

# *Symmetry-aware kinematic skeleton generation of a 3D human body model*

**Shan Luo & Jieqing Feng**

**Multimedia Tools and Applications**  
An International Journal

ISSN 1380-7501

Multimed Tools Appl  
DOI 10.1007/s11042-020-08933-3



**Your article is protected by copyright and all rights are held exclusively by Springer Science+Business Media, LLC, part of Springer Nature. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at [link.springer.com](http://link.springer.com)".**



# Symmetry-aware kinematic skeleton generation of a 3D human body model

Shan Luo<sup>1</sup> · Jieqing Feng<sup>1</sup>

Received: 24 May 2019 / Revised: 25 February 2020 / Accepted: 13 April 2020 /  
Published online: 21 April 2020  
© Springer Science+Business Media, LLC, part of Springer Nature 2020

## Abstract

In this paper, an automatic method is proposed to generate a symmetry-aware kinematic skeleton for a human body model with an arbitrary pose and orientation. First, a template kinematic skeleton with semantics is embedded into the input human body model. Then, the joints of the embedded kinematic skeleton are refined according to the geometry of the human body model and some prior knowledge. Finally, a specific local coordinate system is defined on the kinematic skeleton and is used to distinguish the symmetry of the kinematic skeleton. In this way, the symmetric joints of the kinematic skeleton, e.g., the left knee joint and the right knee joint, can be distinguished. Quantitative and qualitative analysis and comparison show that the proposed method can generate a symmetry-aware kinematic skeleton with accurate joints and has no restrictions on the pose and orientation of the input human body model. Moreover, this paper presents validation of the proposed method in many applications, such as shape alignment, shape deformation, shape co-segmentation and shape correspondence.

**Keywords** Kinematic skeleton · Symmetric ambiguity · Skeleton embedding

## 1 Introduction

Intrinsic symmetry is a common property for bipeds and quadrupeds. Many algorithms in shape correspondence and shape segmentation suffer from symmetric ambiguity because of the intrinsic symmetry of models. Although many methods [24, 32, 38] have been proposed to solve this problem, symmetric ambiguity between non-isometric models is still a challenge since most properties are just isometric-invariant.

For human body models, symmetry distinction is particularly important, whether for the semantic analysis of human body models or the generation of human deformable models [3]. However, symmetric ambiguity for human body models is particularly difficult because of the diversity of human body models due to the changes in poses and the shape variations across individuals. For a human body model in a standard pose (A-pose), the symmetric

---

✉ Jieqing Feng  
jqfeng@cad.zju.edu.cn

<sup>1</sup> State Key Lab of CAD&CG, Zhejiang University, Hangzhou, 310058, China

parts can be distinguished assisted by the directions of the feet, which always face forward in the standard pose. But few straightforward methods have been applied to the human body models in non-standard poses to solve this problem.

If the pose of a human body model is transformed to the standard pose, the symmetric ambiguity can be solved based on the feet directions. As abstract representations of 3D models, skeletons are much easier to be deformed than human body models. Therefore, transforming the skeleton of the human body model to the standard pose may be a good choice to solve the symmetric ambiguity of the human body model. In general, skeletons can be classified into two categories: curve skeletons and kinematic skeletons. Compared with curve skeletons, kinematic skeletons contain only a small number of joints and are more frequently used to deform 3D models. In recent years, some methods [7, 13, 25] have been proposed to generate the kinematic skeletons for 3D models. However, few methods pay attention to solve the symmetric ambiguity problem via the generated kinematic skeleton.

In this paper, we focus on automatic kinematic skeleton generation for a human body model with an arbitrary pose and orientation. This proposed method can distinguish the symmetry of the human body model and generate a semantically symmetry-aware kinematic skeleton. The semantics of the human body model can be obtained indirectly through the semantics of the generated kinematic skeleton, without any learning process. Similar to some kinematic skeleton generation methods, a template skeleton is embedded into the human body model by converting an extracted curve skeleton into a kinematic skeleton. However, the joints of the kinematic skeleton are refined by the geometry of the human body model and some prior knowledge, and the symmetric parts are distinguished by a local coordinate system established based on the assumption that human feet face forward in the standard pose.

In summary, the main contributions in this paper are as follows:

- The proposed method can distinguish the symmetry of the kinematic skeleton, resulting in a semantically symmetry-aware kinematic skeleton.
- The proposed method has no restrictions on the pose and orientation of the human body model.
- The joint positions of the kinematic skeleton are accurate by taking the geometry of the human body model and some prior knowledge into account.

The remainder of our paper is organized as follows. Related works are introduced in Section 2. The proposed method is described in detail in Section 3. Quantitative and qualitative analysis and comparison are given in Section 4. Applications of the generated kinematic skeleton are presented in Section 5. Conclusions are drawn and future work is indicated in Section 6.

## 2 Related work

**Symmetry distinction** Symmetric ambiguity is a common problem when matching or co-segmenting 3D models with intrinsic symmetries. Earlier methods located a set of landmarks on models manually to avoid symmetric ambiguity. To reduce manual operations, some automatic methods were proposed based on the properties of isometric models, such as the geodesic distance preservation [29] and Laplace eigenvectors preservation [24]. However, these methods cannot work when the isometric assumption is not valid. Based on the observation that some orientations are sensitive to the local symmetry, some oriented descriptors are proposed to distinguish the symmetric parts. Yoshiyasu et al. [36] presented

a local depth map by taking the gradient of the average diffusion distance field as the view-up direction. Wang and Fang [32] developed a multi-scale local diffusion map based on heat diffusion distribution. Other descriptors [27, 38] constructed a sign indicator by checking whether the directions satisfy the right-hand rule. These oriented descriptors are discriminative on the symmetric parts of a model. However, the orientations these descriptors take are also isometric-invariant and not stable for non-isometric models. For the symmetry distinction between non-isometric models, Liu et al. [19] extracted the symmetry axis curves first and Yoshiyasu et al. [37] detected a symmetry plane on a low dimensional embedding. However, detecting the symmetry axis or plane is also a complex and time-consuming problem. Marin et al. [22] proposed a symmetry distinction method based on a kinematic skeleton by propagating the foot direction along the leg of the kinematic skeleton under torque-penalizing constraints. It takes about fourteen steps to get the front direction of the kinematic skeleton. In this paper, a more straightforward method is proposed to distinguish the symmetry of the kinematic skeleton.

**Kinematic skeleton generation** There are two types of kinematic skeleton generation approaches. One, which is called skeleton extraction, extracts the joints from the 3D model directly and connects the joints to form a kinematic skeleton. The other, called skeleton embedding, embeds a template skeleton into the 3D model.

For skeleton extraction methods [17, 18, 21], the topological structure and the number of joints are sensitive to the shape of the model, which may be inconsistent for different models. To improve the accuracy of joint locations, some example-based methods [1, 11, 15] were proposed to generate a kinematic skeleton from a set of models in a variety of poses. However, joints locations are highly dependent on the poses of the input models, which may still be inaccurate if the input models are not carefully selected. Moreover, these methods require a set of examples as input, which is hard for users to obtain. In 2019, Xu et al. [34] proposed a method to generate a kinematic skeleton for a 3D articulated model via deep learning. But they aimed at generating a kinematic skeleton tailored for 3D articulated models with different structures or geometries, rather than a kinematic skeleton with pre-defined structure.

Skeleton embedding methods can generate compatible kinematic skeletons and are more suitable for our purpose. To generate the kinematic skeleton, some methods embed a template skeleton into the model directly, while others embed a template skeleton into the model indirectly, i.e., fit a template skeleton to a curve skeleton of the model. For details about curve skeleton extraction, readers can refer to surveys by Cornea et al. [10] and Saha et al. [28].

Baran and Popović [7] proposed an automatic method in which a template skeleton is embedded into the input model directly. However, this method requires that the input model to be well-posed and -oriented. Wang et al. [33] presented a method to generate a consistent skeleton by transferring the structure of the skeleton from the source model to the target model through corresponding points. The quality of the skeleton is decided by the accuracy of corresponding points. Recently, some registration-based methods [6, 12, 22] were proposed to transfer a kinematic skeleton from a template model to the target model. Taking an extra template model and its corresponding kinematic skeleton as input, these methods embed the kinematic skeleton into the target model through registering the template model to the target model. Since a series of optimizations are needed for 3D model registration, the registration-based methods are time-consuming in general.

Indirect methods transform the skeleton generation problem into establishing correspondences between the template skeleton and the extracted curve skeleton. Some methods [5,

[26] solved this problem by a pair of corresponding points specified manually. Pantuwong and Sugimoto [25] established correspondences via the positions of the symmetry junctions while Hajari et al. [13] adopted the degrees of the joints with multiple branches to match them. For these indirect methods, the joints of the generated kinematic skeleton are all located on the curve skeleton, which is not accurate enough since some sharp features of the curve skeleton are lost. Since human bodies are symmetrical, the kinematic skeleton may be generated with symmetric flips in semantics when the input human body model has an arbitrary pose and orientation. However, apart from Pantuwong and Sugimoto [25] determining the orientation of the human body model by the directions of the feet directly, which can be applied to human body models in certain poses, other methods do not pay attention to this problem.

