Animation Cartography - Intrinsic Reconstruction of Shape and Motion

ART TEVS, ALEXANDER BERNER, MICHAEL WAND, IVO IHRKE, MARTIN BOKELOH, JENS KERBER, HANS-PETER SEIDEL
Max-Planck Institut Informatik, and Saarland University, Saarbrücken, Germany

In this paper, we consider the problem of animation reconstruction, i.e., the reconstruction of shape and motion of a deformable object from dynamic 3D scanner data, without using user provided template models. Unlike previous work that addressed this problem, we do not rely on locally convergent optimization but present a system that can handle fast motion, temporally disrupted input, and can correctly match objects that disappear for extended time periods in acquisition holes due to occlusion. Our approach is motivated by cartography: We first estimate a few landmark correspondences, which are extended to a dense matching and then used to reconstruct geometry and motion. We propose a number of algorithmic building blocks: a scheme for tracking landmarks in temporally coherent and incoherent data, an algorithm for robust estimation of dense correspondences under topological noise, and the integration of local matching techniques to refine the result. We describe and evaluate the individual components and propose a complete animation reconstruction pipeline based on these ideas. We evaluate our method on a number of standard benchmark data sets and show that we can obtain correct reconstructions in situations where other techniques fail completely or require additional user guidance such as a template model.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism—*Animation*; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis—*Surface Fitting*

Additional Key Words and Phrases: registration, animation reconstruction, dynamic 3D scanners

ACM Reference Format:

Tevs, A., Berner, A., Wand, M., Ihrke, I., Bokeloh, M., Kerber, J., and Seidel, H.-P. 2011. Animation Cartography - Intrinsic Reconstruction of Shape and Motion. ACM Trans. Graph. xx, x, Article xx (Month 2011), 15 pages. DOI = 10.1145/xxxx.xxxx
http://doi.acm.org/10.1145/xxxxx.xxxxx

1. INTRODUCTION

Recently, a number of techniques have been proposed to scan three-dimensional moving objects in real-time [Würmlin et al. 2002; Zhang et al. 2004; Zitnick et al. 2004; Davis et al. 2005; Weise et al. 2007; König and Gumhold 2008; Vlasic et al. 2009; Bradley et al. 2010]. The output of such an acquisition process is a sequence of unstructured point clouds. The measurement process does not provide any correspondence information and usually only shows a limited part of the object at a time, due to occlusions. This introduces a new problem, the problem of *animation reconstruction*: How can we reconstruct the shape and the motion of a deformable object given that only parts of it can be seen at any given point in time? More precisely, we want to reconstruct the full shape out of the partial observations and establish dense correspondences over time that describe the motion of the object.

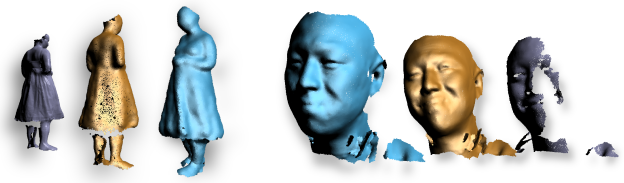


Fig. 1. Animation cartography recovers template model (blue) as well as its motion over time (yellow) from dynamic point cloud data with holes and topological noise (gray).

Some techniques have recently been proposed to solve this problem [Mitra et al. 2007; Wand et al. 2007; Pekelny and Gotsman 2008; Süßmuth et al. 2008; Wand et al. 2009]. However, these approaches employ local numerical optimization to align parts of the object incrementally: The final shape is inferred by a deformable alignment of the geometry in time sequence order. If some of the alignments yield an incorrect result, neither the shape of the deformable object nor the correspondences are reconstructed correctly. In practice, alignment problems are frequently observed. They are caused by fast object movement or vanishing geometry that reappears in later frames in a different pose. Local alignment is not able to handle these situations correctly. The problem obviously becomes much easier if the user provides additional information, such as a template model [Carranza et al. 2003; Sand et al. 2003; Anuar and Guskov 2004; Zhang et al. 2004; Park and Hodgins 2006; de Aguiar et al. 2008; Li et al. 2009]. Nevertheless, numerical tracking can still fail so that manual user intervention becomes necessary. Furthermore, the fixed template restricts the expressiveness of the model, prohibits topological changes, and makes an acquisition of general scenes tedious.

We present a template-free technique (Figure 1) that is able to assemble shapes from partial scans more robustly and under general motion than previous methods. The main idea is motivated by cartography (from which we derive the name of the approach, *animation cartography*). We first track the location of a few landmark points, which we subsequently use to compute dense correspondences, assuming that the deformation of the object is approximately isometric. The output of the algorithm is a *chart* that covers the complete original object. It encodes the intrinsic structure of the reconstructed manifold and dense correspondences to the data points. This intrinsic reconstruction does not yet provide concrete geometry. Therefore, we combine the intrinsic manifold charting with a state-of-the-art extrinsic reconstruction scheme [Wand et al. 2009] that computes actual geometry. By initializing this local nu-

merical optimization scheme with charted correspondences, we obtain much more reliable results.

In order to perform the charting, a number of algorithmic building blocks are necessary, each of which is a novel contribution of this paper: First, we propose a scheme to track salient landmark points. The algorithm automatically detects temporal discontinuities and resorts to a global feature matching algorithm to provide landmark correspondences also in general settings. The second component is the intrinsic charting algorithm that extends the sparse landmark correspondences to dense matches and stitches together partially overlapping charts. Finally, we design a matching pipeline that iteratively performs tracking and chart merging to chart animation sequences. A key challenge in all three steps is that we have to deal with partial data, due to occlusion artifacts. Therefore, intrinsic distances are not reliable. Similarly, the apparent topology of the input data might change, for example if a person temporarily rests his hands touching the body. We account for these problems by employing a novel robust matching model, which can handle such *topological noise* and furthermore quantify the uncertainty under noisy input.

We describe and evaluate the separate building blocks of the algorithm as well as a complete animation reconstruction pipeline that is composed of these components. In experiments with well known benchmark data sets, we show that the new reconstruction pipeline can handle more general input data than previous work.

In summary, the main contributions of this paper are:

—A matching model that is robust to geometric and topological noise and that can quantify the matching uncertainty.

—A landmark tracking algorithm that establishes sparse correspondences fully automatically under both temporally coherent as well as arbitrary, abrupt motion.

—A charting algorithm that computes dense correspondences from sparse landmark tracks, thereby assembling multiple partial charts into one common reconstruction.

—Finally, a complete animation reconstruction pipeline that is significantly more robust than previous techniques. In particular, it can, for the first time, handle abrupt motion and occluded objects that reappear in very different pose without user input.

2. RELATED WORK

In this section, we review previous work related to our approach, in particular techniques for animation reconstruction and global deformable matching.

Animation Reconstruction is the process of recovering the motion of a deformable object from time-varying three-dimensional scanner data, typically point clouds. There are a number of previous methods that require the user to provide a template model that is subsequently deformed in order to match the acquired data [Carranza et al. 2003; Sand et al. 2003; Anuar and Guskov 2004; Zhang et al. 2004; Park and Hodgins 2006; de Aguiar et al. 2008; Li et al. 2009; Bradley et al. 2010].

More recently, a number of template-free techniques have been examined. Mitra et al. [2007] perform rigid alignment between frames, assuming rather slow motion with little local deformation. The technique is elegant and very fast but cannot handle general sequences with missing data and substantial inter-frame deformation. Wand et al. [2007] use deformable matching and a statistically motivated global optimization scheme. The considerable computation costs have been addressed more recently in [Wand et al. 2009] by employing a subspace deformation technique. The technique is able to compute complete template models from partial input data but, as a local optimization technique, it is sensitive to the issues mentioned in the introduction such as large time steps, temporarily disappearing objects, and fragmented frames. We later demonstrate that the technique developed in this paper is significantly more robust in comparison to their previous approach. Very recently, [Popa et al. 2010] propose an improved template-free reconstruction method based on optical flow and cross-parametrization. However, their technique cannot handle fast motion and requires video input for 2D feature tracking (such as a passive stereo acquisition systems).

Li et al. [2009] use a more efficient subspace deformation technique in combination with detail transfer, which was previously examined by Bickel et al. [2007] for the case of wrinkles, to obtain very good results, however, requiring a template model as input. A combination of deformable matching with Mitra et al.'s algorithm is examined in Süßmuth et al. [2008]. Their work however relies on having a complete shape in the first frame, again not improving on the issue of assembling the template model from partial data.

Comparable approaches have also been examined with different regularizing assumptions: Pekelny and Gotsman [2008] use an articulated piecewise rigid model, segmented by the user, Sharf et al. [2008] examine volume and momentum preservation as an alternative. Both are still local optimization techniques, subject to the according limitations.

Global Deformable Matching considers the problem of aligning exactly two deformable shapes. The technique of Li et al. [2008] is based on local matching but increases its robustness by modeling correspondences explicitly as latent variables and optimizing over them. Chang and Zwicker [2008; 2009] provide global and robustified local matching strategies for articulated, piecewise rigid models. Bradley et al. [2008] propose a technique specifically designed for garment capture that uses specific properties of such data sets to control the boundary conditions of a cross parameterization algorithm, thus establishing correct correspondences. Anguelov et al. [2004] introduce intrinsic distances as validation criterion for matching feature points on deformable manifolds. The resulting quadratic assignment problem is solved using Bayesian belief propagation. A similar approach based on dense feature points is proposed by Starck and Hilton [2007]. Leordeanu et al. [2005] propose a simpler technique based on spectral relaxation for solving quadratic feature assignment problems, which has been employed for isometric matching by Huang et al. [2008] and Ahmed et al. [2008]. The two papers introduce landmark coordinates for deriving dense matches from the coarse matches returned by spectral matching, a concept that has previously been invented in the context of routing in sensor networks [Fang et al. 2005].

