Consensus Skeleton for Non-rigid Space-time Registration

Q. Zheng[1] and A. Sharf[1] and A. Tagliasacchi[2] and B. Chen[1] and H. Zhang[2] and A. Sheffer[3] and D. Cohen-Or†[4]

[1] Shenzhen Institutes of Advanced Technology (SIAT), Chinese Academy of Sciences
[2] Simon Fraser University
[3] University of British Columbia
[4] Tel-Aviv University

Abstract

We introduce the notion of consensus skeletons *for non-rigid space-time registration of a deforming shape. Instead of basing the registration on point features, which are local and sensitive to noise, we adopt the curve skeleton of the shape as a global and descriptive feature for the task. Our method uses no template and only assumes that the skeletal structure of the captured shape remains largely consistent over time. Such an assumption is generally weaker than those relying on large overlap of point features between successive frames, allowing for more sparse acquisition across time. Building our registration framework on top of the low-dimensional skeleton-time structure avoids heavy processing of dense point or volumetric data, while skeleton consensusization provides robust handling of incompatibilities between per-frame skeletons. To register point clouds from all frames, we deform them by their skeletons, mirroring the skeleton registration process, to jump-start a non-rigid ICP. We present results for non-rigid space-time registration under sparse and noisy spatio-temporal sampling, including cases where data was captured from only a single view.*

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—object representation, regisration and deformation

1. Introduction

Space-time shape reconstruction of a deforming object from scanned point clouds has been an intensely studied problem in computer graphics and geometry processing recently. An essential and particularly challenging sub-problem is that of space-time registration of the captured shapes across all time frames. Existing techniques work well in the static setup, when the object remains still or rigid during the scanning process and the scans do not incur large amount of missing data. However the problem becomes much more challenging in the dynamic setup with freeform deformation of the scanned object over time [MFO*07, dAST*08, SAL*08, VBMP08, WAO*09], sparse camera views [PG08, LAGP09, LZW*09] or temporal sampling [CZ09]. As a result, significant data gaps, both over time and space, as well as data noise and outliers, can all occur.

In general, if nothing is known *a priori* about the space-time behavior of the scanned object, these problems are effectively intractable, as nothing can be assumed about the shape correlation across time. Hence, a challenge we face is to define a set of assumptions that is sufficient to facilitate correct registration, yet general enough to be applicable to the wide variety of objects we are interested in acquiring. Beyond the capabilities of the registration method, as dictated by the assumptions adopted, the main emphases are placed on efficiency and robustness of results.

Existing algorithms for non-rigid space-time shape registration make varying assumptions. Some rely on geometric or topological priors provided by an *a priori* template [BC08, dAST*08, PG08, VBMP08, LAGP09]. Such a strong constraint allows one to handle highly sparse data, even those acquired from a single camera [PG08] or view [LAGP09]. Some methods are tailor-made for specific classes of shapes only [ASK*05, BPS*08]; some utilize specific knowledge on the type of deformations (e.g., piecewise

† work was performed while visiting SIAT

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

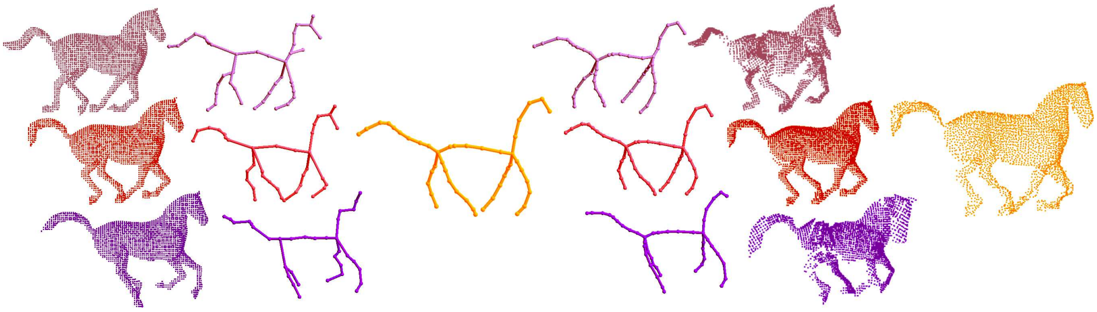

Figure 1: *Overview of our consensusization workflow. Left to right: From an initial set of deforming point clouds, we extract skeletons per frame. We compute the consensus skeleton (middle) and deform it back to the frames' poses. Using the skeleton correspondence, we can deform the point clouds into a common consensus pose and register them together (right).*

rigidity [PG08]) the scanned object can undergo; some assume that the exact shape of the object in some frame is known [SWG08]. Others use much weaker priors [MFO*07, WJH*07,SAL*08,WAO*09]. While more general, these latter methods often require fairly dense spatio-temporal sampling and rely on heavy processing of point [WJH*07] or volumetric [SAL*08, WAO*09] data.

We aim to find a middle-ground, namely an assumption which is both general (no template or piecewise rigidity constraints) and sufficiently robust to facilitate the registration task, even under sparse spatio-temporal acquisition. Our method is inspired by the observation of Sharf et al. [SAL*08] that for a large variety of objects, their volume is *incompressible* during motion. Analogically, the skeletal structure of the objects in motion, including the number and lengths of branches, their connectivity, and associated radius distributions, remains largely consistent over time, though each branch can deform freely. This weak assumption holds true for many classes of objects, including articulated shapes such as humans or animals, yet as we show in this paper is powerful enough to facilitate registration of deforming objects from sparse data.

Based on our skeleton consistence assumption, we develop a non-rigid space-time registration algorithm that is *skeleton-driven*. Instead of basing the registration on point features, which are local and sensitive to noise, as in previous works [WJH*07, dAST*08, WAO*09, LZW*09], we adopt the curve skeleton of a shape as a global and descriptive feature for the task. The simplicity of curve skeletons and their ability to provide effective shape abstractions make them attractive to use in a registration framework (see Figure 1 for an overview).

