EUROGRAPHICS 2009 / P. Dutré and M. Stamminger
(Guest Editors)

Volume 28 (2009), Number 2

Range Scan Registration Using Reduced Deformable Models

W. Chang[1] and M. Zwicker[1,2]

[1]University of California, San Diego
[2]Institut für Informatik und angewandte Mathematik, Universität Bern, Switzerland

Abstract

We present an unsupervised method for registering range scans of deforming, articulated shapes. The key idea is to model the motion of the underlying object using a reduced deformable model. We use a linear skinning model for its simplicity and represent the weight functions on a regular grid localized to the surface geometry. This decouples the deformation model from the surface representation and allows us to deal with the severe occlusion and missing data that is inherent in range scan data. We formulate the registration problem using an objective function that enforces close alignment of the 3D data and includes an intuitive notion of joints. This leads to an optimization problem that we solve using an efficient EM-type algorithm. With our algorithm we obtain smooth deformations that accurately register pairs of range scans with significant motion and occlusion. The main advantages of our approach are that it does not require user specified markers, a template, nor manual segmentation of the surface geometry into rigid parts.

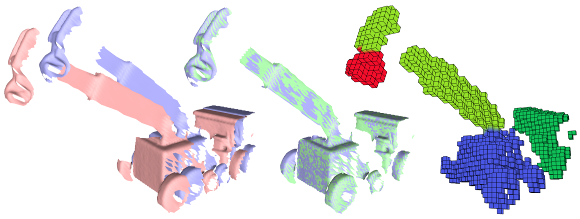
Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms

1. Introduction

Acquiring surface models from the real world is an important task in the area of computer graphics. To create a complete, high-quality surface model, a key processing step is to accurately align range scans of the surface captured at multiple viewpoints. This problem has been solved for the case where the object stays rigid, but the case where the object moves or deforms remains a challenging open problem. In this case, the motion of the object needs to be determined between each pair of scans.

We observe that in the real world, the motion of many objects can be described by a *reduced deformable model* (RDM). This means that the motion can be specified by a small number of deformation parameters. For example, articulated objects consist of a few *bones:* parts of the object that move together as a unit. By moving these bones, the rest of the object moves accordingly. Therefore, the deformation parameters of the RDM are only the position of the bones and the assignment of parts of the object to each bone. Since the parts are not required to be completely rigid, the model is suitable to represent articulated characters.

In this paper, we propose a registration algorithm for aligning range scans of articulated objects. We model the underlying motion with an RDM to limit the number of deformation parameters. We choose a linear blend skinning model (LBS) due to its simplicity and because it allows for efficient optimization algorithms. Traditionally, the deformation parameters of an LBS model are represented as the *rigid transformations* of the bones and their *influence weights*, also known as skinning weights. The weights indicate how strongly the bone influences each vertex, or how much each vertex "belongs" to a particular bone. In order to align range scans using this model, we will solve for its deformation parameters. This means we need to solve for the movement of each bone, i.e., its transformation, and the influence weights at each point on the surface.



(a) Source and Target (b) Registration Result (c) Influence Weights

Figure 1: *Given a pair of range scans, our technique automatically finds the parameters of a linear blend skinning (LBS) model to register the scans accurately.*

However, range scans usually have tens of thousands of 3D points, so the number of unknown weights that we must determine quickly multiplies with the number of bones. In addition, range scans are not in correspondence and typically suffer from missing data and disconnected surface geometry. To address these problems, we propose a novel representation of the influence weights on a grid enclosing the range scan surface. Not only does this reduce the number of variables, but it also provides a regular structure to define the weight function that is decoupled from the incomplete surface representation.

We formulate the registration problem using an objective function that enforces close alignment of the 3D data and includes an intuitive notion of joints. We develop a novel algorithm to solve for the deformation parameters, i.e., the bone transformations and their influence weights, in an alternating fashion. To obtain a coherent deformation and to prevent bones from becoming dislocated, we formulate joint constraints directly on the grid. Also we propose an efficient solution for solving the influence weights on the grid using a combination of discrete labeling with continuous weight optimization.

Using our algorithm, we demonstrate that we can efficiently and accurately register pairs of range scans despite significant motion and missing data. In contrast to other approaches, our method does not require user specified markers, a template, nor a manual segmentation of the surface geometry. In summary, the contributions of our work are:

- A novel method to solve for a RDM using a grid-based representation of the influence weights,
- A generalized deformation energy used to apply joint constraints when solving for the motion,
- A discrete labeling framework to efficiently solve for the influence region of each transformation,
- And the ability to solve for continuous influence weights on range scans.

2. Related Work

Non-Rigid Registration. The registration problem has a long history in the graphics and vision literature. For registering rigid surfaces, the classic iterative closest points (ICP) algorithm [CM91, BM92] and its many extensions have proven to be very effective. Recently, there has been a renewed interest in the registration problem for non-rigid surfaces, and many have extended the basic ICP framework for this case. To examine key differences from our work, we categorize the literature based on how each method *models the motion* of the non-rigid object. Methods that model the motion using thin-plate splines [CR03, BR07] are able to solve for a globally smooth deformation that aligns the surfaces, but are unable to effectively express large or piece-wise rigid deformations. Many techniques model the motion of each individual point using an affine or rigid transformation, and then solve the registration using temporal coherence [HTB03, ARV07], user-placed markers [ACP03, PMG*05], or a template [ASP*04]. Some recent methods model the deforming surface in 4D space-time [MFO*07, WJH*07, SWG08, SAL*08], effectively tracking the motion of each point over time.

The key difference in our work is that we do not model the individual motion of each point, but instead we explicitly solve for skinning weights that determine groups of points moving together. This is similar to the recent work by Huang et al. [HAWG08], which optimizes the deformation by clustering the motion of the surface into rigid parts, and the work by Chang and Zwicker [CZ08], which determines a coherent assignment of transformations to the surface. However, neither approach solves for the skinning weights. Instead, Huang et al. solve for deformed sample positions and interpolate the rest of the shape using prescribed influence weights [SSP07]. The transformation assignment by Chang and Zwicker can be thought of as binary weights, but this leads to unnatural deformations. This problem is alleviated by optimizing continuous skinning weights in our method.

Li et al. [LSP08] also use a reduced set of transformations to model the deformation. However, this work prescribes weights on the surface in the form of a deformation graph [SSP07], and the weights are not allowed to change at all. Our grid-based approach allows us to optimize for the weights dynamically based on the deformation expressed by the examples. Sharp deformations (e.g. a ball joint connecting two rigid parts) can be expressed automatically using our system, whereas the graph-based approach requires that the graph nodes to be placed carefully in advance.