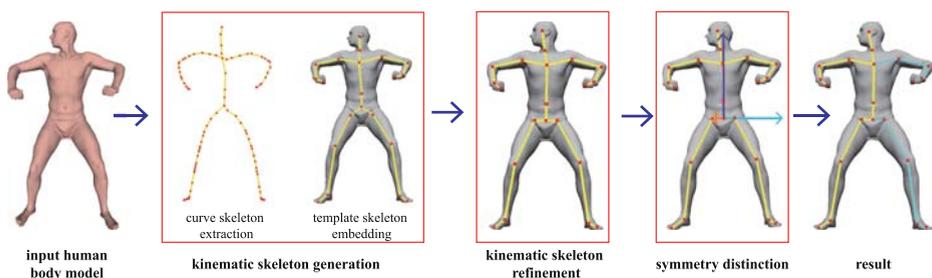
In this paper, an automatic symmetry-aware kinematic skeleton is generated for a human body model with an arbitrary pose and orientation. This method belongs to the indirect skeleton embedding approach. Therefore, the topological structure and the semantics of the kinematic skeleton generated by the proposed method are identical to those of the template skeleton.

### 3 Symmetry-aware kinematic skeleton generation

The input of our method is a triangular mesh of a human body model and a 21-joint template skeleton. In contrast to other methods, each joint of the template skeleton is attached with semantics, as shown in Fig. 2a. The output of our method is a kinematic skeleton with a consistent structure and semantics to the template skeleton. Our method is composed of three steps: kinematic skeleton generation, kinematic skeleton refinement and symmetry distinction. An overview of our pipeline is illustrated in Fig. 1. As shown in the “result” in Fig. 1, the proposed method distinguishes the symmetric parts of the kinematic skeleton successfully.

#### 3.1 Kinematic skeleton generation

In this part, a kinematic skeleton is obtained from a human body model. An indirect skeleton embedding method is adopted to achieve this goal. For the indirect skeleton embedding methods, the main steps are: 1) extracting a curve skeleton, and 2) embedding the template skeleton on the extracted curve skeleton.



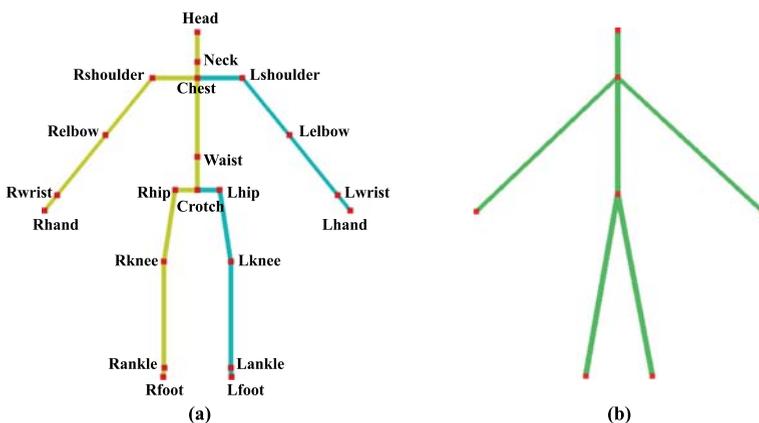
**Fig. 1** Overview of the proposed method. In the final result, the bones on the left part of the kinematic skeleton are shown in blue, while the others are shown in yellow

First, a geometry-based algorithm is adopted to extract the curve skeleton via mesh contraction and simplification [4]. This method is independent of the poses and orientations of human body models and can recover some sharp features in the curve skeleton by the refinement step. After the curve skeleton is extracted, a preprocessing step is adopted to remove additional branches which occur in regions with complex geometry and have no corresponding branches on the template skeleton. The maximum distance between any two nodes on the curve skeleton is normalized to 1 to avoid model size inconsistency. Any branch whose length is smaller than a threshold  $\epsilon$  is considered as an additional branch and will be removed. Since the curve skeleton is extracted by mesh contraction, sometimes a redundant branch occurs at the heel, making the length of the foot branch approximately equal to those of the additional branches on the head. To make the process robust, a multi-resolution approach is adopted to delete the additional branches. First, a small  $\epsilon$  is set as 0.015 to address the “small” additional branches caused by the pure geometric curve extraction algorithm, e.g., the additional branch at the heel. Then, a larger  $\epsilon$  is set as 0.06 to handle the “long” additional branches caused by geometrical details, such as fingers and ears.

Then, the template skeleton is embedded on the extracted curve skeleton. Previous methods established the correspondences via the positions of the symmetry junctions [25] or the degrees of joints with multiple branches [13], which is highly dependent on the structure of the curve skeleton, making the algorithms not robust enough. Inspired by the spectral matching algorithm proposed by Leordeanu and Hebert [16] and applied by Kleiman and Ovsjanikov [14], a matching algorithm is proposed to solve this problem.

To reduce the complexity of the matching step, reduced skeletons that ignore all joints with a degree of two (also called intermediate joints) are used as the input of the matching algorithm. The reduced template skeleton is shown in Fig. 2b. Once the correspondences are established, the intermediate joints can be located based on the length ratios and the local extremal curvatures. The distance between two nodes of the reduced skeleton is defined as the geodesic distance along the original skeleton rather than the Euclidean distance between them.

The matching algorithm measures the similarity between two skeletons by considering both the degree of the node and the distances to other nodes. Let the reduced curve skeleton and the reduced template skeleton be represented as two weighted graphs with  $m$  and  $n$  nodes respectively. Considering the compatibility of corresponding pairs, a matrix  $M$  of



**Fig. 2** a The input template skeleton with semantics for each joint. b The reduced template skeleton

$mn * mn$  is constructed, whose values of diagonal elements represent the affinity of corresponding pairs and values of off-diagonal elements represent the compatibility between two corresponding pairs. Then, the matching problem can be formulated as finding a binary vector  $x$  that represents the one-to-one correspondence to maximize  $x^T Mx$ . To this end, we relax the binary constraint and find a solution discretely based on the eigenvector corresponding to the largest eigenvalue of  $M$ .

In our solution, the elements of  $M$  are defined as follows. For each node  $i$  of the graph, we first compute the node degree  $G(i)$ , distances to other nodes  $D(i, *)$ , and average distance  $D_{avg}(i)$ . For a corresponding pair  $(i, j)$ , we consider the differences in both degrees and average distances, which are defined as:

$$\begin{aligned} E_g(i, j) &= \|G(i) - G(j)\|. \\ E_d(i, j) &= \|D_{avg}(i) - D_{avg}(j)\|. \end{aligned} \tag{1}$$

The affinity of the corresponding pair is computed by the following:  $U(i, j) = \alpha e^{-E_g(i, j)} + e^{-E_d(i, j)}$ , with  $\alpha = 5$  in our experiments.

For the off-diagonal elements in the  $M$ , such as the element related to the corresponding pair  $(i, j)$  and the corresponding pair  $(k, l)$ , we re-define the differences in both degrees and distances:

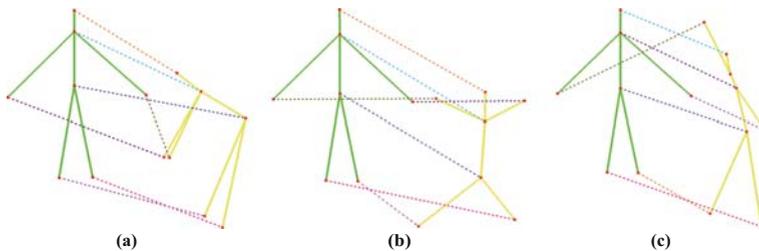
$$\begin{aligned} E_g(i, j, k, l) &= \|E_g(i, j) - E_g(k, l)\|. \\ E_d(i, j, k, l) &= \|D(i, k) - D(j, l)\|. \end{aligned} \tag{2}$$

Then, the compatibility of these two corresponding pairs is defined as:  $U(i, j, k, l) = \alpha e^{-E_g(i, j, k, l)} + e^{-E_d(i, j, k, l)}$ , with  $\alpha = 5$ .

Through eigenvalue decomposition of the matrix  $M$ , we can obtain the eigenvector  $x$  corresponding to the largest eigenvalue. A greedy algorithm is adopted to solve the final corresponding pairs. The greedy algorithm is not terminated until all nodes of the reduced template skeleton find their corresponding nodes on the reduced curve skeleton.

Figure 3a and b show the results of matching between the reduced curve skeleton and the reduced template skeleton. Even if the pose of the reduced curve skeleton differs greatly from that of the reduced template skeleton, the matching algorithm produces correct results.

To verify the robustness of this algorithm, a reduced curve skeleton that has an inconsistent structure with the reduced template skeleton is obtained from a curve skeleton extracted with improper parameters. The matching algorithm still obtains the correct result, as shown in Fig. 3c. To generate a reduced curve skeleton that has a consistent structure with the reduced template skeleton, the non-matched nodes on the reduced curve skeleton are adjusted to its neighboring matched node with the most branches.



**Fig. 3** Matching results. The reduced template skeleton is shown in green, while the reduced curve skeleton is shown in yellow

### 3.2 Kinematic skeleton refinement

Since smoothness is one of the constraints for extracting the curve skeleton, some sharp features of the curve skeleton are lost, making the joint positions insufficiently accurate. In this section, the geometry of the human body model and some prior knowledge are used to refine the joints of the kinematic skeleton to more appropriate positions.