Given a sequence of point clouds acquired over time, we first extract per-frame skeletons and then consolidate them into a skeleton structure that is consistent across time and accounts for all the frames. Since each per-frame skeleton may be incomplete and error-prone, the focal point of our algorithm is the construction of a *consensus skeleton*, or *c*-skeleton for short. It implies the topology of the captured shape and allows for completion of data missing at different points in time based on information available at other time frames. The *c*-skeleton is computed after corresponding and warping all the per-frame skeletons into a common pose. This allows for a consensusization by removing outlier skeleton branches. The subsequent point cloud registration over time is skeleton-driven and via non-rigid ICP, resulting in a consensus point cloud to facilitate shape completion.

The main contribution of our work is the introduction and computation of *c*-skeletons for non-rigid space-time shape registration. The *c*-skeleton statistically combines shape information gathered over time and provides effective handling of imperfections within and incompatibilities between per-frame skeletons, as shown in Figure 3. These artifacts in the extracted skeletons are inherited from the captured point clouds which can differ due to variations in views and occlusions, as shown in Figure 2.

By reducing the problem of explicit shape correlation across a sequence of 3D point clouds to correlation among 1D skeletons, we drastically simplify the most expensive component of space-time registration, that of finding a global consensus shape. With the consensusization step, we also alleviate the error propagation problem in approaches which purely rely on pairwise correspondences.

Finally, the use of curve skeletons, a compact and clean form of shape representation, leads to more robust inference of the shape topology, placing less demand on the temporal sampling rate. Indeed, non-rigid registration schemes which operate on point sets or surface patches either rely on feature points as anchors or require sufficient overlapping regions between registered geometries to ensure accurate registration between consecutive frames. In contrast, we adopt the generally weaker assumption that adjacent per-frame *skeletons* have sufficient overlap. Furthermore, the *c*-skeleton can provide a good initialization and serve as a reference frame to facilitate ICP-based point cloud registration.

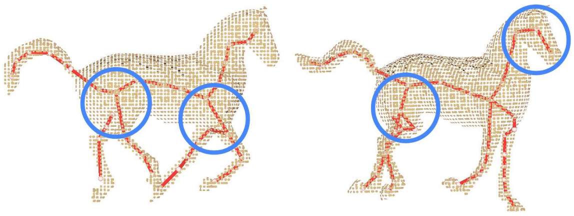

Figure 2: *Differing point clouds over time result in topological and geometric incompatibilities in the extracted skeletons (circled in blue).*

2. Related work

Most solutions to the space-time registration problem rely on an *a priori* shape template [ATD*08, BC08, dAST*08, PG08, SWG08, VBMP08, LAGP09] and deform the template to fit the acquired geometries across all frames. The template defines the topology and sometimes also the coarse geometry of the captured shape. With this crucial prior in hand, these methods perform well even with significant missing data [PG08, LAGP09]. The template can be a coarse mesh [ATD*08, LAGP09], a high-resolution full-body scan [dAST*08], a skinned mesh [BC08, VBMP08], or a skeleton [PG08]. It can be synthesized, learned [ASK*05], or from a perfect reconstruction, often as the first frame. In the latter case [PG08, SWG08], the registration process typically accumulates information only forward in time.

Registration methods which do not use a template include [MFO*07, WJH*07, LZW*09, WAO*09]. They can handle general shapes and deformations, but assume that successive scans are under small deformation and have large over-lapping regions to ensure adequate feature correspondences between consecutive frames. For large deformations, dense spatio-temporal sampling is required. Sharf et al. [SAL*08] base their 4D reconstruction on a volume-time structure and model material flow, assuming the object to be incompressible, but do not provide temporal correspondences. Finding temporal point correspondences is indeed a challenging problem, to the point that some techniques resort to certain level of user intervention [ASK*05, LZW*09] or assistance from video [LZW*09]. The above methods all use some form of global or semi-global optimization to find a globally consistent point registration, but the need to process dense point clouds [MFO*07, WJH*07] or volumetric data [SAL*08, WAO*09] leads to costly computations.

Recent work of Chang and Zwicker [CZ09] uses a linear skinning model to drive a non-rigid registration scheme. Their decoupling of the deformation model from the surface representation shares some similarity with our skeleton-driven approach, allowing for registration under significant motion and occlusion. However, similar to other related techniques [HAWG08, LSP08, CZ08] recently proposed in the context of space-time shape reconstruction, their registration is performed only in a piecewise manner.

Our primary non-rigid registration task is performed on a light skeleton-time structure, instead of relying on point features. We start by computing pairwise skeleton correspondences and then construct a globally consistent *c*-skeleton, using a global alignment scheme similar to the one used by Liao et al. [LZW*09] but applied to skeletons. The skeletons we work with are highly compact (with number of nodes up to 80), greatly improving the efficiency of consensus computation. In contrast, point clouds under typical problem settings often contain tens of thousands of points or more.

There have been works on skeleton extraction from animated mesh sequences [JT05, dATTS08] or sample mesh poses [SY07], utilizing the piecewise rigidity of the motion. Theobalt et al [TdAM*04] extract a hierarchical skeleton from volume data reconstructed from multi-view video, assuming small piecewise rigid motion of the capture object. In the static setting, Tagliasacchi et al. [TZCO09] utilize a rotational symmetry prior for curve skeleton extraction from incomplete point clouds. However, for general shapes and with sparse camera views, per-frame skeletons obtained using their method can still be erroneous. The *c*-skeleton algorithm we develop corrects these errors (Figure 3), without relying on the piecewise rigidity assumption.

3. Overview of skeleton consensusization algorithm

We take as input a sequence of partial point clouds captured over time. First, for each frame, we extract a curve skeleton from the point cloud therein using the algorithm of Tagliasacchi et al. [TZCO09], where the set of parameters chosen is fixed throughout. The rest of our consensus skeleton algorithm can be divided into three distinct steps: pairwise skeleton correspondence (Section 4), multi-skeleton registration (Section 5), and skeleton consensusization (Section 6). Figures 1 and 13 demonstrate the whole pipeline. We now describe these steps briefly.