A problem with all these intrinsic matching strategies is topological noise: Acquisition holes as well as apparent topology changes, such as a closing mouth in a face scan, might strongly distort intrinsic distances such that correspondences cannot be detected reliably. Bronstein et al. [2009] address this problem by mixing geodesic and Euclidean distances, but this solves the problem only in some cases. In general, both extrinsic and intrinsic distances might be very different. In addition, the technique is based on local numerical optimization, which requires pre-alignment of the data. More

recently, Bronstein et al. [2010] propose diffusion distances, which are more robust to a certain amount of topological noise, as this distance measure is sensitive to the cross section of interconnections rather than just the reachability in the case of geodesic distances. However, large scale artifacts such as big acquisition holes or false connections in a large area also change diffusion distances significantly. Unfortunately, these problems are common in our application area (large acquisition holes, arms at the body, closing mouth in a face scan, etc).

Tevs et al. [2009] address this problem in a different way by allowing for outlier geodesic distances within a RANSAC algorithm. This approach requires a minimum set of "witness geodesics" but otherwise does not depend on the geometric extent of topological noise. We put their technique on a sound statistical basis that allows for explicitly calculating matching uncertainties. More importantly, [Tevs et al. 2009] are limited to pairwise matching while we address the more general problem of simultaneously reconstructing 3D topology and correspondences over long sequences. The reconstruction of many-frame correspondences is a non-trivial generalization: Just repeatedly performing pairwise matches exponentially increases the failure probability of randomized matching, thus rendering merging of long sequences practically impossible. We avoid this pitfall by on the one hand using continuous tracking algorithm to detect and use local temporal coherence and on the other hand by explicitly assessing the matching quality, avoiding the incorporation of ambiguous information in the result.

Global Animation Reconstruction refers to methods that aim at incorporating more information than pairwise matching can provide into the reconstruction process. The previously cited global registration methods only consider, with the exception of Ahmed et al. [2008] and Varanasi et al. [2008], the pairwise matching case. Their methods are based on matching feature distances computed by Laplacian diffusion which is not robust against general topological noise. In addition, both methods require color information associated with the point cloud data for matching SIFT/SURF features and thus cannot be applied to purely geometric data sets. The same holds for the technique of Liao et al. [2009], which also does not address the problem of handling temporal gaps or jumps in motion where continuous tracking breaks down. The method of [Popa et al. 2010] also requires image information. It performs intrinsic cross-parametrization, similar to our approach, but has to assume a "gradual change" prior (rather than robust matching densities) to resolve ambiguities, therefore not yet providing a full global animation reconstruction. A very interesting, recent approach by Zheng et al. [2010] aims at reconstructing the temporal correspondences of a skeleton rather than complete geometry, which can then subsequently be used as guidance information for shape alignment. The drawback in comparison to our approach is that although skeletonization improves robustness, it does not represent the full correspondence information, which cannot always be recovered reliably.

3. OVERVIEW

We start with an introduction of concepts that are used throughout the rest of the paper: We first formally define the problem that we are solving and describe the input data we are expecting (Subsection 3.1). Next, we describe the data structures that we use to represent charts, and how intrinsic and extrinsic information is represented (Subsection 3.2). Afterwards, we describe the robust matching model that is used throughout the paper (Subsection 3.3). Finally, we conclude this section with an overview of the individual reconstruction steps and how they are combined in the final reconstruction pipeline (Subsection 3.4).

3.1 Problem Statement

Original animation: The goal of our method is to reconstruct a manifold and its motion from partial observations. Formally, we assume that there has been an original differentiable 2-manifold $\mathcal{M} \subset \mathbb{R}^3$ that underwent a time-variant motion $f_t : \mathcal{M} \to \mathbb{R}^3$. $t \in [1, T]$ is the time parameter. Each f_t is assumed to be injective and differentiable, and each $f_t(\mathcal{M})$ denotes a deformed version of the original manifold $\mathcal{M}$.

Isometry assumption: We equip differentiable manifolds $\mathcal{M} \subset \mathbb{R}^3$ with an intrinsic metric $d_{\mathcal{M}}(\cdot, \cdot)$ that measures the shortest geodesic distance between pairs of points. We assume that the deformation f_t is approximately isometric for each fixed t. This means that

$$\forall t \in \{1..T\} : d_{\mathcal{M}}(\mathbf{x}, \mathbf{y}) = d_{f_t(\mathcal{M})}\left(f_t(\mathbf{x}), f_t(\mathbf{y})\right) + \eta_{\sigma_f}, \quad (1)$$

where $\eta_{\sigma_f} \sim N(0, \sigma_f)$ is an error that is normal distributed with standard deviation σ_f and mean zero. In other words, we assume that the original deformation, even before measurement, has not been perfectly isometric but that there might have been errors that are in the range of σ_f.

Discussion: Assuming (approximate) isometry is an established model [Anguelov et al. 2004; Bronstein et al. 2006; Bradley et al. 2008]. It is sufficiently general to characterize the motion of the surface of many real-world objects, such as scans of people, animals, plants, or clothing. A strongly non-isometric surface deformation would be fatal to such objects. Nevertheless, intrinsic isometry poses a strong constraint on the interpretation of observed data; one can think of a rigidity assumption within the manifold (rather than within the embedding space). Consequently, isometries have only very few degrees of freedom once the manifold they act upon is known [Lipman and Funkhouser 2009; Ovsjanikov et al. 2010; Tevs et al. 2011].

Measurement: A 3D scanner only yields a partial, sampled representation. We assume that the scanner operates at regular time steps $t \in \{1, 2, ..., T\}$ and for each time step, yields a finite set of sample points $D_t \subset \mathbb{R}^3$. We denote the individual points by $\mathbf{d}_t^{(i)}, i = 1, ..., n_t$ and the collection of all input data by just D. To simplify further processing, we assume that parts of objects that have actually been acquired have been sampled with a sample spacing of at most ϵ_s, i.e., for each point of the original surface, there is a sample point in Euclidean distance of at most ϵ_s. Areas with lower sampling density are discarded during preprocessing. Furthermore, we assume that all of $\mathcal{M}$ at some point has been observed with sufficient sampling density (or equivalently, we only try to reconstruct what we have observed).

Reconstruction Tasks: We consider two reconstruction tasks: A *full geometric reconstruction* and the *reconstruction of a chart* of the data. The full geometric reconstruction is the ultimate goal: We want to reconstruct $\mathcal{M}$ and f. Because of acquisition holes, this involves an interpolation of f in areas of missing data. Wand et al. [2007; 2009] propose a variational model that can find plausible interpolations by employing physically motivated prior assumptions on shape and motion. However, their model is non-linear and non-convex and cannot be globally optimized.

In order to compute a suitable initialization for such methods, we propose to perform a simpler reconstruction task first, the reconstruction of a chart of the data. Here, we only reconstruct the shape $\mathcal{M}$ (up to isometries) and correspondences between $\mathcal{M}$ and (most of the) data points D. This means, we either encode for each data point $\mathbf{d}_t^{(i)} \in D$ its preimage $f^{-1}(\mathbf{d}_t^{(i)}) \in \mathcal{M}$, or mark it as unknown, in case the reconstruction was not able to interpret the data point. We can then use this shape and the correspondences to the data points as fixed boundary conditions to stabilize a locally convergent reconstruction. For efficiency, our actual pipeline will not compute explicit correspondences for each single data point but rather use a coarse cloud of correspondence samples that covers the data points in order to encode the correspondence information, as detailed in the next subsection.

3.2 Data Structures

In this subsection, we explain the data structures that we employ to represent the objects defined above.

Sampled manifolds, intrinsic view: From the point of view of intrinsic geometry, we look at manifolds $\mathcal{M}$ simply as metric spaces, i.e., a set of points with a distance measure that determines geodesic distances of pairs of points. We represent these objects as graphs of points: We cover a manifold $\mathcal{M}$ with a finite ϵ_s-sampling $M = \{m_1, ..., m_{n_M}\} \subset \mathcal{M}$. This means that for every point of the original $\mathcal{M}$, one point in M exists at geodesic distance of at most ϵ_s. Furthermore, we build a graph $G = (M, E)$ to approximately encode the metric of $\mathcal{M}$. We include an edge $e \in E$ between points $m_i, m_j \in M$ whenever m_j is among the k-nearest neighbors of m_i or vice versa (in practice, we use $k = 20$). Furthermore, we annotate the edge with this intrinsic distance. The graph distance $d_M(\cdot, \cdot)$ (i.e., the shortest path in the graph) between two arbitrary points will then serve as an approximation of the original geodesic distance $d_{\mathcal{M}}(\cdot, \cdot)$.

Discussion: Obviously, the discrete approximation will distort the distance measure. Smooth geodesics are approximated by zig-zag paths in graphs, which introduces systematic deviations. However, we use this representation consistently for data and all (partial) reconstructions. The systematic errors therefore affect all geodesic paths in the same way so that they remain directly comparable, which is sufficient for our application. Nevertheless, the discretization also causes additional quantization noise in distance estimates. Accordingly, we adapt the expected error of intrinsic distances σ_f to be at least in the range of ϵ_s (in practice, we use $3\epsilon_s$). Hence, σ_f in the following describes the magnitude of both modeling as well as representation noise.

Sampled manifolds, extrinsic view: Sometimes, we want to be able to give an embedding of a sampled manifold M in $\mathbb{R}^3$. This is trivial to encode - we just store for each graph node $m_i \in M$ an additional position vector $\mathbf{x}_0(m_i) \in \mathbb{R}^3$. Following [Wand et al. 2009], we call this embedding of the chart an *urshape*. We denote the urshape of M by $X_0(M)$. Please note that urshapes are not unique but any isometric deformation $f(X_0(M))$ is again a valid urshape.

Charts: A chart combines a (partial) reconstruction of a manifold with correspondences to data points. This means, a chart is a sampled manifold M, and for each $m_i \in M$ we store a list of 3D positions of where node m_i would be located in each data frame. We denote these positions by $\mathbf{x}_t(m_i) \in \mathbb{R}^3$, where t covers a non-empty subset of time steps $t \in 1, .., T$. If the embedding is unknown at a specific time t, we mark $\mathbf{x}_t(m_i)$ as unknown.

