Eurographics Symposium on Geometry Processing 2016
Maks Ovsjanikov and Daniele Panozzo
(Guest Editors)

Volume 35 (2016), Number 5

Mobility Fitting using 4D RANSAC

Hao Li[†1] Guowei Wan[†1] Honghua Li[1] Andrei Sharf[2] Kai Xu[3] Baoquan Chen[1]

[1]Shandong University [2]Ben-Gurion University [3]National University of Defense Technology

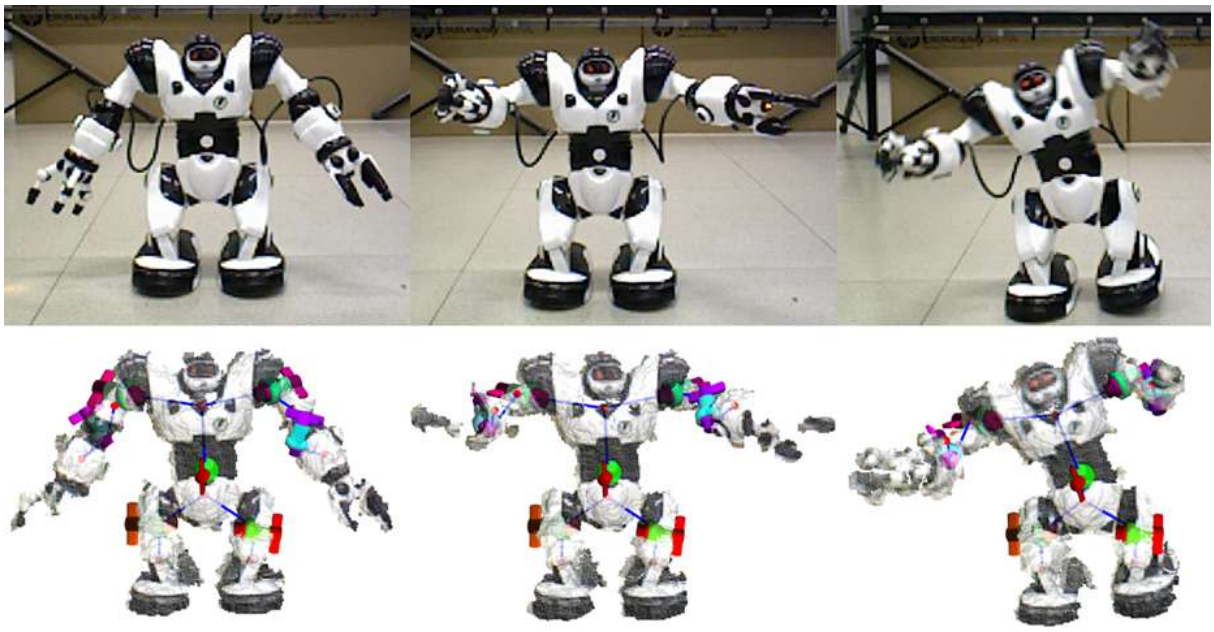


Figure 1: *Mobility fitting of a robot's articulated motion. Top row are excerpts from the robot's motion sequence, while the bottom row depicts the captured dynamic scans with the fitted mobility joints (colored hinges) and their mobility graph.*

Abstract

Capturing the dynamics of articulated models is becoming increasingly important. Dynamics, better than geometry, encode the functional information of articulated objects such as humans, robots and mechanics. Acquired dynamic data is noisy, sparse, and temporarily incoherent. The latter property is especially prominent for analysis of dynamics. Thus, processing scanned dynamic data is typically an ill-posed problem. We present an algorithm that robustly computes the joints representing the dynamics of a scanned articulated object. Our key idea is to by-pass the reconstruction of the underlying surface geometry and directly solve for motion joints. To cope with the often-times extremely incoherent scans, we propose a space-time fitting-and-voting approach in the spirit of RANSAC. We assume a restricted set of articulated motions defined by a set of joints which we fit to the 4D dynamic data and measure their fitting quality. Thus, we repeatedly select random subsets and fit with joints, searching for an optimal candidate set of mobility parameters. Without having to reconstruct surfaces as intermediate means, our approach gains the advantage of being robust and efficient. Results demonstrate the ability to reconstruct dynamics of various articulated objects consisting of a wide range of complex and compound motions.

1. Introduction

The evolution of 3D scanners has made it possible to acquire the geometry of various objects in motion, both man-made and natural.

† The authors assert equal contribution and joint first authorship.

Nevertheless, in comparison to the prevalence of 3D reconstruction techniques, there is still little work that addresses the capturing of dynamics, such as those arising in mechanical interactions, robot dynamics, and even human body motions. Dynamics, more than geometry, succinctly describe the functionality of an articulated object. In his seminal work, Johansson [Joh73] showed that visual perception is tightly coupled with motion cues. Dynamics comprise a wealth of information encompassing motion parameters, joint positions and axes in 3D space, which may be utilized in various motion analysis, recognition and reconstruction tasks.

Reconstruction of 3D motions is a challenging task which is amplified by the often low quality of the scanned dynamic data. 3D scanners are of limited resolution and frame-rate. When applied to rapidly moving geometries the resulting point sets are typically noisy, sparse and largely incoherent. Furthermore, moving parts may generate significant outliers due to ghost effects and self-occlusions yielding holes in the data. To reconstruct the motion parameters (denoted *mobility*), a straightforward approach may attempt to first reconstruct the geometry of dynamic objects and then analyze the mobility based on the animated geometry. Previous works take 4D scan sequences and register in-between frames to reconstruct a coherent geometry [MFO*07, WAO*09, CZ08]. Data-driven methods [PG08, CZ11] pre-define geometry templates and deform them to locally register and track individual frames.

Dynamic objects in the wild typically have a large structural and geometric variety. Hence, templates that accurately fit and reconstruct the dynamic geometry cannot be always assumed. In this work, we take a different approach aiming to recover only the mobility of objects from a dynamic scan sequence. We completely remove the necessity of reconstructing the full surface geometry and topology along its dynamic deformation. Instead, we directly compute the articulated motion parameters (Figure 1), defined by local piecewise-rigid transformations which are prescribed by a limited set of joint types. Due to the fact that joints are typically invariant to form and shape, they may efficiently and robustly computed by generic parametric models that are fitted to the dynamic scans.

Our algorithm computes mobilities by fitting joints to point trajectories in the scanned data through a random consensus sampling, denoted 4D RANSAC. Similar to RANSAC, we fit various joints to random subsets and search for a consensus set which supports them. Then, the method searches for a global consensus mobility model which optimally models the whole articulated motion. The advantage of RANSAC is in its robustness. Although motion data may be largely missing due to occlusions or consist outliers, it can accurately estimate the motion parameters of the model that optimally fits this data.

Our method robustly estimates the parameters of the articulation motion model from a set of scanned 4D dynamic points. Our contribution is a 4D RANSAC scheme, which computes the mobility joints in dynamic scans without intermediately reconstructing the object's geometry. The scheme is global in that it avoids local operations such as pairwise frame registration, per-frame shape reconstruction and accurate trajectory tracking.