Pairwise skeleton correspondence Given the series of per-frame skeletons, we first perform an initial clean-up and resample along each skeleton to regularize the node distributions. To tolerate possible imperfections in the individual skeletons and incompatibilities between them, we need a probabilistic approach for pairwise skeleton correspondence. To this end, we develop a Hidden Markov Model (HMM) technique, which is applied to linearized versions of two skeletons from adjacent frames. Since linearization necessitates the choice of a root node in each skeleton, to account for this, we first build an initial set of correspondences between skeleton nodes using different linearization. We then extract the most consistent subset of skeleton node correspondences via spectral analysis [LH05], which filters out inconsistent correspondence pairs.

Multi-skeleton registration Pairwise correspondences between skeleton nodes serve as soft constraints to drive a

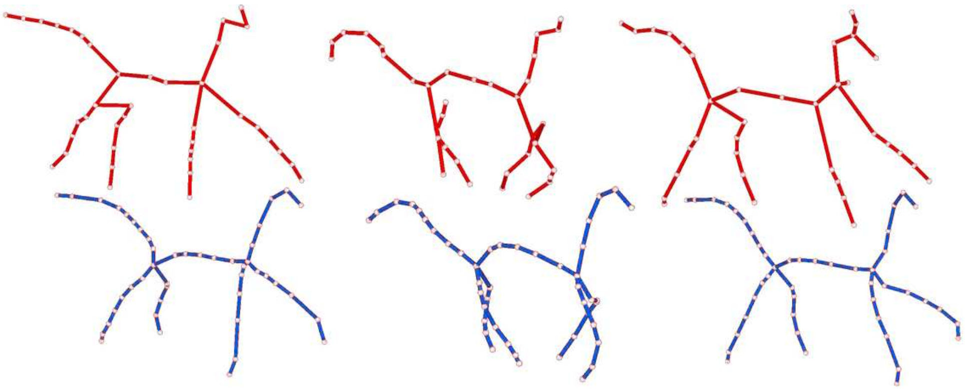

Figure 3: *Consensus skeletons of a horse model. Top: curve skeletons extracted independently from three frames show a variety of incompatibilities, even topological errors (hind legs in the middle frame and neck in the right frame). Bottom: the computed consensus skeleton is deformed back into the poses of the per-frame skeletons.*

global, Laplacian-based deformation process which non-rigidly registers multiple skeletons into a common pose while preserving their local geometry. We modify the classical Laplacian deformation framework [SLCO*04, LZW*09] appropriately to apply to curve skeletons. The issue of rotation handling with Laplacian deformations is dealt with by extending the feature-based image metamorphosis technique of Beier and Neely [BN92] to the 3D setting.

***c*-skeleton construction** Once all skeletons are aligned, corresponding nodes are matched and unified via mean-shift clustering. A new intermediate weighted graph is constructed whose nodes are the cluster centers and whose edges inherit connections between nodes in the skeletons. Appropriate weights are associated to the graph elements to account for how frequent their corresponding skeleton nodes and edges appear in all the frames. The *c*-skeleton is obtained by removing nodes and edges from the graph that are deemed to be infrequent, through an optimization.

4. Pairwise skeleton correspondence

Given two skeletons S_1, S_2, we would like to build a *partial* correspondence between them by pairing up a subset of nodes from each skeleton. Indeed, the skeletons typically do not fully correspond and may contain missing parts on one hand and excessive parts (outliers) on the other. Our method is based on an efficient implementation of the Hidden Markov Model (HMM) which computes an optimal correspondence sequence between two linear sequences of elements.

To linearize a skeleton, we use a depth first traversal of its graph. In our current implementation, we assume that the graph of each curve skeleton is acyclic. Any cycle detected is broken in an arbitrary manner. To execute depth first traversal, however, we are required to set one node in each skeleton as a root, or more precisely, two corresponding root nodes. Since no corresponding roots are given, we employ a multi-pass algorithm in which the HMM algorithm is activated, each pass from different corresponding roots. The multi-pass algorithm generates a large number of corresponding pairs of nodes, most of which agree with each other, but some do not. Thus, in a second step, a consistent subset of corresponding pairs is generated out of the pool of results obtained from the multi-pass HMM step.

Sampling The nodes of a given skeleton are typically sparse and unevenly distributed, which prevent proper correspondence computation. Thus, prior to applying the HMM algorithm, we first remove any skeleton node of the two candidate skeletons if the two adjacent bones either closely form a straight line or one of them is too short. After proper node removal, the skeleton is up-sampled by adding nodes along the skeleton to ensure proper length of each bone.

Hidden Markov Model (HMM) Given a correspondence metric between skeletons, we wish to find a sequential solution which minimizes the correspondence cost. HMM is an appropriate state-space cost minimization for linear sequences. It implicitly considers all possible correspondence assignments by using dynamic programming which breaks the problem into successive stages, where each stage is only dependent on the immediately proceeding stage. We use the basic HMM dynamic programming algorithm, the Viterbi algorithm [Rab89], to compute an optimal correspondence sequence between two skeletons.

In an HMM, the input is a sequential series of observed states, state-to-state transition probabilities, and state-to-observation emission probabilities. The goal is to infer the corresponding sequence of hidden states that is most likely to have generated these observations. In our context, each state is a correspondence pair from a source node to a target

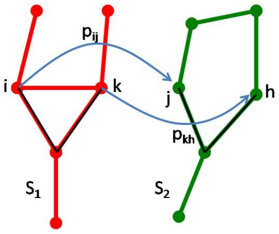

Figure 4: *Distance similarity between pairs (i, j) and (h, k). The chosen paths in both skeletons (blackened) have an equal distance similarity length of two.*

node. As the HMM method deals with a sequence, we set a starting (root) node and traverse each skeleton using DFT. Our emission probability measures the degree to which two paring nodes are consistent, or in other words, how likely two given nodes are in correspondence. The transition probability measures the likelihood that two corresponding pairs are adjacent to each other. See Appendix for details on our HMM formulation.

Correspondence filtering Simply taking all correspondences generated from all the source nodes could lead to inconsistencies among the correspondence pairs. We must filter such inconsistencies to ensure robustness.

