Realtime Reconstruction of an Animating Human Body from a Single Depth Camera

Yin Chen, Zhi-Quan Cheng*, Chao Lai, Ralph R. Martin, Gang Dang

Abstract—We present a method for realtime reconstruction of an animating human body, which produces a sequence of deforming meshes representing a given performance captured by a single commodity depth camera. We achieve realtime single-view mesh completion by enhancing the parameterized SCAPE model. Our method, which we call *Realtime SCAPE*, performs full-body reconstruction without the use of markers. In Realtime SCAPE, estimations of body shape parameters and pose parameters, needed for reconstruction, are decoupled. Intrinsic body shape is first precomputed for a given subject, by determining shape parameters with the aid of a body shape database. Subsequently, per-frame pose parameter estimation is performed by means of linear blending skinning (LBS); the problem is decomposed into separately finding skinning weights and transformations. The skinning weights are also determined offline from the body shape database, reducing online reconstruction to simply finding the transformations in LBS. Doing so is formulated as a linear variational problem; carefully designed constraints are used to impose temporal coherence and alleviate artifacts. Experiments demonstrate that our method can produce full-body mesh sequences with high fidelity.

Index Terms—Realtime reconstruction, Human animation, Depth camera, SCAPE.

1 INTRODUCTION

Realtime reconstruction of animating full-body performances is of use in a range of applications requiring 3D personalized avatars, for example movie production and game control.

Here, we present an approach to markerless realtime reconstruction of an animating human, captured using a single commodity depth camera such as the Microsoft Kinect [1]. Single-view capture offers several advantages over multi-view techniques, including lower price, simpler calibration, and more flexible setup. However, there are several technical challenges in using such an approach. Firstly, depth data from a single low-price camera are typically very noisy, and suffer from significant missing regions due to self-occlusion. Secondly, computing the deformation giving the pose for each frame is inherently a nonlinear problem, so is hard to solve in real time, especially if there is rapid motion between adjacent frames. Lastly, to reconstruct a smooth full-body animation from low quality depth data, temporal coherence needs to be carefully taken into account in pose estimation—yet without markers or manual assistance to build inter-frame correspondences, coherence is difficult to ensure.

- **Corresponding author E-mail: cheng.zhiquan@gmail.com*
- *Yin Chen, Chao Lai, and Gang Dang are with the Computer School, National University of Defense Technology, Changsha, Hunan, China, 410073. Homepage: http://www.computer-graphics.cn.*
- *Zhi-Quan Cheng is with the Avatar Science Company, Changsha, Hunan, China, 410013.*
- *Ralph R. Martin is with the School of Computer Science & Informatics, Cardiff University, Cardiff, Wales, UK, CF24 3AA.*

We address these challenges by treating full-body reconstruction from single-view data as a parameterized template fitting problem. In particular, we extend the SCAPE *(Shape Completion and Animation of PEople)* approach [2] to provide realtime performance. The original SCAPE method was devised for reconstruction of a complete human body from a set of markers attached to the target subject. Directly using unmodified SCAPE for full-body reconstruction is very time-consuming: e.g. see the markerless method in [3].

Fortunately, estimation of body *shape* and *pose* parameters can be *decoupled* when using the SCAPE model. The intrinsic body shape of a performing subject does not change, and so body shape parameters can be estimated offline beforehand, leaving just the pose parameters to be determined for each frame of a motion sequence. We take advantage of this approach, but to enable realtime reconstruction, we further enhance SCAPE, which formulates pose parameter computation in terms of linear blending skinning (LBS) deformation [4]. The LBS approach represents pose using skinning weights and transformations. The skinning weights are again fixed with respect to time, so can also be learnt offline from a human database, reducing realtime reconstruction to the solution of a *linear* variational problem to determine a set of transformations. To provide high-quality output with temporal coherence and avoiding deformation artifacts, carefully designed constraints are also imposed.

In summary, the contribution of Realtime SCAPE is a method for accurate, realtime, geometry and motion reconstruction of an animating human from a single low-cost depth camera: see Figure 1. Its key features

Fig. 1. Frames from a performance, showing: photograph of the pose, depth data (left), and watertight mesh produced in realtime (right).

are:

- Two stages of parameter decoupling, permitting pose estimation at realtime speed.
- Constrained pose transformation recovery to suppress deformation artifacts and ensure temporal coherence.
- Robust reconstruction results, even for challenging performances, e.g. those including 360° rotations of the human body.

2 RELATED WORK

Human body reconstruction has been studied both theoretically and algorithmically in computer vision and graphics. Existing approaches can be classified as single- or multi-view, according to the number of cameras used. We focus on single camera methods and related recent advances; see [3], [5], [6], [7] for comprehensive reviews.

Shape/geometry reconstruction. The Kinect [1] is a representative low-cost depth camera, producing low-quality data with a high rate. The GPU-based KinectFusion method [8] can be used for both tracking and static surface reconstruction. In particular, we utilize KinectFusion to capture *static* initial body shape data as a 3D mesh, which is used *offline* to determine parameters of an intrinsic body shape model particular to the subject.

Even large gaps in captured data can be overcome by use of template-based registration, which leads to a template fitting problem [2], [9], [10], [11], [12], [13], [14]. Earlier work often tracked marker points for correspondence estimation [2], [9], but more recently, markerless reconstruction methods [10], [11], [12], [13], [14] have made great progress. The single-view method in [12] is a good example, but it requires high-quality data, and is unable to handle relatively low-quality depth data such as that provided by a Kinect device. Our method is also markerless, and can robustly reconstruct human geometry and motion from low-quality data.

We use a SCAPE model as the basis for shape and pose reconstruction [2]. Two important lines of research have emerged in this area, those using 2D images [15], [16], and those using a single depth camera [3], [17], [18]. The latter category is most similar to our work: it estimates body shape using image silhouettes and depth data using a single Kinect device. However, the method in [3] takes approximately one hour to produce a result, which is far too slow for many practical applications, and underlines the difficulties in reconstructing human geometry and motion from single-view data in real time.

Pose/motion capture. A skeleton provides a compact object representation, summarizing both geometrical and topological information, and so is frequently adopted as a proxy in place of capturing accurate geometry when estimating motion from a single camera. Weiss [19] combines motion capture with physically-based simulation to obtain skeleton-based motion results using a traditional 2D camera, but manual labeling of key frames is required. The same group's later work [20] uses a depth camera, and provides a more accurate solution based on an iterative process of tracking and detection. Related research estimate 3D pose in realtime by using trained randomized decision trees [21], a context-sensitive regression forest [22], or one-shot skeleton fitting using Vitruvian manifold methods [23]. These methods, as well as those in [24], [25], [26], [27], [28], all rely on a database of prerecorded human motions. However, such a database cannot include every possible pose which may occur in a human performance. Note further that the main goal of such skeleton tracking methods is to estimate the motion in terms of parameters describing

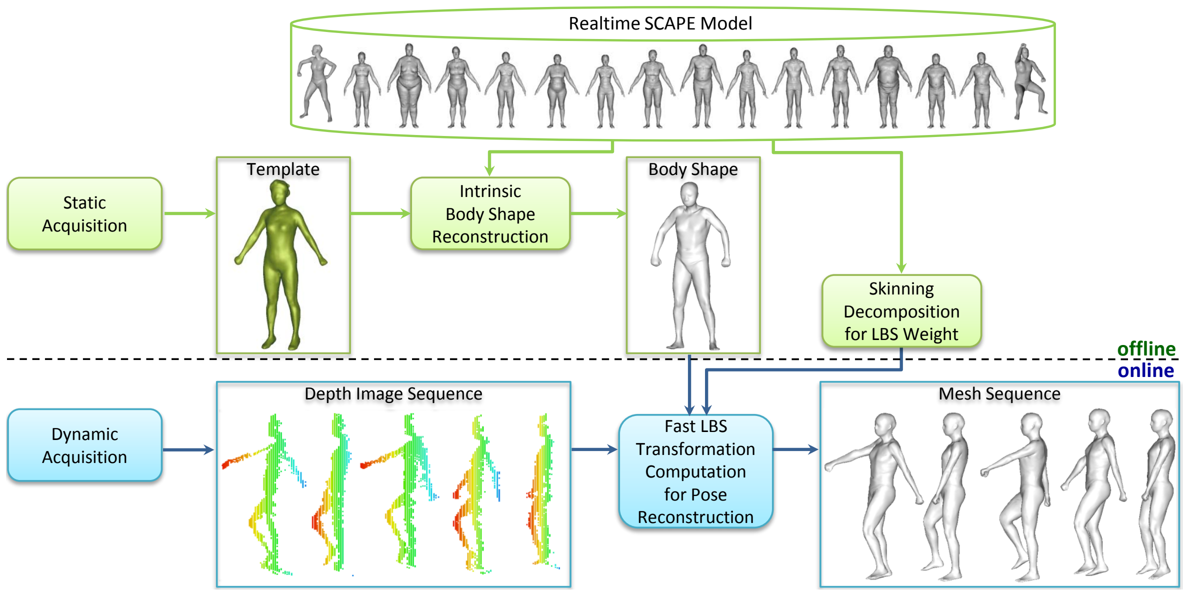


Fig. 2. Framework. Top: *Realtime SCAPE* model. Center: offline template acquisition, intrinsic body shape reconstruction, and weight computation for use in linear blending skinning (LBS). Bottom: online animating human body reconstruction, matching deformed intrinsic body shape to each dynamic data frame, via rapid computation of the LBS pose transformations.

the skeleton, whereas our goal is to perform surface reconstruction from each frame of depth data. Thus, methods such as those in [20], [21], [24], [25], [26], [29], cannot be compared directly to ours. These differences in goals mean that they are complementary rather than competing.