2. Related Work

Dynamic geometry processing has been an active field of research in recent years in both computer vision and graphics. It is beyond the scope of this paper to fully review it and refer to Chang et al. [CLM*12] for a comprehensive survey of this topic. Instead, we focus our discussion on articulated motion analysis and reconstruction techniques.

Rigid motion segmentation. Motion trajectories are a fundamental representation of motion sequences besides RGBD depth images and 3D scans. Their segmentation serve as means to reduce their complexity and analyze their features. Many approaches to motion segmentation rely on sparse interest points correspondence or on dense optical flow based techniques.

To segment moving objects in real time using a calibrated stereo camera, a three-point RANSAC for rigid body detection is used in [AKI05]. Thus, rigid motion candidates are detected by fitting rigid transformations to a sparse set of feature points that are tracked across the sequence.

A scene flow algorithm introduced in [HRF13] shows how to combine depth and color information to estimate the dense 3D motion of each point in an RGB-D frame. To refine the motion, they perform a rigid motion clustering following [HRF12]. Thus, they sample possible point correspondences and fit them with rigid motions in a RANSAC manner, followed by discarding outliers using a Markov random field.

An expectation-maximization (EM) framework is introduced in [SB15] for motion segmentation of RGB-D sequences. Motion segments and their 3D rigid-body transformations are formulated as an EM problem, thus it determines the number of rigid parts, their 3D rigid- body motion, and the image regions that map these parts.

Dynamic scan reconstruction. Researchers have considered dynamic scans as a 4D surface reconstruction problem in space-time. Mitra et al. [MFO*07] use kinematic properties of the 4D space-time surface to track points and register frames of a rigid object together. SÍuSSmuth et al. [SWG08] and Sharf et al. [SAL*08] explicitly reconstruct the 4D space-time surface using an implicit surface. In [WAO*09], multiple scan frames are aligned together by solving surface motion in terms of a displacement field. A common shape that deforms and matches data is computed.

A template model is used in [LAGP09] as a coarse motion prior to reconstruct the non-rigid motion. The template deforms using a deformation graph and is registered to the scanned data reconstructing the coarse geometry. Popa et al. [PSDB*10] reconstruct mesh animations using optical flow and a gradual change prior. Animation reconstruction is computed in [TBW*12] using reliable landmark correspondences which are extended to define a dense matching across time. Nevertheless, reconstructing the full 4D deforming geometry from scans is a challenging problem. The solution lies in a high dimensional domain and requires a dense sampling in both time and space. In practice, the reconstruction problem is under constrained as scans are typically sparse with large missing parts due to self occlusions and noise.

Articulated motion processing. Registration of dynamic point sets is at the core of many 4D reconstruction techniques for both rigid [GMGP05] and non-rigid [HAWG08] motions. The difference between general non-rigid and articulated motions is that rigid parts of the surface yield a constrained motion which help in registration and trajectory processing. In fact, Huang et al. [HAWG08] use clustering of rigid transformations to improve registration convergence in non-rigid surface deformations.

Mean shift clustering is used in [JT05] to cluster rotations extracted from an articulated mesh deformation. This is used to automatically identify skin bones and their transformations for mesh animation sequences. Rigid transformation clustering has been also used in the context of symmetry detection [MGP06]. A similar approach to RANSAC is applied here, as the surface is sampled and matching feature pairs are accumulated to identify potential symmetries.

The problems of piece-wise rigid point registration and part segmentation are described as tightly coupled in [CZ08]. They propose to transform the registration problem into a discrete labeling problem, where the goal is to find an optimal set of rigid transformations for aligning the shapes' parts. We observe that many previous works focus on solving the coupled problems of surface processing and motion reconstruction. Instead, we take a different approach and directly analyze the joints and their mobility. Our method utilizes a predefined set of mobility priors which are directly fitted to the 4D data, leading to robust motion reconstruction and segmentation.

Pekelny and Gotsman [PG08] present an articulated motion reconstruction method which assume a given segmentation of the shape into rigid parts and an underlying skeletal structure. Chang and Zwicker [CZ11] present an algorithm which simultaneously optimizes scan alignment and model reconstruction using a reduced deformable model. Their method formulates the problem as a skinning problem, searching for a global set of transformations and weights. Both works are similar to us in that they attempt to reconstruct the articulated motion utilizing global priors. Nevertheless, we avoid the problem of coherent geometry reconstruction and instead, directly reconstruct only the dynamics in terms of joints and their motion parameters. To this end, we take a RANSAC approach, taking advantage of its robustness and fast processing.

Similar to us, Mufti et al. [MMH12] introduce a spatio-temporal RANSAC algorithm for dynamic scan processing. Nevertheless, their setup consists of a moving ToF camera scanning a 3D outdoor scene and their goal is the detection of the planar (static) ground. Thus, their method focus on computing reliable spatio-temporal planar hypotheses.

3. Overview

The motion of an articulated model is typically governed by a set of joints which define the relative piecewise rigid motions between parts pairs. Nevertheless, parts may deform at different intervals, in more than one way, independently or together, resulting in complex motions.

Our input consists of a scanned articulated motion sequence.

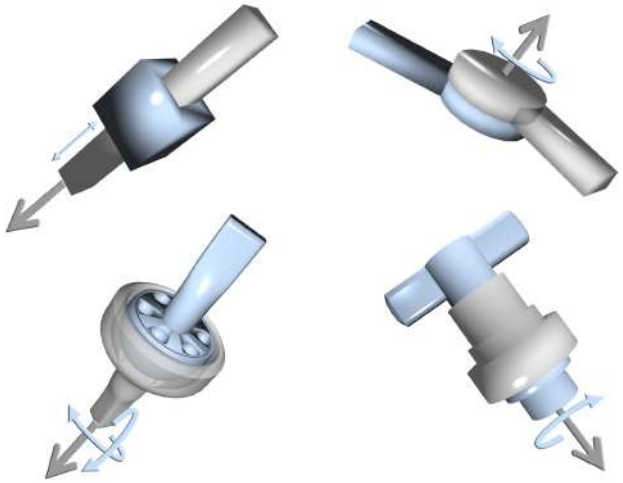


Figure 2: *Our joint types consist of (top left, CW): slider, planar hinge, orthogonal hinge and ball joint.*

Each frame in this sequence contains a 3D point set that samples the scene at one time point. In a preprocessing step, we compute inter-frame correspondence between points, utilizing both geometry and color cues, and concatenate these correspondences to form trajectories. To reduce their complexity, we cluster similar trajectories together, based on their transformation similarity.

We devise a random sampling consensus method in 4D which fits joints to point trajectories by considering three basic types of joints: hinge, slider, and ball joint (Figure 2). Thus, we randomly select a *space-time subset* by selecting few random trajectories and a subset of their interval. For each random selection, we compute its best fitting mobility model among the predefined set of joints. Typically, articulated motions are defined by joints connecting part pairs. Thus, our random sampling consensus aims at fitting mobility models to pairs of relative trajectory motions. We represent the trajectory motions using a relative scheme which accounts only for the local transformation as defined by a single joint.