We define the consistency between two correspondences p_{ij} and p_{kh} as a weighted sum of three terms:

$$c_{p_{ij},p_{kh}} = w_D D(p_{ij}, p_{kh}) + w_T T(p_{ij}, p_{kh}) + w_A A(p_{ij}, p_{kh}),$$

where the first term $D(p_{ij}, p_{hk}) = min(\frac{d_1(i,k)}{d_2(j,h)}, \frac{d_2(j,h)}{d_1(i,k)})$ represents distance similarity and is used to measure invariability of bone lengths in skeletons S_1 and S_2; here $d_1(i,k)$ is the path distance between the i-th and the k-th nodes in skeleton S_1 (similarly for $d_2(j,h)$). This term is used to deal with loops where multiple paths exist for an edge. Here path distance is defined as the sum of squared Euclidean distances of all edges along the path. We compute all possible paths between two nodes and separately choose a path from S_1 and a path from S_2 that make $D(p_{ij}, p_{hk})$ the largest.

The second term $T(p_{ij}, p_{kh})$ represents the similarity of topological changes along the two paths, from u_1^i to u_1^k and from u_2^j to u_2^h. We use $c_1(i,k)$ to measure topological change of the chosen path from the i-th node to the k-th node in skeleton S_1. As removing a node with degree two does not change skeleton topology, $c_1(i,k)$ is equal to the number of nodes with degree larger than two in the chosen path.

The last term $A(p_{ij}, p_{kh}) = 1 - \beta/\pi$ measures the direction similarity of the two vectors, $\overline{u_1^i u_2^j}$ and $\overline{u_1^k u_2^h}$, where β is the angle between them.

The weights of the above three terms are user defined and we use $w_D = 0.7, w_T = 0.1, w_A = 0.1$ in all experiments.

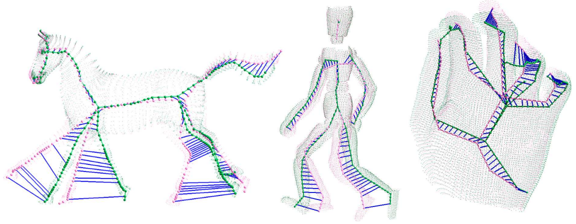

Figure 5: *Results of pairwise skeleton correspondence for several models: horse, humans, hand, and sweaters. The correspondences have high quality even for skeletons with noises and cycles.*

Similar to the work of non-rigid registration of Huang et al. [HAWG08], the initial correspondences (or the union set) are transformed to a spectral domain, where high quality correspondences are selected [LH05]. We first construct a covariance matrix M whose xy-th entry measures the consistency of the x-th (p_{ij}) and the y-th (p_{hk}) pairs of correspondence. The entries of M are defined as follows:

$$M_{xy} = \begin{cases} (\frac{c_{xy}-0.4}{0.6})^{0.8} & \text{if } c_{xy} > 0.4, \\ 0 & \text{otherwise.} \end{cases}$$

Here c_{xy} is as defined above; $M_{xy} = 0$ if the x-th and y-th correspondence pairs are not consistent. The x-th entry of the principal eigenvector of M gives the consistence score for the x-th correspondence pair. For more details of this algorithm, we refer the reader to the original paper [LH05].

We iteratively move a correspondence (i, j) featuring the highest score from the union set to the consistence set of correspondences, and remove from the union set correspondences that are not consistent with the correspondence (i, j). The process stops until the union set is empty.

In Figure 5, we show some results of pairwise skeleton correspondence. We see that with noise, large deformations (e.g., the horse example), and even cycles in the original skeletons, (e.g., the hand examples), our correspondence method works quite robustly. In particular, the ability of our method to tolerate large poses changes, as opposed to registration based on matching point features, highlights an advantage of the skeleton-based approach.

5. Multi-skeleton registration

Having obtained the pairwise skeleton correspondences, we next perform non-rigid registration of the multiple per-frame skeletons. The multi-skeleton registration is based on a simultaneous warping of the skeletons. The basic warp mechanism is based on the Laplacian warping technique [SLCO*04], where pairwise skeleton correspondences are used to constrain the system. Note, however, that the Laplacian of a vertex in a skeleton is defined by its 1-ring neigh-

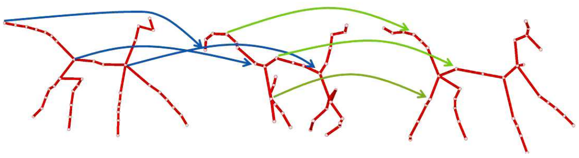

Figure 6: *Simultaneous global alignment of skeletons. The corresponding pairs do not necessarily have the same common nodes across the sequence. However, the Laplacian warpings force all parts to agree to the set of corresponding pairs, leading to a global registration with respect to a reference skeleton.*

borhood, which is a degenerated ring consisting of only its adjacent vertices along the skeleton graph.

To align all the skeletons in the set, we apply a global alignment in the spirit of [LZW*09]. The idea is to warp simultaneously all the skeletons such that all corresponding pairs agree on their final location. An arbitrary skeleton is picked as a reference frame to set hard constraints for its vertices. As shown in Figure 6, the corresponding pairs do not necessarily have common nodes along the sequence. However, the Laplacian warpings force all parts to agree to the set of corresponding pairs, leading to a global non-rigid registration with the reference skeleton.

For the global alignment problem, we solve the following system for node positions:

$$argmin_{U'} E_L(U') + E_P(U')$$

$$E_L(U') = \sum_{1 \leq f \leq F} \sum_{i \in N_f} \|L(u^i_f) - L({u'}^i_f)\|^2$$

$$E_P(U') = \sum_{(u^i_{f_s}, u^j_{f_t}) \in P} \|{u'}^i_{f_s} - {u'}^j_{f_t}\|^2 + \sum_{i \in N_r} \|u^i_r - {u'}^i_r\|^2,$$

where U' represents new node position, F is the set of all input frames, N_f is the total number of nodes in frame f, $L(n)$ gives the Laplacian coordinate of node n, and n^i_f denotes the i-th node of the f-th skeleton. In addition, r is the index of the reference frame; the second component of $E_P(U')$ is there to ensure non-deformation of the reference frame.