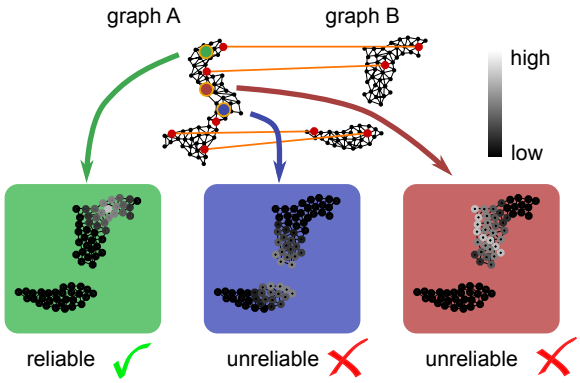


Fig. 2. Matching probabilities (schematic drawing): For the green point (left) the variance is low so that the match is accepted as reliable. The matching point for the blue point would be located in the hole of chart B, leading to a high variance that indicates an unreliable match (for large holes, a uniform distribution remains). The red point finally, has a proper neighborhood, however, the landmark coordinates do not constrain the match sufficiently. Again, this leads to a high variance and the match is detected to be not reliable.

Discussion: The definition of a chart has been chosen to account for a later technical problem: To limit computational costs, we will not be able to include every data point into the chart. Therefore, we allow for using a coarse set of points to represent M and store correspondences implicitly, by storing 3D positions for $m_i \in M$. Each m_i will form (partial) tracks that move over time but, in general, will not exactly coincide with data points but rather cover the data points. Please also note that a chart does not necessarily have a full geometric embedding, as the temporal coverage might be sparse and different at every node. However, as we will see later, our final pipeline will actually maintain a fully embedded urshape $X_0(M)$ for every chart in order to interface with the extrinsic reconstruction.

Landmark coordinates: Some of the chart points are *landmark points*. These points are special as they correspond to features of the input data that we were able to recognize and track over time. As any other chart point, the spatial location of landmarks might not be known for the full time sequence $t = 1..T$ but only for a (non-empty) subset. Given a set of landmarks $L = \{l_1, .., l_n\} \subseteq M$, we define the *landmark coordinates*[1] $\mathbf{d}_L(m)$ of an arbitrary node $m \in M$ as the vector of intrinsic distances between m and the landmark points:

$$\mathbf{d}_M^{(L)}(m) = [d_M(l_1, m), .., d_M(l_n, m)]^T \qquad (2)$$

Discussion: The main idea of our algorithm is that if two charts share a number of landmarks, we can compute dense correspon-

[1]To be precise, there is a difference between distances and coordinates: distances are non-linear functions of coordinates. [Fang et al. 2005] show how this non-linearity can be removed for developable surfaces, but we are not aware of such a solution for general 2-manifolds. However, in our application, the non-linearity is not an issues as we only need to test for the likelihood of equality rather than compute routing paths as in their original paper. We therefore stick to the simple but slightly imprecise notion of calling the vectors of distances "landmark coordinates".

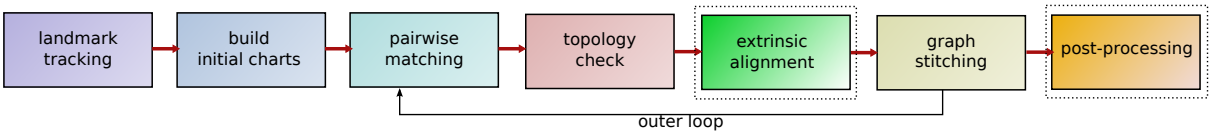


Fig. 3. Overview of the animation cartography pipeline. Landmark tracks, Algorithm 4.1.1, are used to build initial charts (i-charts), Algorithm 4.2.1, which are recursively combined in a pairwise chart merging loop, Algorithm 4.2.2-4.3.2. The final result is used to initialize a numerical bundle adjustment algorithm for post-processing. Dotted blocks indicate extrinsic matching components based on previous work of [Wand et al. 2009]. A more detailed description of how each of the part fits in the overall approach is given in Section 5.

dences for the remaining chart points by comparing landmark coordinates. The main challenge is to do this in a way that is robust to topological and geometric noise, unlike previous work [Fang et al. 2005; Ahmed et al. 2008; Huang et al. 2008]. For this, we introduce a robust probabilistic matching model in the next subsection.

3.3 Robust Intrinsic Matching

A central problem of our approach is to determine correspondences between points from different charts. Let M_A and M_B be two charts that share a set L of landmarks for which we know the correspondences. We now want to compute where a point $a \in M_A$ could correspond to in M_B. For this, we compute a probability distribution over all points of M_B that quantifies the likelihood of $a \in M_A$ matching a point $b \in M_B$ (denoted as $a \sim b$):

$$\Pr(a \sim b|L) = \frac{1}{Z} \prod_{j=1}^{|L|} \left(\lambda \cdot e^{-\frac{\left(\mathbf{d}^{(L)}_{M_A}(a)[j] - \mathbf{d}^{(L)}_{M_B}(b)[j]\right)^2}{2\sigma_f^2}} + (1-\lambda) \right), \tag{3}$$

$\mathbf{d}^{(L)}_{M_A}(a)[j]$ and $\mathbf{d}^{(L)}_{M_B}(b)[j]$ are the jth component of the landmark coordinate vector of a and b, respectively, and the term $1/Z$ is just the normalization constant. Equation 3 models the matching problem by considering the geodesic distance to each landmark point independently. For each connection, we assume a normal-distributed error in case that the geodesic is correct. However, it may happen that acquisition holes or pseudo connections ("closing mouth") distort the geodesics such that the distance is arbitrarily wrong. In this case, we do not have any information about the correct distance so that we resort to a uniform distribution. The parameter λ is the probability for geodesics being correct. We use a global constant failure probability of 10%, i.e., $\lambda = 0.9$.

Improvement: In practice, we can make the model more robust by limiting the product to take into account only nearby landmark points (in a geodesic sense) for each model point. In practice, we use the 5 nearest landmarks. Limiting the influence helps because the likelihood of geodesic path being distorted increases with distance to the point considered.

Discussion: This model is similar to the robust RANSAC approach by Tevs et al. [2009], where only the k-best geodesics are considered for matching. However, our new model provides some important improvements: It provides a continuous probability density that describes the likelihood of matching point a on M_B. If landmark points are chosen in a good configuration, the density is more sharply peaked than for landmark points in a bad configuration (with geodesics almost parallel, for example). The probability density does not only encode the maximum likelihood match, but we have the complete distribution that quantifies the uncertainty. In particular, we examine the variance of $\Pr(a \sim b|L)$ w.r.t. to $\mathbf{x}_t(b)$ in order to determine how *certain* a match is. If the variance is high, the match is not reliable. Please note that the variance automatically increases if the outlier probability $1-\lambda$ increases. In this case, more correct landmark matches are required to reduce the variance again (the the uniform density "floor" of the distribution converges to zero with $(1-\lambda)^{|L|}$). Furthermore, if the error in the normal distribution is large, combining multiple landmark correspondences reduces the variance because multiplying the Gaussians will lead to a more peaked distribution. Another important improvement over the previous model is that we do not need to fix a constant k of reliable geodesics but we can use the more natural formulation that geodesics have a certain failure probability; the resulting uncertainty is automatically taken into account, including the case that even some of the k best matching geodesics could be wrong.

3.4 Pipeline Overview

The full animation reconstruction pipeline consists of a number of components. We will discuss each individual component separately in the next Section (Section 4) and the composition of the full pipeline afterwards, in Section 5. Here, we give a brief overview for orientation (see also Figure 3).

The reconstruction starts by *landmark tracking*. In this step, the input data is examined for feature regions and a KLT-like tracking scheme [Lucas and Kanade 1981], adapted to 3D geometry, is used to find landmark tracks. Out of these tracks, initial charts are built, which we refer to in short as *i-charts*. Our algorithm usually extracts a number of such i-charts that end when tracks discontinue due to abrupt motion or occlusion. Therefore, the next step matches disconnected i-charts to general partial charts of the animation, which we refer to in short as *p-charts*. The matching step involves an additional topology check to remove incorrect edges from the sampled manifolds and an extrinsic refinement step to improve the matching precision, followed by a graph stitching step that connects the two manifold representations (charts). This scheme is iterated in an outer loop until a full chart of the complete animation is obtained. Finally, we input the full chart as boundary conditions in a standard numerical optimization to obtain the final results; we use the method of [Wand et al. 2009].

4. ALGORITHMIC BUILDING BLOCKS

This section discusses the individual algorithmic primitives that are used in our reconstruction pipeline. We opt for an isolated discussion for two reasons: First, it makes it easier to structure the rather complex reconstruction system. Second, several of the individual components might be useful as algorithmic primitives in other ge-

ometry processing contexts so that it is valuable to look at them separately.

We structure the building blocks in three parts: Landmark tracking related algorithms (Subsection 4.1), intrinsic charting algorithms (Subsection 4.2) and extrinsic matching techniques (Subsection 4.3).

4.1 Landmarks

We start by discussing the concept of landmarks and their tracking. Landmarks are the key concept for solving the reconstruction problem because they allow us to characterize dense correspondences between surfaces by fixing only a small number of landmark correspondences. This reduces the combinatorial complexity of the matching problem to a level that makes the reconstruction feasible.

Algorithm 4.1.1: Continuous Landmark Tracking

Input: Temporal sequence of data points D
Output: Set of landmark tracks L. Each of these tracks is a smoothly moving feature.

The first component is a tracker for *continuous* landmark tracks. It gets the raw data D from the scanner as input. The task is to (1) identify feature regions, (2) track features over time, and (3) recognize when tracks end due to incoherent motion.

We solve the first problem (1) by running slippage analysis [Gelfand and Guibas 2004]. It looks at every frame D_t of the data and determines for each point $\mathbf{d}_t^{(i)}$ whether a region of radius r around $\mathbf{d}_t^{(i)}$ can be stably aligned to itself under a rigid motion (in practice, we use r = 10% of the bounding box size of the object). For flat areas, for example, the alignment is unstable because the patch could just slip along the plane. We keep only the unslippable regions and perform a coarse r-sampling to distribute *feature points* uniformely. Again, we use a Poisson-disc algorithm to obtain a good uniform distribution.