We evaluate each mobility model by computing its consensus w.r.t. the complete 4D data set and use a voting scheme to measure the fitting quality of each mobility model. Our method is iterative, repeating the random selection and consensus voting steps until all mobilities are reconstructed in the data. Specifically, we iterate until no mobility models can be found with sufficient support.

The reconstructed mobility joints are organized as a *mobility graph*, which represents the dynamic structure of the scanned object. The graph is an abstract representation encoding joints as nodes and their adjacency relations as edges connecting the nodes (see Figure 1(bottom row)).

4. Technical Details

Scanned motions are represented by a sequence of frames sampling the articulated motion in space and time. Using consumer level RGB-D depth cameras (e.g. Kinect®, Primesense®, and others),

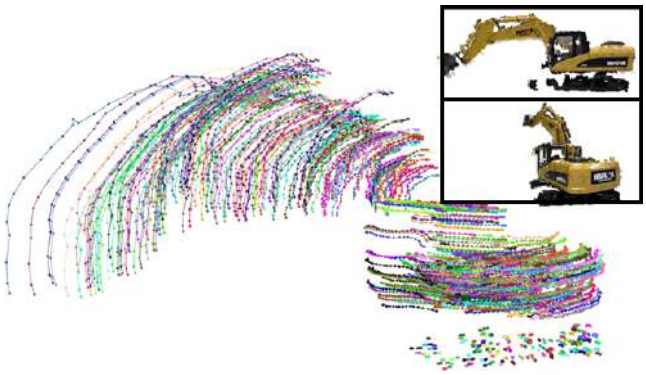


Figure 3: *Trajectories of an excavator's articulated motion are illustrated by colored lines. Initial and final states are shown on top right.*

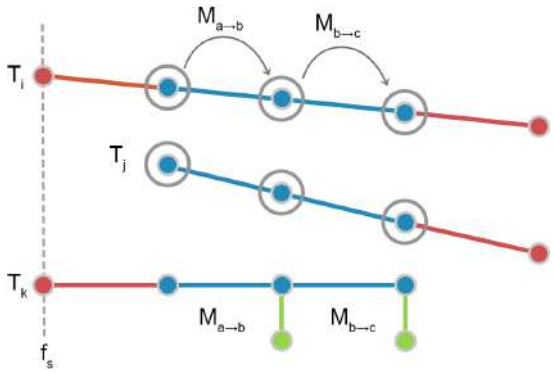


Figure 4: *2D illustration of trajectory representation. Three trajectories (T_i, T_j, T_k) sharing a common life span (blue). We randomly select a trajectory triplet (here in 2D the pair T_i, T_j) and compute its RTM: $M = (\mathbf{M}_{a\to b}, \mathbf{M}_{b\to c})$. The residual $r(T_k, M)$ measures the average positional offset when applying M on T_k (in green).*

captured frames encode both depth and color channels as raw point sets.

Trajectory generation. We compute dense motion trajectories connecting points in adjacent frames using scene flow [JSGJC15]. Our algorithm does not assume accurate trajectories and is designed for robustness to trajectory noise. In fact, scanned data is typically very noisy and trajectory noise has a similar magnitude.

Given the RGB-D data of two consecutive frames S_i and S_{i+1}, the algorithm produces a 3D motion vector $\mathbf{v}$ for each point $\mathbf{p} \in S_i$ to S_{i+1}, by computing closest point correspondences. Correspondences are typically noisy due to poor data quality and occlusions. Thus, from the raw dense correspondences we extract a reliable sparse set by rejecting correspondences with low matching scores in terms of their SIFT image features [Low04]. We represent SIFT by a 128-dimension vector, and define dissimilarity as the Euclidean distance between SIFT descriptors. For all experiments, a threshold of $\varepsilon_{\text{SIFT}} = 0.5$ on SIFT distance is utilized to prune dense correspondences.

We build motion trajectories by concatenating sparse correspondences sharing common points. Trajectories may not exist during the entire sequence due to occlusions or disappearance from camera view, and thus may have different life spans. We denote a trajectory life span $T = (\mathbf{p}_s, \ldots, \mathbf{p}_{s+k})$, where $\mathbf{p}_s$ is the trajectory point in frame s and $\mathbf{p}_{s+k}$ in frame $s+k$. Figure 3 shows the motion trajectories of an excavator. Although they are noisy due to artifacts in the sceneflow algorithm, they capture the overall motion well, due to our SIFT filtering step.

Trajectories simplification. Trajectory data is typically large in the order of the sampled points. Furthermore, it may be of various lengths due to disconnections and intersections. Our goal is to simplify the trajectory data and reduce its complexity by a compact set of representative trajectories. Although our mobility fitting can perform on the raw trajectories its performance significantly benefits from reducing their size, similar to Yan et al. [YSL*14].

Thus, we cluster together trajectories based on a similarity metric which accounts for the articulated motion. We represent a trajectory T using a rigid trajectory-model (RTM) which encodes a series of rigid transformations M representing its articulated motion:

$$M = (\mathbf{M}_{s\to s+1}, \ldots, \mathbf{M}_{s+k-1\to s+k}), \tag{1}$$

where $\mathbf{M}_{a\to b}$ is a local rigid transformation from frame f_a to f_b.

To compute $\mathbf{M}_{a\to b}$ in two consecutive frames a, b, we randomly select trajectory points triplets $\{\mathbf{p}_i^a\}$ in frame a and their correspondences $\{\mathbf{p}_i^b\}$ in frame b. The rigid transformation $\mathbf{M}_{a\to b} = (\mathbf{R}, \mathbf{t})_{a\to b}$ is estimated by:

$$\mathbf{M}_{a\to b} = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{i=1}^{N} \|\mathbf{p}_i^b - (\mathbf{R}\mathbf{p}_i^a + \mathbf{t})\|^2. \tag{2}$$

We solve this optimization problem using the least-squares fitting algorithm [AHB87]. Note that since rigidity preserves isometry, we reject triples which do not preserve pairwise distances along the trajectories.

Given a rigid transformation model M and a trajectory $T = (\mathbf{p}_s, \ldots, \mathbf{p}_{s+k})$, we define the residual of T w.r.t. M as:

$$r(T, M) = \frac{1}{k} \sum_{i=0}^{k-1} ||\mathbf{M}_{s+i\to s+i+1}\mathbf{p}_{s+i} - \mathbf{p}_{s+i+1}||, \tag{3}$$

which measures the average 3D positional offset of a trajectory when applying transformation M to T along (f_s, f_{s+k}) (see Figure 4).

Naturally, we randomly select trajectory triplets and compute their RTM's using Equation 1 (see Figure 4). In total n RTM candidates are generated. We define a trajectory-model association signature (TMS) for a trajectory T, as a n-dimensional vector A, which measures how well each RTM represents T:

$$A(i) = \begin{cases} 1, & \text{if } r_i < \varepsilon_r, \\ 0, & \text{if } r_i \geq \varepsilon_r, \end{cases} \tag{4}$$

where $r_i = r(T, M_i)$ is the residual of T w.r.t. the i-th RTM M_i, using $\varepsilon_r = 0.004$ in all experiments.