Finally, the work by Pekelny and Gotsman [PG08] solves the problem using a manually specified segmentation to indicate each rigid part of the scanned object, but our approach does not require such user intervention.

Skinning from Examples. Since we attempt to directly model a RDM, our work is closely related to techniques that model surface deformation from motion parameters. Given a mesh animation with associated skeleton parameters, many algorithms improve on the basic LBS model to accurately represent complex surface deformation [LCF00, SCFRC01, KJP02, WP02, MG03, WPP07]. Also, it is possible to extract a model using only a mesh animation [AKP*04, JT05, SY07, dATTS08]. Our work differs from these techniques because the input to our algorithm is only range scan data with no given correspondence or motion information.

3. Problem Formulation

Deformation Model. We propose to align depth scans of a moving object by modeling the motion with a reduced deformable model (RDM). Among the many RDMs available today, we choose LBS due to its simplicity and ease of optimization. In addition, we adapt the “abstract” notion of skeletal structure, where each “bone” of the skeleton is described implicitly by the weight function associated with each bone transformation [JT05]. In this case, a deformation is described by a function $D(\mathbf{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ representing the deformation of space as a weighted sum of rigid transformations

$$D(\mathbf{x}) = \sum_j w_j(\mathbf{x}) T_j(\mathbf{x}) = \sum_j w_j(\mathbf{x}) \left(R_j(\mathbf{x}) + \mathbf{t}_j\right). \quad (1)$$

Here, $T_j \in SE(3)$ is a rigid transformation (rotation R_j and translation $\mathbf{t}_j$), and we denote applying T_j to $\mathbf{x}$ as $T_j(\mathbf{x})$. Also, $w_j(\mathbf{x})$ is a spatially varying weight function that defines the continuous region of influence for the bone j. We also assume that the weights are non-negative and normalized:

$$\sum_j w_j(\mathbf{x}) = 1, \text{ where } w_j(\mathbf{x}) \geq 0 \;\forall j \;\forall \mathbf{x} \in \mathbb{R}^3.$$

Grid Based Representation. Because our input data consists of incomplete range scans without correspondence information, it is not possible to define the weight functions directly on the surfaces. Instead, we propose representing the weight functions on a regular grid enclosing the scanned geometry. Let us denote a regular grid by specifying values on a set of grid points

$$G = \{(\mathbf{p}_c, \mathbf{v}_c)\}_{c=1}^{m},$$

where c is the index of the grid point, $\mathbf{p}_c$ is the position of grid point c, $\mathbf{v}_c$ is the vector of weights at c, and m is the number of grid points. Alternatively, we can think of the grid as a collection of grid *cells*, denoted in capital letters $C \in G$, where each grid cell contains eight grid points, and adjacent grid cells share four grid points. For the 2D case, this situation is illustrated in Figure 2.

To compute the value of the weight function $w_j(\mathbf{x})$ on the surface geometry, we find the grid cell C that contains $\mathbf{x}$ and perform trilinear interpolation of the values $\mathbf{v}_c$ defined at the cell's grid points $c \in C$. This allows us to specify the weight function everywhere within the grid, and leads to a compact representation of the vertex weight function that is well-defined even in the presence of missing data. In addition, we adapt the grid to the scanned geometry. This is accomplished by using only those grid cells that contain points from the range scan. This reduces the numbers of variables needed to solve, and localizes the deformation to the surface geometry. We can also control the smoothness of the weight function by adjusting the resolution of the grid.

Registration. Given this underlying model of the motion, our goal is to find the deformation parameters that best align the input range scans. The parameters of our model are the set of transformations $\mathcal{T} = \{T_j\}$ and the weight functions $\mathcal{W} = \{w_j(\mathbf{x})\}$. We formulate an objective function $E(\mathcal{T},\mathcal{W})$ that describes the quality of a registration as a function of these unknown parameters:

$$\begin{aligned} E(\mathcal{T},\mathcal{W}) = \; & \alpha E_{\text{dist}}(\mathcal{T},\mathcal{W}) + \beta E_{\text{joint}}(\mathcal{T},\mathcal{W}) \\ & + \mu E_{\text{smooth}}(\mathcal{W}) + \nu E_{\text{norm}}(\mathcal{W}). \end{aligned} \tag{2}$$

The first term E_{dist} measures the accuracy of the alignment by evaluating the distance between the registered surfaces, as discussed in the beginning of Section 4.1. The second term E_{joint} enforces a joint constraint that encourages transformations to act intuitively at joints. We discuss this term in more detail in Section 4.1. The final two terms E_{smooth} and E_{norm} are regularization constraints on the weight functions, which we discuss in Section 4.2.

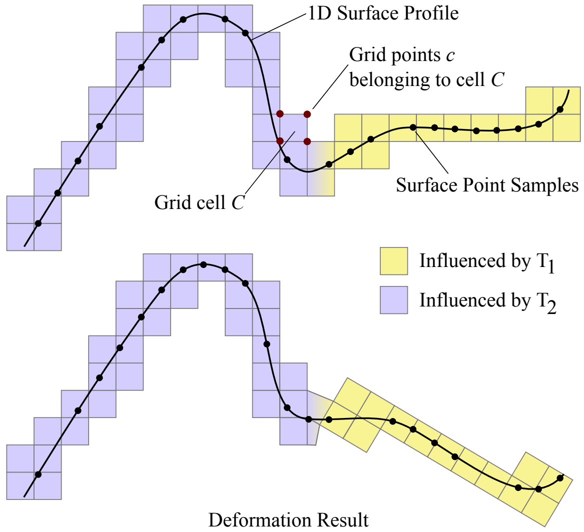


Figure 2: *We represent the motion of a range scan using LBS, which consists of a fixed number of bone transformations and their associated influence weights. We decouple the weights from the incomplete surface representation by defining them on a grid enclosing the range scan data.*

Algorithm 1 Registration Overview

Generate grid & initial segmentation (Section 4.3)
Generate initial correspondences using spin image features and RANSAC (Section 4.3)
repeat
    Solve for the transformations $\mathcal{T}$ (Equation 5)
    Update correspondences
    Solve for discrete weights $\mathcal{W}$ (Equation 11)
    Update correspondences
until Stopping criteria are met
Solve for continuous weights $\mathcal{W}$ (Equation 8)

4. Optimization Algorithm

We find the LBS parameters by minimizing the objective function:

$$\operatorname*{argmin}_{\mathcal{T},\mathcal{W}} E(\mathcal{T},\mathcal{W}). \tag{3}$$

We solve this problem in an iterative fashion, where we update the model parameters $\mathcal{T}$ and $\mathcal{W}$ until convergence. At each iteration, we perform an alternating optimization similar to the EM algorithm [DLR77] where we

1. (T-step) fix the current weights $\mathcal{W}$ to be constant and solve for the rigid transformations $\mathcal{T}$, then
2. (W-step) using the rigid transformations $\mathcal{T}$ from the T-step, solve for the weight functions $\mathcal{W}$ on the grid.