The main tracking step (2) is performed by simple rigid ICP (Figure 4): We extract the r-neighborhood of each feature point and align it to the next frame using point-to-plane ICP, always initialized to the (known) position of the previous frame. If the algorithm converges, we align the same geometry again to the next frame, and iterate until the alignment diverges. The landmark track is given by the trajectory of the center of the aligned region (the feature point) over time. Divergence is determined (3) by not converging to a fixed point within 32 iterations or by a translational motion by more than r within one frame (which is likely to be wrong, because there was no initial overlap of the geometry with the new target).

We start new tracks automatically: For each new frame, we recompute the non-slippable regions and insert a new point whenever it is not r-covered with feature points that are being tracked.

Discussion: This scheme could be considered a geometric variant of the well-known KLT feature tracker for images [Lucas and Kanade 1981]. It works quite effectively in our situation because scanned data usually contains a lot of coherent motion (but not everywhere) with small motion between frames. Locally, within a small spatial and temporal environment, the motion is usually almost rigid. Our scheme does not lead to perfect results but might create both false negatives and positives, which have to be handled by the robust matching model.

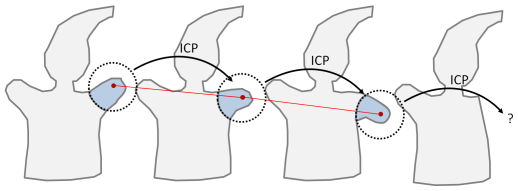


Fig. 4. We align small patches of points to successive frames via ICP to generate tracks. A track is stopped when ICP fails to compute a stable result.

Algorithm 4.1.2: Connecting Broken Tracks

Input: Two points clouds $A, B \subset \mathbb{R}^3$, feature points $F_A \in A$. Optionally: seed correspondences L between A and B.
Output: Correspondences between all $a \in F_A$ and points in B (set to "unknown" if unreliable)

Usually, the tracking algorithm is not able to cover the whole animation sequence but rapid motion or occlusion disrupt some or all of the tracks. Therefore, we need an algorithm to reconnect broken tracks.

We consider two point clouds $A, B \subset \mathbb{R}^3$. They might already have a small number of landmark tracks L in common, but the set L can be empty. If a few continuous tracks are present, we include these as initialization, so that the algorithm finds the correct solution more quickly and more reliably. We now form candidate landmark correspondences by connecting each landmark node of A to every other node in B (here: landmarks and ordinary nodes). From this set, we have to find a consistent subset. We employ a forward search algorithm based on our robust matching scores, extending the RANSAC-like algorithm of [Tevs et al. 2009]: We tag each candidate correspondence with a descriptor matching score (using local curvature histograms as rigidly invariant descriptors of r-neighborhoods, as in their paper) and select a starting correspondence by importance sampling according to these scores. Given one correspondence, we can select a random second starting point from A and compute for all points in B the likelihood of this match being correct. This probability is given by multiplying the descriptor matching score with the robust distance matching score for the intrinsic distances of all previously selected correspondences, Equation 3. We then draw the next correspondence using importance sampling according to this compond density and iterate the process until no more candidates are found that have a low variance in the resulting probability density, indicating that no more reliable matches can be found. We switch from probabilistic sampling to choosing only the best fit (highest density) after 3 matches have been fixed to improve the convergence speed. The whole forward search/RANSAC loop is iterated multiple times (typically, 100 trials), and the result with the largest number of established matches is used as the final result.

Discussion: In principle, we could just always apply this algorithm to find landmark tracks, omitting the continuous tracking phase altogether. However, RANSAC-based matching might fail with a small probability. Therefore, several independent matching operations have success probability that declines exponentially with the number of matches. By making use of temporal coherence, we can make our algorithm substantially more robust, or in other words, dramatically reduce the computational costs (because the number of RANSAC-rounds would have to be increased exponentially to make up for this).

4.2 Intrinsic Charting

We now assume that we know landmark correspondences and turn our attention to the problem of establishing dense correspondences among charts, and subsequently merging these into compound reconstructions. We look at a number of different algorithmic steps: Creating a single frame chart from scratch (as initialization), merging two charts given landmarks, and checking the topology of merged charts. Afterwards, we use these more elementary algorithms to formulate higher level algorithms that build i-charts and p-charts.

Algorithm 4.2.1: Building Single-Frame Charts

Input: Point cloud D_t
Output: Single frame chart M_t

We build initial charts (i.e., just sampled manifolds) for a single frame directly from data D_t: In order to limit the computational costs, we resample D_t to an (Euclidean) sample spacing of ϵ_s, using a Poisson-disc sampler. This yields the vertices of a sampled manifold M_t. Afterwards, we form the graph G_t by building a k-nearest neighbor graph on M_t, with respect to Euclidean distances. We then also use the Euclidean distance of the points as edge length. For a smooth manifold, this is a first-order approximation of the true (but unknown) distances, which is sufficiently accurate for the sampling resolution we employ. Afterwards, we remove all vertices and edges in connected components with fewer than 100 points in order to delete small outliers patches and undersampled data.

Algorithm 4.2.2: Merging Two Charts Given Landmarks

Input: Charts M_A, M_B
Output: A single chart of $M_A \cup M_B$

Let us assume that we have two charts M_A and M_B and a set of landmarks L that the two charts have in common. Our goal is now to compute dense correspondences and then stitch together the charts accordingly to form a single sampled manifold.

Probabilistic Correspondences: We go through all points of $a \in M_A$ and compute a probability distribution $Pr(a \sim b|L)$ for all points $b \in M_B$ according to Equation 3. If the landmarks are placed well to constrain the matching point and if redundant landmark coordinates are all consistent, a single narrow peak indicates the expected position. If only a small number of inconsistent distances are present, this scheme still leads to one pronounced maximum. In case of insufficient or completely inconsistent information, we obtain a spread-out distribution with high variance, which can be detected (see Figure 2).

Reliability: We use the variance of the distribution of the matching score as a reliability measure for the correctness of a match. We assume that M_B has an extrinsic embedding $X_0(M_B)$ and annotate each point $x_b \in X_0(M_B)$ with the probability $Pr(a \sim b|L)$. Then, we compute the mean and covariance of this distribution in 3D by a PCA analysis. As uncertainty criterion, we look at the largest eigenvalue of the PCA[2] (largest standard deviation). In our implementation, we consider matches unreliable if this value is larger than $3\epsilon_s$. Unreliable correspondences will be excluded from the output.

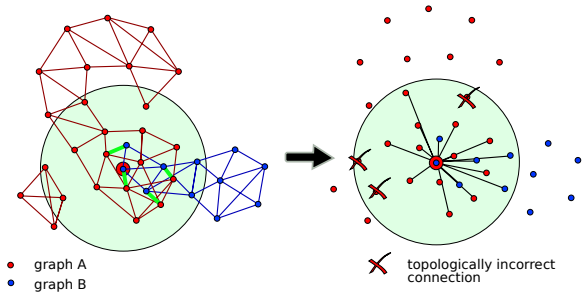


Fig. 5. Graph merging: The blue and the red graph are merged. Points within the $c \cdot \epsilon_s$ search range (light green) are potential neighbors of the center point. We exclude points that are not reachable by walking on the joint red and blue graph without leaving the search range. Nearest neighbor correspondences (green) can be used as "bridges" to walk from blue to red and back.

Improvements: We can further reduce the risk of wrong correspondences if we perform a *bijective consistency check*. Intuitively, we aim at establishing correspondences that are valid either way, whether matching from A to B or vice versa. In our probabilistic framework, this is realized by constructing a probability in graph A that the matched point in B matches back to the region around the original point in A with a high probability. Computationally, we importance sample the matching distribution in B to determine a set of potentially matching points. We then determine their matching probability on A. Assuming statistical independence of the individual potential matches, we multiply their distributions in A to obtain a probability density that the match is bijective. If the original point in A has a high probability of being matched back, we accept the match, otherwise it is discarded.

Point-to-point correspondences: Finally, we need to convert the matching densities to actual point-to-point correspondences. We have two choices for this step: The simplest is the *nearest-neighbor* approach. We just connect $a \in M_A$ to the point $b \in M_B$ that maximizes $Pr(a \sim b|L)$. This is simple and robust but comes with an error of $\mathcal{O}(\epsilon_s)$. The second option is an *extrinsic* approach[3]: We assume that M_A and M_B both have an extrinsic urshape $X_0(M_A)$ and $X_0(M_B)$. We then use the nearest neighbor estimates (first strategy) to initialize an extrinsic optimization that aligns the two urshapes by pairwise deformable matching (see Algorithm 4.3.1). From the urshapes, we recompute a new sampled manifold, as described below:

Graph merging: Having two aligned urshapes, we can easily recompute a new sampled manifold. We just connect each point to its extrinsic k-nearest-neighbors (in a Euclidean sense) in the overlaid urshapes.

Technical details: We need to avoid connecting parts that accidentally have a similar Euclidean position but are actually far away in an intrinsic sense. This can happen because the extrinsic optimization does not perform any collision detection (Subsection 4.3). We therefore do not consider all points as candidates for the k-nearest neighbors but only those that can be reached by a short walk along

[2]The check could also be implemented purely intrinsically by looking at the variance of geodesic distances; however, in our pipeline, an extrinsic urshape will always be available.

[3]In our implementation, we use a combination of the nearest-neighbor and the extrinsic approach; an implementation of the purely intrinsic formulation is still subject to future work.