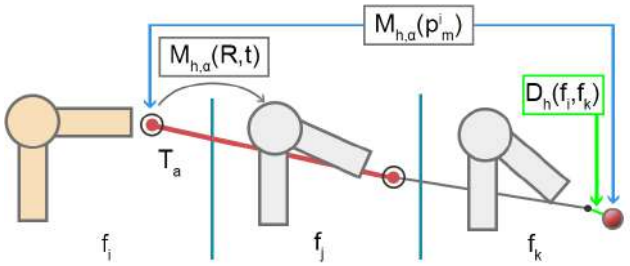


Figure 5: *2D illustration of hinge fitting. We randomly select two frames (f_i, f_j) and compute the hinge fitting $h = (\mathbf{a}, \mathbf{c})$, in yellow. To compute the consensus set, we measure the distortion $D_h(f_i, f_k)$ for a new frame f_k w.r.t f_i by transforming f_i using $\mathbf{M}_{h,\alpha}$.*

Finally, using the TMS signature A, the clustering algorithm performs by incrementally merging closest clusters in terms of their *Jaccard distance*, following Roldo and Fusiello [TF08]:

$$d_J(A_1, A_2) = 1 - \frac{A_1 \cap A_2}{A_1 \cup A_2}. \tag{5}$$

For each cluster, we update its TMS by the intersection of TMS's of merged clusters. The merging procedure stops when the distance between two clusters is $d_J(A_i, Aj) \geq 0.7$ which means two clusters A_i and A_j share less than 30% common RMTs . We then remove small clusters containing less than 5% of the total number of trajectories. A consolidated RTM $\hat{M}$ is then computed for each cluster by computing the transformations between frames within a cluster. In the following, we refer to a consolidated RTM as a representative trajectory (or trajectory for simplicity).

Mobility fitting. We consider joint types that include general hinges, sliders and ball joints. We randomly sample trajectory data and fit a mobility model to it following [Reu76]. A trajectory of length 2 uniquely defines a hinge and slider joint and of length 3 for a ball joint. Fitted mobilities are tested against the entire dataset, yielding a set of conforming trajectories, denoted *consensus set*. The mobility with the largest consensus set is selected and corresponding trajectories are removed. The process repeats until no mobility models can be found with sufficient support. Assuming that mobility models always connect between two rigid parts, we analyze the local motion between pairs of trajectories denoted as the relative mobility. Given two representative trajectories (or simply trajectories) T_A and T_B, we arbitrarily select T_A as the reference and compute the relative trajectory representation of T_B w.r.t T_A denoted $T_{B|A}$. Let $(\mathbf{p}_s^B, \ldots, \mathbf{p}_{s+k}^B)$ be the points of trajectory T_B, then the relative trajectory representation $T_{B|A} = (\mathbf{p}_s^{B|A}, \ldots, \mathbf{p}_{s+k}^{B|A})$ can be computed as

$$\mathbf{p}_s^{B|A} = \mathbf{p}_s^B, \tag{6}$$

$$\mathbf{p}_{s+i}^{B|A} = (\mathbf{M}_{s\to s+i}^B - \mathbf{M}_{s\to s+i}^A) * \mathbf{p}_s^B, 1 \leq i \leq k. \tag{7}$$

Hinge joint. A hinge connects two rigid bodies, allowing a relative rotational motion between them about a fixed axis defined by its direction $\mathbf{a}$ and position $\mathbf{c}$. The transformation $\mathbf{M}_{h,\alpha} = (\mathbf{R}, \mathbf{t})_{h,\alpha}$,

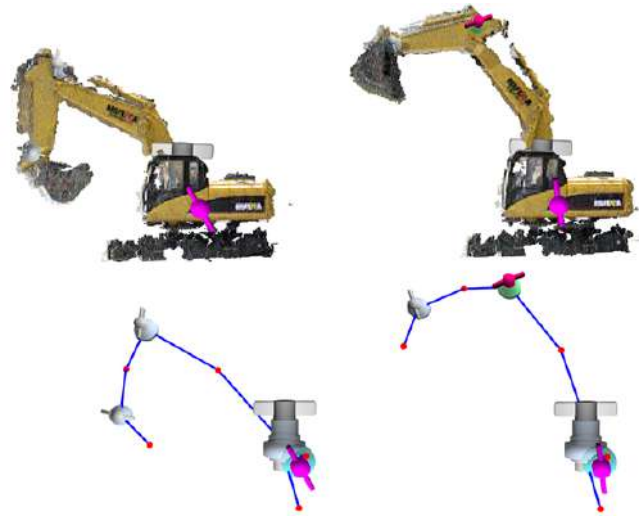


Figure 6: *Excavator motion defined by 4 hinges. Top row are excerpts from the scanned sequence with correctly reconstructed hinges. Bottom row is the full motion graph following the reconstructed joints.*

representing the rotation about hinge $h = (\mathbf{a}, \mathbf{c})$ by angle α is defined by:

$$\mathbf{R}_{h,\alpha} = \mathbf{R}_{\mathbf{a},\alpha}, \tag{8}$$

$$\mathbf{t}_{h,\alpha} = \mathbf{c} - \mathbf{R}_{\mathbf{a},\alpha}\mathbf{c}, \tag{9}$$

where

$$\mathbf{R}_{\mathbf{a},\alpha} = \cos\alpha \mathbf{I}_3 + (1 - \cos\alpha)\mathbf{a}\mathbf{a}^T + \sin\alpha[\mathbf{a}]_\times \tag{10}$$

is the rotation about axis $\mathbf{a}$ located at origin, and $[\mathbf{a}]_\times$ denotes the cross-product matrix.

Given the hinge transformation $\mathbf{M}_{h,\alpha} = (\mathbf{R}, \mathbf{t})$ between two frames $\{f_i, f_j\}$, according to Euler's rotation theorem, the hinge direction $\mathbf{a}$ is the eigenvector of $\mathbf{R}$ with eigenvalue of 1. The hinge location $\mathbf{c}$ can be computed by solving the linear equation $(\mathbf{I} - \mathbf{R})\mathbf{c} = \mathbf{t}$. Since, this is under-constrained, the solution could be any point on the hinge. Thus, we project the centroid of all relative trajectories onto the hinge axis to obtain the absolute hinge location.

For a hinge $h = (\mathbf{a}, \mathbf{c})$, we compute its support by measuring its fitting error to the full lifespan trajectory. Thus, taking points from other frames f_k relative to f_i, we measure:

$$D_h(f_i, f_k) = \min_\theta \|\mathbf{M}_{h,\theta}\mathbf{p}_m^i - \mathbf{p}_m^k\|, \tag{11}$$

where $\{\mathbf{p}_m^i\}$ and $\{\mathbf{p}_m^k\}$, $m = 1, \ldots, N$, are corresponding trajectory points in f_i and f_k respectively.

The point $\mathbf{p}_m^k \in f_k$ supports $h = (\mathbf{a}, \mathbf{c})$ if its fitting error is below a threshold $\varepsilon_h = 0.05$. As a fast reject strategy, we discard a hinge if its supporting points are less than 80% of the trajectory's lifespan (see Figure 5).