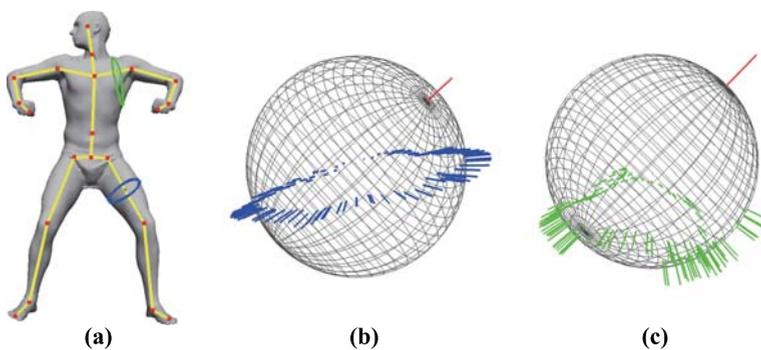
#### 3.2.1 Refinement by the geometry of the human body model

Centeredness is one of the constraints of skeleton generation. The embedded kinematic skeleton can be further adjusted to the center of the human body model based on the geometry of the human body model.

In general, for an elliptical cylindrical shape, the skeleton point is the center of the cross-section curve at the point. Since most parts of the human body are approximately elliptical cylindrical, some skeleton points of the human body model can be obtained by the centers of the cross-section curves. Then, the generated kinematic skeleton can be refined by these skeleton points.

The approximately elliptical cylindrical parts of a human body model can be recognized by the semantics of the kinematic skeleton obtained above. For each part to be processed, we uniformly sample  $n$  points on the corresponding bone and compute the cross-section curve at each sampling point. Then, the reliability of the center points is analyzed. Finally, the joint positions are adjusted according to the linear fitting results of the reliable center points.

The reliability of a center point is judged by the quality of the cross-section curve. Since Prominent Cross-Sections (PCS) [30] is also a way to extract the curve skeleton, the center points of the prominent cross-sections can be regarded as reliable skeleton points. Therefore, we adopt a similar idea and use the sectional Gauss map, which is the Gaussian sphere constructed by the normals associated with the points on a cross-section curve, to determine whether a cross-section curve is effective. Examples of the sectional Gauss map are shown in Fig. 4. PCS is an iterative result of making the points on the Gaussian sphere to be in one plane as far as possible. Therefore, we measure the quality of a cross-section curve by the flatness of the points on the Gaussian sphere. We define the flatness of the points on the Gaussian sphere as:  $f = \lambda_3 / (\lambda_1 + \lambda_2 + \lambda_3)$ , where the  $\lambda$  represent the eigenvalues of the PCA analysis on the points of the Gaussian sphere and  $\lambda_1 \geq \lambda_2 \geq \lambda_3$ . If the flatness  $f$  is



**Fig. 4** **a** Cross-section curves of the human body model. **b** The sectional Gauss map of the blue cross-section curve in **(a)**. **c** The sectional Gauss map of the green cross-section curve in **(a)**

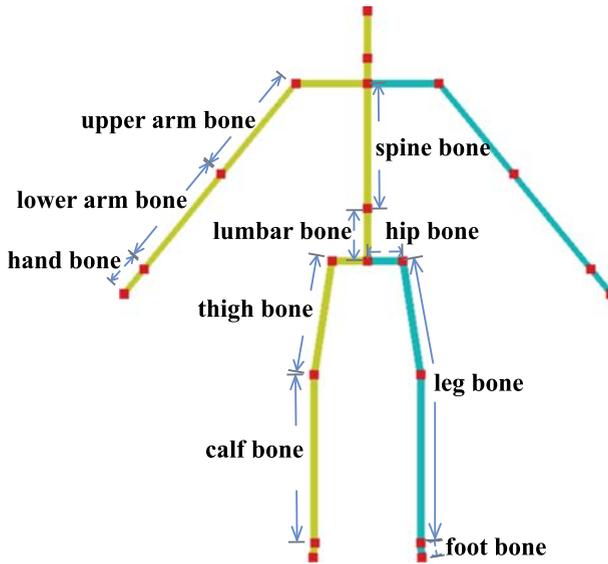


Fig. 5 The definitions of some bones on the kinematic skeleton

smaller than a threshold  $\epsilon_1$ , the cross-section curve is considered to be effective, then the center point is a reliable point. In our implementation,  $\epsilon_1$  is set as 0.1.

### 3.2.2 Refinement by some prior knowledge

In this section, we will use some prior knowledge to refine the joints to more reasonable positions. For better explanation, the definitions of some bones on the kinematic skeleton are shown in Fig. 5.

Since mesh contraction and simplification are adopted to extract the curve skeleton, occasionally the crotch joint is nearly on the model or even outside the model, as shown

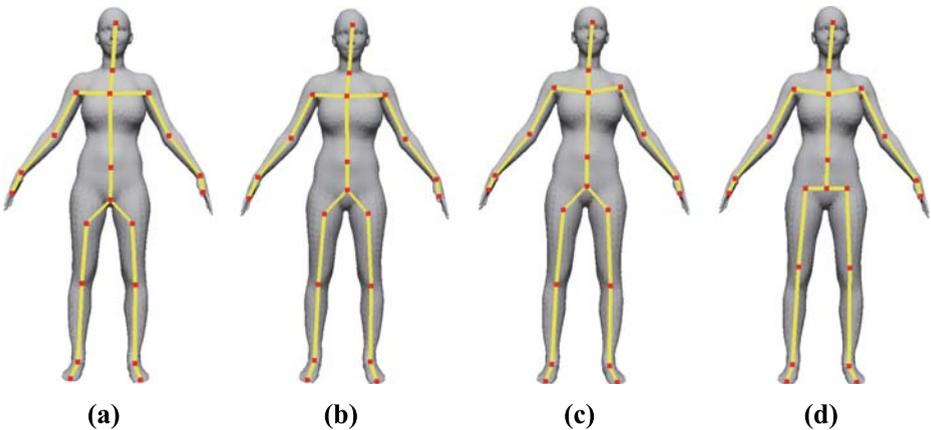


Fig. 6 Joint refinement. **a** The generated kinematic skeleton. **b** Refinement of crotch joint. **c** Refinement of arm joints. **d** Refinement of leg joints

in Fig. 6a. From observation, it is reasonable that the distance between the crotch joint and the human body model would be not less than the thigh thickness. If the distance between the crotch joint and the human body model is less than the thigh thickness, the crotch joint is adjusted towards the waist joint until the distance is not less than the thigh thickness. Figure 6b shows the refined crotch joint. The thickness of a point on the bone is defined as the average distance from the cross-section curve, and the thigh thickness is defined as the average thickness of the sampling points on the thigh bone.

Affected by the torso, the position of the shoulder joint is somewhat lower than its ideal position, as shown in Fig. 6a. Thus, the shoulder joint should be adjusted away from the elbow joint to a more reasonable position. However, the adjustment distance is not fixed and varies with the pose of the human body model. After experiments, we find that the adjustment distance of the shoulder joint is closely related to the pose. Let  $\theta$  be the angle between the upper arm bone and spine bone; the adjustment distance will decrease when  $\theta$  increases. Here, an empirical formula is proposed for the adjustment distance  $d$  as follows:

$$d = \begin{cases} 2r \cos \theta & \text{if } \theta \leq \pi/2; \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

where  $r$  is the average thickness of the upper arm bone.

The wrist joint is refined to the point with the minimum thickness around it. For a human body model with a straight arm, there will be no salient geometric features near the elbow joint. In this case, the elbow joint is adjusted according to the length ratio of the template skeleton if the shoulder joint or wrist joint is adjusted. Figure 6c shows the refined arm joints.

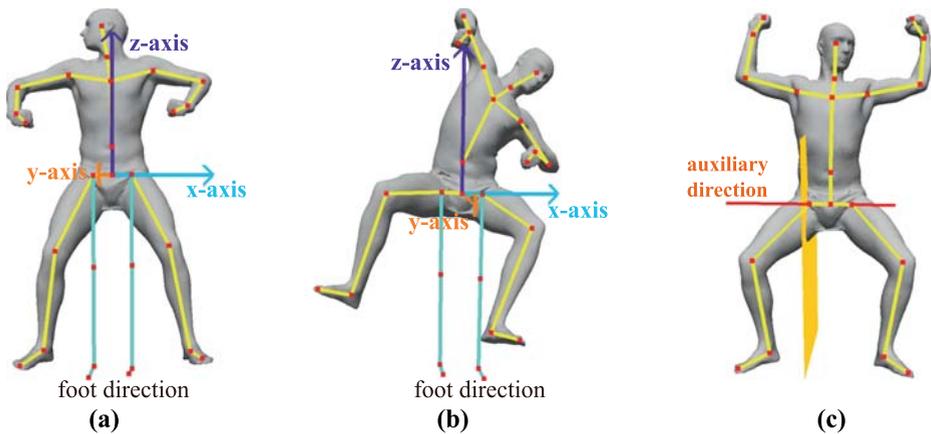
For a human body model with an arbitrary pose, it is difficult to locate the hip joints due to the lack of obvious geometric features. From observation, the lumbar bone is generally perpendicular to the hip bone. Therefore, the hip joint should be adjusted away from the knee joint to make the lumbar bone perpendicular to the hip bone. To achieve this, a hip direction which is perpendicular to the lumbar bone is obtained first by projecting the hip joints to the normal plane of the lumbar bone. Then, the hip joint is located on the midpoint of the common perpendicular line of the hip direction and thigh bone direction. When a leg of the human body model is straight, the corresponding knee joint is adjusted based on the length ratio once the hip joint is located. Figure 6d shows the refined leg joints.