Rotating the Laplacians Recall that Laplacians are not rotation invariant and in general the method [SLCO*04] allows only rather small rotations. To improve the performance of the Laplacian warping we estimate the rotation of the Laplacian vectors based on a space-deformation. We build upon the feature-based image metamorphosis technique of Beier and Neely [BN92] and extend it to 3D where the skeleton bones serve as the constraining vectors.

In the Beier and Neely technique, every vector defines a local coordinate system, and the coordinate of a point in the 2D space is defined by a weighted average of the local coordinates defined by the vectors. However, our vectors, defined by the skeletal bones, are in 3D, and cannot define a coordinate uniquely. Thus, we define the local coordinate system by two consecutive bones (the vertices of the bones are in correspondence pairs as well). Each pair of bones defines a plane, and a point in 3D has a unique relative coordinate over the plane. The third coordinate is given by the length of the perpendicular projection from the point to the plane. Thus, we calculate the 3D projection coordinate of the target node.

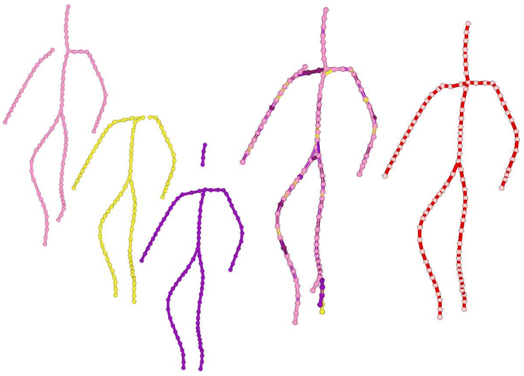

Figure 7: *Computing a c-skeleton from three registered skeletons of running Ben. Left to right: the noisy skeletons in different colors registered in a common pose, their superimposed skeleton and the unified c-skeleton after clustering and removing outliers.*

Each point in the 2D space is influenced by its control feature lines with different weights, determined by the Euclidean distance to the point. Instead of using lines as control primitives, we warp a skeleton by pairs of planes defined by consecutive pairs of bones, which define the source and target positions of the three points defining the local plane.

6. Skeleton consensusization

Once all skeletons are aligned, corresponding nodes and bones are matched and unified via clustering. Then we would like to discard spurious parts in the skeletons that appear only in a rather insignificant number of frames; this usually means that they are outliers. To this end, the appearance frequency of the skeleton bones is measured, indicating their confidence (or popularity) across all frames.

Skeleton clustering Since the registered skeletons are not perfectly aligned, we first cluster nearby nodes, unifying all the skeletons into one coherent structure. We use mean-shift clustering with a Gaussian kernel $\theta(\gamma) = e^{-\gamma^2/h^2}$, where γ is the Euclidean distance between two nodes and h is a constant. If $\gamma \geq 4h$, $\theta(\gamma)$ can be considered as zero. We assign appropriate γ values to either encourage (e.g., for corresponding pair) or penalize (e.g., for two nodes of the same skeleton) clustering two nodes together.

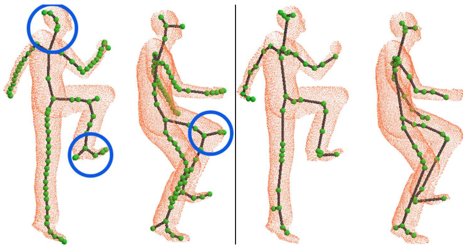

Figure 8: *Results of running Ben. Left: input scans of two poses with independently extracted initial skeletons. Right: consensus skeletons deformed to individual poses.*

The result of the mean-shift clustering process is a “union" of the skeletons into a graph from which the final *c*-skeleton will be computed. The nodes of the graph are cluster centers and they later define the nodes of the *c*-skeleton. Connectivity between the graph nodes is defined by, or more specifically, inherited from, the connectivity between skeleton nodes corresponding to the graph nodes. The graph is weighted by the *confidence* or popularity of the nodes and edges. The confidence given to a graph node is the size of its corresponding cluster of skeleton nodes, while the confidence of an edge is measured by the frequency of appearance of its corresponding skeleton bones across all frames.

Outlier removal The unified skeleton, the graph defined above, requires further pruning to delete outlier nodes and branches. Outliers are graph edges or nodes with low confidence. Algorithmically, we modify the edge weights in the graph so that the search for the *c*-skeleton can be solved by a constrained minimum-weight spanning tree problem. Specifically, we define the modified weight at an edge *e* by:

$$\psi(e) = \varepsilon - l(e) \cdot c(e),$$

where $l(e)$ is the Euclidean length of e, $c(e)$ is the confidence value at e, and ε is a scaling parameter to ensure that all the edge weights are positive. The outliers are then removed by building a connected minimum spanning tree over nodes with sufficiently high confidence.

7. Results

In this section, we show results of our skeleton consensusization algorithm, as well as skeleton-driven non-rigid point cloud registration. We have experimented with real data, e.g., dancing mannequins (Figure 10) acquired by a structured light scanner, and on synthetic models using a virtual scanner. We scanned each model from one or two views per frame, which resulted in an imperfect point cloud for which the individual extracted skeletons were also rather imperfect. Such imperfections can be observed from numerous examples shown in the paper, e.g., in the top rows of Figure 1 and Figure 10, as well as in Figures 2, 7, and 8.