Slider joint. A slider connects two rigid bodies and allows a relative translational motion between them along a direction $\mathbf{v}$. Suppose the rigid transformation between two frames $\{f_i, f_j\}$ is $\mathbf{M}_{i\to j} = (\mathbf{R}, \mathbf{t})$. Ideally the sliding direction is $\mathbf{v} = \mathbf{t}/\|\mathbf{t}\|$, and the rotation component $\mathbf{R}$ are the identity matrix.

For a slider joint $\mathbf{v}$, we compute its support by measuring its fitting error to the full lifespan trajectory. Thus, taking points from other frames f_k relative to f_i, we measure:

$$D_{\mathbf{v}}(f_i, f_k) = \min_{v} \|\mathbf{p}_m^i + t\mathbf{v} - \mathbf{p}_m^k\|. \tag{12}$$

The point $\mathbf{p}_m^k \in f_k$ supports $\mathbf{v}$ if the fitting error is below a threshold $\varepsilon_{\mathbf{v}} = 0.05$. Similarly, slider joints are discarded if their support is below 80% of the trajectory's lifespan.

Ball joint. A ball joint connects two rigid bodies and allows a rotation between them about a fixed pivot $\mathbf{c}$. Suppose the rigid transformations between three frames $\{f_i, f_j, f_k\}$ are $\mathbf{M}_{i\to j} = (\mathbf{R}_1, \mathbf{t}_1)$ and $\mathbf{M}_{i\to k} = (\mathbf{R}_2, \mathbf{t}_2)$, the pivot point $\mathbf{c}$ can be computed by solving the following linear system:

$$\begin{pmatrix} \mathbf{I} - \mathbf{R}_1 \\ \mathbf{I} - \mathbf{R}_2 \end{pmatrix} \mathbf{c} = \begin{pmatrix} \mathbf{t}_1 \\ \mathbf{t}_2 \end{pmatrix}. \tag{13}$$

For a ball pivot model candidate $\mathbf{c}$, we compute its support by measuring its fitting error to the full lifespan trajectory. Thus, taking points from other frames f_k relative to f_i, we measure:

$$D_{\mathbf{c}}(f_i, f_k) = \|\mathbf{p}_m^i - \mathbf{c}\| - \|\mathbf{p}_m^k - \mathbf{c}\|. \tag{14}$$

The point $\mathbf{p}_m^k \in f_k$ supports $\mathbf{c}$ if the fitting error is below a threshold $\varepsilon_{\mathbf{c}} = 0.05$. Similarly, ball joints are discarded by early rejection if their support is below 80% of the trajectory's lifespan.

Consensus voting. We fit a joint to a new trajectory candidate using the respective fitting error minimization equations 11, 12, 14 corresponding to hinge, slider and ball joints. For generality, we refer to these joints as RTM's denoted M. For each joint, we test it against all the trajectories in the data and considering their full lifespan.

Thus, for a given mobility model M and a trajectory T_k we measure its fitting residual $r(T_k, M)$ using equation 3. A trajectory is added to the consensus set of a joint M if the majority of points along its lifespan (here 80%) are within the error fitting threshold of M. Only mobility models with a consensus set larger than certain threshold (here 10% of all raw trajectories) are selected and their corresponding consensus trajectories removed from the dataset.

If two mobility models are identical, we merge their consensus set. Hinges are identical if they share the same axis; sliders are identical if their directions are the same; ball joints are identical if they share the same pivot. The process repeats until no model with sufficient support can be found, yielding a set of mobility models associated with their supporting 4D trajectory points.

Mobility graph. A by-product of our mobility fitting is a joint based skeleton which connects adjacent joints, yielding a global kinematic chain that governs the articulated motion. The skeleton graph is an abstract representation encoding joints and trajectory clusters as two types of nodes and their mobility relations as edges (in Figure 6 joints and red nodes connected by blue edges).

To compute the skeleton, each joint node connects to trajectory clusters in its consensus set. Naturally, these clusters define a piecewise rigid motion of shape parts (red nodes in our graph). Thus, each joint node connects to two or more rigid nodes forming a graph structure.

Example	Seq	Traj (ms)	Fit(ms)	Joints#
Robot	69	6,112	876	7H
Excav.	155	12,875	288	4H
Crane	167	13,574	344	1H, 1S
Tripod	101	9,547	234	1B
Chair	26	5,984	46	1H
Human	169	13,632	587	2B, 2H
Arm	26	6,145	74	3H, 1S
Chain	16	1,608	89	10H
Hand	151	12,755	1,672	14H
Two obj.	96	9,653	458	3H, 1S
Comp. robot	90	417,600	943	6H
Comp. car	90	846,000	327	1B, 2H

Table 1: *Summary of experiments presented in this paper: sequence length in frames (Seq), trajectory processing time (Traj), joint fitting time (Fit), and joints numbers for each type – hinge(H), slider(S), ball(B).*

5. Results and discussion

To demonstrate our method, we experiment with different objects performing various articulated motions. To evaluate performance and scalability we show different kinematic configurations, ranging from simple motions to more complex combinations of joints which generate compound motions.

In our tests, dynamics are captured using a standard Kinect device, which captures both depth and color frames of the dynamic sequence. In the preprocessing step, we use the scene flow algorithm of [JSGJC15] to extract dense trajectories from the dynamic sequence.

We test our method on an Intel® i7-6700K 4.00GHz with 16GB RAM. Experiments information is summarized in Table 1. Note that in all our experiments, processing times are very fast staying below 200ms per frame. In fact, trajectory processing (Traj. col) took the majority of time in comparison to 4D RANSAC (Fit col.). Thus, our method is very efficient, making it a good candidate in the future for achieving real-time performance rates.

In Figure 1 we capture the motion of a toy robot, whose parts move simultaneously, as seen in the top row. Articulated motion is defined by a 7 joint skeleton, generating intricate trajectories due to the simultaneous motion. Our method has managed to reconstruct the 7 hinge joints that define this motion (see accompanying video for full sequence). The bottom row demonstrates the reconstructed mobilities by positioning the joints in their correct 3D positions and orientations in each frame.

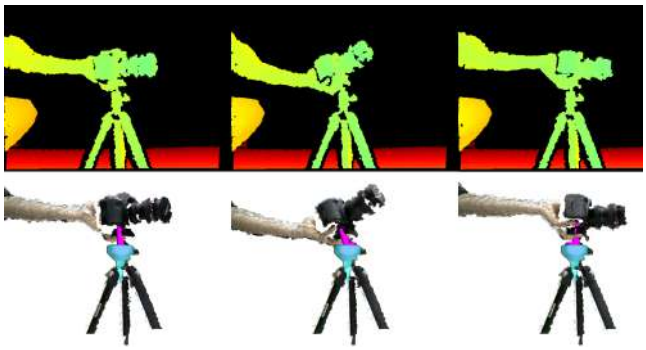


Figure 7: *Mobility reconstruction of a manually operated tripod. Top row are three frames from the scanned tripod articulation. Bottom row shows the fitted ball joint with the scanned point clouds.*

Figure 8: *Mobility reconstruction of a manually operated rotating chair. Top row are three frames from the scanned motion. Bottom row shows the reconstructed hinge joint fitted to scans.*

Figure 6 (top) shows a complex motion of a toy excavator performing common excavation actions. Motions are generated by 4 simultaneously operating hinges. The base and the excavator's arm are rotating together. Furthermore, the excavator base defines a compound joint consisting of two hinges with different orientations: one defining the rotation w.r.t the excavator's base and another the arm's lift. Note that reconstructed joints have the correct position and orientation w.r.t. the scanned data (top). Joints stay coherent over time, maintaining their relative position and only adjusting their angle. In the bottom row we show the reconstruction of mobility joints together with their kinematic graph.