After these joint refinements are made based on prior knowledge, the joints of the kinematic skeleton are located at more reasonable positions.

### 3.3 Symmetry distinction

Since the matching algorithm does not take the symmetric ambiguity into consideration, the kinematic skeleton may be semantically symmetrically flipped for symmetric parts (legs and arms), making the semantics of the kinematic skeleton inconsistent with that of the human body model.

Because the correction is focused on semantics, any method which is semantic-independent cannot be used to solve this problem. In this paper, a semantic-based local coordinate system is established on the kinematic skeleton to distinguish the symmetry of the kinematic skeleton. The direction determination of the local coordinate system is based on two observations: 1) human feet face forward in the standard pose, and 2) the waist joint is above the crotch joint. Figure 7a and b show the local coordinate systems established based on the above observations.



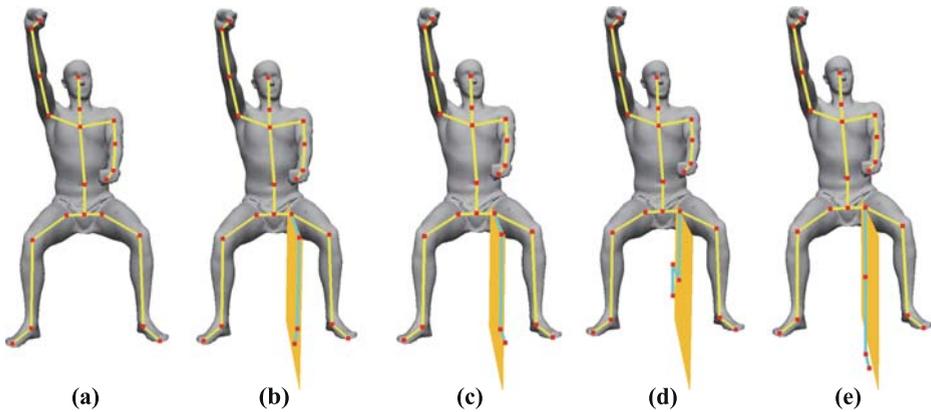
**Fig. 7** a,b The local coordinate system for the human body model. c Definition of an auxiliary direction for the human body model

For a human body model, the positive  $z$ -axis direction is defined from the crotch joint to the waist joint. Then, the leg skeletons and foot skeletons are rotated to the standard pose to obtain the feet directions. The positive  $x$ -axis direction is defined via the cross product of the positive  $z$ -axis direction and the average direction of the feet directions. Finally, the positive  $y$ -axis direction is determined via the cross product of the positive  $x$ -axis direction and the positive  $z$ -axis direction. In the above local coordinate system, the positive  $x$ -axis points to the left part of the human body model.

To rotate the leg bones and foot bones to the standard pose, an auxiliary direction is first defined that is perpendicular to the plane formed by the leg bone and foot bone in the standard pose. We project the hip joints to the normal plane of the  $z$ -axis and define the direction between the projected hip joints as the auxiliary direction, as shown by the red line in Fig. 7c. Then, the leg bone and foot bone should be rotated to the plane which is perpendicular to the auxiliary direction and across the hip joint, as the yellow plane shown in Fig. 7c.

Once the auxiliary direction is determined, the leg bone and foot bone undergo a step-by-step rotation to the standard pose. (1) The leg bone and foot bone are rotated around the axis which is parallel to the  $z$ -axis and across the hip joint simultaneously such that the plane determined by the thigh bone and calf bone is perpendicular to the auxiliary direction. (2) The foot bone is rotated around the calf bone such that the foot bone is perpendicular to the auxiliary direction. When the thigh bone and calf bone are on a straight line, these two steps are combined to rotate the leg bone and foot bone simultaneously such that the plane determined by the leg bone and foot bone is perpendicular to the auxiliary direction. (3) The thigh bone is rotated around the auxiliary direction to the negative  $z$ -axis direction. (4) The calf bone is rotated around the auxiliary direction to the negative  $z$ -axis direction. Since the rotation angle of the joint is limited for the human body shape, the rotation angles in step (4) should not be greater than 180 degrees, while the others should not be greater than 90 degrees. A step by step example of the rotation of the bones to the standard pose is shown in Fig. 8. A similar process is used for the other leg.

Once the local coordinate system is established, the positive  $x$ -axis direction is used to distinguish the symmetry of the kinematic skeleton. The hip (shoulder) joints that do not



**Fig. 8** Step-by-step rotation of the leg bone and the foot bone to the standard pose

produce confusion in any pose are taken to determine the left and right legs (arms) of the human body model.

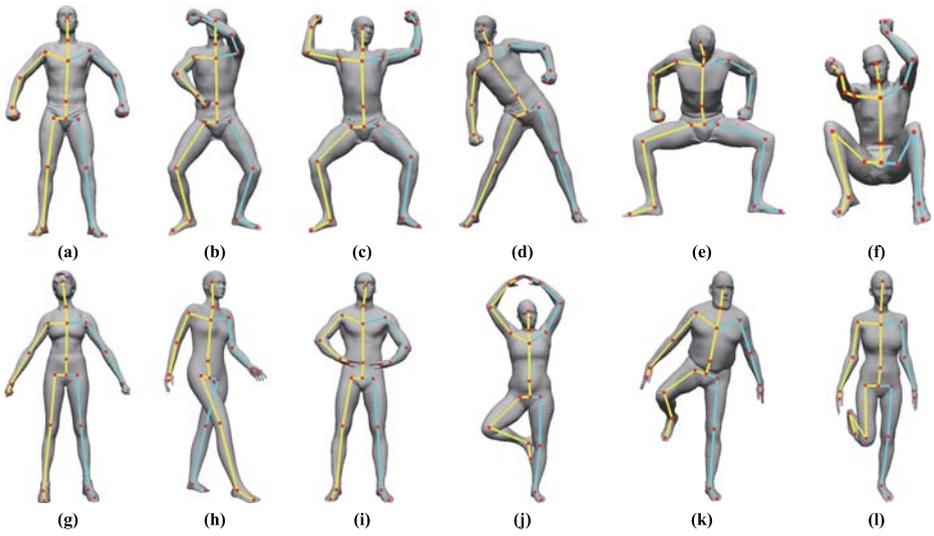
After the steps above, all joints of the kinematic skeleton are attached with correct semantics, which can be used to recognize the semantics of the human body model. A similar but not identical local coordinate system should be established if the method is extended to animals, but this is not explained here because this paper mainly focuses on a human body model.

## 4 Results and discussion

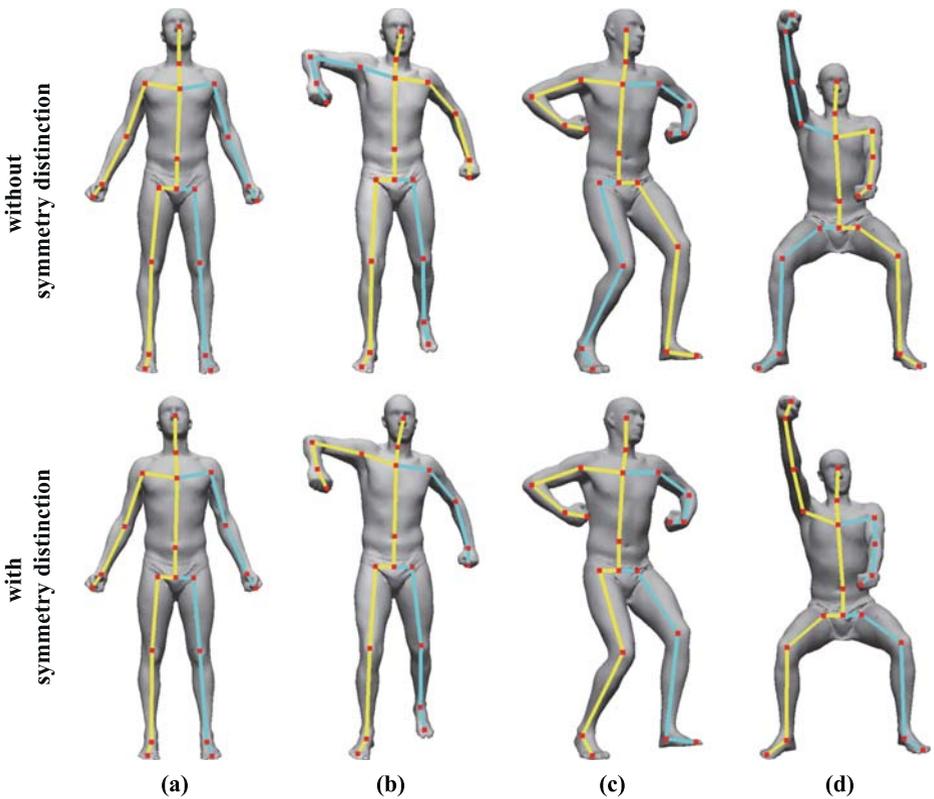
The proposed method generates a symmetry-aware kinematic skeleton with semantics for a human body model with an arbitrary pose and orientation. The method was implemented on a desktop PC with an Intel Core i5-4590 CPU, 16 GB memory, Windows 7 OS and a single thread. Three datasets were adopted for the test: the SCAPE dataset [3], which provides 71 human body models of the same person in different poses; the MPI FAUST dataset [8], which contains 100 human body models of 10 people in 10 different poses; and the Princeton Segmentation Benchmark (PSB) [9], which provides 20 human body models, in which 2 models are not applicable to the proposed method due to the topological connection of hands and legs.

