewicz. Global Non-rigid Alignment of 3-D Scans. *ACM Trans. Graph.*, 26(3), July 2007. 2

[3] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3D Shape Regression for Real-time Facial Animation. *ACM Trans. Graph.*, 32(4):41:1–41:10, July 2013. 2

[4] W. Chang and M. Zwicker. Range Scan Registration Using Reduced Deformable Models. *Comput. Graph. Forum*, 28(2):447–456, 2009. 2

[5] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of SIGGRAPH*, 1996. 1, 4

[6] B. L. Curless. *New Methods for Surface Reconstruction from Range Images*. PhD thesis, Stanford University, 1997. 4

[7] M. Dou and H. Fuchs. Temporally enhanced 3D capture of room-sized dynamic scenes with commodity depth cameras. In *Virtual Reality (VR), 2014 iEEE*, pages 39–44, March 2014. 2

[8] M. Dou, H. Fuchs, and J.-M. Frahm. Scanning and tracking dynamic objects with commodity depth cameras. In *Mixed and Augmented Reality (ISMAR), 2013 IEEE International Symposium on*, 2013. 2

[9] P. Henry, D. Fox, A. Bhowmik, and R. Mongia. Patch Volumes: Segmentation-based Consistent Mapping with RGB-D Cameras. In *International Conference on 3D Vision (3DV)*, 2013. 1

[10] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. A. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. J. Davison, and A. Fitzgibbon. KinectFusion: Real-Time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *Proceedings of ACM Symposium on User Interface Software and Technolog (UIST)*, 2011. 1

[11] 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*, I3D '07, pages 39–46, New York, NY, USA, 2007. ACM. 3

[12] H. Li, B. Adams, L. J. Guibas, and M. Pauly. Robust Single-view Geometry and Motion Reconstruction. *ACM Trans. Graph.*, 28(5):175:1–175:10, Dec. 2009. 2

[13] H. Li, L. Luo, D. Vlasic, P. Peers, J. Popović, M. Pauly, and S. Rusinkiewicz. Temporally Coherent Completion of Dynamic Shapes. *ACM Transactions on Graphics*, 31(1), January 2012. 2

[14] H. Li, R. W. Sumner, and M. Pauly. Global Correspondence Optimization for Non-Rigid Registration of Depth Scans. *Computer Graphics Forum (Proc. SGP'08)*, 27(5), July 2008. 2

[15] H. Li, E. Vouga, A. Gudym, L. Luo, J. T. Barron, and G. Gusev. 3D Self-Portraits. *ACM Transactions on Graphics (Proceedings SIGGRAPH Asia 2013)*, 32(6), November 2013. 2

[16] H. Li, J. Yu, Y. Ye, and C. Bregler. Realtime Facial Animation with On-the-fly Correctives. *ACM Transactions on Graphics*, 32(4), July 2013. 2

[17] N. J. Mitra, S. Flöry, M. Ovsjanikov, N. Gelfand, L. Guibas, and H. Pottmann. Dynamic Geometry Registration. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 173–182, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. 2

[18] 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 International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011. 1, 4

[19] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 32(6):169, 2013. 1

[20] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. In *BMVC*. BMVA, 2011. 2

[21] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and Robust Hand Tracking from Depth. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1106–1113, June 2014. 2

[22] H. Roth and M. Vona. Moving Volume KinectFusion. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2012. 1

[23] T. Schmidt, R. Newcombe, and D. Fox. DART: Dense Articulated Real-Time Tracking. *Proceedings of Robotics: Science and Systems, Berkeley, USA*, 2014. 2

[24] A. Sharf, D. A. Alcantara, T. Lewiner, C. Greif, A. Sheffer, N. Amenta, and D. Cohen-Or. Space-time surface reconstruction using incompressible flow. *ACM Trans. Graph.*, 27(5):110, 2008. 2

[25] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. *ACM Trans. Graph.*, 26(3):80, 2007. 2, 4, 6

[26] J. Süssmuth, M. Winter, and G. Greiner. Reconstructing Animated Meshes from Time-varying Point Clouds. In *Proceedings of the Symposium on Geometry Processing*, SGP '08, pages 1469–1476, Aire-la-Ville, Switzerland, Switzerland, 2008. Eurographics Association. 2

[27] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The Vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 103–110. IEEE, 2012. 2

[28] A. Tevs, A. Berner, M. Wand, I. Ihrke, M. Bokeloh, J. Kerber, and H.-P. Seidel. Animation Cartography&Mdash;Intrinsic Reconstruction of Shape and Motion. *ACM Trans. Graph.*, 31(2):12:1–12:15, Apr. 2012. 2

[29] M. Wand, B. Adams, M. Ovsjanikov, A. Berner, M. Bokeloh, P. Jenke, L. Guibas, H.-P. Seidel, and A. Schilling. Efficient Reconstruction of Nonrigid Shape and Motion from Real-time 3D Scanner Data. *ACM Trans. Graph.*, 28(2):15:1–15:15, May 2009. 2

[30] M. Wand, P. Jenke, Q. Huang, M. Bokeloh, L. Guibas, and A. Schilling. Reconstruction of Deforming Geometry from Time-varying Point Clouds. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 49–58, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association. 2

[31] T. Whelan, M. Kaess, J. Leonard, and J. McDonald. Deformation-based loop closure for large scale dense RGB-D SLAM. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 548–555, Nov 2013. 1

[32] T. Whelan, J. McDonald, M. Kaess, M. Fallon, H. Johannsson, and J. J. Leonard. Kintinuous: Spatially Extended KinectFusion. In *Workshop on RGB-D: Advanced Reasoning with Depth Cameras, in conjunction with Robotics: Science and Systems*, 2012. 1

[33] M. Ye and R. Yang. Real-Time Simultaneous Pose and Shape Estimation for Articulated Objects Using a Single Depth Camera. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2353–2360, June 2014. 2

[34] M. Zeng, F. Zhao, J. Zheng, and X. Liu. A Memory-Efficient KinectFusion Using Octree. In *Computational Visual Media*, volume 7633, pages 234–241. 2012. 1

[35] M. Zeng, J. Zheng, X. Cheng, and X. Liu. Templateless Quasi-rigid Shape Modeling with Implicit Loop-Closure. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 145–152, June 2013. 2

[36] Q. Zhang, B. Fu, M. Ye, and R. Yang. Quality Dynamic Human Body Modeling Using a Single Low-cost Depth Camera. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2

[37] 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. Graph.*, 33(4), July 2014. 2, 6