As noted in both recent [18], [27] and earlier [24], [25], [26], [29] work, performances including such motions as 360° human rotation present a severe challenge. For example, [28] uses body-worn inertial sensors to help in such cases. Similar problems also arise in the state-of-the-art skeleton extraction approach taken by the Kinect SDK [1]. This shortcoming was successfully overcome in [20], by taking advantage of temporal coherence between neighboring frames. We also use temporal cues to allow such performances to be robustly handled by our method, without the need for complementary sensors.

Linear blending skinning Linear blending skinning (LBS) [4] is a popular deformation model, providing fast performance and good deformation qualities. [30] proposed an automatic algorithm to extract an LBS model from a set of example poses based on rigid bones; it borrowed the term *skinning decomposition* from [31] to refer to the inverse problem of fitting an LBS model to measured data. The latter is formulated as a constrained optimization problem in which the least-squares errors of vertex positions reconstructed by LBS are minimized; a linear solver iteratively updates a weight map and the bone transformations. However, the speed of this approach is far from sufficient for realtime work. We build on these ideas, and further decouple pose deformation using the human database to significantly increase performance.

3 Overview

Fig. 2 illustrates our framework, which has three main components: a modified SCAPE model (our *Realtime SCAPE* model), an offline preprocessing module, and a module for online reconstruction from the single depth camera.

SCAPE [2] describes the human body using coupled shape and pose parameters. We modify the original SCAPE model (see Section 4) in *Realtime SCAPE* to meet the needs of realtime reconstruction. The shape model is revised to include offline construction of a template, based on scanned data, to capture the subject's individual body shape. To improve speed, the pose representation used in the original SCAPE approach is replaced by LBS decomposition [30], [31]. This LBS decomposition is represented by sparse rigid transformations and weights. The weights are also learnt offline for use in online pose determination, reducing the dimensionality and difficulty of the geometry and motion reconstruction problem. The only parameters remaining to be estimated in real time are a set of rigid transformations.

During offline preprocessing (see Section 5), KinectFusion [8] is used to provide an initial mesh representing a particular subject. The subject stands in a static T-pose. Depth data is captured and registered into a single coordinate system, by moving the camera

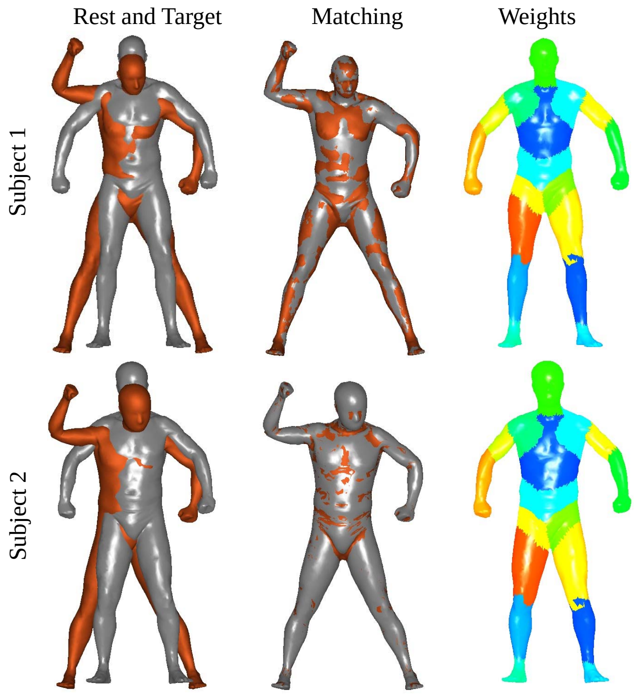


Fig. 3. Reconstructed poses matched to real data, for two subjects. Left: rest pose (grey) and target pose (orange). Center: match between reconstructed pose (grey) and target pose (orange). Right: bones for which each triangle has largest weight.

around the subject until sufficient data have been acquired. This mesh, together with intrinsic attributes of weight, height and gender, are used to determine the body shape parameters in the shape deformation model which describe this particular individual. We call this the subject's *intrinsic body shape* model. LBS skinning weights are also determined.

Online motion capture of the subject is then performed using the Kinect, which provides a depth image sequence with resolution 320×240 at 30 frames per second. We use a linear variational approach to determine the transformation parameters, which are used together with the learnt weight parameters of LBS to reconstruct the motion of the performer from the single depth camera (Section 6).

4 Realtime SCAPE model

4.1 SCAPE overview

SCAPE [2] is a decoupled deformation model which separately accounts for shape variation between different people, and changes in pose.

- *Shape* is parameterized by $\Theta = U\theta + \mu$, where μ is mean human body shape, and U are eigenvectors found by principal component analysis (PCA). Both μ and U can be directly determined by using a reference human database. The parameter vector θ of linear coefficients characterizes a particular subject.
- *Pose* is parameterized by a set of pose matrices Q, which determine the articulated pose.

These two sets of parameters may be combined to reconstruct realistic results for various humans in different poses.

The SCAPE model [2] deforms a body template $\mathcal{M}$ to fit a particular mesh $\mathcal{M}^{sp}$, corresponding to a subject s in the database in pose p. In detail, consider some triangle in $\mathcal{M}$ with vertices $(v_{k_1}, v_{k_2}, v_{k_3})$. Shape and pose deformations are applied in turn to transform it into its counterpart in $\mathcal{M}^{sp}$. Deformations are computed in terms of the triangle's local coordinate system, obtained by translating point v_{k_1} to the global origin. Thus, deformations are applied to triangle edges $e_{k_n} = v_{k_n} - v_{k_1}, n = 2, 3$. Given Q, Θ, for each template triangle, SCAPE can thus determine a mesh for a specific person and pose by finding the set of vertex locations $v_1, \cdots, v_{|V|}$ (where $|V|$ is the number of mesh vertices) that minimizes the reconstruction error for the observed triangle edges:

$$\arg\min_{v_1,\cdots,v_{|V|}} \sum_k \sum_{n=2,3} \|Q_k^{sp}\Theta_k^{sp} e_{k_n} - (v_{k_n} - v_{k_1})\|^2. \quad (1)$$

4.2 Realtime SCAPE using LBS-based pose deformation

In our enhancements to SCAPE for realtime performance, we replace the pose deformation matrices Q by the LBS technique [4]. To learn our modified Realtime SCAPE model parameters, we used the CAESAR human database [32], which includes 2400 subjects in $|P| = 70$ poses. Each subject is represented by a closed mesh, fitted to a template $\mathcal{M}$ with 12,500 vertices and 25,000 faces.

LBS synopsis. In LBS, pose is represented using transformations of rigid bones relative to a rest pose, and skinning weights. For a subject s, the weight w_{ij} represents the influence of the j-th bone on the i-th vertex. Each vertex is associated with no more than $|N|$ bones, and there are $|B|$ bones in total, If v_i^r is the position of the i-th vertex in the rest pose, and each R_j^p and T_j^p are a 3×3 rotation matrix and 3×1 translation vector transforming the j-th bone in the p-th pose, then the deformed i-th vertex, v_i^p, is given by:

$$v_i^p = \sum_{j=1}^{|B|} w_{ij}(R_j^p v_i^r + T_j^p), \tag{2a}$$

subject to :

$$w_{ij} \geq 0, \quad \forall i,j, \tag{2b}$$

$$\sum_{j=1}^{|B|} w_{ij} = 1, \quad \forall i, \tag{2c}$$

$$|\{w_{ij} | w_{ij} \neq 0\}| \leq |N|, \quad \forall i, \tag{2d}$$

$${R_j^p}^{\mathbf{T}} R_j^p = I, \quad |R_j^p| = 1, \quad \forall p,j. \tag{2e}$$

Eqns. (2b–2d) ensure physically meaningful bone-vertex influences, while Eqn. (2e) ensures that R_j^p is a proper rotation matrix.