In Figure 7, we show mobility reconstruction of a ball joint of a tripod motion. In contrast to the above toy examples operated by a remote controller, this object's articulation is manually operated. Therefore, articulation speed is non-constant adding a noise factor to trajectory data. Our method reconstructs the accurate joint position in the model as well as coherently tracks its articulation.

Figure 8 shows our mobility reconstruction on a scanned office chair with the back seat rotating around its axis. Although the chair seat is almost completely missing due to material reflections, we correctly fit the hinge there. This example is an excellent demonstration of our direct mobility reconstruction idea. While

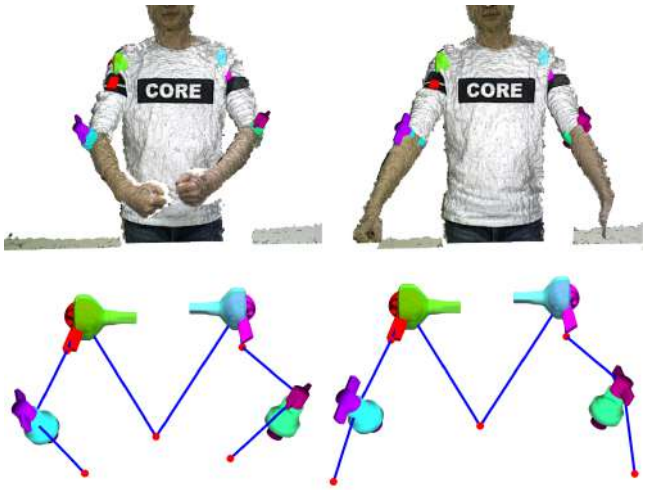


Figure 9: *Mobility reconstruction of a scanned human while moving hands, arms and shoulders.*

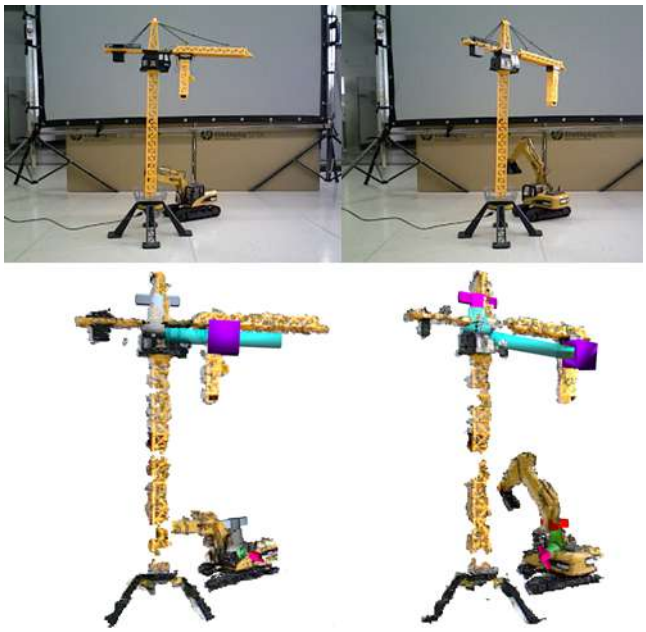
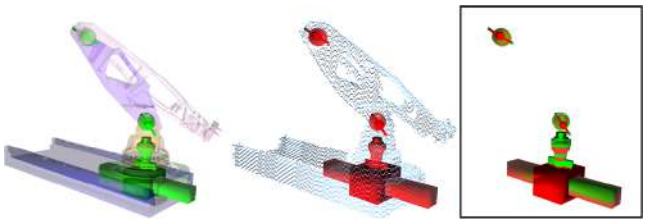


Figure 10: *Mobility reconstruction of a crane and an excavator moving simultaneously in the scene.*

reconstruction of the surface geometry would fail here, we directly compute the articulated motion parameters using our robust 4D RANSAC.

We experiment also with organic data, reconstructing the mobility from a scanned human motion, see Figure 9. Here, we capture a motion sequence of the upper torso of a human. Since human motion is closely related to articulated motion due to human bone



(a) (b) (c)

Figure 11: *Synthetic mechanic arm motion. (a) is the 3D animation with ground truth joints (green). (b) is the virtually scanned motion with reconstructed mobilities (red). (c) is a comparison between ground truth and reconstructed joints in terms of their position and orientation.*

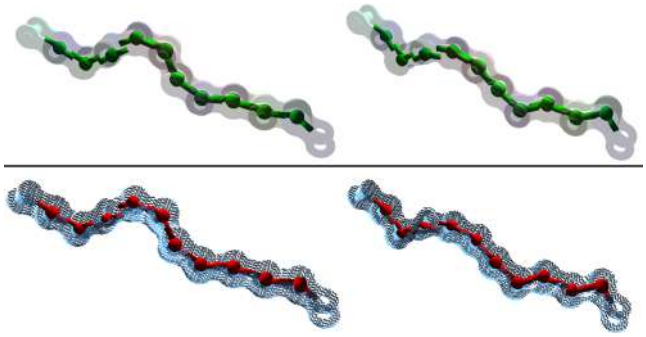


Figure 12: *A 3D chain model defined by* 10 *hinges which operate simultaneously and generate a complex compound motion. Top row shows frames from the ground truth animation and bottom row depicts our reconstructed hinges positioned in the virtually scanned data.*

structure, we correctly reconstruct joints using our robust technique.

Finally, our method is not limited by the number of operating objects in the scene nor their joints as long as they are captured properly. In Figure 10 we show a case of two articulated objects, a crane and an excavator, operating simultaneously in a scene with their mobility reconstructed.

Quantitative evaluation. We employ a virtual scanner and scan articulated 3D animations, yielding a raw dynamic point set. This yields a full simulation and proper evaluation of our result accuracy against ground truth. For a quantitative evaluation, we use 3D models with predefined ground truth mobilities. We save the model's joints positions and angles through the animation and compute new mobilities from the scanned points, using our method.

In Figure 11, a mechanic arm motion shows a comparison between ground truth joints (left col., green) and ours reconstructed from the virtual scanned motion (mid col., red). Qualitatively, our algorithm is accurate as both ground truth and reconstructed joints perfectly overlap (right col.).