The entire optimization procedure is illustrated in Figure 3 and Algorithm 1. In the following sections, we discuss the T-step and the W-step in further detail.

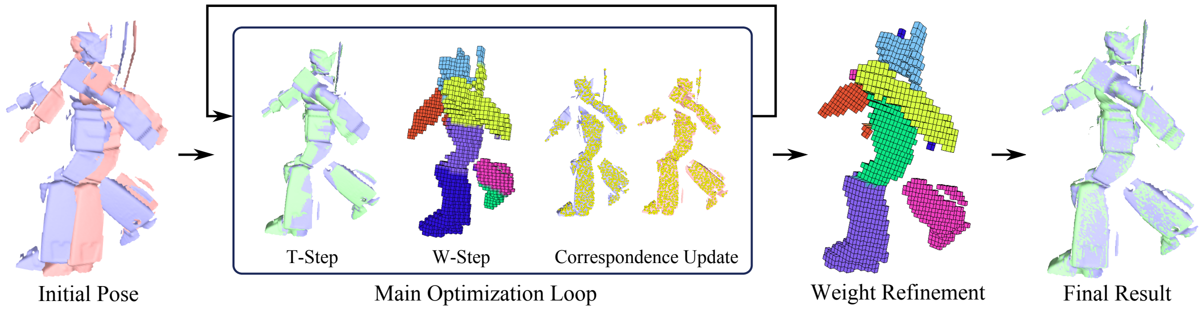


Figure 3: *Overview of our optimization algorithm. The main optimization loop alternates between solving for the transformations (T-step), solving for the weights (W-step), and updating correspondences. We obtain the final registration result after a weight refinement step that smooths unnatural deformations.*

4.1. Solving the T-step

The goal of the T-step is to obtain the rigid transformations $\mathcal{T}$ that minimize the objective. In this step the weights $\mathcal{W}$ are fixed, therefore the optimization includes only the distance term E_{dist} and the joint constraint E_{joint} term.

Distance Term. The distance term E_{dist} measures how close the deformation aligns the two shapes together. This term uses point correspondences between the source and target shapes. First, we generate a uniform sampling of points on the source shape. We then update the corresponding points on the target each time the transformations or weights are updated in the optimization. Initially, the corresponding points are found using feature descriptors (see Section 4.3). During the optimization, these points are updated using a closest point strategy with a modified distance measure. This measure computes a weighted average of two components: the distance between the points, and the distance between the normals [Joh97]. Given these correspondences, the total correspondence error is measured using the point-to-plane distance metric. To make this term more robust to outliers and missing geometry, we do not include a correspondence when the target point lies on a boundary, the distance between the points exceeds a threshold, or the angle between the normals exceeds a threshold [PG08]. The values of these thresholds are discussed further in Section 5.

Joint Constraints. A joint is a region that is influenced by more than one bone, and we naturally expect that the bones always meet at these joints. Therefore, to solve for natural bone transformations, we employ the joint constraint in our optimization to keep the shape intact and prevent bones from sliding away from each other (Figure 4).

Specifically, this constraint enforces that the joint region is transformed to the same location by all bones that influence it. In order to determine which points in space are part of a joint, we use the product of the weight functions. For example, the joint region for a pair of bones T_i and T_j is given by the points $\mathbf{x} \in \mathbb{R}^3$ where $w_i(\mathbf{x})w_j(\mathbf{x}) > 0$. For every such point, we would like to enforce that $T_i(\mathbf{x}) = T_j(\mathbf{x})$. Thus, we measure the total misalignment for each potential joint between bone i and j as:

$$E_{\text{joint}} = \sum_{i,j} \tau_{ij} \int_{\mathbf{x}\in\mathbb{R}^3} w_i(\mathbf{x})w_j(\mathbf{x}) \|T_i(\mathbf{x}) - T_j(\mathbf{x})\|^2 d\mathbf{x}, \quad (4)$$

where $\tau_{ij} = \left(\int_{\mathbf{x}\in\mathbb{R}^3} w_i(\mathbf{x})w_j(\mathbf{x})d\mathbf{x}\right)^{-1}$ is a normalization constant. This term is similar to the deformation energies and regularization terms proposed elsewhere [BPWG07,SSP07], which simply differ in the choice of weight functions. While other techniques prescribe the weight functions (e.g. [LSP08]), our formulation allows joint constraints for arbitrary weight functions. This is especially useful since we allow the weight function to change each time the weights are optimized in the W-step.

To compute the joint constraint term, we pull the unknown elements of the transformations out of the integral, since they are constants with respect to $\mathbf{x}$. We then analytically evaluate the integrals of the weight functions in each cell in closed form, and these form coefficients of a quadratic function of the unknowns T_i. Also, in our implementation, we specify the joint constraint separately for each cell by integrating only within the cell.

Optimization. Our final goal in the T-step is to minimize the objective consisting of the distance and the joint terms,

$$\operatorname*{argmin}_{\mathcal{T}} \alpha E_{\text{dist}}(\mathcal{T},\mathcal{W}) + \beta E_{\text{joint}}(\mathcal{T},\mathcal{W}), \quad (5)$$

where α and β are coefficients to weight each term. Since we require the transformations $\mathcal{T}$ to be rigid, this becomes a non-linear optimization problem. Similar to Botsch et al. [BPWG07], we solve this problem iteratively using the Gauss-Newton method, linearizing both the residuals and the rotation matrices.

4.2. Solving the W-step

In the W-step, we keep the transformations $\mathcal{T}$ fixed and solve for the weight functions $\mathcal{W}$. Since the weight function is defined on a regular grid and interpolated in between the grid points, we just need to solve for the weight values at each grid point. To provide regularization, the W-step includes the smoothness term E_{smooth} and the normalization term E_{norm}.