Skinning decomposition. Following [30], the transformations and weights may be determined by solving a constrained least squares optimization problem; the example poses in the human database are used as data to learn the set of weights:

$$\arg\min_{w,R,T} \sum_{p=1}^{|P|} \sum_{i=1}^{|V|} \|v_i^p - \sum_{j=1}^{|B|} w_{ij}(R_j^p v_i^r + T_j^p)\|^2, \tag{3}$$

subject to the constraints in Eqns. (2b–2e).

Each subject s has a variety of poses in the human database. The subject's body surface is initially automatically decomposed with faces allocated to $|B|$ rigid bones ($|B| = 17$ in practice), using a rigging technique [33]. As all shapes in the database have the same topology, decomposition of one subject can be directly transferred to all other subjects. We define neighbors for each bone. The weights of a vertex v belonging to bone b are non-zero weights only for b and its neighboring bones. Since each bone has at most 3 neighboring bones, $|N| = 4$.

The weights are determined by iteratively solving Eqn. (3). Since we have initial vertex clusters for each bone, we can initialize each R_j and T_j using the method in [30]. Then, for every pose of s, the LBS weights W and transformations R, T are iteratively updated by alternating two steps, until convergence, or a maximum number of iterations (experimentally set to 20) has been reached. These steps are:

Weight computation. The bone transformations are fixed, and W optimized by solving a constrained least squares problem as in [30].

Transformation computation. The weights W are fixed, and optimization is performed to find the bone transformations, via LBS minimization as in Eqn. (3). The objective function is now:

$$\min_{R,T} E = \min_{R,T} \sum_{p=1}^{|P|} \sum_{i=1}^{|V|} \|v_i^p - \sum_{j=1}^{|B|} w_{ij}(R_j^p v_i^r + T_j^p)\|^2 \tag{4}$$

subject to: ${R_j^p}^{\mathbf{T}} R_j^p = I, \quad \det R_j^p = 1, \quad \forall p,j.$

We solve Eqn. (4) iteratively after linearizing the rotation matrices. Specifically, when optimizing R, we use the standard approximation $R_{\text{new}} \approx (I + \hat{R})R_{\text{old}}$, where the vector $r = (r_1, r_2, r_3)$ is a linear approximation for a small rotation $\hat{R}$:

$$\begin{pmatrix} 0 & -r_3 & r_2 \\ r_3 & 0 & -r_1 \\ -r_2 & r_1 & 0 \end{pmatrix}. \tag{5}$$

This quickly converges to a local optimum of the objective function in Eqn. (4). This approach converts the LBS optimization problem into a linear variational problem which can be rapidly solved.

Our experiments using the CAESAR human database [32], (e.g. see Fig. 3) indicate that essentially identical weights are obtained for all human subjects, and hence do not need redetermination for new subjects.

Decoupled Realtime SCAPE. In our Realtime SCAPE model, the PCA parameters θ describing shape deformation are learnt as described in Section 5. Pose deformation is represented in terms of sparse rigid bone transformations and the weight map, greatly reducing the dimensionality of the learning problem. The learnt model contains $|B| \times |P|$ rotation transformations plus a weight vector, where the same weight map W is used for *all* subjects in *any* pose, while the rotation R_s^p is similar for all subjects in a given pose p.

Our tests have shown that the Realtime SCAPE model with LBS decomposition can accurately approximate all test subjects in a variety of poses. Example matches between the reconstructed pose and real data are shown in Fig. 3, illustrating the high quality of results obtained. As the same weight map is used for all subjects, it can be computed once during offline Realtime SCAPE analysis, and saved for direct application during online motion reconstruction, helping to meet the realtime goals.

5 Offline Intrinsic Body Shape Reconstruction

We start by scanning the subject in an initial static T-pose, using KinectFusion [8] to create a mesh, which is used for offline reconstruction of the subject's intrinsic body shape. An objective function is used to determine various body shape attributes (represented in PCA space), while minimizing the difference between the target shape and the mesh:

$$\min E_{\text{shape}} = \arg\min_{\theta}(E_{\text{ap}} + \lambda_1 E_{\text{diff}}), \tag{6}$$

where λ_1 is experimentally set to 2. The two terms have the following meanings:

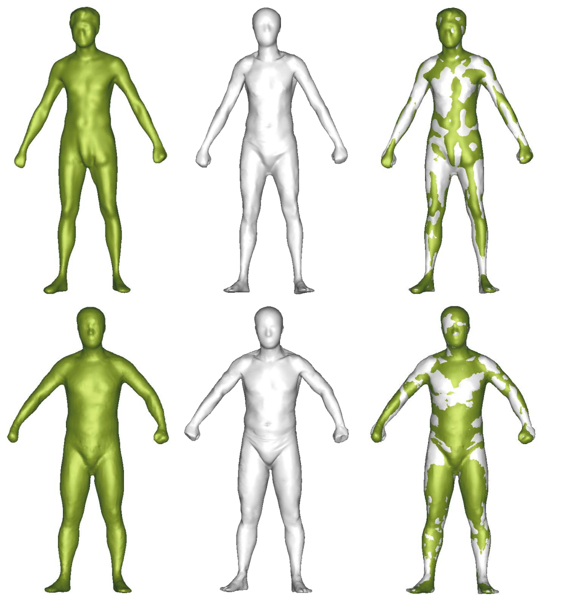


Fig. 4. Left: pre-scanned template. Center: intrinsic body shape reconstructed from it, taking into account known attributes of height, weight, and gender. Right: match between template and intrinsic body shape.

- E_{ap} is an attribute-preserving term which tries to enforce the known height, weight, and gender of the subject. Tthe method in [15] is followed to constrain shape deformation to variation in a subspace orthogonal to these three attributes.
- E_{diff} measures the difference between the target shape and the mesh, using the bi-directional objective function from [16].

Finding the vector θ of linear coefficients that characterizes the input subject provides the model of the subject's intrinsic body shape. Intrinsic shapes for two subjects are shown in Fig. 4. As can be seen, the reconstructed body shapes are plausible and fit the scanned data well.

6 Realtime full-body capture

We now explain how the Realtime SCAPE model provides online full-body reconstruction from a single depth camera. It reconstructs complete geometry, even when the input data suffers from self-occlusion, as well as the motion for an animating subject.

In the model, the parameters θ, W, R, and T model the specific shape and pose. We must determine suitable values to provide a mesh sequence consistent with successive depth images. The *shape* parameters θ for the particular subject are determined during initial offline processing, as explained in Section 5. The LBS weight map W is fixed for all subjects, and is learnt during Realtime SCAPE analysis, as explained in Section 4. The remaining unknown variables to be found per depth image are the transformations R, T.

6.1 Transformation formulation

The transformation is determined by optimizing a function with four terms which represent:

1) how well the reconstructed mesh fits the current frame's depth data,
2) the constraint that neighboring bones remain connected,
3) inertia of rigid bone rotation,
4) orientation preservation for certain bones.

Mathematically, this leads to the formulation:

$$
\begin{aligned}
\min_{R,T} E = \min_{R,T} \sum_{t=1}^{|t|} \sum_{i=1}^{|V|} \{ \| \hat{v}_i^t - \sum_{j=1}^{|B|} w_{ij} (R_j^t v_i^r + T_j^t) \|^2 + \\
\alpha_1 \sum_{j=1}^{|B|} \sum_{l=1}^{|B|} \frac{w_{ij} w_{il}}{\tau_{jl}} \| R_j^t v_i^r + T_j^t - R_l^t v_i^r - T_l^t \|^2 + \\
\alpha_2 \sum_{j=1}^{|B|} \| R_j^t - R_{j_{\text{parent}}}^t R_{j_{\text{local}}}^t) \|^2 + \\
\alpha_3 \sum_{j=1}^{|B_s|} \| R_j^t d_j^t - R_{j_{\text{parent}}}^t d_j^t \|^2 \}. \quad (7)
\end{aligned}
$$

The weights α_1, α_2 and α_3 are experimentally set to 10, 5 and 1 respectively. We now explain each term in detail.

Goodness of fit. The reconstructed mesh should agree with the observed depth map. Fitting error is measured in terms of the correspondence between each mesh point $v_i^t = \sum_{j=1}^{|B|} w_{ij}(R_j^t v_i^r + T_j^t)$, and $\hat{v}_i^t$, the closest point in the depth data in frame t.

Joint constraints. A joint is any mesh region influenced by more than one bone. Joint constraints serve to keep bones connected. We formulate them as in [34]; $\tau_{jl} = \sum_{j=1}^{|B|} \sum_{l=1}^{|B|} w_{ij} w_{il}$ is a normalization factor. In order to determine which vertices belong to a joint, we use products of weight functions: the joint region for a pair of bones j and l comprises those vertices v_i for which $w_{ij} w_{il} > 0$.