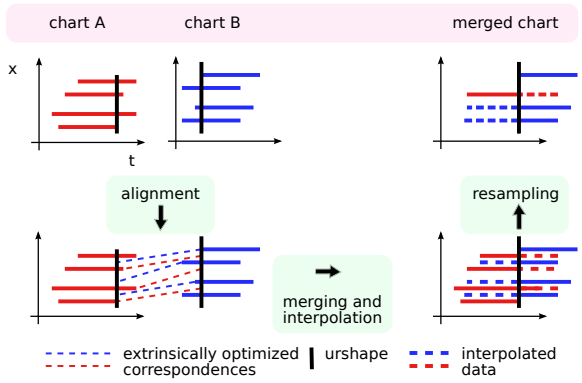


Fig. 6. Resampling merged charts: We first merge two charts A, B (upper left) by nearest neighbor matching (lower left). The match is refined by extrinsic, numerical alignment, which leads to partially represented correspondences (lower right). When we resample the representation, we need to perform neighborhood-based interpolation to retain this information (upper right).

the graph of the sampled manifold: We allow only points within a Euclidean distance $c \cdot \epsilon_s$ (for $k = 20$ we use accordingly $c = 3$), walking on the graph edges of M_B and M_A and using the nearest neighbor correspondences between M_A and M_B as "bridges" to switch between the graphs (see Figure 5 for an illustration).

Discussion: In summary, this algorithmic building block allows us to merge two charts into a single one if we find suitable landmark correspondences. It might fail to recognize corresponding parts if the landmarks are unable to reliably identify the dense matches. However, as part of the output, the algorithm will mark these regions and not provide correspondence information. Furthermore, as described above, the algorithm needs an extrinsic urshape for both charts in order to compute an accurate matching. Otherwise, a nearest neighbors solution is possible but it introduces a small error in each operation so that iterative merging would become inaccurate over time.

Algorithm 4.2.3: Resampling a Chart

Input: A chart M.
Output: A resampled version M' which is a minimal ϵ_s-covering of M

The next operation we need to provide is to reduce the complexity of a chart by resampling. The motivation for this is that we will need to perform many chart merging operations that will constantly increase the sampling density in overlapping parts, which at some point becomes a problem in terms of computational costs.

Resampling itself is very easy: We just use the Poisson-disc sampler to remove nodes from the graph that are still covered by nearby nodes within a distance of no more than $\epsilon_s/2$. The remaining challenge is to maintain the correspondence information between the chart and the actual data. At this point we need to remember that charts encode correspondences by attaching sets of extrinsic positions of points to which they correspond. Therefore, removing chart points deletes valuable correspondence information (see Figure 6 for an illustration of this problem).

We propose again an extrinsic scheme to counter this problem by *interpolation*: We keep the original chart M and chart M' resampled to a sample spacing of ϵ_s. Each $m' \in M'$ is also a node in the original M. We look at all neighbors $N_{\epsilon_s} \subseteq M$ of m' that are located within an (intrinsic) distance of ϵ_s. For each time step t that is covered by the chart, we then retrieve their extrinsic embeddings. If we find at least three such points, we compute a local tangent space by fitting a least squares-optimal rigid alignment $\mathbf{T}$ of the points at time t to the corresponding urshape points [Horn 1987]. We then estimate the correspondence of m' at time t as $\mathbf{T}^{-1}(x_0(m'))$, i.e., by just transforming the urshape point back to the corresponding tangent space.

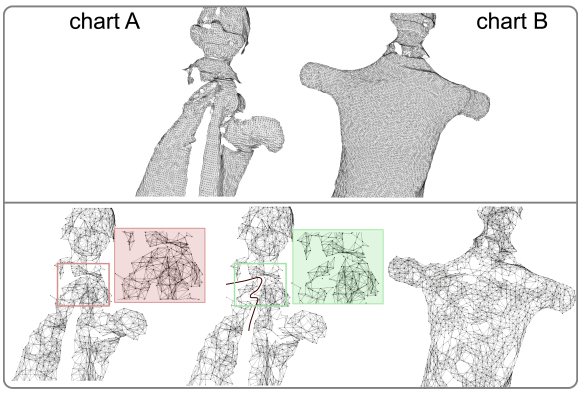


Fig. 7. A difficult case for chart merging: the two charts of the puppet data set have a very different topology. In input data (top row) on the left, the hand of the puppet is merged with the body. On the right, the puppet is fully visible and hand and body are disconnected. The lower row shows the chart topology before performing the topology consistency check (left), after topology clean-up (middle) and for the second chart (right). Note how the hand got disconnected from the body in the left chart while the topology of the chart on the right is unaffected.

Landmarks: A special situation occurs for landmark nodes. Since landmarks carry global matching information that is valid across different charts, these nodes cannot be deleted, moved or interpolated to different positions in the graph. Hence we just copy the landmark nodes into the resulting chart and their position does not need to be interpolated over the temporal support.

Discussion: The scheme performs a first order accurate interpolation which yields satisfying results for the dense sampling we are employing in our implementation. The scheme could easily be implemented intrinsically, without having an extrinsic urshape, using the intrinsic distances as weights for the tangent space approximation but this is not necessary for our pipeline.

Algorithm 4.2.4: Detecting Apparent Topology Changes

Input: A chart M
Output: Augmented chart M' so that intrinsic paths are never shorter than extrinsic paths

If we use the chart merging algorithms described above to assemble a more comprehensive chart from simpler ones, we are still facing a major problem: It might happen that the *apparent topology* of the chart changes, for example if the mouth closes in a face scan. Charts build from closed mouth data have an incorrect metric structure and incorrect topology: They do not show a hole in the mouth region and the distance between the lips is too short. We therefore need to detect this situation and adapt the graph of the chart accordingly.

Invariant: We have only one criterion to detect such mistakes: the intrinsic distance between corresponding points must always be larger or equal to the largest extrinsic distance that has ever been observed. This criterion is used in [Wand et al. 2007] to build a straightforward "edge-stretch" test: It just checks if extrinsic embeddings of points connected by a common edge violate this invariant, and if so, delete the edge. This works in practice but it is not very robust; it requires a delicate trade-off of elasticity and edge-stretch tolerance. We adopt this basic idea and also propose an improved, more robust algorithm.

Basic version: We can implement the basic stretch test easily by just comparing the Euclidean embedding (correspondences to data) of neighboring nodes at all time steps. Because the extrinsic correspondences are stored only sparsely, we also have to resort to interpolation from neighbors at both endpoints (as described in Algorithm 4.2.3) to make this robust.

Improved version: The improved version looks at the problem from a more global perspective: The main idea is to look at shortest intrinsic paths and all of their known temporal correspondences in Euclidean space. If we find a sub-path for which the endpoints are at a larger Euclidean distance than the geodesic distance of the path, we know that a part of this path violates our invariant, hence one or more edges on that path must be deleted. In order to search for those paths, we compute the geodesic paths between all pairs of nodes in the graph and compare the Euclidean and the intrinsic distance for all time steps. If we find a violation (intrinsic distance being too small), we walk inwards along the path until we find the smallest interval that still violates the distance criterion. We also stop shrinking the interval if correspondences are not known. This usually does not yet give us the desired result because the intervals in which the error occurs can be quite large. We therefore perform a voting scheme in order to identify edges which are responsible for the violation. Each edge gets a vote if it shows up in a path that violates the distance criterion. After that, we determine the set of all edges that obtained a maximum number of votes and delete them from the graph. Then we iterate this scheme until no more violating paths are found.

Speedup: Computing all pairs of paths is obviously too slow. We therefore restrict the search to paths of a bounded length and use only a subsample of starting points instead of all points.

Discussion: This strategy is more robust in finding problematic connections than the edge-stretch test. However, due to the greedy deletion algorithm, it might still delete a larger set of edges than absolutely necessary. Due to subsampling, it is also possible to miss smaller topological problems. Nevertheless, our experience is that the improved strategy is significantly more robust than the previous technique. An example of the topological consistency filter on the hand puppet data set [Li et al. 2009] is shown in Fig. 7; the simple edge-stretch test fails here.

4.3 Extrinsic Matching

And finally we take an overview over the extrinsic matching algorithms. This is only a very short summary of the previous work of Wand et al. [2007; 2009]; we refer the reader to the original papers for implementation details. We describe the previous work first so that the difference of our new approach becomes clear. Furthermore, we will use the numerical approach for refinement. This approach is very common in optimization: we first use a coarse global optimization algorithm to estimate a good initialization for a more precise (but not globally optimal) local optimization scheme.

Algorithm 4.3.1: Pairwise Local Matching

Input: Two point clouds $A, B \subset \mathbb{R}^3$
Output: A deformed version $f(A)$ of A that fits the shape of B

The main idea of the extrinsic matching algorithm is to compute a deformation field $f : A \to \mathbb{R}^3$ that minimizes a matching energy:

$$E_{match}(f) = E_{dist}(f(A), B) + E_{elastic}(f) \quad (4)$$

E_{match} combines two energy functions: The first, E_{dist}, measures the distance of point cloud B from the deformed $f(A)$. It sums up the point-to-plane distance between points from $f(A)$ and points from B. In order to support partial matching reliably, a number of heuristics are employed, such as checking the angle of the correspondence vectors to the surface normals. The energy $E_{elastic}$ penalizes the elastic energy of the deformation field f, trying to keep it as-rigid-as possible. In the optimum, minimal bending and stretching is introduced while still matching the data well. The two terms are usually weighted to control the trade-off. We use the elastic subspace matching model of [Wand et al. 2009], but several other choices are possible, see for example the seminal work of [Allen et al. 2003; Häehnel et al. 2003].

Algorithm 4.3.2: Animation Fitting

Input: Temporal point sequence D
Candidate reconstruction f, M, partially initialized.
Output: Improved reconstruction f, M

The pairwise matching model of Equation 4 is extended in Wand et al. [2007; 2009] to a global animation fitting approach that fits animation sequences with multiple frames to data. For this, an augmented energy function is employed:

$$E_{anim}(f) = E_{dist}(f(M), D) + E_{elastic}(f) + E_{temp}(f) \quad (5)$$

It now operates on a whole animation sequence. It computes the distance to the data at all frames (summation over time) and it also sums up the elastic energy in all time steps. Furthermore, it adds a new term E_{temp} that takes into account the temporal behavior of the time-dependent motion field f. It penalizes acceleration such that smooth motion is preferred. This energy can be optimized using partially initialized data, where some correspondences $f_t(m), m \in M$ are not known. The method first fixes the known correspondences and fills in the missing data and then perform a global energy minimization. This interpolates missing data in a temporally coherent fashion and distributes the remaining error globally. Another way to view this is as a numerical bundle adjustment to improve the reconstruction accuracy.