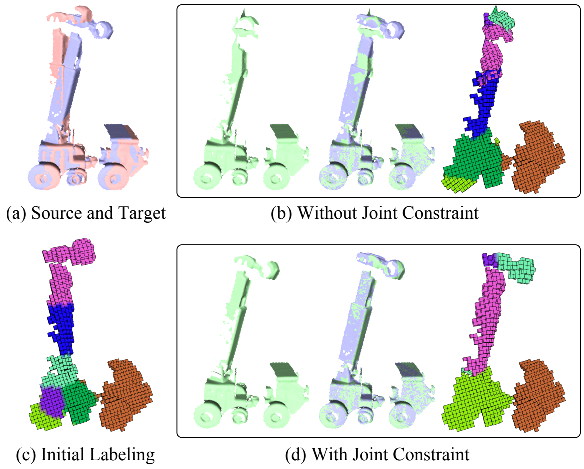


Figure 4: *The joint constraint helps to overcome undesired local minima in the optimization by preserving the joint location between neighboring transformations. The initial alignment (a) with labeling (c) does not register correctly when the joint constraint is removed (b). By adding the joint constraint, however, we converge to the correct result (d).*

However, we leave out the joint constraint term E_{joint} for computational efficiency. In theory, this means that the overall error could increase after each W-step iteration, but we have not experienced any problems in practice.

The smoothness term penalizes variation between the weights of neighboring grid points. It is defined as

$$E_{\text{smooth}}(\mathcal{W}) = \sum_{(c,d)\in E_G} \|\mathbf{v}_c - \mathbf{v}_d\|^2, \quad (6)$$

where E_G is the set of all edges between neighboring grid points c and d. The normalization term encodes our preference that the weights sum to 1 at each grid point

$$E_{\text{norm}}(\mathcal{W}) = \sum_c \left(1 - \sum_j v_{cj}\right)^2, \quad (7)$$

where v_{cj} is the j^{th} component of $\mathbf{v}_c$. Thus, in the W-step our goal is to optimize

$$\begin{aligned} &\underset{\mathcal{W}}{\text{argmin}} \quad \alpha E_{\text{dist}}(\mathcal{T},\mathcal{W}) + \mu E_{\text{smooth}}(\mathcal{W}) + \nu E_{\text{norm}}(\mathcal{W}) \\ &\text{subject to} \quad v_{cj} \geq 0 \quad \forall\, j,\ \forall\, \mathbf{v}_c \in G. \end{aligned} \quad (8)$$

We have observed that optimizing this objective directly often converges to a local minimum that does not correspond to an intuitive solution. With continuous weights, the system is underconstrained because even a single transformation can be expressed as a weighted combination of one, two, three, or more transformations. Thus, instead of obtaining weight functions that are localized to each rigid part, the weights tend to be very smooth and influence large parts of the shape. At the same time, the transformations $\mathcal{T}$ are overfitted in the successive T-steps.

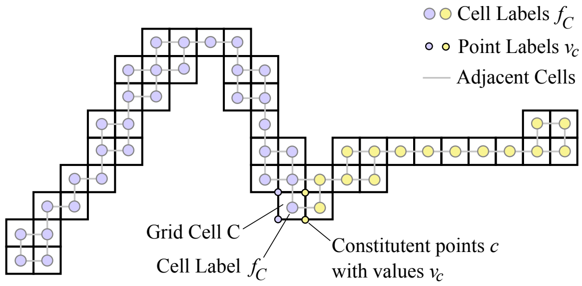


Figure 5: *We solve for the influence weights by formulating it as a labeling problem on the grid cells.*

Therefore, we use a two step procedure that starts out with a more aggressive regularization. Since articulated models are composed of nearly rigid parts, we expect that the weight function is mostly 0 or 1, except at joints where nearby transformations blend together. So instead of solving for a vector of continuous influence weights at each point, we first solve a *discrete labeling* problem. We find a single label at each cell that indicates which bone the cell belongs to. This setup is illustrated in Figure 5. Later in a weight refinement step, we adjust the weights using a continuous optimization step.

Labeling. In the labeling step we assign a label (a bone index) to each cell. We assign labels to each cell as opposed to each grid point. The reason for this is that in the subsequent optimization we will evaluate a correspondence error for each label. This is straightforward if each label determines a transformation of a cell.

We replace the continuous smoothness term with a discrete version

$$\sum_{(C,D)\in A_G} V(f_C, f_D), \quad (9)$$

where f_C is the bone index (or label) assigned to cell C, and A_G is the set of adjacent grid cell pairs $C, D \in G$. The discrete smoothness term assigns a constant penalty when neighboring cells have different labels:

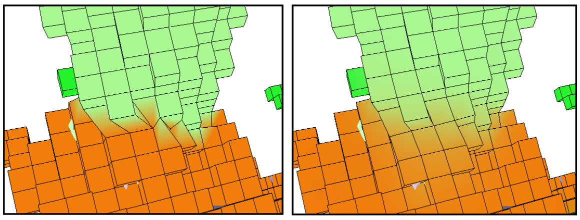
$$V(f_C, f_D) = \begin{cases} 0 & \text{if } f_C = f_D \\ 1 & \text{otherwise.} \end{cases} \quad (10)$$

This term controls the degree to which the labels form contiguous regions over the grid. In addition, it also satisfies E_{norm} and the non-negativity constraint automatically. Thus, the overall optimization objective becomes

$$\underset{f_C \forall C}{\text{argmin}} \ \alpha \sum_{C\in G} W(C, f_C) \ + \ \mu \sum_{(C,D)\in A_G} V(f_C, f_D). \quad (11)$$

Here, $W(C, f_C)$ is the discrete version of E_{dist}, and it is essentially the same as in the continuous case, measuring the point-to-plane distance in each grid cell.

This labeling approach dramatically reduces the complexity of the problem. We solve it efficiently using the graph cuts algorithm. First, we construct an instance of graph cuts where the grid cells $C \in G$ are the sites and adjacent cells



(a) Before weight refinement (b) After weight refinement

Figure 6: *In many cases, discrete labeling causes unnatural deformations between neighboring transformations. With the final weight refinement step, we solve for a smooth blend between neighboring transformations at the joint.*

$(C,D) \in A_G$ are neighbors. We then apply the α-expansion algorithm [BVZ01, KZ04] to find the cell labeling that minimizes this objective. Once the optimal labeling is obtained, we update the weights v_{cj} at each grid point c in each cell C by assigning a binary value

$$v^*_{cj} = \begin{cases} 1 & \text{if } j = f_C \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

where f_C is the label assigned to cell C.