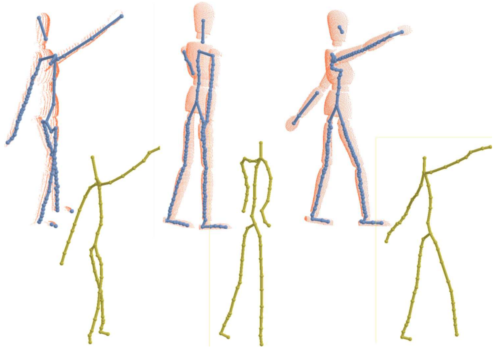

Figure 9: *Handling sparse temporal sampling. Top: three consecutive frames captured for the mannequin under very sparse temporal sampling and the extracted skeletons. Bottom: the consensus skeletons computed and warped to the poses of the individual skeletons.*

Model	Size	*c*-skeleton	*c*-point cloud	frames	pre/post process
mannequin	9k	8.2s	28.6s	13s	34.4s/156s
horse	4k	8.6s	9.6s	12s	35.2s/95.7s
ben	41k	14.1s	59.8s	21s	63.2s/234.1s

Table 1: *Running times of our algorithm*

We have implemented our algorithm on a 3.4GHz PC with 1.5GB of RAM. Table 1 summarizes our experiments timings (in seconds) normalized per frame for consensus skeleton computation (*c*-skeleton) and point cloud registration (*c*-point cloud). Pre/post processing refer to skeleton extraction, ICP point cloud registration and outlier removal respectively.

Figure 10 shows a sequence of two dancing mannequins scanned in different poses from only a single view. The extracted skeletons demonstrate the challenging task of the consensusization process of complex topological scenes with noise. These initial skeletons are either broken or have erroneous branches. This is largely caused by the sparsity (single view) and the noisiness of the input data. Note the missing left hand and the disconnected limbs in the top row. The initial deficiencies of the skeletons are effectively fixed through consensusization, as shown in the deformed consensus skeletons. This conforms with the intuition that multiple views combined together can significantly enrich point representation, hence better skeletonization results. Additional results are shown in Figure 8 for the running Ben example.

Figure 9 demonstrates the ability of our algorithm in handling highly sparse temporal sampling. Note that the three consecutive frames were captured with large deformations of the mannequin in-between frames. There are almost no overlap between the arms of the mannequin in adjacent frames, while our consensusization algorithm still is able to compute the correct skeleton correspondence and global alignment.

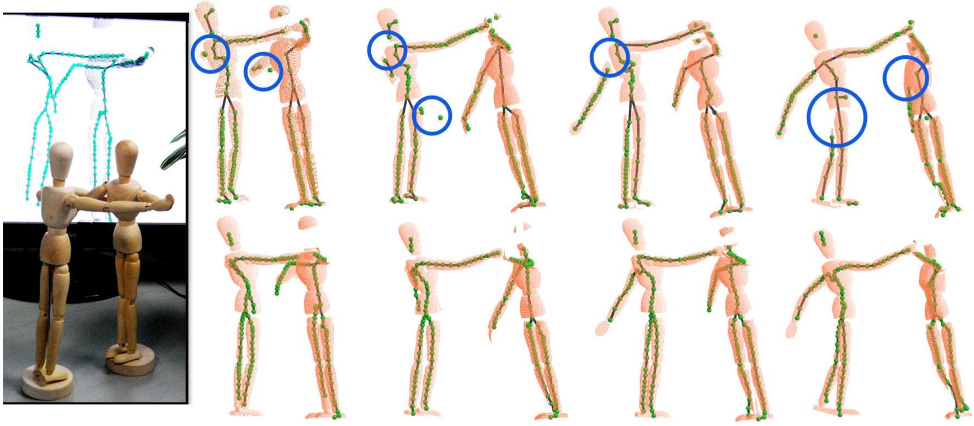

Figure 10: *Consensus skeleton extraction from a topologically complex scene of two dancing mannequins (left). Top: single-view scans of four dancing poses with initially extracted skeletons. Bottom: consensus skeletons deformed to individual poses. Note that the deformed consensus skeleton is generally smoother and better connected than independently extracted skeletons.*

Skeleton-driven point cloud registration Finally, Figures 1, 11, 13 show results for skeleton-driven non-rigid point cloud registration. Based on the skeleton consensusization process, we can obtain a *consensus point cloud* by registering the point clouds from all frames using non-rigid deformation into a common pose. We deform the consensus skeleton back onto each frame using the rotated Laplacians deformation and consensus-to-original skeleton correspondence. Using standard linear skinning [BP07] we can deform each point cloud by its skeleton into a common pose, as shown in Figure 11(left). The point cloud consensusization process is then carried out by performing pairwise registrations, via the classic soft or non-rigid ICP [PG08], and gradually building up the hierarchy by combing all frames. Our registration process is guided by the *c*-skeleton in two ways. First, a subset of points which correspond to a bone (edge) of the *c*-skeleton, which we refer to as a patch, is regarded as a rigid part during registration and the set of patches corresponding to the same bone serve to constrain the ICP computation. Secondly, the skeletal deformations computed from the skeleton registration step (Section 5) are used to deform the point cloud and initialize ICP for patch registration.

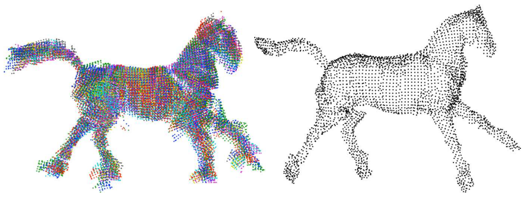

Figure 11: *Skeleton-driven point cloud registration of the horse model shows good initial alignment (left), but misalignments are still present. Right is the result using ICP a-priori initialized by our registration.*

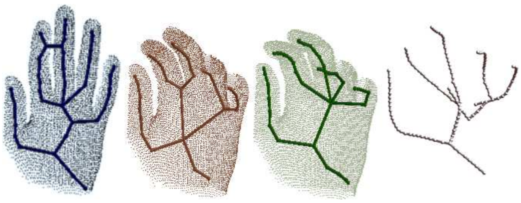

Figure 12: *Initially extracted skeletons with large topological dissimilarity and the resulting c-skeleton (right).*

Limitations The performance of our method has to do with the quality of the initially extracted skeletons. Both *c*-skeleton computation and *c*-skeleton deforming back to original frames assume some similarity between skeletons. In particular, pairs of skeletons extracted from consecutive frames should contain a reasonable degree of similarity for the HMM to perform well enough. If extracted skeletons are too noisy, non-informative or consist very different geometry and topology, our *c*-skeleton may become incorrect. (see Figure 12). Furthermore, when the extracted skeleton is not correctly embedded back in the point cloud, parts of the point cloud might be dangling. When dangling parts are too far from their consensus position, it may cause ICP to converge to a erroneous local minimum.