The method was fully evaluated in three aspects: generating kinematic skeletons of human body models with arbitrary poses and orientations, symmetry distinction of the kinematic skeleton and joint localization.

Figure 9 shows the results of the proposed method for a set of human body models with various poses and orientations, among which each human body model is taken as input respectively. These models have no fixed orientation, and their original orientations are adopted as input. More results for different orientations are also given in Fig. 14. As shown in Fig. 9, the proposed method correctly generates kinematic skeletons for all of these human body models. Therefore, the proposed method does not limit the pose and orientation of the human body model.



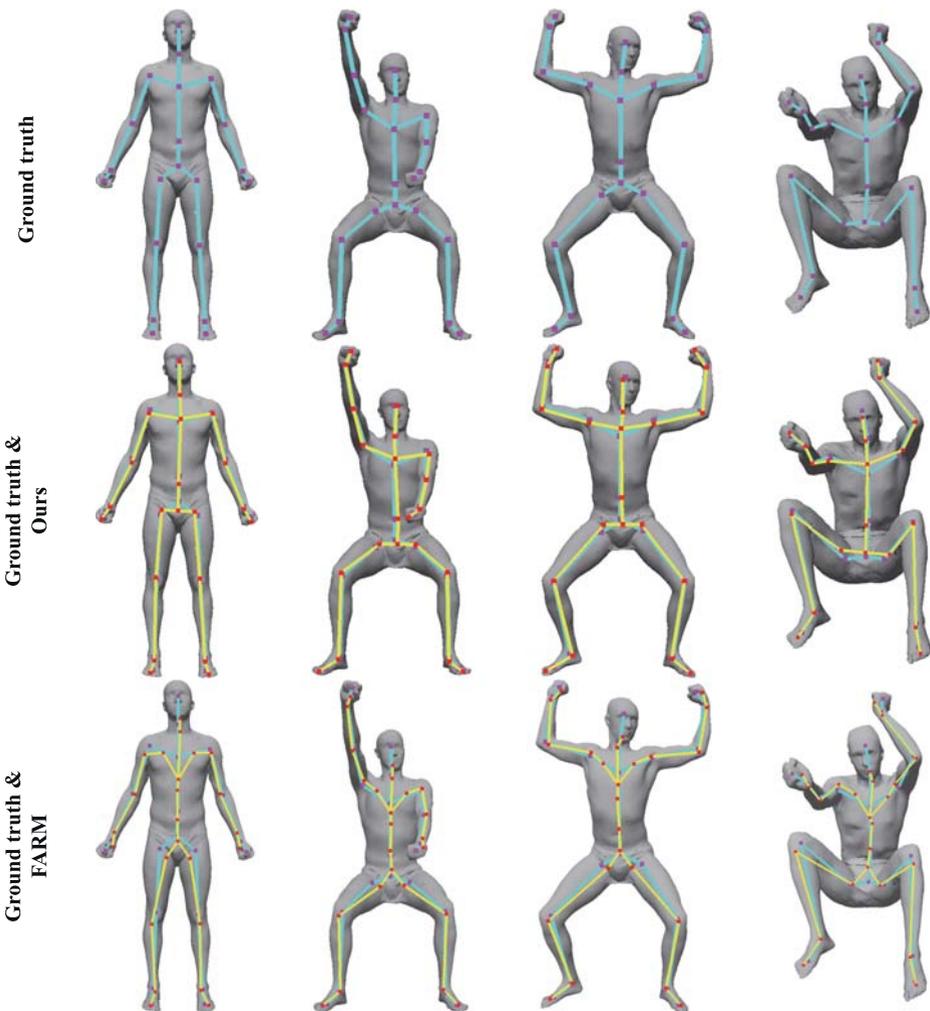
**Fig. 9** Kinematic skeleton generation results by the proposed method. The bones on the left part of the kinematic skeleton are shown in blue, while the others are shown in yellow



**Fig. 10** Kinematic skeleton generation results without symmetry distinction (top row) and with symmetry distinction (bottom row)

The symmetry of the generated kinematic skeleton is distinguished by a local coordinate system. Without the symmetry distinction, the kinematic skeleton may suffer from symmetric ambiguity in semantics, as shown in the top row of Fig. 10. It can also be seen in Fig. 9 that the proposed method successfully distinguishes the symmetric parts of the kinematic skeleton. For all the 189 human body models tested, our algorithm can correctly distinguish the symmetric parts.

For joint localization, Fig. 9 shows that all joints of the kinematic skeleton are located at reasonable positions, even for the part without salient features, such as the elbow joints and the knee joints in Fig. 9g. To quantitatively evaluate the accuracy of the joints, the generated kinematic skeleton is compared to the kinematic skeletons that are provided manually by using some software. Four human body models with different poses and orientations are



**Fig. 11** Top row: the average kinematic skeletons (ground truth) generated manually. Middle row: the comparison of our kinematic skeletons and the ground truth. Bottom row: the comparison of the kinematic skeletons generated by FARM [22] and the ground truth

analyzed in this part. To improve the reliability, the kinematic skeletons are generated by 6 graduate students whose research interests are computer graphics and its applications separately, and the average kinematic skeleton for each human body model is taken as the ground truth. The ground truth is shown in the top row of Fig. 11. A visual comparison between our kinematic skeleton and the ground truth is shown in the middle row of Fig. 11.

Table 1 shows the average errors for each joint of these four human body models. Besides the errors between our kinematic skeleton and the ground truth, the average errors and the maximum errors between the hand-build kinematic skeletons and the ground truth are also provided in Table 1. Although the errors of our kinematic skeletons are a bit larger than the average errors of the hand-build kinematic skeletons, the errors of most joints are lower than the maximum errors of the hand-build kinematic skeletons. For boundary joints and joints with less features, e.g., the chest joint and the crotch joint, it's really difficult to obtain identical positions, even for manual operations. However, the proposed method can generate a kinematic skeleton which is comparable to the hand-build kinematic skeletons. Therefore, the joints of our kinematic skeleton are accurate.

**Comparisons** The proposed method is compared with three kinematic skeleton generation methods: the method proposed by Marin et al. [22], called FARM, the method proposed

**Table 1** Quantitative evaluation of the generated kinematic skeletons

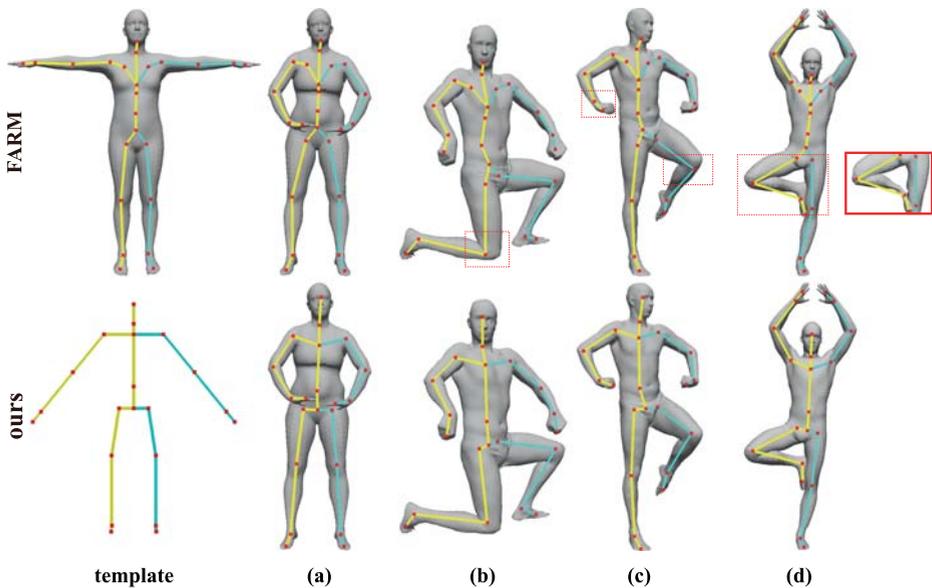
Joint	Avg. error	Hand-build		FARM [22]
		Avg. error	Max. error	Avg.error
Lshoulder	0.0160	0.0111	0.0224	0.0106
Rshoulder	0.0171	0.0096	0.0176	0.0161
Lelbow	0.0066	0.0056	0.0105	0.0098
Relbow	0.0080	0.0053	0.0087	0.0119
Lwrist	0.0048	0.0043	0.0085	0.0168
Rwrist	0.0056	0.0038	0.0084	0.0188
Lhand	0.0156	0.0059	0.0096	0.0209
Rhand	0.0117	0.0071	0.0162	0.0190
Lhip	0.0147	0.0103	0.0150	0.0254
Rhip	0.0196	0.0124	0.0207	0.0257
Lknee	0.0074	0.0069	0.0114	0.0152
Rknee	0.0074	0.0071	0.0123	0.0222
Lankle	0.0102	0.0103	0.0180	0.0102
Rankle	0.0094	0.0104	0.0187	0.0072
Lfoot	0.0208	0.0099	0.0263	0.0082
Rfoot	0.0185	0.0130	0.0293	0.0107
Neck	0.0118	0.0115	0.0257	0.0188
Crotch	0.0189	0.0144	0.0259	0.0262
Chest	0.0181	0.0151	0.0306	–
Head	0.0178	0.0148	0.0346	–
Waist	0.0128	0.0124	0.0213	–