Reusing Labels. If the smoothness weight is set too high, or if some transformations are very similar, the smoothness term will dominate the distance term. In these situations, two separate parts may be labeled with the same bone index. This often results in unused labels that are not assigned to any grid cell, with little chance that the label will be reintroduced. Therefore, we give a second chance for these unused labels by introducing them into regions with the highest registration error. The optimization then proceeds to the T-step, giving a chance to adapt the transformations before the next W-step.

In our implementation, we first find the regions with the highest registration error according to $W(C, f_C)$, and split each region randomly in half. The split is performed in the same fashion as we initialize the weights (Section 4.3). We continue this process until the registration error is below a threshold (typically 0.1 times the vertex sampling distance) or until there are no remaining unused labels.

Weight Refinement. The binary weight assignment according to Equation 12 has the limitation that the surface may deform unnaturally at joints, as illustrated in Figure 6. In order to smooth these regions, we perform an optional weight refinement step by solving for the continuous weights according to the original optimization objective in Equation 8. Since we would like the weights to be as similar as possible to the discrete weights, we add an additional term to Equation 8 that pulls the weights to their discrete counterparts:

$$E_{\text{disc}} = \sum_c \left(v_{cj} - v^*_{cj}\right)^2, \tag{13}$$

where v^*_{cj} is the binary weights defined using the discrete labeling result. We solve the resulting non-negative least squares (NNLS) problem using the approach by Schaefer and Yuksel [SY07]. This technique solves the least squares problem without the non-negativity constraint and repeatedly removes variables with the smallest negative values, effectively forcing their value to zero, until it arrives at a non-negative solution.

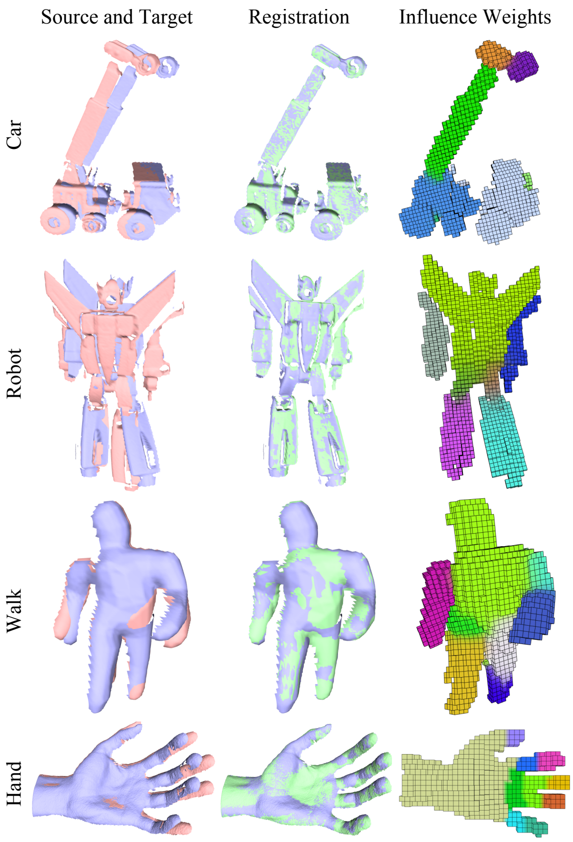


Figure 7: *Some registration examples from the car, robot, walk, and hand datasets. The red shape is the source, the blue shape is the target, and the green shape is the source deformed to match the target.*

4.3. Initialization

To start our algorithm, we automatically initialize the weight functions and find initial correspondences using feature descriptors and RANSAC. The user specifies the total number of bones used to estimate the motion of the range scans. This parameter does not need to be exact, since the graph cuts optimization tends to combine regions and throw away extra labels.

Initial Segmentation. To initialize the weight function, we generate a random segmentation of the range scan, similar to the k-means clustering algorithm [Llo82]. First, we randomly pick initial bone locations for each bone j on the source shape. In this step, we ensure that these locations are sufficiently spaced apart by using best-candidate sam-

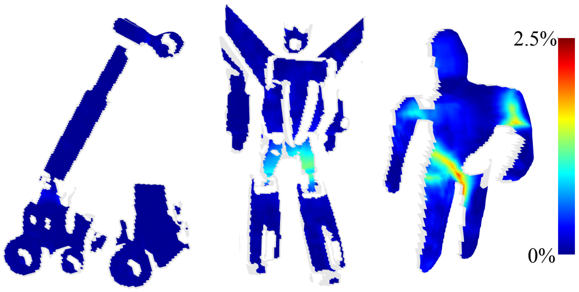


Figure 8: *Registration Error. The color-coded visualization shows the distance to the closest point as a percentage of the bounding box diagonal.*

pling [Mit91]. Then, we initialize binary weight values $\mathbf{v}_c$ at each grid point. To do this, we find the bone j whose location is closest to the grid point, and then we set the j^{th} weight component to be 1 and the rest to 0. Even though this segmentation is often incorrect (e.g. Figure 4c), the subsequent weight optimization is able to adjust the weights to express the motion accurately.

Initial Correspondences Using RANSAC. When the input scans are initially close together, we can use closest point correspondences. To provide better correspondences when the object has moved significantly, we use spin images as a feature descriptor and use Johnson's spin-image matching engine to determine reliable correspondences [Joh97]. We use these initial correspondences for only the first iteration of the T-step and use closest point correspondences for the rest of the optimization.

Since spin images are local descriptors, matching with them alone gives rise to many incorrect correspondences due to repeated shape features. To deal robustly with such outliers, we treat each labeled region independently and perform RANSAC [FB81] to estimate a rigid transformation and remove spurious correspondences. This step proved to be useful in our experiments: on average anywhere from 20% to 40% of correspondence candidates were removed as outliers. In some cases, even 60%∼70% were outliers, most likely because the scan contained a large amount of symmetry or because regions in the initial segmentation overlapped multiple rigid parts. Occasionally this method confused symmetric parts, because it still relies on the comparison of local feature descriptors. However, it provided reasonably accurate starting guesses in almost all of our experiments.

5. Experimental Results

We tested our algorithm with five datasets: one synthetic dataset of a walking figure generated by Pinocchio [BP07], two real datasets of a car and a robot [PG08], a dataset of a moving hand [WLG07], and a dataset of human body scans [ACP02]. Each dataset consists of a sequence of depth scans of a moving object. All tests were performed on a single core of an Intel Core 2 Duo 2.66GHz with 2GB of RAM.