8. Concluding remarks

We have presented a technique for non-rigid registration. Unlike common techniques which are based on local point features or local shape descriptor, here the registration is based on a global feature, namely the shape skeleton, which is not sensitive to surface fine details or noise. We presented a technique to combine a set of skeletons, unifying them into

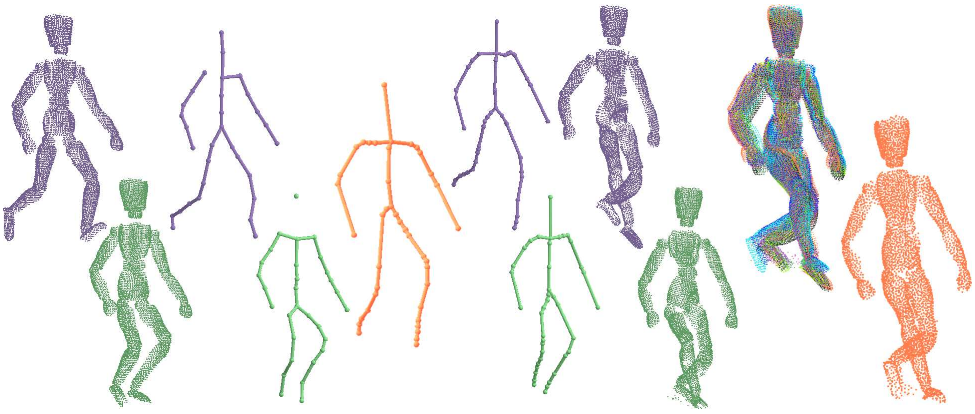

Figure 13: *Another example of our c-skeleton pipeline. Left to right: We compute the c-skeleton from a set of noisy skeletons (left-middle); we deform the c-skeleton onto original frame poses and compute the skeleton-driven point cloud registration. On the right, we show the registered superimposed point cloud (top), and the final ICP perfect registration (bottom).*

a consensus skeleton, while ignoring outliers. The consensus skeleton serves as robust constraints for mapping and aligning the points from each frame into a common pose, facilitating the fine registration by a local non-rigid ICP.

The multi-skeleton registration is based on a HMM pairwise correspondence. We believe that this technique can be further developed into a robust pairwise correspondence useful for many other applications. In the future we would also like to explore the possibility to combine parts of skeletons extracted from partial views or captured by scans taken from at different scale. Pairwise matching or correspondence that is scale invariant is challenging, but we believe that it can significantly improve fidelity to fine details of models.

Acknowledgements We thank Zhan Song, Ke Xie and Fei-Long Yan for capturing the mannequin data, and Gao-Jin Wen, Liang-Liang Nan, Zhang-Lin Cheng and Yan-Yang Li for the fruitful discussions. This work was supported in part by the following grants: National Natural Science Foundation of China (60903116, 60902104), National High-tech R&D Program of China (2009AA01Z302), CAS Visiting Professorship for Senior International Scientists, CAS Fellowship for Young International Scientists, Shenzhen Basic Research Foundation (JC200903170443A), Shenzhen Science & Technology Foundation (SY200806300211A) and NSERC (No. 611370).

References

[ASK*05] Anguelov D., Srinivasan P., Koller D., Thrun S., Rodgers J., Davis J.: SCAPE: shape completion and animation of people. *ACM Trans. on Graphics 24*, 3 (2005), 408–416.

[ATD*08] Ahmed N., Theobalt C., Dobrev P., Seidel H. P., Thrun S.: Robust fusion of dynamic shape and normal capture for high-quality reconstruction of time-varying geometry. In *IEEE Conf. on Comp. Vis. and Pat. Rec.* (2008), pp. 1–8.

[BC08] Ballan L., Cortelazzo G. M.: Marker-less motion capture of skinned models in a four camera set-up using optical flow and silhouettes. In *Proc. of 3D Data Processing, Visualization, and Transmission* (June 2008).

[BN92] Beier T., Neely S.: Feature-based image metamorphosis. *SIGGRAPH Comput. Graph. 26*, 2 (1992), 35–42.

[BP07] Baran I., Popović J.: Automatic rigging and animation of 3d characters. *ACM Trans. on Graphics 26*, 3 (2007), 72.

[BPS*08] Bradley D., Popa T., Sheffer A., Heidrich W., Boubekeur T.: Markerless garment capture. *ACM Trans. on Graphics 27*, 3 (2008), 99–108.

[CZ08] Chang W., Zwicker M.: Automatic registration for articulated shapes. *Eurographics Symp. on Geom. Processing 27*, 5 (2008), 1459–1468.

[CZ09] Chang W., Zwicker M.: Range scan registration using reduced deformable models. *Computer Graphics Forum (Special Issue of Eurographics) 28*, 2 (2009), 447–456.

[dAST*08] de Aguiar E., Stoll C., Theobalt C., Ahmed N., Seidel H.-P., Thrun S.: Performance capture from sparse multi-view video. *ACM Trans. on Graphics 27*, 3 (2008).

[dATTS08] de Aguiar E., Theobalt C., Thrun S., Seidel H.-P.: Automatic conversion of mesh animations into skeleton-based animations. *Computer Graphics Forum (Special Issue of Eurographics) 27*, 2 (2008), 389–397.

[HAWG08] Huang Q., Adams B., Wicke M., Guibas L. J.: Non-rigid registration under isometric deformations. *Eurographics Symp. on Geom. Processing 27*, 5 (2008), 1449–1457.

[JT05] James D. L., Twigg C. D.: Skinning mesh animations. *ACM Trans. on Graphics 24*, 3 (2005), 399–407.

[LAGP09] Li H., Adams B., Guibas L. J., Pauly M.: Robust single-view geometry and motion reconstruction. *ACM Trans. on Graphics 28*, 5 (2009).