The maximum geodesic distances of the human body models are unit scaled, and the Euclidean distances between corresponding joints are taken as the errors

by Baran and Popović [7], called Pinocchio and the method proposed by Hajari et al. [13]. FARM focuses on 3D model registration, while the kinematic skeleton generation is one of its straightforward applications. Pinocchio is a famous method for embedding a template skeleton into the model directly. And the method proposed by Hajari et al. [13] specifically focuses on a human body model and embeds a template skeleton into the human body model indirectly. Other indirect methods [5, 25, 26] prefer kinematic skeleton generation methods for models in various categories rather than focusing on a human body model. Therefore, we do not compare our method with these methods.

A visual comparison between the kinematic skeletons generated by FARM and the proposed method is shown in Fig. 12. Since FARM is a registration-based method, the kinematic skeleton can be generated from the deformed template model directly. All the results of FARM shown in this paper are generated by taking SMPL [20] as the template model. As shown in Fig. 12, both FARM and the method proposed in this paper can distinguish the symmetry of the generated kinematic skeleton successfully. Because the joint positions of the kinematic skeleton generated by FARM are determined by the vertex positions of the template model, the quality of 3D model registration determines the accuracy of joint positions. FARM can generate kinematic skeletons reasonably for most of the tested models. An example is shown in Fig. 12a. But it may also generate kinematic skeletons with some inaccurate joint positions due to inaccurate 3D model registrations, as shown in Fig. 12b–d. However, for all these human body models, the method proposed in this paper can generate kinematic skeletons with joints in reasonable positions. What's more, for each human body model shown in Fig. 12, it takes more than thirty minutes for FARM to generate a kinematic skeleton, while the proposed method can obtain a kinematic skeleton within one minute.

To compare the quality of the generated kinematic skeletons quantitatively, the kinematic skeletons for the human body models shown in Fig. 11 are also obtained by FARM. And a visual comparison between the kinematic skeletons generated by FARM and the ground



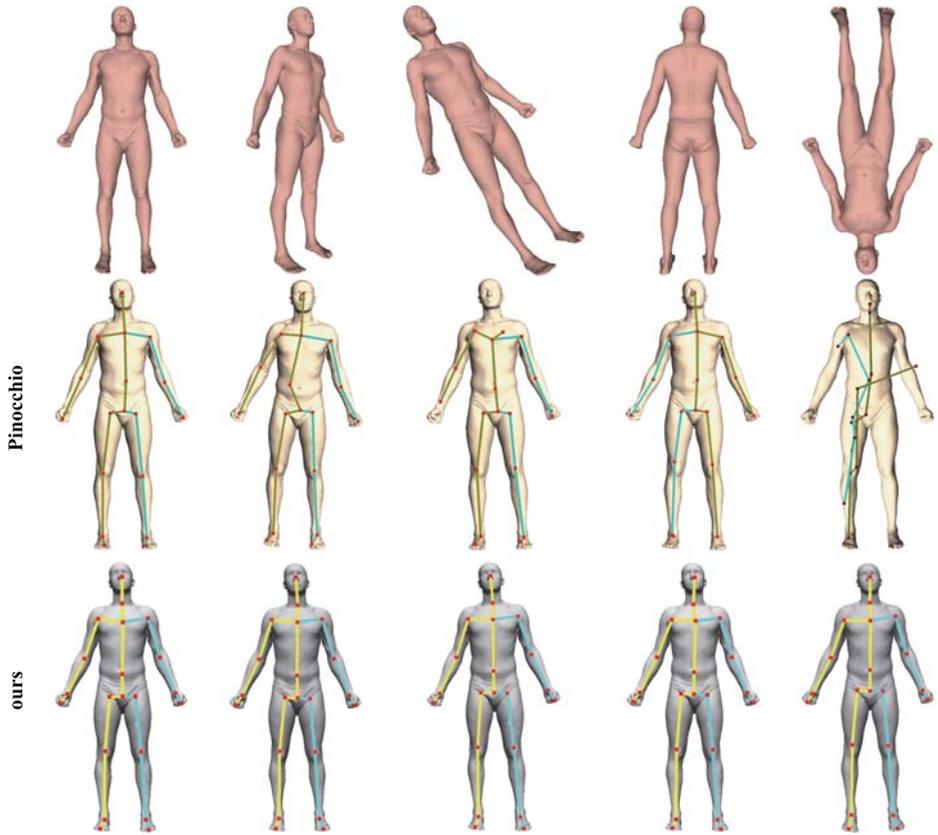
**Fig. 12** Top row: the kinematic skeletons generated by FARM. Bottom row: the kinematic skeletons generated by our method

truth is shown in the bottom row of Fig. 11. Because the structure of the template skeleton obtained by FARM is inconsistent with that of the template skeleton adopted in this paper, only the joints with the same semantics are analyzed for FARM. A quantitative comparison of the errors for joint positions is reported in Table 1. For most of the joints, the joint positions of the kinematic skeleton generated by the proposed method are more accurate. For the joints with lower accuracy than those obtained by FARM, such as the shoulder joints, the errors of the proposed method are also less than the max errors for the hand-build kinematic skeletons.

Figure 13 shows the comparison of the kinematic skeleton generation results between Pinocchio and our method. Since Pinocchio requires the input model to be given in the same orientation and pose as the template skeleton, the orientation of the model is first adjusted manually. By contrast, for our method, the original human body model is used as input. For human body models with poses similar to the template skeleton, Pinocchio can generate good results. However, for models with different poses, the results of Pinocchio may be inaccurate or even incorrect. However, the proposed method can generate correct results for all of these human body models. We also try to take the 18-joint template skeleton used by Pinocchio as the input of the proposed method and obtain satisfactory results, as shown in

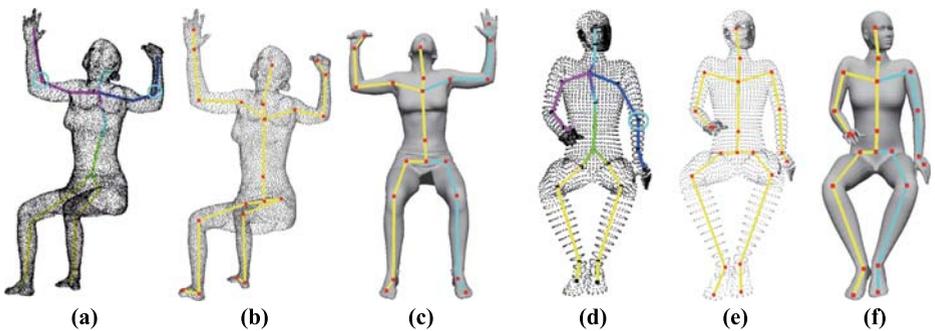


**Fig. 13** Top row: the skeleton embedding results by Pinocchio. Middle row: the kinematic skeleton generation results of our method by using the template skeleton of Pinocchio. Bottom row: the kinematic skeleton generation results of our method by using our 21-joint template skeleton



**Fig. 14** Top row: human body models with different orientations. Middle row: the kinematic skeletons generated by Pinocchio. Bottom row: the kinematic skeletons generated by our method

the middle row of Fig. 13. However, Pinocchio cannot change the structure of the template skeleton directly because some of the parameters are obtained by learning.