Inertia of local rotation. Physics determines that each bone should maintain its state of rest or uniform local rotation unless acted upon by an external force. As Fig. 5(right) shows, bones in the articulated body are connected in a tree structure. The rotation of bone j in frame t combines its own local rotation with the rotation of its parent in the tree: $R_j^t = R_{j_{\text{parent}}}^t R_{j_{\text{local}}}^t$. To provide inertia, $R_{j_{\text{local}}}^t$ for frame t remains unchanged from frame $t-1$, $R_{j_{\text{local}}}^t = R_{j_{\text{local}}}^{t-1}$, so is directly computed from $R_{j_{\text{local}}}^{t-1}$ at frame $t-1$. Bones are computed in top-down tree order, therefore $R_{j_{\text{parent}}}^t$ is already known at frame t, while $R_{j_{\text{root}}}^t$ remains fixed as an identity transformation. (The root does not correspond to any body part and merely serves as a reference for other body parts—see Fig. 5).

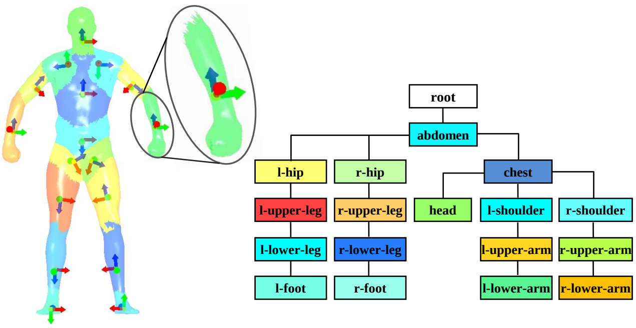


Fig. 5. Body representation. Left to right: mesh regions associated with bones, close-up of a special bone with axis shown by a red arrow, and bone tree.

Main-axis orientation invariance. Seven particular bones: those for the head, feet, forearms, and lower legs, are treated specially. The corresponding body parts are approximately cylindrical, and have limited freedom of movement. Each can only rotate about a main axis in its local reference frame, with one degree of freedom. Thus, each has a chosen axis attached to it whose direction d is resistant to variation during the motion. This axis attempts to merely follow changes induced by its parent, and refrains from introducing changes of its own: ideally $R_j^t d_j^t$ should be close to $R^t_{j_{\text{parent}}} d_j^t$. This constraint helps prevent *candy-wrapper* artifacts, where parts of the body near joints are unnaturally twisted like a candy wrapper, a problem discussed in [35].

These four terms play different roles during online reconstruction. The fitting and joint constraint terms are essential, and have already been used in previous reconstruction algorithms, such as [34]. While using these two obvious terms alone leads to a basically correct mesh, the results typically suffer from both jitter, and candy-wrapper artifacts. Clear improvements result from adding the inertia term to give temporal smoothness, and the final term to solve the candy-wrapper problem, as can be seen in Fig. 6.

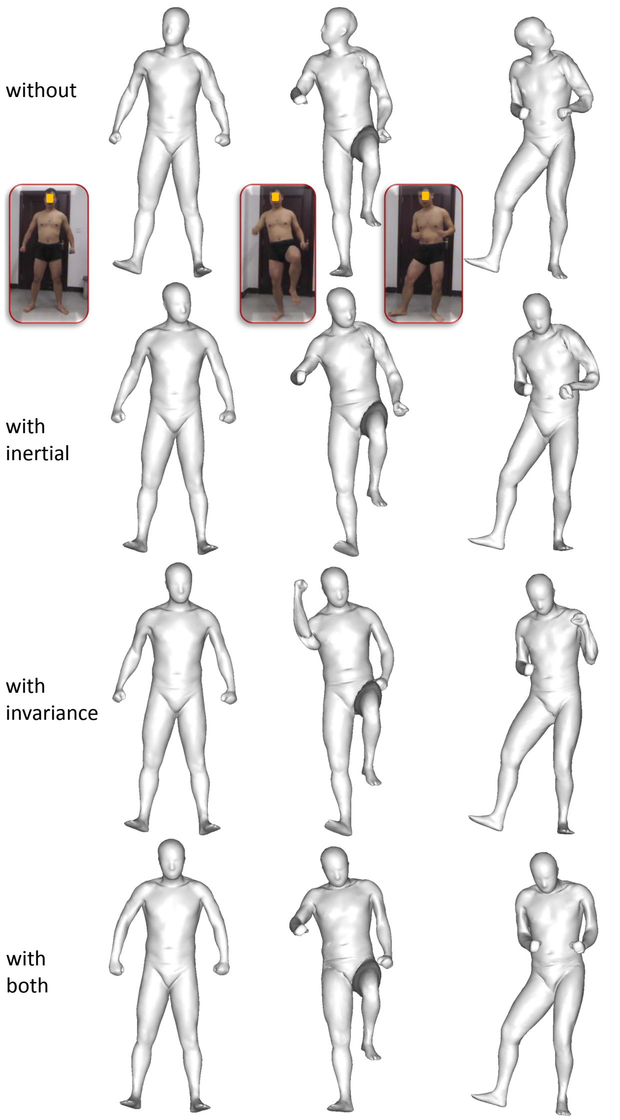


Fig. 6. Effects of the last two terms in Eqn. 7. Top: without additional terms: head orientation jitter and left shoulder candy-wrapper artifact present. Row 2: inertia term only. Row 3: main-axis orientation invariance term only. Bottom: both additional terms: jitter and artifacts are absent.

6.2 Reconstruction of animating subject

During online reconstruction, the performer starts from a predetermined static T-pose, then moves in front of the single depth camera. We compute R^t, T^t by minimizing the function in Eqn. 7, using the solution in frame $t-1$ to initialize computation of a local minimum in frame t.

Utilizing the expected temporal coherence of the transformation in this way helps to quickly determine the solution. In detail, given the transformation R_j^{t-1} in the previous time step for some rigid bone j, we solve R_j^t iteratively in a similar way to Eqn. 5. We approximate the rotation via $R_j^t \approx (I+\hat{R})R_j^{t-1}$, where $r = (r_1, r_2, r_3)$ is a vector linearizing a small rotation $\hat{R}$; see Eqn. 5, leading to a linear solution for R_j^t. On average, 3.5 iterations are required to compute the optimized R_j^t, which is fast enough for online processing. T can be directly computed once R has been found.

After finding R, T for each frame, the SCAPE reconstruction is found by Eqn. (2a), using the precomputed skinning weights W and intrinsic body shape in T-pose defined by shape parameter θ.

The whole framework for online pose parameter calculation is listed in Algorithm 1; further details are now discussed. The resolution of the Kinect depth images is 320×240. To reduce the time for kd-

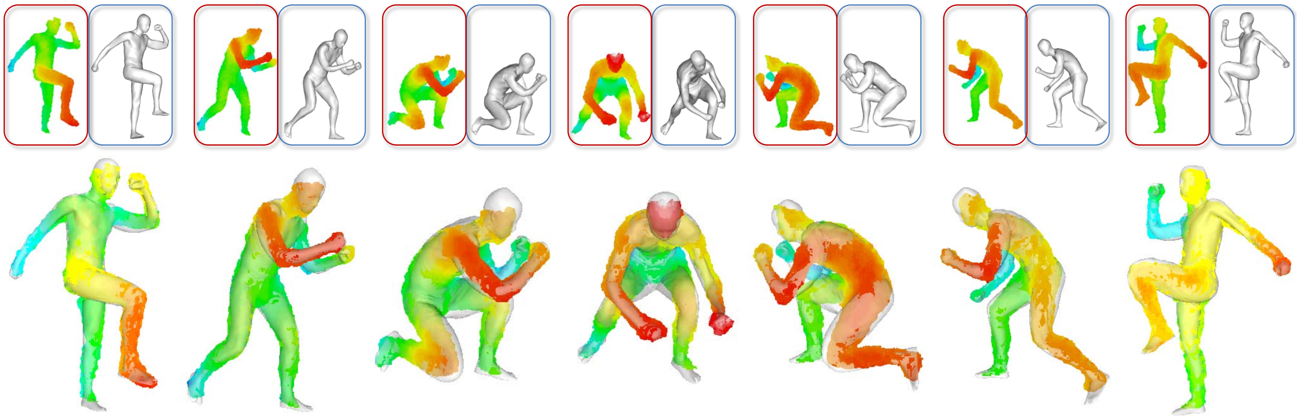


Fig. 7. Example reconstruction results. Top: dynamic depth images and corresponding complete meshes. Bottom: reconstructed meshes overlaying the depth data. These are pseudocolor depth images: red is nearest, and blue furthest from the reader.