Discussion: Once again, it is very important to stress that this optimization is only reliable if the model is suitably initialized. In particular, the data term is highly non-convex so that model parts covered by data need to be prepositioned close to matching data points. We use the existing technique because of its ability to interpolate missing data and because the numerical optimization, as a continuous method, does not suffer from precision limitations (unlike some of our intrinsic algorithms, as discussed next). There is a small inconsistency, though: The extrinsic methods assume elastic deformations (minimizing bending and stretching), while the intrinsic methods assume isometry (minimizing stretching only). For

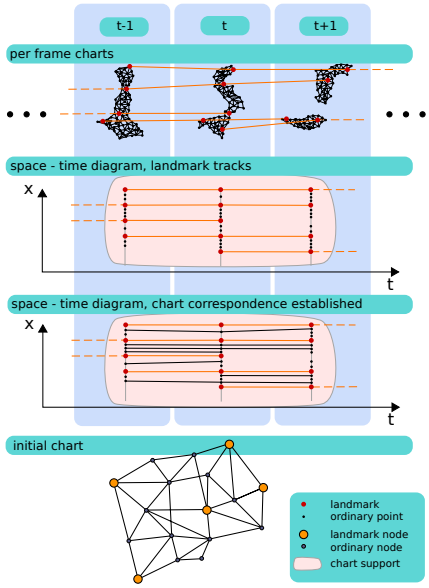


Fig. 8. i-chart construction as space time diagram (x-axis: time, y-axis: spatial location). We first build single frame charts and track landmarks (first row). The second row shows a block of tracked landmark points. These are used to perform robust matching of landmark coordinates to establish correspondence between ordinary points (third row). The resulting chart is represented as a time independent graph that encodes the intrinsic structure (last row). Each point stores correspondences to the raw data (not shown).

the target data, this is an acceptable restriction: The stricter assumption of elastic behavior is a reasonable regularizer, as validated extensively in previous work [Häehnel et al. 2003; Wand et al. 2007; Süßmuth et al. 2008; Li et al. 2009; Wand et al. 2009]. Nevertheless, the charting algorithm itself could alternatively be formulated in a purely intrinsic fashion. We will discuss this briefly in the following.

5. RECONSTRUCTION PIPELINE

We now use the building blocks developed in the previous section to setup a complete animation reconstruction pipeline. We divide the algorithm into three conceptual stages: *i-chart building*, *p-chart merging*, and *final optimization*.

5.1 Building i-Charts

As a first step of our algorithm, we run the continuous landmark tracking Algorithm 4.1.1 to determine a set of landmark tracks L. Afterwards, we first build a separate single frame chart for every frame of the input using Algorithm 4.2.1. From this correspondence information, we build initial charts (i-charts). This is done by merging single frame charts using the continuous landmarks tracks L.

We first select a subset of starting frames to build the initial charts (our current implementation uses every tenth input time step as starting frame). For each starting frame, we build one i-chart. We first fix the landmark set L to the landmark sets that overlap the starting frame. We then walk both forward and backward in time and use the chart merging to merge the data into a larger i-chart. For this step, however, we ignore the stitching of the graphs and use only the reference frame as chart's urshape. This provides us with a temporal correspondence information of the chart and a suitable urshape (i.e. does not contain any "artificial" errors which might be introduced with our stitching pipeline).

In each merging step, we exclude unreliable correspondences, and also exclude newly starting tracks that were not continuously present from the starting frame. Therefore, the amount of area covered will typically decrease with time distance to the starting frame. When the ratio of matched nodes to the number of nodes in the base frame falls below a threshold (we use 40% in all our results) we stop the temporal extension of the i-charts. Finally, we equip the newly created chart with an urshape; we just use the starting frame. Figure 8 summarizes the process.

Discussion: We have designed this procedure to make sure that an i-chart does not contain the same piece of geometry twice at different positions in the chart: We never include any area in an i-chart that could already have been represented elsewhere within the same chart, but where we would not yet have been able to recognize this fact. This is guaranteed by not introducing new landmarks and by only collecting reliable correspondences. Therefore, the coverage of i-charts is typically still fragmented. Patching these fragments together is the task of the next step, *p-chart merging*.

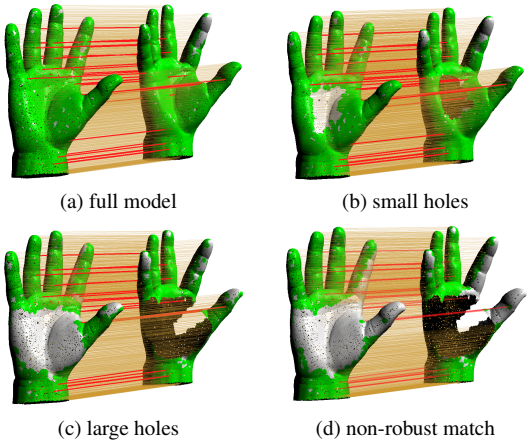
5.2 Outer Loop: Building p-Charts

We now build p-charts by stitching together separate i-charts as well as p-charts that have already been generated earlier during this process. The stitching is done by using the global matching Algorithm 4.1.2. It first tries to establish landmark correspondences. If a sufficient number of matches is found, the two charts are assembled by chart merging (Algorithm 4.2.2), followed by subsampling (Algorithms 4.2.3). The topology check (Algorithm 4.2.4) is performed before graph merging in order to reduce the error accumulation which might be introduced by merging two urshapes with false connections.

Merging by global matching has a certain risk because the RANSAC matching algorithm might fail to give good results with some small probability. We can minimize the risk by using good matching candidates first. Each i-chart and newly generated p-chart is kept in a pool of matching candidates. In order to decide on which pairs to match first, we use the following score:

$$w_{score} = \lambda_1 w_{overlap} + \lambda_2 w_{match} + \lambda_3 w_{common}, \qquad (6)$$

$w_{overlap}$ is the temporal overlap of the charts, i.e. the number of overlapping frames of the two charts normalized by their maximum length. w_{common} is the normalized number of common landmarks in both charts. w_{match} is the average number of matched nodes during all previous i-chart or p-chart merging operations, thus quantifying how well the matching worked out in the history of this chart. The weight parameters are set to $\lambda_1 = 3, \lambda_2 = 2$ and $\lambda_3 = 1$, putting most emphasize on overlap. This heuristic scoring encourages the merging of charts that actually do overlap and are not likely to be bad matches. In addition, we also monitor the outcome of a match. Chart merging is considered a failure if only a small number of correspondences have been established in relation to the overall number of nodes (in practice, we use 30% as threshold). In case of failure, the p-chart is not added to the pool and only one of the two participating charts is kept. We keep the "better" one judging by Equation 6 (omitting the overlap which is not defined for a single chart).



(a) full model (b) small holes

(c) large holes (d) non-robust match

Fig. 9. Effect of robust matching: Matching a (synthetic) hand model with a simulated acquisition hole. The results (a)-(c) use robust matching so that still large areas are covered with reliable correspondences (green). In the non-robust result in the lower right (d), significantly more area outside the hole region cannot be matched.

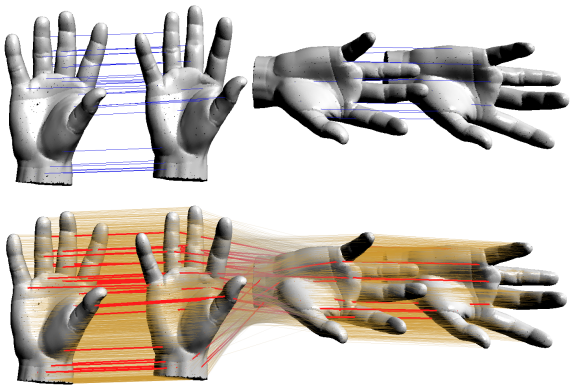


Fig. 10. Applying landmark tracking to the hand data set (synthetic). Upper row: The blue tracks are obtained by continuous tracking; they end automatically at the abrupt turn in the middle. Bottom row: The situation is recognized automatically and the RANSAC algorithm establishes additional landmark correspondences. The orange lines indicate the final dense correspondences.

5.3 Final Optimization

The outer loop described above is run until only one chart is left in the pool, which is the final reconstructed chart, and the primary output of our method.

We use this chart to drive a final numerical bundle adjustment according to Algorithm 4.3.2. This yields a full motion sequence where the urshape of the final chart is deformed to plausibly fit all of the data and move smoothly over time for frames or parts where no data is available. We show these reconstructions as results in Section 6 and in the accompanying video.

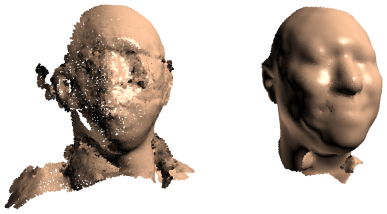


Fig. 11. Comparison of the algorithm of Wand et al. [2009] (left) and our reconstruction (right) for the "Face 2 shuffled" data set. In the case of Wand et al., large deformations between individual frames prevent a proper alignment of the data.

6. RESULTS

We evaluate our algorithm on a number of data sets. First, we use the "Saskia", "Abhijeet" and "Kicker" data sets of Vlasic et al. [2009], which have been acquired using a photometric stereo approach. We also include the "Face" and "Puppet" data sets of Li et al. [2009], which have been acquired with the motion-compensated structured light acquisition method of [Weise et al. 2007]. Finally, we also include the "woman" a dataset that we acquired ourselves using a Swissranger SR4000 [MESA] time-of-flight depth camera. In addition to the original data sets we create a shuffled version of the "Face" data set by deleting subsequences of frames and rearranging the remaining data blocks. This data set is specifically designed to test the performance of our landmark continuation technique, Algorithm 4.1.2. In addition, we also use a synthetic data set of a gesticulating hand, created in Poser 7, to separately evaluate the two main new pipeline stages, landmark tracking and robust charting. To fully appreciate the results of our technique we recommend to watch the accompanying video. A brief summary is shown in Figure 12.