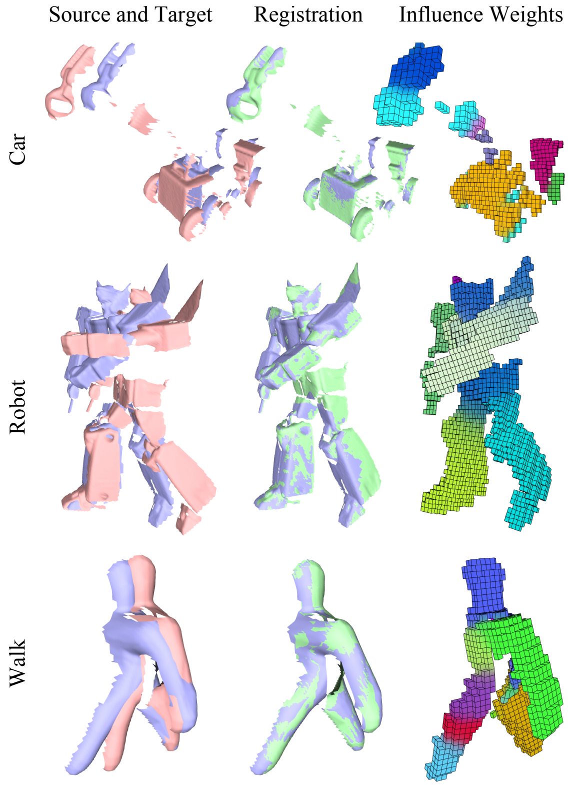


Figure 9: *More difficult registration tests, each with a significant amount of motion and occlusion.*

Figure 7 shows some registered example pairs and the resulting influence weight function on the grid. In the color-coded visualization of the skinning weights, we can see that our algorithm determines intuitive weights for each rigid part. In addition, we were able to obtain an accurate registration with small registration error, visualized for some examples in Figure 8.

Our algorithm also worked well with a severe amount of occlusion in the examples. The car example shown in Figure 9 is completely missing the arm in one of the pairs. Also, one of the robot pairs is completely missing the torso region, and only a small part of the whole surface is observed in the example of the walking figure. Nevertheless, we are able to successfully align all three examples. We also tested our algorithm using a few examples from the human body scan dataset [ACP02], shown in Figure 10. Our method produces accurate registrations, and the quality is comparable to that of previous work [HAWG08]. To test the robustness of our method, we degraded the data by manually cutting holes into the geometry, shown in the bottom of Figure 10. We were able to obtain a successful registration even in this case.

We tested our implementation with pairs of adjacent frames in the datasets. In order to evaluate the registration results, we verified the registration of each pair visually. For the car dataset, out of a total number of 89 registra-

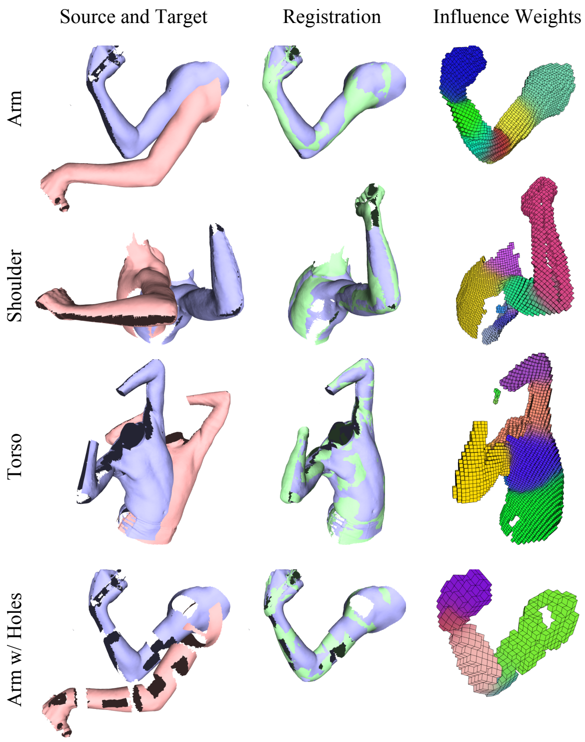


Figure 10: *Registration examples from the human body dataset. The bottom example, a modified version of the top example, shows that our method is robust to missing data.*

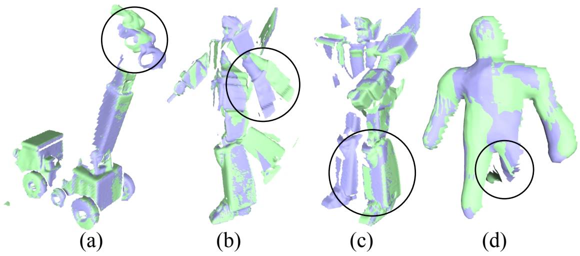


Figure 11: *Less successful registration examples, showing errors due to local minima in (a), an undesired connection between the left wing and left arm in (b), both legs mapping to the left leg in (c), and insufficient data causing misalignment in (d).*

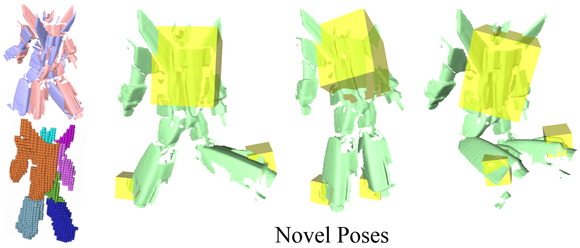


Figure 12: *Interactive posing using our system. The weights (bottom left) were optimized by registering two scans (top left). The user can interactively select and drag constraints on the shape (shown as boxes) to generate a novel pose.*

tion pairs, 85 examples were registered correctly with only 4 pairs exhibiting an objectionable misalignment. The algorithm worked well with the more complex robot dataset as well, with 67 out of 89 pairs registered correctly. In addition, 167 of the 189 pairs were registered correctly in the synthetic walking figure dataset, and 31 out of 46 for the hand dataset. Some results that were less successful are shown in Figure 11. In these cases, the maximum correspondence distance was too low or high, causing local minima or wrong part mappings (Figure 11a,c), the grid resolution was too low or high, causing separate parts to be attached or separated (Figure 11b), or there was too much missing geometry, causing bad part mappings or misalignments (Figure 11d). After retrying with different parameter settings, we were able to reduce the misalignment rate to 0, 4, and 6 examples for the car, robot, and walking figure dataset, respectively. The remaining examples remained problematic because of too much missing data or bad part mappings.