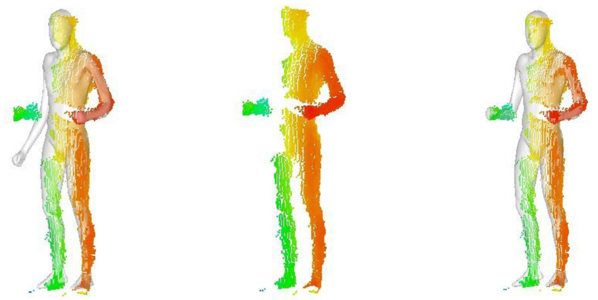


Fig. 8. Comparison. Left: result using the method of [3]. Right: our result. Our reconstructed meshes are better aligned with the depth data (center) in the presence of self-occlusion.

Algorithm 1 Calculation of pose parameters for each frame

Input: Depth image of frame I^t
Output: Pose parameters $\beta^t = (R^t, T^t)$

```
Initialize pose parameters β^t ← β^{t−1}
Build kd-tree for point cloud P^t from I^t
i ← 0
repeat
    Render the depth image of the model M(θ, β^t) specialised to this person and pose, to get the visible vertex set V^t
    Build kd-tree for V^t
    Classify P^t into P_1^t and P_2^t, V^t into V_1^t and V_2^t
    Build correspondences from V^t to P^t
    Set up linear equation for ΔR^t and ΔT^t
    Solve the equation
    Update R^t, T^t and M(θ, β^t)
    if ||Δr^t||_max < ε_1 and ||ΔT^t||_max < ε_2 then
        break
    else
        i ← i + 1
    end if
until i > n_max
```

tree construction and k-nearest-neighbour search, we subsample to half this resolution. There are about 5000 points in the final set P of valid human surface points. To construct the kd-tree, we use the *flann* library. The human template mesh contains 6252 vertices and 12500 faces; in the view of camera, about one third of the template vertices are visible. To determine the visible vertices, the VBO technique is used to determine the depth image of the template. We then compare the depth of each vertex to the corresponding pixel of the rendered depth image, and keep vertices whose depth differences are less than 0.002 m. In the linear equation, for each of the 17 bones, 3 unknowns determine its rotation increment and 3 give its translation increment. There are 4 constraint terms. Denoting the visible vertices of the template by V, we divide them into two sub-classes: for V_1, the depth of V_1 is close to the corresponding pixel of the captured depth image, while V_2 are the remainder. For vertices in V_1, we just constrain their depths (z coordinates). For vertices in V_2, we search for the closest point in P using the kd-tree and choose pairs whose distance is less than a threshold of 0.02m as correspondences. We use the same strategy to classify P into P_1 and P_2 and build up correspondences from P_2 to V to improve robustness. This gives $|V1|+3(|V_2|+|P_2|)$ equations for the goodness of fit term. The joint constraints lead to 3×18 equations since we have 18 joints. The rotational inertia term leads to 9×17 equations since we have 17 bones. The main-axis orientation invariance term leads to 3×7 equations since there are 7 special bones. The total number of linear equations is the sum of the above. We use the conjugate gradient algorithm to solve the linear system, which terminates when *either* the largest rotation angle increment $\|\Delta r^t\|_{\max}$ of any bone is less than a threshold ϵ_1 and the largest translation vector increment $\|\Delta T^t\|_{\max}$ is less than a threshold ϵ_2 *or* the number of iterations exceeds a limit $n_{\max}$. We set $\epsilon_1 = 5°$, $\epsilon_2 = 0.025$ m and $n_{\max} = 7$ in all experiments. Table 1 demonstrates the efficiency of our algorithm, providing average computational

TABLE 1
Times per frame for each step of Realtime SCAPE model processing.

	offline		online							
Step	θ	W	2	5	6	7	8	9	10	11
Time(ms)	3000	1000	4.2	0.5	1.5	0.3	1.8	0.9	0.6	0.4

times for each major component.

The output for a performance is a reconstructed mesh sequence that both fits the single-view depth data, and is consistent with the Realtime SCAPE model. As shown by Figs. 1 and 7, our method can automatically and accurately model any parts of each frame which are occluded. Fig. 7 shows sample input depth image data (top) and overlaid reconstructed poses (bottom).

7 Evaluation and discussion

Our method has been implemented using Visual C++ and OpenGL on a desktop PC with a 3.4GHz CPU.

Table 1 indicates average times for each computational step recorded during all tests carried out for this paper. The parameters θ representing body shape and W representing LBS weights can be pre-computed offline in a few seconds. The online times refer to the steps of Algorithm 1 by line number. The total calculation time for each frame is t_2 plus the number of iterations times the sum of the other steps. On average, 3.5 iterations are needed, so overall about 25 ms per frame are needed to compute the LBS transformation variables R, T.

7.1 Evaluation

Firstly, we compared our method to alternative SCAPE-based methods. Fig. 8 shows one sample frame result produced using our method and the one in [3]. The latter failed to correctly model the person's right forearm because the corresponding depth data is disconnected due to self-occlusion. Both methods use preprocessing to initially determine shape parameters from a T-pose, then match the intrinsic body shape in the rest T-pose to the sampled frame data. The main difference lies in the approach to pose reconstruction: we use an LBS-based pose deformation model, while [3] utilizes linear regression deformation and the traditional SCAPE model [2]. As this comparison shows, our surface reconstruction process is more robust than the one in [3], especially in the presence of self-occlusion. This is mainly because our method reconstructs the pose using an LBS-based top-down tree representation, and does not treat the isolated left arm depth data as an outlier. A further, very significant, advantage of our system over the one in [3] is that our method takes just about 25 ms to reconstruct each pose, while the latter takes about an hour.

Secondly, a comparison was made with a skeleton-based character animation approach to realtime motion reconstruction from a single-view depth camera. This used skeleton extraction plus shape rigging [33]. Again, the intrinsic body shape built offline was used as the mesh for the given subject; the method in [33] was employed to automatically embed the skeleton in the intrinsic body shape. The online process used the Kinect SDK [1] to produce skeletal motion data as a basis for shape rigging to drive motion reconstruction in realtime. Although skeleton-based character animation can also produce a deformable mesh sequence, it has limitations. Firstly, motion accuracy is mainly determined by the skeleton extraction algorithm, which uses a model learnt from a pre-defined database. In particular, the skeleton for each frame is determined independently, and temporal coherence is not enforced. Secondly, alignment between the skeleton and the input depth data is not guaranteed; often the skeleton extraction algorithm does not output a skeleton accurately lying within the data. Thirdly, even if this were accurate, accuracy of the output mesh with respect to the depth data would still be affected by the rigging scheme. Finally, jitter and candy-wrapper problems would occur without taking any special precautions. A visual comparison between the results of our method and such a skeleton-based character animation approach is shown in Fig. 9. Accurate alignment between the skeleton and the data has been performed to obtain reasonable results. As expected, surface matching, jitter and candy-wrapper issues all arise in the skeleton-based method. Overall, the aims and output of skeleton-based character animation and our reconstruction are very different: our goal is an accurate surface model, while the former merely concentrates on capturing a sequence of skeletons (typically to drive animation of a different character). They should be seen as complementary rather than competing techniques.

Thirdly, we compared our method with the latest database approach [27] for the challenging $360°$ human rotation performance using the data provided by [27]. Existing approaches [18], [25], [26], [27] provide results with limited success, or even fail completely, as the depth data are very similar when the actor is facing the camera or has his back towards it. When using low-quality depth data, this results in unreliable pose recognition, even when based on database retrieval. Figure 10 shows that our Realtime SCAPE method can successfully handle such data. This is due to careful technical choices in our approach: (i) using SCAPE [2] as a basis gives us the advantage of SCAPE's capability for robust single-view mesh completion, which guarantees that an acceptable *entire* surface of the human body is reconstructed even