6.1 Synthetic Tests

We first examine the two most important algorithmic components of our pipeline separately before we test the complete pipeline. Figure 9 shows a hand model in two different poses with an increasing amount of missing data. Green area indicates that the variance of the matching distributions indicates a reliable match. The robust matching model is able to find reliable matches for most of the non-hole area and does not create false-positives. If we turn off the robustness, the coverage is substantially reduced.

Figure 10 shows a tracking result on a hand sequence that in the middle undergoes an abrupt motion. The landmark tracks are correctly interrupted at this point and the RANSAC matching is invoked to build new landmark correspondences. Finally, the dense chart merging is used to obtain globally consistent dense correspondences.

6.2 Real-world Scanner Data

The different real-world data sets (see Figure 12 and the video) present a number of challenges to our reconstruction pipeline. For the "Saskia" data set, the legs of the person often appear to be connected to the skirt, giving evidence for a different topology than the correct interpretation of the legs being separately moving objects. In addition, we have significant amounts of missing data in the leg region. Another difficulty is presented by the arms moving upwards

Table I. Statistics for processing the individual data sets.

data set	Saskia	Face	Face shuffled	Puppet	Abhijeet*	Kicker*	Woman*
# frames in seq.	116/116	200/200	141/106	100/100	92/112	20/20	100/100
avg. # data points / frame	20500	10300	10300	9800	20000	16000	8300
avg. # ord. points	1250	1300	1300	1250	1800	2000	3000
avg. # landmarks	82	66	66	86	81	79	34
comp. time i-charts	5h	3h	2h	3h30min	4h30min	1h18min	21min
comp. time outer loop	3h	1h20min	1h	1h40min	1h	19min	2h
comp. time post-proc.	30min	1h	1h	25min	12min	8min	4min

in the beginning of the sequence. Since the scanner is not able to resolve the gap between the arms and the body the surface seems to undergo elastic deformation. Even though this violates our model our algorithm is able to process the data. Also note how the legs are reconstructed as separate entities, see Fig. 1 (left). This is only possible using the improved version of Algorithm 4.2.4, which detects topology changes. The basic variant proposed in earlier work fails to recognize the individual parts.

The main challenge in the "Face" data set, which has previously been used for template-based animation reconstruction, is presented by large amounts of missing data, due to the single camera scanner setup. Important features of the head, such as both ears, are never present simultaneously in any of the input frames. The nose is often visible from one side only. Nevertheless, our algorithm successfully assembles a complete urshape of the person, including both ears and a closed nose surface, See Figure 1 (right) and Figure 12. In addition, the neck region appears to be disconnected for a large part of the sequence. Our algorithm is able to correctly connect the neck to the head. A small artifact remains: The data set contains a few small disconnected outlier patches (collar of the shirt) that are attached to the main figure in the reconstruction. Here, the available data is insufficient to handle these pieces correctly.

The "Puppet" data set is an example for a strongly deformable object. It has also previously been used for template-based reconstruction. The data set is challenging due to its strongly changing apparent topology (hand connected to body), see Fig. 7. The incorrect topology even persists for as many as 40 frames. Even with large amounts of missing data in the folds and widely varying apparent surface topology we recover the complete sequence. Again, a using the improved algorithm to resolve the topology is essential; the previous algorithm leads to incorrect reconstructions.

The "Face shuffled" data set shows a test sequence for our landmark continuation strategy. We cut the original "Face" sequence into 5 blocks of 10,29,29,10, and 28 data frames each. In between these contiguous blocks of data we deleted 12,10,8, and 5 frames of the original data frames. They are present as empty frames in the modified data set, disrupting landmark tracking altogether. As shown in Fig. 11 (right), our algorithm is still able to reconstruct a complete template model and its motion over the full sequence, even interpolating the missing frames with plausible information (see video). For comparison, we show a result computed with the sequential alignment algorithm of Wand et al. [2009], Fig. 11 (left), which, as expected, is not able to perform a useful reconstruction for this type of incoherent motion.

The "Abhijeet" data set is particularly challenging. As shown in the video, the topology is ambgious and the geometry shows systematic low-frequency artifacts. Parts of the arm are displaced by more than the diameter of the arm itself, and incorrect sheets of surface show up, probably due to the photometric acquisition approach that

Fig. 12. Original data (left) and reconstruction with parameterization of the "Saskia", "Puppet", "Face", "Abhijeet", "Kicker" and "Woman" data sets for different poses.

cannot estimate depth reliably in this highly occluded situation. The situation is particularly bad for the first 20 frames, where the arms are merged into the body and drag large sheets of phantom geometry with them when disconnecting. When we omit these frames, we obtain qualitatively correct results for the remaining 94 frames, with stable correspondences. The main artifact is that the arms in some frames twist and squeeze. However, the data supporting this is rather week and already outside the scope covered by our matching model. We therefore think that this example shows quite well both the limitations of matching model itself as well as the robustness of the computational pipeline. We compare our results to those of previous local optimization [Wand et al. 2009]; again, the local algorithm produces inferior results with significant artifacts in both shape and motion (see the accompanying video for details).

The "Woman" data set represents a stress test and partial failure case for our approach. The time-of-flight data is extremely noisy, which is a major challange for the landmark tracking algorithm. In addition, the apparent topology is again constantly changing, including a full topological connection of the arms with the upper body in the the beginning of the sequence. We obtain only sparse tracking information so that our algorithm was not able to reconstruct dense correspondences reliably over the body for all frames but some data remains uncharted. Hence, the final optimization produces a smooth interpolation that in some parts does not follow the data. The quality of the reconstructed geometry is low; the correspondence noise does not permit resolving high frequency details in the final reconstruction, but the result is qualitatively correct. For such kind of very low-resolution data, additional cues such as a simultaneously recorded video with color information is probably necessary to permit better reconstructions.

In Table I, we show statistics of algorithm run times and other characteristic data for the different sequences. The first row shows the number of reconstructed frames versus the available data frames. Note that for the "Face shuffled" data set more frames are reconstructed than are present in the original data. The second row shows the average number of data points per frame of the input sequence, while row three shows the number of nodes in a typical chart. The average number of detected landmarks per frame is shown in row four. Note that this number varies widely over frames. Finally, the computation times for the different steps of our algorithm are given. The computations were run on a 2xQuadCore Intel Xeon X5550 with 2.67 GHz. Datasets marked with a "*" were computed on 2xHexaCore Intel Xeon X5650 with 2.67 GHz.

6.3 Discussion and Limitations

As shown by the example scenes, the new algorithm is able to handle more general input data that could not be reconstructed automatically by previous techniques. Not relying on temporal coherence is an important step for practical applications. Although scanners are available that scan at very high frame rates, the fact that geometry often vanishes in acquisition holes and reappears in a different pose is a strongly limiting factor in practice to previous algorithms. We can also show that the algorithm is quite robust. Even for data sets with strong noise or artifacts outside our modeling assumptions, we still obtain qualitatively correct results.

As most complex reconstruction systems, our method has a number of parameters. However, we were able to fix most of these parameters for all of the data sets, as described in the text. We only adapted the sampling resolution ϵ_s to minimize the computational costs. In addition, we have increased the number of landmarks in the robust matching scores from 5 to 6 in the "Abhijeet" data set as this lead to slightly better results. Finally, we have adapted the regularizer weights for stiffness and acceleration penalty in the final numerical optimization best visual (aesthetic) impression.

Our method still has a number of limitations that require further research: A problem is the handling of "unreliable" data. Our current pipeline dismisses this data in the construction of initial i-charts but during p-chart merging, we currently do not delete uncharted data because this could reintroduce large holes in the charts but rather rely on extrinsic alignment to match these pieces. This problem can be addresses by a good scheduling of the merging operations, which are commutative but not associative: the order in which pairs are merged matters. The current heuristic tries to minimize the negative impact by aiming at large overlap, but better orders (possibly including options for backtracking from bad matches) might exist.

A second issue is the detection of topological changes. Although we can handle more scenes than previous techniques, we still encounter problems in some situations. In particular, if large acquisition holes and topological changes coincide, this can lead to incorrect results where the local topology is not resolved correctly. An example is the face scan from [Wand et al. 2007]. Our technique cannot resolve the opening of the mouth because of large acquisition holes opening up around the lips simultaneuously with the opening of the mouth. The missing area is too large to be handled by even robust intrinsic matching. In this case, a purely extrinsic technique or a template based techniques [Li et al. 2009] has an advantage over our approach.

Finally, the combination of elastic and isometric matching is sometimes a limiting factor: for objects with very strong deformations, this introduces a bias towards rigidity, leading to insufficient bending. A purely intrinsic formulation of the charting could probably reduce these problems. However, this seems to be a minor issue in practice that can usually be resolved by reducing the strength of the elastic regularizer appropriately.

7. CONCLUSIONS

We have presented a global optimization technique for animation reconstruction from dynamic point cloud sequences as produced by dynamic range scanning devices. Our method is based on the concept of cartography and uses an intrinsic framework for a more reliable and robust matching of partial deformable shapes in vastly different poses. Iteratively applying this technique automatically yields a completed template model, its motion over the course of the acquired sequence and a consistent parameterization. Our technique uses a landmark tracking scheme that uses temporal coherence if available but can fully automatically resort to an efficient randomized global matching algorithm if required by the data. We can thus recover from scanner shortcomings such as large scale occlusion and we can handle fast motion in the scene. We also improve the robustness in detecting topological changes. Overall, we are able to process sequences under significantly more general conditions than previous work.

For future work some interesting avenues are opened up by our research. First, the problem of finding a globally consistent intrinsic description of a moving scene can be applied to other problem areas, e.g. the computer vision problem of robustly detecting occlusion boundaries in video sequences. Stereo or multi-view stereo applications could be approached this way. Another interesting development would be the formulation of resolution-enhancing tech-

niques like interpolation and surface approximations that currently require extrinsic embeddings in a completely intrinsic framework. This would enable our technique to be applied in a completely shape invariant way.