Creating Novel Poses. Since we solve for the skinning weights, our technique is especially useful for creating novel poses. Unlike previous work, we use the output of our algorithm directly to perform both forward and inverse kinematics, either by specifying or solving for the bone transformations. In fact, by substituting the correspondences with user-given constraints, we can use our T-step optimization directly to perform inverse kinematics. We created an interactive application that allows the user to define and manipulate end effectors to create novel poses. Figure 12 shows that interesting poses can be generated using our application.

Parameters. The number of bones, the number of correspondences, the grid resolution, and the maximum correspondence distance for each dataset is reported in Table 1. The grid resolution is specified as the number of divisions along the longest axis of the bounding box, and the maximum correspondence distance is specified as a multiple of the range scan sample spacing. Using a coarse grid resolution covers larger holes between parts. However, if the grid is too coarse, then it becomes harder to localize the weights to smaller parts. Also, a coarse grid may cause undesired parts to be connected, causing difficulties in the registration. For the maximum correspondence distance, a larger distance increases the running time for searching initial correspondences, because a greater number of spin images must be compared. This allows for larger movement, but also may cause similar parts (e.g. arms, legs) to be mapped incorrectly. In addition to the maximum distance, we use a dynamic threshold for the maximum normal angle (decreasing from 80 to 20 degrees) similar to [PG08].

For the weights of each error term, the weight of E_{dist} was $\alpha = 1.0$ for both the T-step and the W-step. For the joint constraint term E_{joint}, we decreased the weight from 1.0 to 0.05 in 5 iterations of the main optimization loop accord-

	Car	Robot	Walk	Hand
Bones	7	7	10	12
Corresp.	1200	1200	1000	1500
Vertices	5389	9377	4502	34342
Max Dist	20	40	20	30
Grid Res.	60	65	50	40
Grid Cells	1107	1295	1014	814
Grid Points	2918	3366	2553	1884
Setup	0.185s	0.234s	0.136s	0.078s
RANSAC	8.089s	20.001s	5.517s	N/A
Align	9.945s	19.644s	23.092s	49.918s
Weight	6.135s	10.713s	10.497s	3.689s
Total Time	24.355s	50.591s	39.242s	53.684s

Table 1: *Average performance statistics for our tests. Timings (in seconds) represent the total time spent in each stage.*

ing to the function $e^{k\ln(0.05)/5}$, where k is the iteration number starting from 0. This allows the registration to be refined precisely during the latter part of the optimization. For the weight smoothness term E_{smooth}, a constant between 0.5 and 1.0 times the grid spacing size worked well. For E_{norm} we used $\nu = 1.0$, and for E_{disc} we chose a value between 0.1 and 0.5 depending on how close we wanted to be to the discrete weights. Finally, for generating the spin images, we set bin size equal to the grid spacing, the image size to 15, and the maximum support angle to 90 degrees.

Performance. We report averaged statistics and timings of our method in Table 1 using our implementation. The setup time is spent creating and initializing the data structures, the RANSAC step generates the initial correspondences using spin image matching, the align time is the total time the algorithm processes the main loop, and the weight time is spent performing the final NNLS weight refinement step.

6. Discussion and Future Work

Our algorithm is currently limited to registering pairs of shapes. A promising area for future work is to extend it to the multiple scan case. This would allow us to reconstruct a complete 3D articulated model. Preliminary experiments show that accumulating the geometry successively in each frame causes registration error to also accumulate and throw off the registration. Formulating a simultaneous optimization could avoid this problem.

We are currently using the LBS deformation model for its simplicity. In case LBS exhibits artifacts, such as the well known "candy wrapping" effect, our method could be adapted to optimize other RDMs, for example dual-quaternion blending [KCZO07]. However, this would likely make the optimization procedure more computationally expensive.

Our method requires sufficient separation between different parts of the shape in the source scan. If two parts are joined together, this causes the grid cells to be connected between these parts, and the joint constraint will constrain the parts to move together. In the future, we plan to address these topological issues by using a spatially varying smoothness weight that allows the deformation to disregard problematic joints. Also it would be interesting to use a multiresolution hierarchical grid to represent the skinning weights.

We believe that improving the initial correspondences will make the algorithm more robust to problematic cases. Using a more discriminative feature detection, or using spectral clustering (as proposed by Huang et al. [HAWG08]) could help our algorithm to avoid local minima and wrong part mappings in the registration. Our discrete labeling step also resembles clustering algorithms [CSAD04, HAWG08], which could be used as an alternative to the graph cut optimization.

7. Conclusions

We have presented a method to register deforming range scans by modeling the motion with a reduced deformable model (RDM). We have chosen linear blend skinning for its simplicity, but more sophisticated approaches could be used if necessary. A key idea of our approach is to represent the weight functions on a 3D grid surrounding the scanned geometry. This allows us to apply linear blend skinning (LBS) to range scans with occlusions and missing data. We solve for the parameters of the deformation model using an EM-type algorithm.

We have demonstrated that our approach is able to register articulated shapes robustly and with significant occlusions and missing data. The advantages of our technique are that it does not require any manual segmentation, user specified markers, nor a surface template. We believe that our work is a significant step forward in automatically reconstructing fully rigged 3D articulated models from range scans.

Acknowledgements. We wish to thank Y. Pekelny and C. Gotsman for sharing the car and robot datasets, T. Weise for the hand dataset, I. Baran for the walking figure dataset, and B. Allen for the body scan data. Additional thanks to G. Debunne for providing the libQGLViewer library, D. M. Mount and S. Arya for providing the ANN library, and Y. Boykov, O. Veksler, R. Zabih for providing an implementation of their graph cuts algorithm. Also we thank Q.-X. Huang for kindly providing us with comparisons and H. Li for providing feedback. Finally we thank the anonymous reviewers and the members of the UCSD Vision and Graphics Labs for their valuable feedback.

References

[ACP02] Allen B., Curless B., Popović Z.: Articulated body deformation from range scan data. *ACM SIGGRAPH 21*, 3 (2002), 612–619.

[ACP03] Allen B., Curless B., Popović Z.: The space of human body shapes: reconstruction and parameterization from range scans. In *ACM SIGGRAPH* (2003), pp. 587–594.

[AKP*04] Anguelov D., Koller D., Pang H., Srinivasan P., Thrun S.: Recovering articulated object models from 3d range data, 2004.