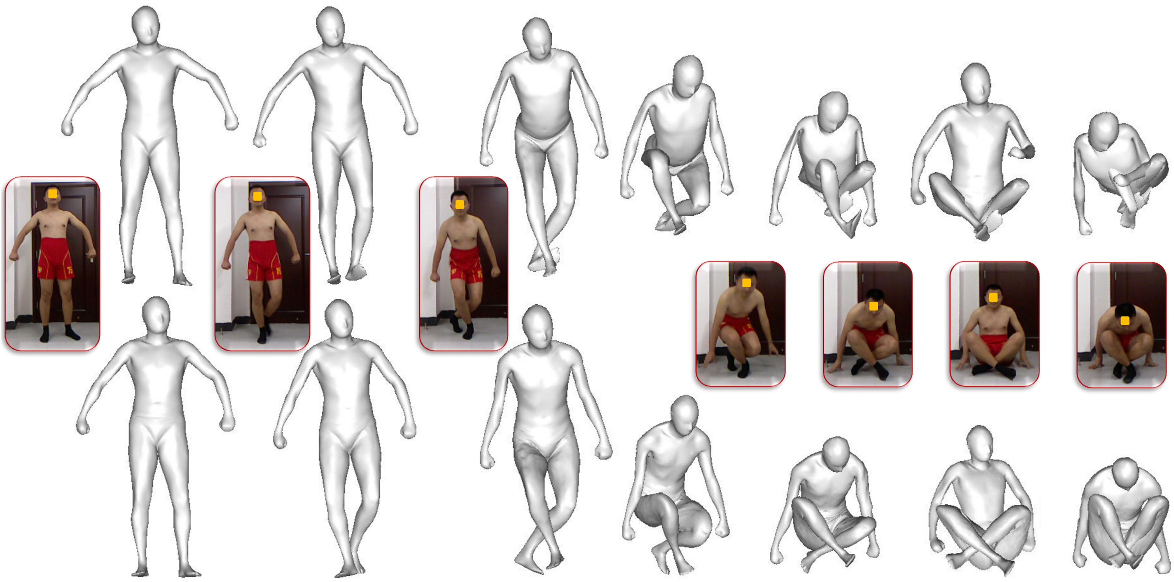


Fig. 9. Comparison, showing poses modelled. Top: skeleton-based character animation results using Kinect-provided skeletons. Bottom: Realtime SCAPE results.

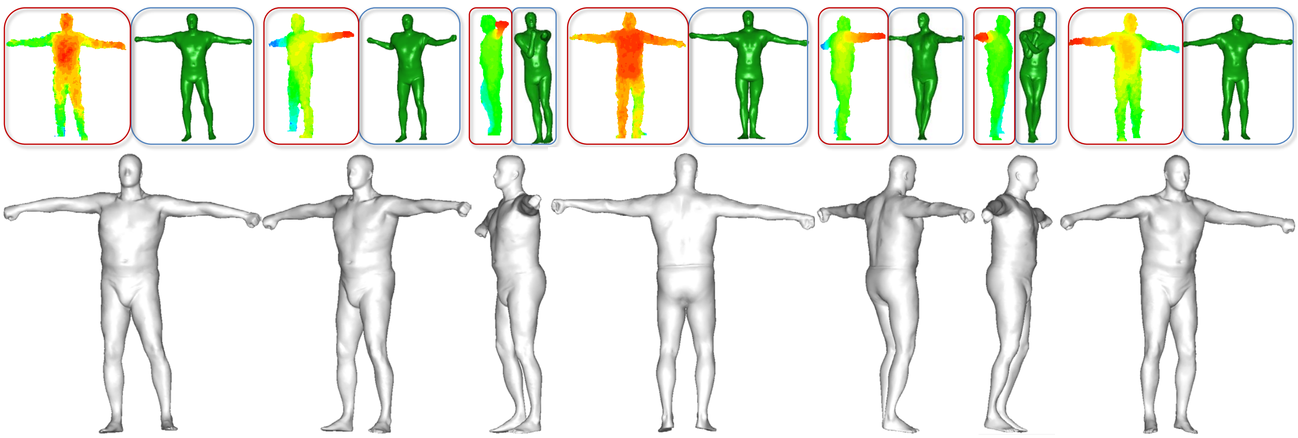


Fig. 10. Reconstructed 360° human rotation performance; data from [27]. Above: depth data (left) and results using the method in [27] (right) for each frame. Below: reconstruction results using our method.

from partial depth data, and (ii) in the rigid bone transformation computation (Eq. 7), any occluded part retains its previous transformation state, due to the use of temporally consistent transformations which are incrementally updated.

Fourthly, we evaluated the robustness of our method using ground truth data: see Fig. 11. Ground-truth for an animating subject was obtained from a deformation transfer approach [33], producing a sequence of dynamic closed-manifold body meshes. The motion capture process was simulated by creating an artificial depth map (with 320×240 resolution) from a single viewpoint. Our reconstruction approach aligned the intrinsic body shape to the depth images. These experiments demonstrated that the geometry and motion of the animating subject could be correctly reconstructed, without use of markers or user assistance. Quantitatively, the maximum L_2 distance error between the reconstructed meshes and the ground truth for all frames was about 0.03 units, while the average distance error for all frames was about 0.001 units, where the unit is the diagonal of the bounding box diagonal for the subject.

Fifthly, we compared our method to one based on cylindrical models with ICP tracking [20]. Figure 12 shows that our method works better; in this case the input data came from the Stanford EVAL dataset [29]. This is because of two reasons. Firstly, our SCAPE model more accurately models the human body than a set of cylindrical models. Secondly, constraints are used in our optimization framework to avoid artifacts. Figure 13 shows further results using the dataset from [20]; again our approach produces better results.

Finally, we measured reconstruction accuracy on the

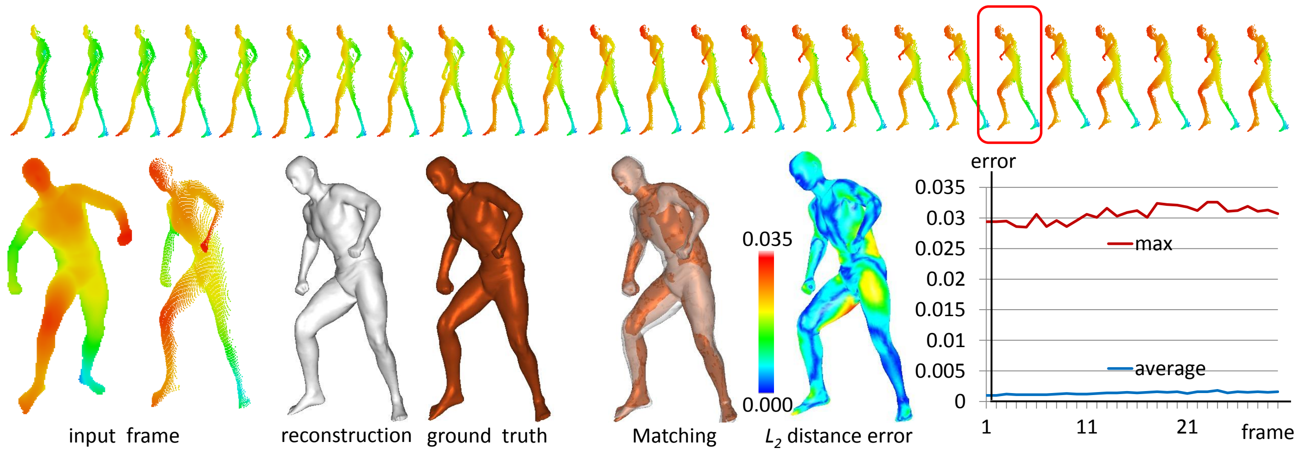


Fig. 11. Ground truth comparison for a synthetic full-body example. Above: input frame sequence. Below: depth images from two selected viewpoints, the reconstructed mesh and corresponding ground truth, match between the reconstructed mesh and ground truth, and color-coded L_2 distance error between the reconstruction and ground truth. The graph shows the maximum and average distance errors for each frame as a fraction of the diagonal length of the bounding box.

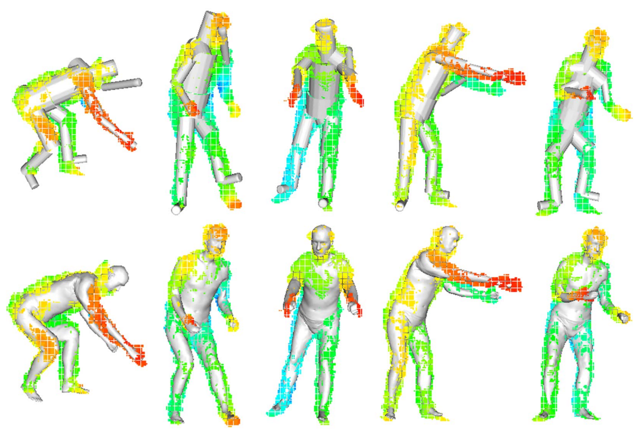


Fig. 12. Reconstructed poses from a sequence; data from [29]. Above: depth data and results using cylindrical models with ICP tracking for each frame. Below: depth data and results using our method, which shows better agreement.