**Fig. 15** Comparison between the method proposed by Hajari et al. [13] (a,d) (pictures taken from [13]) and our method (b,c,e,f)

**Table 2** Columns  $t_1$ ,  $t_2$ , and  $t_3$  show the running time of kinematic skeleton generation, kinematic skeleton refinement and symmetry distinction, respectively

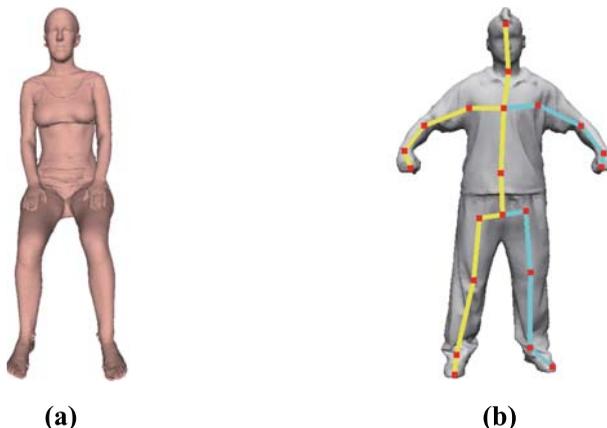
Dataset	Size	Times(s)			Total
		$t_1$	$t_2$	$t_3$	
SCAPE	25K	13.270	0.761	0.0000	14.031
FAUST	13K	4.657	0.430	0.0000	5.087
PSB	20K	8.799	0.816	0.0000	9.615
	30K	21.052	1.165	0.0000	22.217

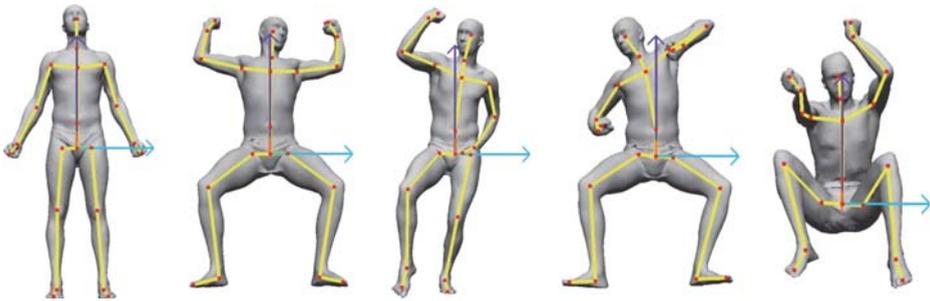
Pinocchio only addresses models with the same orientation as the template skeleton. For the same model with different orientations, the results will be quite different, as shown in Fig. 14. However, our method is insensitive to the orientation of the human body model.

Figure 15 shows the comparison of the kinematic skeleton results between the method proposed by Hajari et al. [13] and our method. The method proposed by Hajari et al. [13] locates the joints on the curve skeleton directly, making some joints not accurate enough, for example, the knee joints and elbows joints in Fig. 15a and the shoulder joints in Fig. 15d. Furthermore, the method proposed by Hajari et al. [13] does not consider the symmetry distinction problem. By contrast, our method generates a symmetry-aware kinematic skeleton with more accurate joints.

**Runtime** Table 2 shows the running time of the proposed method for the human body model with different face sizes. The data of the SCAPE dataset and the FAUST dataset in Table 2 are the average running time for all models in the dataset. Most of the computation time is devoted to curve skeleton extraction, while other steps take less time.

**Limitations** For the human body model shown in Fig. 16a, the geometries of the hand and the thigh are connected. In this case, the topology of the model is inconsistent with that of the template skeleton. The proposed method cannot process the model correctly.

**Fig. 16** a A failure case. b The kinematic skeleton of a dressed human body model



**Fig. 17** Shape alignment results based on the local coordinate system

Some prior knowledge is adopted to refine the joints of the kinematic skeleton. For a dressed human body model, some observations and prior knowledge may be inaccurate, and thus the generated kinematic skeleton will not be as good as the tested human body models, as shown in Fig. 16b. In addition, the length ratio is adopted to deal with the cases where a leg or an arm of the human body model is straight. This ratio is statistically significant. It holds for most of the people. However, the ratio varies for some special people, e.g., the high jumpers. In these cases, the ratio of the template skeleton can be adjusted interactively if accurate joint points are required. How to determine the related joint positions automatically via machine learning will be further investigated in our future work.

Our method assumes that the human feet are always in the forward direction in the standard pose. Consequently, if the input model does not contain feet or the feet can face back simultaneously because of the strong flexibility of some people, our method cannot distinguish the symmetry correctly.

## 5 Applications

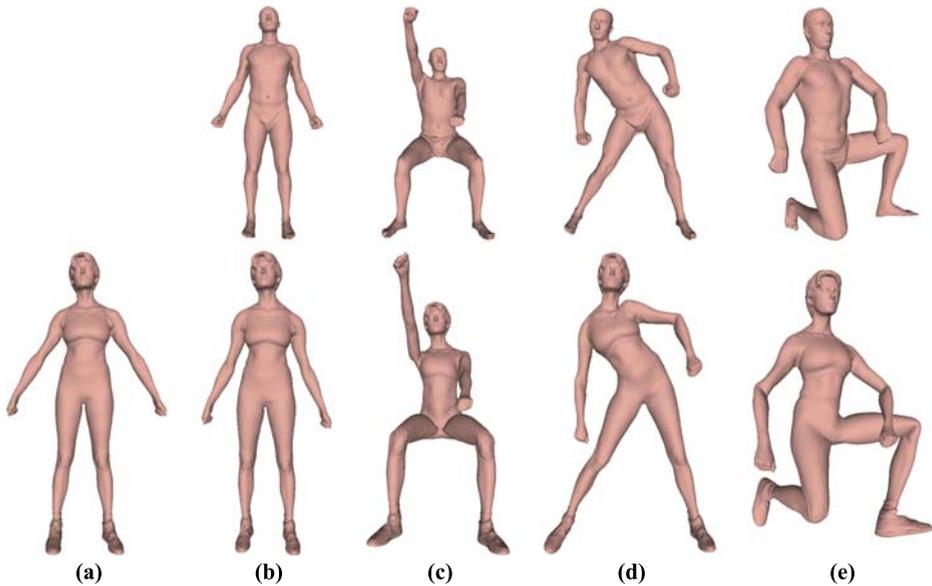
In this section, we demonstrate the effectiveness of the kinematic skeleton generated by the proposed method for some graphics applications.

### 5.1 Shape alignment

To correct the semantics of symmetric parts of a kinematic skeleton, a semantic based local coordinate system is established in the proposed method. Since the local coordinate system is semantically related to the human body model, the coordinate system can be used to align the human body models by aligning the coordinate axes, making human body models have a consistent orientation. Figure 17 shows some human body models with the coordinate system alignment.

### 5.2 Skeleton-based deformation

As an abstract representation of a 3D model, a kinematic skeleton is often used to deform or animate the 3D model. Some shape deformations directly based on the kinematic skeleton generated by the proposed method are performed, which are shown in the bottom row of Fig. 18. The skinning procedure is accomplished by using the method proposed by Baran and Popović [7].



**Fig. 18** Shape deformation results. Top row: human body models in the SCAPE dataset. Bottom row: **a** an original human body model in the PSB dataset; **b–e** the deformed human body models based on the poses in the top row

Since our method can generate the kinematic skeletons for human body models in different poses and orientations, the poses can be transferred to different human body models via the generated kinematic skeletons. The models on the top row of Fig. 18 are human body models in the SCAPE dataset. The model in Fig. 18a is a human body model in the PSB dataset. Once all the kinematic skeletons are generated, we can deform the kinematic skeleton of the model in Fig. 18a to the kinematic skeletons generated from the models in the SCAPE dataset. The deformed human body models are shown in the bottom row of Fig. 18. Therefore, the generated kinematic skeleton is of high quality and is appropriate for the human body deformation.

### 5.3 Shape co-segmentation

Symmetric ambiguity is a difficult problem to be solved for 3D models with intrinsic symmetry in shape co-segmentation. Because the kinematic skeletons generated by the proposed method are semantically distinguished for symmetric parts and structurally consistent for human body models, they can be used to distinguish the symmetric parts for human body models in shape co-segmentation.

Figure 19a shows the consistent region segmentation results of the method proposed by Kleiman and Ovsjanikov [14]. The symmetric parts of human body models are not distinguished by this method. Although a symmetry-breaking algorithm is proposed to obtain a one-to-one part matching result, the totally symmetric flips that map the entire left side of a human body model to the right side cannot be avoided. With the symmetry-aware kinematic skeleton generated by the proposed method, the symmetric parts can be distinguished by the semantics of the nearest bones on the kinematic skeleton. As shown in Fig. 19b, the symmetric parts of the models are distinguished and matched correctly.

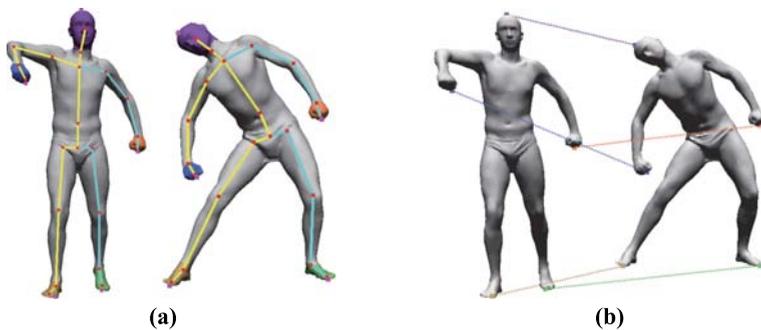


**Fig. 19** **a** Consistent region segmentation results by Kleiman and Ovsjanikov [14]. **b** Consistent region segmentation results refined by our kinematic skeletons

### 5.4 Shape correspondence

Shape correspondence is a fundamental problem in computer graphics with a wide range of applications such as shape deformation, shape analysis, etc. There are many good methods [24, 32, 36] to find correspondences between isometric models based on some isometric-invariant features, and the symmetric ambiguity problem can also be solved by some oriented descriptors. However, for non-isometric models, it is difficult to obtain correspondences since features between the non-isometric models are not invariant. Some methods [2, 35] establish the correspondences between non-isometric models by optimizing the minimum distortion with a set of landmarks. However, finding a sparse set of feature correspondences automatically is not an easy task.