[ARV07] AMBERG B., ROMDHANI S., VETTER T.: Optimal step nonrigid icp algorithms for surface registration. In *CVPR* (2007).

[ASP*04] ANGUELOV D., SRINIVASAN P., PANG H.-C., KOLLER D., THRUN S., DAVIS J.: The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces. In *NIPS* (2004).

[BM92] BESL P. J., MCKAY H.: A method for registration of 3-d shapes. *PAMI 14*, 2 (1992), 239–256.

[BP07] BARAN I., POPOVIĆ J.: Automatic rigging and animation of 3d characters. In *ACM SIGGRAPH* (2007), p. 72.

[BPWG07] BOTSCH M., PAULY M., WICKE M., GROSS M.: Adaptive space deformations based on rigid cells. *EUROGRAPHICS 26*, 3 (2007), 339–347.

[BR07] BROWN B. J., RUSINKIEWICZ S.: Global non-rigid alignment of 3-d scans. In *ACM SIGGRAPH* (2007), p. 21.

[BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *PAMI 23*, 11 (2001), 1222–1239.

[CM91] CHEN Y., MEDIONI G.: Object modeling by registration of multiple range images. *Proceedings of the IEEE International Conference on Robotics and Automation 3* (1991), 2724–2729.

[CR03] CHUI H., RANGARAJAN A.: A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding 89*, 2-3 (2003), 114–141.

[CSAD04] COHEN-STEINER D., ALLIEZ P., DESBRUN M.: Variational shape approximation. In *ACM SIGGRAPH* (2004), pp. 905–914.

[CZ08] CHANG W., ZWICKER M.: Automatic registration for articulated shapes. *Computer Graphics Forum (Proceedings of SGP) 27*, 5 (2008), 1459–1468.

[dATTS08] DE AGUIAR E., THEOBALT C., THRUN S., SEIDEL H.-P.: Automatic conversion of mesh animations into skeleton-based animations. *Computer Graphics Forum 27*, 2 (2008), 389–397.

[DLR77] DEMPSTER A. P., LAIRD N. M., RUBIN D. B.: Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological) 39*, 1 (1977), 1–38.

[FB81] FISCHLER M. A., BOLLES R. C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *CACM 24*, 6 (1981), 381–395.

[HAWG08] HUANG Q.-X., ADAMS B., WICKE M., GUIBAS L. J.: Non-rigid registration under isometric deformations. *Computer Graphics Forum (Proceedings of SGP) 27*, 5 (2008), 1449–1457.

[HTB03] HÄHNEL D., THRUN S., BURGARD W.: An extension of the icp algorithm for modeling nonrigid objects with mobile robots. In *IJCAI* (2003), pp. 915–920.

[Joh97] JOHNSON A.: *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, August 1997.

[JT05] JAMES D. L., TWIGG C. D.: Skinning mesh animations. In *ACM SIGGRAPH* (2005), pp. 399–407.

[KCZO07] KAVAN L., COLLINS S., ZÁRA J., O'SULLIVAN C.: Skinning with dual quaternions. In *I3D* (2007), pp. 39–46.

[KJP02] KRY P. G., JAMES D. L., PAI D. K.: Eigenskin: real time large deformation character skinning in hardware. In *SCA* (2002), pp. 153–159.

[KZ04] KOLMOGOROV V., ZABIH R.: What energy functions can be minimizedvia graph cuts? *PAMI 26*, 2 (2004), 147–159.

[LCF00] LEWIS J. P., CORDNER M., FONG N.: Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *ACM SIGGRAPH* (2000), pp. 165–172.

[Llo82] LLOYD S.: Least squares quantization in pcm. *IEEE Transactions on Information Theory 28*, 2 (1982), 129–137.

[LSP08] LI H., SUMNER R. W., PAULY M.: Global correspondence optimization for non-rigid registration of depth scans. *Computer Graphics Forum (Proceedings of SGP) 27*, 5 (2008), 1421–1430.

[MFO*07] MITRA N. J., FLORY S., OVSJANIKOV M., GELFAND N., GUIBAS L. J., POTTMANN H.: Dynamic geometry registration. In *SGP* (2007), pp. 173–182.

[MG03] MOHR A., GLEICHER M.: Building efficient, accurate character skins from examples. *ACM SIGGRAPH* (2003), 562–568.

[Mit91] MITCHELL D. P.: Spectrally optimal sampling for distribution ray tracing. *SIGGRAPH* (1991), 157–164.

[PG08] PEKELNY Y., GOTSMAN C.: Articulated object reconstruction and markerless motion capture from depth video. *EUROGRAPHICS 27*, 2 (2008).

[PMG*05] PAULY M., MITRA N. J., GIESEN J., GROSS M., GUIBAS L. J.: Example-based 3d scan completion. In *SGP* (2005), p. 23.

[SAL*08] SHARF A., ALCANTARA D. A., LEWINER T., GREIF C., SHEFFER A., AMENTA N., COHEN-OR D.: Space-time surface reconstruction using incompressible flow. *ACM SIGGRAPH ASIA* (2008).

[SCFRC01] SLOAN P.-P. J., CHARLES F. ROSE I., COHEN M. F.: Shape by example. In *I3D* (2001), pp. 135–143.

[SSP07] SUMNER R. W., SCHMID J., PAULY M.: Embedded deformation for shape manipulation. In *ACM SIGGRAPH* (2007), p. 80.

[SWG08] SÜSSMUTH J., WINTER M., GREINER G.: Reconstructing animated meshes from time-varying point clouds. *Computer Graphics Forum (Proceedings of SGP) 27*, 5 (2008), 1469–1476.

[SY07] SCHAEFER S., YUKSEL C.: Example-based skeleton extraction. In *SGP* (2007), pp. 153–162.

[WJH*07] WAND M., JENKE P., HUANG Q., BOKELOH M., GUIBAS L. J., SCHILLING A.: Reconstruction of deforming geometry from time-varying point clouds. In *SGP* (2007), pp. 49–58.

[WLG07] WEISE T., LEIBE B., GOOL L. J. V.: Fast 3d scanning with automatic motion compensation. In *CVPR* (2007).

[WP02] WANG X. C., PHILLIPS C.: Multi-weight enveloping: least-squares approximation techniques for skin animation. In *SCA* (2002), pp. 129–138.

[WPP07] WANG R. Y., PULLI K., POPOVIĆ J.: Real-time enveloping with rotational regression. In *ACM SIGGRAPH* (2007), p. 73.