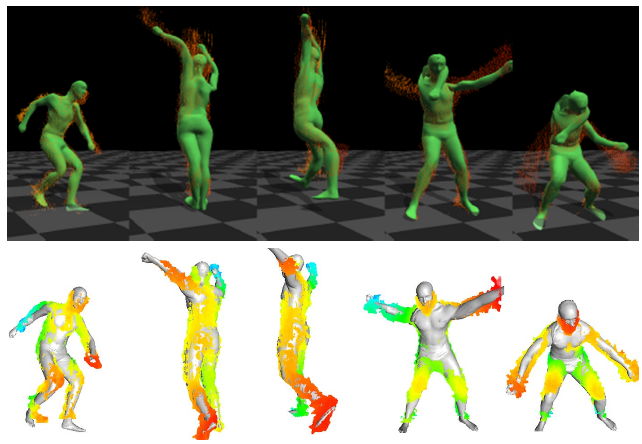


Fig. 13. Reconstruction of a sequence from [20]. Above: depth data and results for selected frames using the method in [20]. Below: depth data and results using our method.

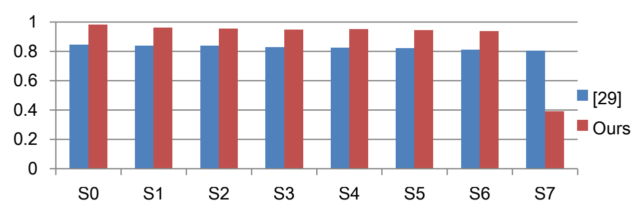


Fig. 14. Tracking accuracy of the approach in [29] and our approach, for the Stanford EVAL dataset.

Stanford EVAL dataset [29] for a set of depth sequences, including handstands, kicks, and sitting down on the floor. We evaluated tracking accuracy using joint accuracy, as described by [29]: we estimated 3D joint positions using our system, and compared these to the true joint positions provided in the dataset using motion capture markers. We counted a joint as detected correctly if the system estimated the 3D joint location to lie within 10 cm of the true joint location. Quantitative results are given in Fig. 14, showing accuracy histograms for all motion sequences (S0 to S7) for Human 0 in the dataset. For S0 to S6, about 82% accuracy was achieved by [29], while we achieved about 94% accuracy. However, for the more tricky example S7 involving a handstand, our approach failed to reconstruct accurate results, for reasons we shortly explain. In this example, our accuracy rate fell to 39%, worse than the 80% achieved by [29].

7.2 Discussion

The major advantage of our method over existing single view human shape completion methods such as [2], [3] is speed, while still producing high quality geometry. This is achieved by careful factoring of the computation. In preprocessing, intrinsic body shape parameters are precalculated, as are weights for the LBS representation. During online motion reconstruction, only transformation parameters remain to be determined. These can be found quickly via a linearized variational solution, as changes between neighboring frames are small.

However, our method has certain limitations. The prior template built by KinectFusion [8] requires sufficiently dense data to produce the initial static reference pose. An unsuitable template may result due to misalignments if the subject does not hold still during scanning, which takes about 30 s. This is a little long for comfort, but not unreasonable.

Clothing increases the difficulty of human body reconstruction. Fig. 15 shows reconstruction results for a female body with fairly tight fitting clothes; clearly a skirt or loose fitting clothing would be trickier to handle. With tight clothing Realtime SCAPE can reconstruct accurate poses and high-quality shapes. As the performers in the Stanford EVAL dataset [29] are dressed in such clothing, we can reconstruct good models for this data.

Fast and sudden motions, such as kicking (see Fig. 16), are potentially trickier to handle. Some such motions are present in the Stanford EVAL dataset [29]; for example, frames 274 and 275 in sequence S4 for Human 0 have large differences. In our approach, this mainly affects affects speed, as more iterations are needed to compute the transformation parameters (Eq. 7). Even so, surface reconstruction takes only about 35 ms per frame in this case.

The handstand examples, S6 and S7 in the Stanford EVAL dataset, present more serious challenges for our approach. S6 was correctly reconstructed, but our approach broke down for S7, as shown in Fig. 17. This is because if parts of the body are out of view for a period of time, and also undergo deformations, our assumption of smooth and continuous movement breaks down. This is an inherent limitation of single-view systems, in which some parts are invisible at any given moment.

We currently do not take any steps to prevent global self-intersection of the deforming meshes. Nevertheless, as the visual results show, our method can robustly reconstruct complex poses, mainly due to the suitability of the modified SCAPE model for guiding motion reconstruction. Avoiding self-intersections entirely would require an additional collision detection and avoidance step in motion estimation, which would add a significant computational burden in an online process.

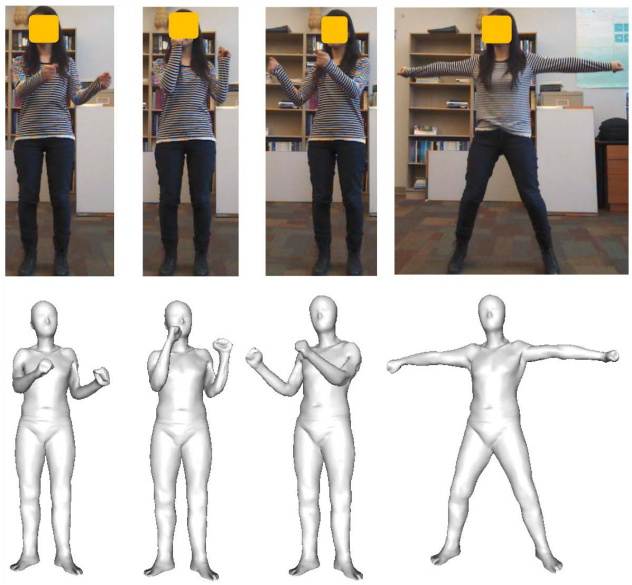


Fig. 15. Reconstruction results for a clothed woman.

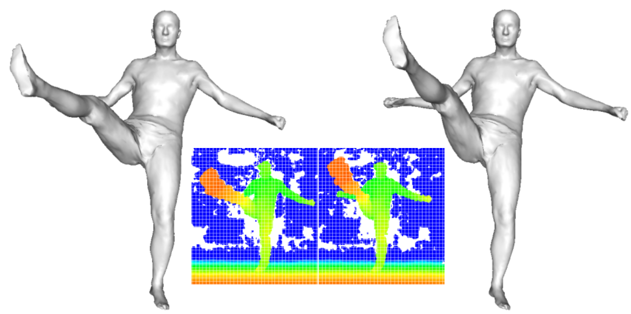


Fig. 16. Fast sudden kicking motion, in adjacent frames S4-274 and S4-275 of Human 0 in the Stanford EVAL dataset.

Ultimately, the problem of full-body animation is very challenging. We believe, however, that our method has advanced the possibilities of what can be achieved with low quality depth data, providing a capability for dynamic human modeling in real time.

8 Conclusions

We have presented Realtime SCAPE, a markerless, automatic human full-body geometry and motion reconstruction method, using a single depth camera. The key to its success is that Realtime SCAPE uses two levels of decoupling. Firstly SCAPE decomposition allows intrinsic body shape to be determined offline, separately from pose estimation. Secondly pose deformation based on linear blending skinning decomposes into problems of weight determination, again,

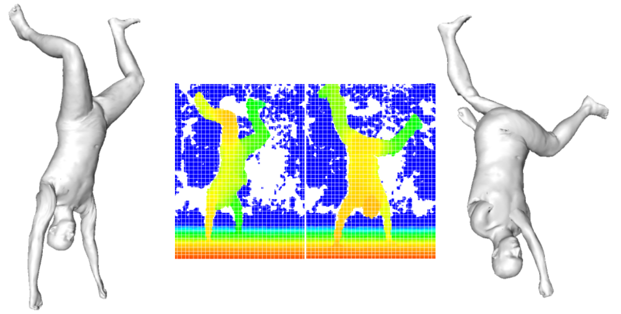


Fig. 17. Handstand examples, frames S6-237 and S7-216 of Human 0 in the Stanford EVAL dataset.

carried out offline, and transformation determination, computed online. The latter is formulated as a linear variation problem, providing realtime performance. We have demonstrated the speed and geometric plausibility of our method on a wide range of subjects with a variety of motions, achieving realistic reconstruction of dynamic motion with complete geometry in all except the most challenging cases.

Future work is needed to address reconstruction of animated human bodies with loose clothing. We also wish to evaluate our method in a multi-view setting where more of the body can be seen at the same or alternating time instances. Further plans include integrating a dynamic model to ensure stable motion estimates for occluded limbs and topology changes, more realistic deformation modeling by use of a more accurate skinning method, and a means to automatically reset the system after failures if it gets stuck in a local minimum.

Acknowledgment.

We are grateful to Microsoft for the Kinect SDK, and Will Chang and Thomas Helten for sharing source code, executable programs, and test data.

This work was supported by the Natural Science Foundation of China (No. 61103084, 61272334) and RIVIC, the One Wales Institute for Visual Computing.

References

[1] Kinect, "Microsoft Xbox," 2013. http://www.xbox.com/Kinect.