With the kinematic skeleton generated by the proposed method, a sparse set of correspondences can be obtained easily. Given two human body models, kinematic skeletons can be generated by the proposed method first. Then, some parts of the human body model can be segmented according to the joint positions of the kinematic skeleton, as shown in Fig. 20a. For each part, the point with maximum HKS [31] values can be defined as a feature point. Feature points corresponding to parts with the same semantics are regarded as corresponding points, as shown in Fig. 20b.



**Fig. 20** **a** Segmented parts of human body models based on the generated kinematic skeletons. **b** A sparse set of correspondences between human body models

Recently, some shape correspondence methods [23, 27] have been proposed based on the functional map framework. The results of the symmetry-aware shape co-segmentation and the sparse point correspondences can also be used as constraints of the functional map framework to avoid symmetric ambiguity and improve the accuracy of dense shape correspondence.

## 6 Conclusion and future work

We have proposed an automatic symmetry-aware kinematic skeleton generation method that is specifically designed for a human body model. Arbitrary pose and orientation of the input human model are allowed. The joint positions of the generated kinematic skeleton are accurate since the geometry of the human body model and some prior knowledge are taken into consideration. Furthermore, the proposed method can distinguish the symmetry of the generated kinematic skeleton. In addition to animation applications, our symmetry-aware kinematic skeleton generation method can also be used in shape correspondence, shape co-segmentation and other applications that need to distinguish the symmetric parts of a human body model.

In the future, we will focus on skeleton generation with more details, such as a skeleton with fingers and toes. Moreover, a similar method can also be extended to animals by taking a corresponding template skeleton as input, which can also be considered in future work.

**Acknowledgements** This work was supported by the National Natural Science Foundation of China under Grants No. 61732015 and No. 61472349, and Key Research and Development Program of Zhejiang Province under Grants No. 2018C01090.

## References

1. Aguiar ED, Theobalt C, Thrun S, Seidel H (2008) Automatic conversion of mesh animations into skeleton-based animations. *Comput Graph Forum* 27(2):389–397
2. Aigerman N, Poranne R, Lipman Y (2015) Seamless surface mappings. *ACM Trans Graph* 34(4):72:1–13
3. Anguelov D, Srinivasan P, Koller D, Thrun S, Rodgers J, Davis J (2005) Scape: shape completion and animation of people. *ACM Trans Graph* 24(3):408–416
4. Au OKC, Tai CL, Chu HK, Cohen-Or D, Lee TY (2008) Skeleton extraction by mesh contraction. *ACM Trans Graph* 27(3):44:1–10
5. Aujay G, Hétry F, Lazarus F, Depraz C (2007) Harmonic skeleton for realistic character animation. In: *ACM SIGGRAPH/Eurographics symposium on computer animation*, pp 151–160
6. Avril Q, Ghafourzadeh D, Ramachandran S, Fallahdoust S, Ribet S, Dionne O, de Lasa M, Paquette E (2016) Animation setup transfer for 3d characters. *Comput Graph Forum* 35(2):115–126
7. Baran I, Popović J (2007) Automatic rigging and animation of 3d characters. *ACM Trans Graph* 26(3):72:1–8
8. Bogo F, Romero J, Loper M, Black MJ (2014) Faust: dataset and evaluation for 3d mesh registration. In: *IEEE Conference on computer vision and pattern recognition*, pp 3794–3801
9. Chen X, Golovinskiy A, Funkhouser T (2009) A benchmark for 3d mesh segmentation. *ACM Transa Graph* 28(3):73:1–8
10. Cornea ND, Silver D, Min P (2007) Curve-skeleton properties, applications, and algorithms. *IEEE Trans Vis Comput Graph* 13(3):530–548
11. Fechteler P, Hilsmann A, Eisert P (2016) Example-based body model optimization and skinning. In: *Proceedings of the 37th annual conference of the European association for computer graphics*, pp 5–8
12. Feng A, Casas D, Shapiro A (2015) Avatar reshaping and automatic rigging using a deformable model. In: *Proceedings of the 8th ACM SIGGRAPH conference on motion in games*, pp 57–64

13. Hajari N, Cheng I, Basu A (2016) Robust human animation skeleton extraction using compatibility and correctness constraints. In: IEEE International symposium on multimedia, pp 271–274
14. Kleiman Y, Ovsjanikov M (2019) Robust structure-based shape correspondence. *Comput Graph Forum* 38(1):7–20
15. Le BH, Deng Z (2014) Robust and accurate skeletal rigging from mesh sequences. *ACM Transact Graph* 33(4):84:1–10
16. Leordeanu M, Hebert M (2005) A spectral technique for correspondence problems using pairwise constraints. In: IEEE International conference on computer vision, pp 1482–1489
17. Lien JM, Keyser J, Amato NM (2006) Simultaneous shape decomposition and skeletonization. In: ACM symposium on solid and physical modeling, pp 219–228
18. Liu PC, Wu FC, Ma WC, Liang RH, Ouhyoung M (2003) Automatic animation skeleton construction using repulsive force field. In: Pacific Conference on computer graphics and applications, pp 409–413
19. Liu T, Kim VG, Funkhouser T (2012) Finding surface correspondences using symmetry axis curves. *Comput Graph Forum* 31(5):1607–1616
20. Loper M, Mahmood N, Romero J, Pons-Moll G, Black MJ (2015) Smpl: a skinned multi-person linear model. *ACM Trans Graph* 34(6):248:1–16
21. Ma J, Choi S (2014) Kinematic skeleton extraction from 3d articulated models. *Comput Aided Des* 46(1):221–226
22. Marin R, Melzi S, Rodolà E, Castellani U (2020) Farm: functional automatic registration method for 3d human bodies. *Comput Graph Forum* 39(1):160–173
23. Ovsjanikov M, Ben-Chen M, Solomon J, Butscher A, Guibas L (2012) Functional maps: a flexible representation of maps between shapes. *ACM Trans Graph* 31(4):30:1–11
24. Ovsjanikov M, Mérigot Q, Pătrăucean V, Guibas L (2013) Shape matching via quotient spaces. *Comput Graph Forum* 32(5):1–11
25. Pantuwong N, Sugimoto M (2012) A novel template-based automatic rigging algorithm for articulated-character animation. *Comput Anim Virt Worlds* 23(2):125–141
26. Poirier M, Paquette E (2009) Rig retargeting for 3d animation. In: Graphics interface, pp 103–110
27. Ren J, Poulenard A, Wonka P, Ovsjanikov M (2018) Continuous and orientation-preserving correspondences via functional maps. *ACM Trans Graph* 37(6):248:1–16
28. Saha PK, Borgefors G, di Baja GS (2016) A survey on skeletonization algorithms and their applications. *Pattern Recogn Lett* 76(1):3–12
29. Sahillioğlu Y, Yemez Y (2013) Coarse-to-fine isometric shape correspondence by tracking symmetric flips. *Comput Graph Forum* 32(1):177–189
30. Sellamani S, Muthuganapathy R, Kalyanaraman Y, Murugappan S, Goyal M, Ramani K, Hoffman CM (2010) Pcs: prominent cross-sections for mesh models. *Comput-Aided Des Appl* 7(4):601–620
31. Sun J, Ovsjanikov M, Guibas L (2009) A concise and provably informative multi-scale signature based on heat diffusion. *Comput Graph Forum* 28(5):1383–1392
32. Wang M, Fang Y (2016) Local diffusion map signature for symmetry-aware non-rigid shape correspondence. In: ACM international conference on multimedia, pp 526–530
33. Wang K, Razzaq A, Wu Z, Tian F, Ali S, Jia T, Wang X, Zhou M (2016) Novel correspondence-based approach for consistent human skeleton extraction. *Multimed Tools Appl* 75(19):11741–11762
34. Xu Z, Zhou Y, Kalogerakis E, Singh K (2019) Predicting animation skeletons for 3d articulated models via volumetric nets. In: 2019 International conference on 3D vision, pp 298–307
35. Yang Y, Fu XM, Chai S, Xiao SW, Liu L (2018) Volume-enhanced compatible remeshing of 3d models. *IEEE Trans Vis Comput Graph* 25(10):2999–3010
36. Yoshiyasu Y, Yoshida E, Yokoi K, Sagawa R (2014) Symmetry-aware nonrigid matching of incomplete 3d surfaces. In: IEEE Conference on computer vision and pattern recognition, pp 4193–4200
37. Yoshiyasu Y, Yoshida E, Guibas L (2016) Symmetry aware embedding for shape correspondence. *Comput Graph* 60(5):9–22
38. Zhang Z, Yin K, Foong KW (2013) Symmetry robust descriptor for non-rigid surface matching. *Comput Graph Forum* 32(7):355–362



**Shan Luo** received her BSc in computer science from Jilin University in 2015. She is now a PhD student in State Key Lab of CAD&CG, Zhejiang University of China. Her research interest includes human body shape analysis and synthesis.



**Jieqing Feng** is a professor in the State Key Lab of CAD&CG, Zhejiang University, Peoples Republic of China. He received his BSc in applied mathematics from the National University of Defense Technology in 1992 and his Ph.D. in computer graphics from Zhejiang University in 1997 respectively. His research interests include geometric modeling, stereo vision and computer graphics application in solar thermal energy simulation and modeling.