In Figure 12 we demonstrate the scalability of our method using a virtual scan of a motorcycle chain with 10 hinges operating simul-

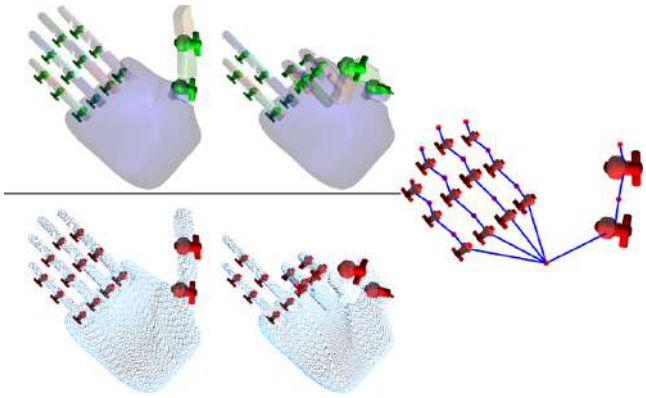


Figure 13: *Mobility reconstruction of a complex robot hand motion composed of* 15 *different joints.*

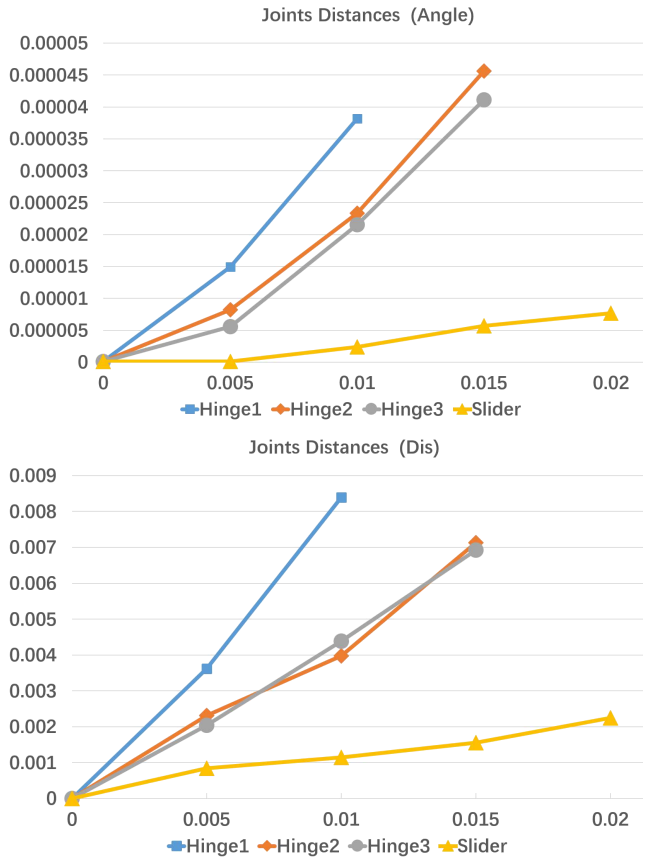


Figure 14: *Error fitting of our method compared with ground truth (in robot-arm example). Graphs measure average error per sequence for each of the 3 hinges and 1 slider. y-axis is the measured error, x-axis is the noise in position in % of bounding box diagonal. Top is average angle error and bottom is position distance error.*

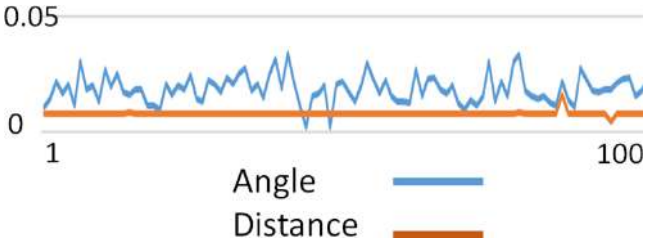


Figure 15: *Robustness to random selection.*

taneously. Hinges are green in the ground truth simulation in the top row. Our method has successfully reconstructed all hinges accurately (bottom row, in red) without a significant increase in the processing times per frame. Similarly, we demonstrate our method's scalability to both joints size and motion length on a complex articulated robot hand (Figure 13). The hand consists of 15 different joints operating simultaneously in a motion sequence of 15sec (150 frames at 10fps).

We measure the algorithm's accuracy, as the sum of distances in terms of vector orientation and angle between the reconstructed and ground truth joints. To evaluate robustness, we measure the accuracy with increased noise using our virtually scanned data. Thus, we gradually insert noise in the scanned points position (percentage of the bounding box diagonal) and measure the accuracy of mobility reconstruction w.r.t. ground truth. The results of these experiments are summarized in Figure 14. The plots show that our method is robust as accuracy error grows slowly with increasing noise-level. Our system was able to detect the correct joints with sufficient consensus up to 2% noise. Furthermore, to evaluate the method's robustness to random selection, we run our method with different random selections for 100 times (see Figure 15). The accuracy w.r.t. ground truth stayed nearly constant with position error staying below 4*mm* and angle error below 0.8 degrees.

Comparison. To evaluate our work, we compare the results with that of Chang and Zwicker's global registration approach, which utilized a reduced deformable model to simultaneously optimize scan alignment and model reconstruction. We run our algorithm on the same data sets of articulated robot and truck as in [CZ11]. Our mobility reconstruction is qualitatively compared to theirs in Figure 16. We found that both methods reconstruct mobility joints in these examples with equivalent quality. Nevertheless, our method is more efficient due to the fact that our algorithm avoids the expensive coherent geometry reconstruction. Since authors in [CZ11] did not provide any accuracy and scalability evaluation, we could not compare these terms.

Limitations. In terms of limitations, our method assumes an articulated type of motion where parts transform rigidly. Thus, an extension to elastic deformations seems non-straightforward. In the case of the stuffed animal toy in Figure 17, our approach fails to discover the rigid segments, as well as the mobility joints. The reason lies in the fact that there are no obvious articulations in the motion sequence and joints can not fit reliably. Furthermore, the performance bottleneck of our algorithm is the trajectory size. We are required

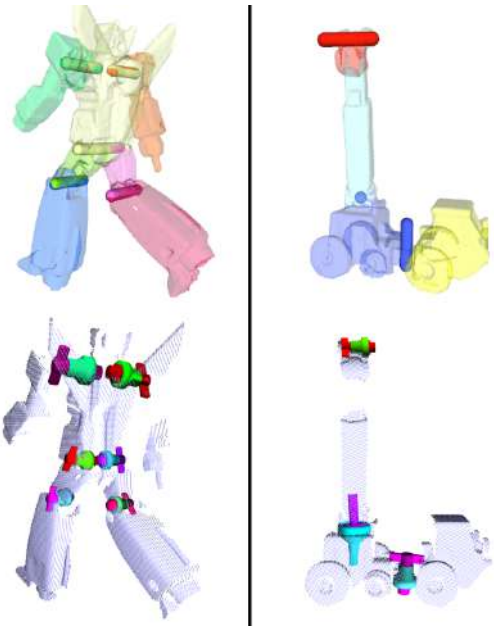


Figure 16: *Comparison of Chang and Zwicker [CZ11] (top tow) and our method (bottom row).*

Figure 17: *Applying our method on a hand puppet motion with no clear articulation yields no significant joints and consensus trajectory sets (right).*

to significantly simplify trajectories in the preprocessing step for efficiency.