[LH05] Leordeanu M., Hebert M.: A spectral technique for correspondence problems using pairwise constraints. In *Proc. Int. Conf. on Comp. Vis.* (2005), pp. 1482–1489.

[LSP08] Li H., Sumner R. W., , Pauly M.: Global correspondence optimization for non-rigid registration of depth scans. *Eurographics Symp. on Geom. Processing 27*, 5 (2008), 1421–1430.

[LZW*09] Liao M., Zhang Q., Wang H., Yang R., Gong M.: Modeling deformable objects from a single depth camera. In *Proc. Int. Conf. on Comp. Vis.* (2009), p. to appear.

[MFO*07] Mitra N. J., Flöry S., Ovsjanikov M., Gelfand N., Guibas L., Pottmann H.: Dynamic geometry registration. In *Eurographics Symp. on Geom. Processing* (2007), pp. 173–182.

[PG08] Pekelny Y., Gotsman C.: Articulated object reconstruction and markerless motion capture from depth video. *Computer Graphics Forum 27*, 2 (2008), 399–408.

[Rab89] Rabiner L. R.: A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE 77*, 2 (1989), 257–286.

[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 Trans. on Graphics 27*, 5 (2008), 1–10.

[SLCO*04] Sorkine O., Lipman Y., Cohen-Or D., Alexa M., Rössl C., Seidel H.-P.: Laplacian surface editing. In *Eurographics Symp. on Geom. Processing* (2004), pp. 179–188.

[SWG08] Süssmuth J., Winter M., Greiner G.: Reconstructing animated meshes from time-varying point clouds. *Eurographics Symp. on Geom. Processing 27*, 5 (2008), 1469–1476.

[SY07] Schaefer S., Yuksel C.: Example-based skeleton extraction. In *Eurographics Symp. on Geom. Processing* (2007), pp. 153–162.

[TdAM*04] Theobalt C., de Aguiar E., Magnor M. A., Theisel H., Seidel H.-P.: Marker-free kinematic skeleton estimation from sequences of volume data. In *ACM Symposium on Virtual reality software and technology* (2004), pp. 57–64.

[TZCO09] Tagliasacchi A., Zhang H., Cohen-Or D.: Curve skeleton extraction from incomplete point cloud. *ACM Transactions on Graphics, (Proceedings SIGGRAPH 2009) 28*, 3 (2009), Article 71, 9 pages.

[VBMP08] Vlasic D., Baran I., Matusik W., Popović J.: Articulated mesh animation from multi-view silhouettes. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers* (2008), pp. 1–9.

[WAO*09] Wand M., Adams B., Ovsjanikov M., Berner A., Bokeloh M., Jenke P., Guibas L., Seidel H.-P., Schilling A.: Efficient reconstruction of non-rigid shape and motion from real-time 3D scanner data. *ACM Trans. on Graphics 28*, 2 (2009).

[WJH*07] Wand M., Jenke P., Huang Q., Bokeloh M., Guibas L., Schilling A.: Reconstruction of deforming geometry from time-varying point clouds. In *Eurographics Symp. on Geom. Processing* (2007), pp. 49–58.

Appendix. HMM Formulation

The HMM requires emission probabilities, i.e., the likelihood that a given hidden state will produce a given output, and transition probabilities, i.e., the likelihood of a transition from one hidden state to another.

To define these terms, let us first define the similarity metric between two nodes, u_1^i and u_2^j, of skeletons S_1 and S_2:

$$S(u_1^i, u_2^j) = |e_1^i - e_2^j|,$$

where e_1^i (respectively e_2^j) is the degree of u_1^i, the i-th node of S_1 (respective, u_2^j, the j-th node of S_2).

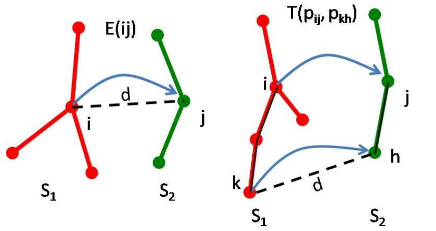

Figure 14: *Emission and transition costs. Left: emission cost of a pair of nodes is a function of their (degree) similarity and their distance. Right: transition cost from a correspondence pair (i, j) to a new one (k, h).*

The emission $E(u_1^i, u_2^j)$ is the weighted sum of their similarity cost and distance:

$$E(i, j) = S(u_1^i, u_2^j) + w_e||u_1^i - u_2^j||,$$

where w_e balances the scales of the two terms.

The transition cost $T(p_{ij}, p_{hk})$ from a previous correspondence pair (i, j) to a new one (k, h) is determined by three terms:

$$T(p_{ij}, p_{kh}) = S^p(p_{ij}, p_{kh}) + S(k, h) + \delta,$$

where the first term measures the similarity of two correspondence pairs. If both k, h, or neither, are descendants of i, j respectively, then the term is defined as:

$$S^p(p_{ij}, p_{kh}) = w_g|g(p_{ij})^2 - g(p_{kh})^2|,$$

otherwise, $S^p(p_{ij}, p_{kh}) = \propto$. Here $g(p_{ij})$ is the geodesic distance between the two nodes along the skeleton graph and we set the weight $w_g = 100$ to balance the scale with the other terms. The second term is the similarity cost of the two nodes of the second pair. The last term δ penalizes mappings that are not one-to-one: $\delta = 0$ if p_1 and p_2 have no common node, otherwise $\delta = 1$.

If the cost is larger than a threshold, we disable the state or the transition between the two states. Hence, we can skip a source node if it cannot find a corresponding node, allowing for partial correspondence.

To transform costs to probabilities, we use an exponentially descending function

$$\zeta(c_i) = e^{-(1.5*\sqrt{c_i/max(C)})^4},$$

where C is the set of cost values, $c_i \in C$. Using function ζ, we first convert emission costs to emission probabilities. For a given correspondence p_{ij}, it can possibly transit to a group of states whose source node is the $(i+1)$-th node in skeleton S_1. We also convert transition costs to transition probabilities using ζ. After that, we employ the Viterbi algorithm to find a depth-sorted sequence of skeleton S_1 (with a randomly selected root node) that has the largest possibility.