Acknowledgements

This work has been partially supported by the DFG "Cluster of Excellence Multi-Modal Computing and Interaction". We would also like to thank Vlasic et al. and Li et al. for making their data available.

REFERENCES

AHMED, N., THEOBALT, C., ROSSL, C., THRUN, S., AND SEIDEL, H.-P. 2008. Dense correspondence finding for parametrization-free animation reconstruction from video. *Proc. of CVPR*, 1–8.

ALLEN, B., CURLESS, B., AND POPOVIĆ, Z. 2003. The space of human body shapes: Reconstruction and parameterization from range scans. *22,* 3, 587–594.

ANGUELOV, D., SRINIVASAN, P., PANG, H.-C., KOLLER, D., THRUN, S., AND DAVIS, J. 2004. The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *NIPS*.

ANUAR, N. AND GUSKOV, I. 2004. Extracting animated meshes with adaptive motion estimation. In *Proc. of VMV*. 63–71.

BICKEL, B., BOTSCH, M., ANGST, R., MATUSIK, W., OTADUY, M., PFISTER, H., AND GROSS, M. 2007. Multi-scale capture of facial geometry and motion. *ACM Trans. Graph. 26,* 3, 33.

BRADLEY, D., HEIDRICH, W., POPA, T., AND SHEFFER, A. 2010. High resolution passive facial performance capture. *ACM Trans. Graph. 29,* 3, to appear.

BRADLEY, D., POPA, T., SHEFFER, A., HEIDRICH, W., AND BOUBEKEUR, T. 2008. Markerless garment capture. *ACM Trans. Graph. 27,* 3, 99.

BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. 2006. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proc. National Academy of Sciences (PNAS) 103,* 5, 1168–1172.

BRONSTEIN, A. M., BRONSTEIN, M. M., AND KIMMEL, R. 2009. Topology-invariant similarity of nonrigid shapes. *Intl. Journal of Computer Vision (IJCV) 81,* 3, 281–301.

BRONSTEIN, A. M., BRONSTEIN, M. M., KIMMEL, R., MAHMOUDI, M., AND SAPIRO, G. 2010. A gromov-hausdorff framework with diffusion geometry for topologically-robust non-rigid shape matching. *Intl. Journal of Computer Vision (IJCV).* in press.

CARRANZA, J., THEOBALT, C., MAGNOR, M. A., AND SEIDEL, H.-P. 2003. Free-viewpoint video of human actors. In *ACM Trans. Graph.* 569–577.

CHANG, W. AND ZWICKER, M. 2008. Automatic registration for articulated shapes. *Computer Graphics Forum (Proc. of SGP) 27,* 5, 1459–1468.

CHANG, W. AND ZWICKER, M. 2009. Range scan registration using reduced deformable models. *Computer Graphics Forum (Proc. Eurgraphics) 28,* 2, 447–456.

DAVIS, J., NEHAB, D., RAMAMOORTHI, R., AND RUSINKIEWICZ, S. 2005. Spacetime stereo: A unifying framework for depth from triangulation. *IEEE PAMI 27,* 2, 296–302.

DE AGUIAR, E., STOLL, C., THEOBALT, C., AHMED, N., SEIDEL, H.-P., AND THRUN, S. 2008. Performance capture from sparse multi-view video. *ACM Trans. Graph. 27,* 3, 98.

FANG, Q., GAO, J., GUIBAS, L. J., DE SILVA, V., AND ZHANG, L. 2005. Glider: Gradient landmark-based distributed routing for sensor networks. In *24th Conference of the IEEE Communications Society (InfoCom)*.

GELFAND, N. AND GUIBAS, L. J. 2004. Shape segmentation using local slippage analysis. In *Proc. of SGP*. 214–223.

HÄEHNEL, D., THRUN, S., AND BURGARD, W. 2003. An extension of the icp algorithm for modeling nonrigid objects with mobile robots. In *Proc. Int. Joint Conf. on Artificial Intelligence (IJCAI)*. 915–920.

HORN, B. K. P. 1987. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A 4,* 4, 629–642.

HUANG, Q.-X., ADAMS, B., WICKE, M., AND GUIBAS, L. J. 2008. Non-rigid registration under isometric deformations. *Computer Graphics Forum (Proc. SGP) 27,* 5, 1449 – 1457.

KÖNIG, S. AND GUMHOLD, S. 2008. Image-based motion compensation for structured light scanning of dynamic scenes. *Int. J. of Int. Sys. Tech. App. 5,* 3/4, 434 – 441.

LEORDEANU, M. AND HEBERT, M. 2005. A spectral technique for correspondence problems using pairwise constraints. In *Proc. of ICCV*. Vol. 2. 1482–1489.

LI, H., ADAMS, B., GUIBAS, L. J., AND PAULY, M. 2009. Robust single-view geometry and motion reconstruction. *ACM Trans. Graph. 28,* 5, 175.

LI, H., SUMNER, R. W., AND PAULY, M. 2008. Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum (Proc. SGP) 27,* 5, 1421–1430.

LIAO, M., ZHANG, Q., WANG, H., YANG, R., AND GONG, M. 2009. Modeling deformable objects from a single depth camera. In *Proc. Int. Conf. on Computer Vision (ICCV)*.

LIPMAN, Y. AND FUNKHOUSER, T. 2009. Möbius voting for surface correspondence. In *Proc. of SIGGRAPH '09*. 1–12.

LUCAS, B. D. AND KANADE, T. 1981. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*. 674–679.

MITRA, N. J., FLORY, S., OVSJANIKOV, M., GELFAND, N., GUIBAS, L., AND POTTMANN, H. 2007. Dynamic geometry registration. In *Proc. of SGP*. 173–182.

OVSJANIKOV, M., MÉRIGOT, Q., MÉMOLI, F., AND GUIBAS, L. 2010. One point isometric matching with the heat kernel. In *Proc. of SGP*. 1555–1564.

PARK, S. I. AND HODGINS, J. K. 2006. Capturing and animating skin deformation in human motion. *ACM Trans. Graph. 25,* 3, 881–889.

PEKELNY, Y. AND GOTSMAN, C. 2008. Articulated object reconstruction and markerless motion capture from depth video. *Computer Graphics Forum (Proc. Eurographics) 27,* 2, 399–408.

POPA, T., SOUTH-DICKINSON, I., BRADLEY, D., SHEFFER, A., AND HEIDRICH, W. 2010. Globally consistent space-time reconstruction. *Computer Graphics Forum (Proc. SGP) 29,* 5, 1633–1642.

SAND, P., MCMILLAN, L., AND POPOVIĆ, J. 2003. Continuous capture of skin deformation. *ACM Trans. Graph. 22,* 3, 578–586.

SHARF, A., ALCANTARA, D. A., LEWINER, T., GREIF, C., SHEFFER, A., AMENTA, N., AND COHEN-OR, D. 2008. Space-time surface reconstruction using incompressible flow. *ACM Trans. Graph. 27,* 5, 110.

STARCK, J. AND HILTON, A. 2007. Correspondence labelling for wide-timeframe free-form surface matching. In *Proc. of ICCV*. 1–8.

SÜSSMUTH, J., WINTER, M., AND GREINER, G. 2008. Reconstructing animated meshes from time-varying point clouds. *Computer Graphics Forum (Proc. SGP) 27,* 5, 1469–1476.

TEVS, A., BERNER, A., WAND, M., IHRKE, I., AND SEIDEL, H.-P. 2011. Intrinsic shape matching by planned landmark sampling. *Computer Graphics Forum (Proc. Eurographics) 30.* 543–552.

TEVS, A., BOKELOH, M., WAND, M., SCHILLING, A., AND SEIDEL, H.-P. 2009. Isometric registration of ambiguous and partial data. In *Proc. of CVPR*. 1185–1192.

VARANASI, K., ZAHARESCU, A., BOYER, E., AND HORAUD, R. P. 2008. Temporal surface tracking using mesh evolution. In *Proc. of ECCV*. 30–43.

VLASIC, D., PEERS, P., BARAN, I., DEBEVEC, P., POPOVIĆ, J., RUSINKIEWICZ, S., AND MATUSIK, W. 2009. Dynamic shape capture using multi-view photometric stereo. *ACM Trans. Graph. 28,* 5, 174.

WAND, M., ADAMS, B., OVSJANIKOV, M., BERNER, A., BOKELOH, M., JENKE, P., GUIBAS, L., SEIDEL, H.-P., AND SCHILLING, A. 2009. Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data. *ACM Trans. Graph. 28,* 2, 1–15.

WAND, M., JENKE, P., HUANG, Q., BOKELOH, M., GUIBAS, L., AND SCHILLING, A. 2007. Reconstruction of deforming geometry from time-varying point clouds. In *Proc. of SGP*. 49–58.

WEISE, T., LEIBE, B., AND GOOL, L. V. 2007. Fast 3d scanning with automatic motion compensation. In *Proc. of CVPR*. 1–8.

WÜRMLIN, S., LAMBORAY, E., STAADT, O. G., AND GROSS, M. H. 2002. 3d video recorder. In *PG '02: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*. IEEE Computer Society, Washington, DC, USA, 325.

ZHANG, L., SNAVELY, N., CURLESS, B., AND SEITZ, S. M. 2004. Space-time faces: High resolution capture for modeling and animation. *23,* 3, 548–558.

ZHENG, Q., SHARF, A., TAGLIASACCHI, A., CHEN, B., ZHANG, H., SHEFFER, A., AND COHEN-OR, D. 2010. Consensus skeleton for non-rigid space-time registration. *Computer Graphcis Forum (Special Issue of Eurographics) 29,* 2, to appear.

ZITNICK, C. L., KANG, S. B., UYTTENDAELE, M., WINDER, S., AND SZELISKI, R. 2004. High-quality video view interpolation using a layered representation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*. ACM, New York, NY, USA, 600–608.