6. Conclusions and future work

In this paper we introduce a method for mobility reconstruction from scanned dynamic points. Our algorithm takes a random sampling approach (4D RANSAC) which robustly reconstructs the mobility joints in the data. Typically, articulated motions are defined by a discrete set of piecewise-rigid transformations. Our algorithm

leverages this fact by searching a restricted space of deformations. Thus we pre-define a discrete set of mobility joints which we fit to the 4D data. A key observation in our approach is to avoid the common pitfall of reconstructing the transforming surface geometry. Instead, we directly reconstruct mobilities from point trajectories. Results show the ability to reconstruct a variety of articulated motions of various objects. Additionally, evaluation shows the accuracy of our method and its robustness to noise and scalability.

In the future we would like to further advance this technique in two ways. First, we would like to explore accelerations of our technique using modern GPU parallel computations. Second, we would like to adapt the 4D RANSAC for online processing of streaming data. This fits well in the state-of-the-art real-time scanning of dynamic activities. In this context it would require turning our random sampling and consensus voting scheme into a progressive one.

Acknowledgments. We thank the reviewers for their valuable comments. We would also like to acknowledge our research grants: National 973 Program (2015CB352501), NSFC (61232011, 61572507, 61532003), NSFC-ISF (61561146397), ISF (1106/11), The Fundamental Research Funds of Shandong University (2015JC051), Shenzhen Knowledge innovation program for basic research (JCYJ20150402105524053).

References

[AHB87] ARUN K., HUANG T., BLOSTEIN S.: Least-squares fitting of two 3-d point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (1987), 698–700. 4

[AKI05] AGRAWAL M., KONOLIGE K., IOCCHI L.: Real-time detection of independent motion using stereo. In *Proceedings of the IEEE Workshop on Motion and Video Computing* (2005), pp. 207–214. 2

[CLM*12] CHANG W., LI H., MITRA N. J., PAULY M., WAND M.: Dynamic geometry processing. In *Eurographics 2012 STAR* (2012). 2

[CZ08] CHANG W., ZWICKER M.: Automatic registration for articulated shapes. In *Computer Graphics Forum (Special Issue of SGP)* (2008), pp. 1459–1468. 2, 3

[CZ11] CHANG W., ZWICKER M.: Global registration of dynamic range scans for articulated model reconstruction. *ACM Trans. on Graph 30*, 3 (2011), 697–706. 2, 3, 9

[GMGP05] GELFAND N., MITRA N. J., GUIBAS L. J., POTTMANN H.: Robust global registration. In *Proc. of Symp. on Geom. Proc.* (2005), p. Article No. 197. 3

[HAWG08] HUANG Q.-X., ADAMS B., WICKE M., GUIBAS L. J.: Non-rigid registration under isometric deformations. In *Proc. of Symp. on Geom. Proc.* (2008), pp. 1449–1457. 3

[HRF12] HERBST E., REN X., FOX D.: Object segmentation from motion with dense feature matching. In *ICRA Workshop on Semantic Perception, Mapping and Exploration* (2012). 2

[HRF13] HERBST E., REN X., FOX D.: Rgb-d flow: Dense 3-d motion estimation using color and depth. In *IEEE International Conference on Robotics and Automation* (2013), pp. 2276–2282. 2

[Joh73] JOHANSSON G.: Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics 14*, 2 (1973), 201–211. 2

[JSGJC15] JAIMEZ M., SOUIAI M., GONZALEZ-JIMENEZ J., CREMERS D.: A primal-dual framework for real-time dense rgb-d scene flow. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on* (2015), pp. 98–104. 4, 6

[JT05] JAMES D. L., TWIGG C. D.: Skinning mesh animations. In *Proc. of SIGGRAPH* (2005), pp. 399–407. 3

[LAGP09] LI H., ADAMS B., GUIBAS L. J., PAULY M.: Robust single-view geometry and motion reconstruction. In *Proc. of SIGGRAPH Asia* (2009), pp. 175:1–175:10. 2

[Low04] LOWE D. G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* (2004), 91–110. 4

[MFO*07] MITRA N. J., FLORY S., OVSJANIKOV M., GELFAND N., GUIBAS L., POTTMANN H.: Dynamic geometry registration. In *Proceedings of the Symposium on Geometry Processing* (2007), pp. 173–182. 2

[MGP06] MITRA N. J., GUIBAS L. J., PAULY M.: Partial and approximate symmetry detection for 3D geometry. *ACM Trans. on Graph 25*, 3 (2006), 560–568. 3

[MMH12] MUFTI F., MAHONY R., HEINZMANN J.: Robust estimation of planar surfaces using spatio-temporal ransac for applications in autonomous vehicle navigation. *Robot. Auton. Syst. 60*, 1 (Jan. 2012), 16–28. 3

[PG08] PEKELNY Y., GOTSMAN C.: Articulated object reconstruction and markerless motion capture from depth video. *Computer Graphics Forum (Special Issue of Eurographics) 27*, 2 (2008), 399–408. 2, 3

[PSDB*10] POPA T., SOUTH-DICKINSON I., BRADLEY D., SHEFFER A., HEIDRICH W.: Globally consistent space-time reconstruction. *Comput. Graph. Forum* (2010), 1633–1642. 2

[Reu76] REULEAUX A. F.: *The Kinematics of Machinery: Outlines of a Theory of Machines*. Macmillan, 1876. 5

[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. In *ACM SIGGRAPH Asia 2008 papers* (2008), SIGGRAPH Asia '08, pp. 1–10. 2

[SB15] STÜCKLER J., BEHNKE S.: Efficient dense rigid-body motion segmentation and estimation in rgb-d video. *Int. J. Comput. Vision 113*, 3 (July 2015), 233–245. 2

[SWG08] SÜSSMUTH J., WINTER M., GREINER G.: Reconstructing animated meshes from time-varying point clouds. In *Proceedings of the Symposium on Geometry Processing* (2008), pp. 1469–1476. 2

[TBW*12] TEVS A., BERNER A., WAND M., IHRKE I., BOKELOH M., KERBER J., SEIDEL H.-P.: Animation cartography: Intrinsic reconstruction of shape and motion. *ACM Trans. Graph. 31*, 2 (2012), 12:1–12:15. 2

[TF08] TOLDO R., FUSIELLO A.: Robust multiple structures estimation with j-linkage. In *Proceedings of the 10th European Conference on Computer Vision: Part I* (2008), ECCV '08, pp. 537–547. 5

[WAO*09] WAND M., ADAMS B., OVSJANIKOV M., BERNER A., BOKELOH M., JENKE P., GUIBAS L., SEIDEL H.-P., SCHILLING A.: Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data. *ACM Trans. Graph. 28*, 2 (May 2009), 15:1–15:15. 2

[YSL*14] YAN F., SHARF A., LIN W., HUANG H., CHEN B.: Proactive 3d scanning of inaccessible parts. *ACM Trans. Graph.* (2014), 157:1–157:8. 4