[2] D. Anguelov, P. Srinivasan, D. Koller, S. Thrun, J. Rodgers, and J. Davis, "SCAPE: shape completion and animation of people," *ACM Transations on Graphics (special issue of SIGGRAPH)*, vol. 24, no. 3, pp. 408–416, 2005.

[3] A. Weiss, D. Hirshberg, and M. J. Black, "Home 3D body scans from noisy image and range data," in *International Conference on Computer Vision*, pp. 1951–1958, 2011.

[4] J. P. Lewis, M. Cordner, and N. Fong, "Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation," in *SIGGRAPH*, pp. 165–172, 2000.

[5] R. Poppe, "Vision-based human motion analysis: An overview," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 4–18, 2007.

[6] H. Li, L. Luo, D. Vlasic, P. Peers, J. Popović, M. Pauly, and S. Rusinkiewicz, "Temporally coherent completion of dynamic shapes," *ACM Transactions on Graphics*, vol. 31, no. 1, pp. 2:1–2:11, 2012.

[7] A. Tevs, A. Berner, M. Wand, I. Ihrke, M. Bokeloh, J. Kerber, and H.-P. Seidel, "Animation cartography - intrinsic reconstruction of shape and motion," *ACM Transactions on Graphics*, vol. 31, no. 2, 2012.

[8] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 559–568, 2011.

[9] B. Allen, B. Curless, and Z. Popović, "The space of human body shapes: reconstruction and parameterization from range scans," *ACM Transations on Graphics (special issue of SIGGRAPH)*, vol. 22, no. 3, pp. 587–594, 2003.

[10] D. Vlasic, I. Baran, W. Matusik, and J. Popović, "Articulated mesh animation from multi-view silhouettes," *ACM Transations on Graphics (special issue of SIGGRAPH)*, vol. 27, no. 3, pp. 97:1–97:9, 2008.

[11] E. de Aguiar, C. Stoll, C. Theobalt, N. Ahmed, H.-P. Seidel, and S. Thrun, "Performance capture from sparse multi-view video," *ACM Transations on Graphics (special issue of SIGGRAPH)*, vol. 27, no. 3, pp. 98:1–98:10, 2008.

[12] H. Li, B. Adams, L. J. Guibas, and M. Pauly, "Robust single-view geometry and motion reconstruction," *ACM Transations on Graphics (special issue of SIGGRAPH Asia)*, vol. 28, no. 5, pp. 175:1–175:10, 2009.

[13] D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz, and W. Matusik, "Dynamic shape capture using multi-view photometric stereo," *ACM Transations on Graphics (special issue of SIGGRAPH Asia)*, vol. 28, no. 5, pp. 174:1–174:11, 2009.

[14] A. Jain, T. Thormählen, H.-P. Seidel, and C. Theobalt, "MovieReshape: tracking and reshaping of humans in videos," *ACM Transactions on Graphics (SIGGRAPH ASIA)*, vol. 29, no. 6, pp. 148:1–148:10, 2010.

[15] P. Guan, A. Weiss, A. O. Balan, and M. J. Black, "Estimating human shape and pose from a single image," in *IEEE International Conference on Computer Vision*, pp. 1381–1388, 2009.

[16] A. O. Balan, L. Sigal, M. J. Black, J. E. Davis, and H. W. Haussecker, "Detailed human shape and pose from images," in *CVPR*, 2007.

[17] Y. Chen, Z. Liu, and Z. Zhang, "Tensor-based human body modeling," in *CVPR*, pp. 105–112, 2013.

[18] M. Ye, H. Wang, N. Deng, X. Yang, and R. Yang, "Real-time human pose and shape estimation for virtual try-on using a single commodity depth camera," *IEEE Transactions on Visualization and Computer Graphics (IEEE Virtual Reality)*, vol. 20, no. 4, pp. 550–559, 2014.

[19] X. Wei and J. Chai, "VideoMocap modeling physically realistic human motion from monocular video sequences," *ACM Transaction of Graphics*, vol. 29, no. 4, pp. 42:1–42:10, 2010.

[20] X. Wei, P. Zhang, and J. Chai, "Accurate real-time full-body motion capture using a single depth camera," *ACM Transaction of Graphics*, vol. 31, no. 6, pp. 188:1–188:12, 2012.

[21] J. Shotton, A. W. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *CVPR*, pp. 1297–1304, 2011.

[22] R. B. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. W. Fitzgibbon, “Efficient regression of general-activity human poses from depth images,” in *ICCV*, pp. 415–422, 2011.

[23] J. Taylor, J. Shotton, T. Sharp, and A. W. Fitzgibbon, “The Vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation,” in *CVPR*, pp. 103–110, 2012.

[24] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, “Real time motion capture using a single time-of-flight camera,” in *CVPR*, pp. 755–762, 2010.

[25] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt, “A data-driven approach for real-time full body pose reconstruction from a depth camera,” in *International Conference on Computer Vision*, pp. 1092–1099, 2011.

[26] M. Ye, X. Wang, R. Yang, L. Ren, and M. Pollefeys, “Accurate 3D pose estimation from a single depth image,” in *International Conference on Computer Vision*, pp. 731–738, 2011.

[27] T. Helten, A. Baak, G. Bharaj, M. Müller, H.-P. Seidel, and C. Theobalt, “Personalization and evaluation of a real-time depth-based full body tracker,” in *Proceedings of the 3rd joint 3DIM/3DPVT Conference (3DV)*, 2013.

[28] T. Helten, M. Müller, H.-P. Seidel, and C. Theobalt, “Real-time body tracking with one depth camera and inertial sensors,” in *ICCV*, pp. 1105–1112, 2013.

[29] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, “Real-time human pose tracking from range data,” in *ECCV (6)*, pp. 738–751, 2012.

[30] B. Le and Z. Deng, “Smooth skinning decomposition with rigid bones,” *ACM Transaction of Graphics (SIGGRAPH ASIA)*, vol. 31, no. 6, pp. 199:1–199:10, 2012.

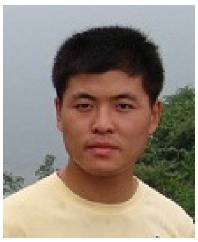
[31] L. Kavan, P.-P. Sloan, and C. O’Sullivan, “Fast and efficient skinning of animated meshes,” *Computer Graphics Forum*, vol. 29, no. 2, pp. 327–336, 2010.

[32] SAE, “Caesar 3D anthropometric database,” 2013. http://store.sae.org/caesar/index.htm.

[33] I. Baran and J. Popović, “Automatic rigging and animation of 3D characters,” *ACM Transaction of Graphics (SIGGRAPH)*, vol. 26, no. 3, 2007.

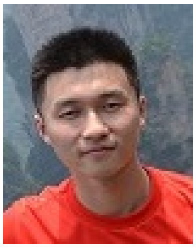
[34] W. Chang and M. Zwicker, “Range scan registration using reduced deformable models,” *Computer Graphics Forum*, vol. 28, no. 2, pp. 447–456, 2009.

[35] L. Kavan, R. McDonnell, S. Dobbyn, J. Zara, and C. O’Sullivan, “Skinning with dual quaternions,” in *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pp. 53–60, 2007.



Yin Chen received BSc and MSc degrees in 2008 and 2010 respectively from the Computer School at the National University of Defense Technology, China, where he is currently a PhD student. His research interests include computer graphics and digital geometry processing.

Zhi-Quan Cheng received BSc, MSc, and PhD degrees in 2000, 2002, and 2008 respectively from the Computer School at the National University of Defense Technology, China. He is currently a researcher in the Avatar Science Company, Hunan. His research interests include computer graphics and virtual reality.



Chao Lai received BSc and MSc degrees in 2008 and 2011 respectively from the Computer School at the National University of Defense Technology, China, where he is currently a Ph.D student. His research interests include visual computing and computer graphics.

Ralph R. Martin Ralph R. Martin received a PhD degree from Cambridge University in 1983, with a dissertation on “Principal Patches for Computational Geometry”, and since then, has worked at Cardiff University where he is now a full professor. He has co-authored more than 250 papers and 12 books covering such topics as solid modelling, surface modelling, intelligent sketch input, vision based geometric inspection, geometric reasoning and reverse engineering. He is a Fellow of the Learned Society of Wales, the Institute of Mathematics and Its Applications, and the British Computer Society. He is on the editorial boards of Computer Aided Design, Computer Aided Geometric Design, Graphical Models, and Computers & Graphics; he has also been active in the organisation of many conferences.

Gang Dang received BSc, MSc, and PhD degrees in 1994, 1997, and 2011 respectively from the Computer School at the National University of Defense Technology, China, where he is currently an associate professor. His research interests include computer graphics and digital geometry processing.