

Automatic Conversion of Human Mesh into Skeleton Animation by Using Kinect Motion

Abdul Razzaq, Zhongke Wu, Mingquan Zhou, Sajid Ali, and Khalid Iqbal

Abstract—Skeleton-base-animation methods have been commonly used in the character animations. The process of making skeleton for character animation is a long-winded task requiring manual tweaking. This paper presents a novel method to create an automatic animated skeleton from 3D human geometric model through mesh contraction. An automatically generated skeleton is animated by using Kinect captured human motion. The method first, extract a 1D curve-skeleton from the mesh through mesh contraction with constraints. Secondly, the hierarchical joint-based skeleton (armature) has been generated, using the extracted 1D curve-skeleton of the input mesh automatically. Third, the real-time human motion is captured by using the Kinect device. The Kinect captured motion also converts into a standard skeleton motion BVH (Biovision hierarchical) format. Finally, the Kinect motion is retargeted to animate the resulting skeleton of the mesh through joint mapping. The main objective of the proposed approach is to minimize labor-intensive process of skeleton adjustment for character animation. The results of a mesh generated skeleton and plausible skeleton animation to demonstrate the efficiency of the proposed work. The mesh generated skeleton and Kinect motion skeleton both can be directly useful for mesh skinning, mesh rigging, and motion retargeting to create satisfactory character animation.

Index Terms—3D mesh, kinect motion, skeleton extraction, motion retargeting, skeleton animation.

I. INTRODUCTION

Prompt creation of realistic 3D human character animation has become very popular and challenging task in movies, computer games, simulations, education training, medical applications, forensic animation, avatar animation and surveillance systems. Animation of 3D objects is one of the most attractive parts of the computer graphics. In character animation, the use of skeleton has an extensive role. The skeleton has been utilized in many applications such as character rigging through skeleton embedding [1], mesh skinning using skeleton [2] and skeleton-driven soft body animation [3]. The skeleton-driven animation is a very effective method for animating characters.

Generally, the skeleton has been made by using the commercial animation packages such as Maya, 3D Max

Studio and Blender. These packages require manual tweaking of the skeleton, such as bones and vertex weighting of character, manual creation of a skeleton hierarchy for getting desired animation effects. In present animation systems, object representation and its skeleton are disjointed, which often create problems in animation of the objects. The character animation based on skeleton still require experienced work and laborious process.

To address these problems, a new framework has been proposed to generate an automatic animated skeleton from 3D human geometric mesh model. In this study, first extract a curve-skeleton from a polygon mesh by using mesh Laplacian contractions. The extracted curve-skeleton is converted into a joint-based-skeleton automatically. This method is to make a skeleton, which has an accurate joint position, follow typology of the human skeleton. The flow of method divided into four phases.

In first phase, mesh contraction has been applied to contract the given mesh model. The mesh contraction process preserves the original topology and connectivity of the mesh model. The mesh contraction process based on Laplacian smoothing with constraints. Contract the mesh iteratively until to obtain a 1D shape of that model. The 1D curve-skeleton is derived from the contracted mesh through edge contraction operation. The methods of Au *et al.* [4] used as starting point of the contracting mesh model.

Second phase, present an automatic conversion of the refined curve-skeleton into a joint-based skeleton. Because the suitable extracted curve-skeleton should leave enough information for the overall structure, while maintaining a certain level of details for the model. The extracted curve-skeleton is refined by mesh-skeleton mapping.

The third phase of this work is to capture a real-time motion of the human body by using Microsoft Kinect. The process for converting Kinect captured data into commonly used motion data format BVH (Biovision) is also described in this phase.

In the fourth phase, the generated skeleton has been animated by retargeting 3D motion data through joint mapping. The captured motion of the Kinect and CMU motion adopt for motion retargeting.

This work is an extension of the existing studies of static curve skeletons into animated skeleton. The present study is also contributes to capture the real-time motion by using Xbox Kinect and converted into a commonly used motion format automatically. The goal of present framework is to generate robust animated skeleton from the mesh which reduced the manual tweaking of the skeleton. The movements authenticity of the resulting skeleton is evaluated through motion retargeting.

It is expected that mesh generated skeleton has applications in a variety of fields such as, television

Manuscript received May 22, 2014; revised September 12, 2014. This work is partially supported by the National Natural Science Foundation of China (NO: 61170170).

Abdul Razzaq, Zhongke Wu, Mingquan Zhou and Sajid Ali are with the Engineering Research Center of Virtual Reality and Applications, Ministry of Education, P.R.C. College of Information Science & Technology, Beijing Normal University, China (e-mail: razzaq144@hotmail.com, zwu@bnu.edu.cn, mqzhou@bnu.edu.cn and sajid_ila@yahoo.com).

Khalid Iqbal is with the Department of Computer Science, COMSATS Institute of Information Technology, Attock Campus, Pakistan (e-mail: khalidqbal@ciit-attock.edu.pk).

commercials, animation movie industries and computer games. The present research could be used in character rigging and automatic skinning of the 3D mesh.

Recently, extraction of the kinematic skeleton from mesh is found in [5]. This method bears some similarity with our method. However, this study has limitations such as extracted shapes of skeletons are star like skeleton dissimilar to the standard skeleton. This method also required inserting some extra joints for skeleton animation.

The remainder of the paper is organized as follows: Section II reviewed the related work. The framework of the proposed work is presented in Section III. Extraction of the curve-skeleton is explained in Section IV. An automatic construction of the joint-based skeleton by using a refined 1D curve skeleton is described in Section V. Section VI is devoted to Kinect human motion capturing and conversing. The process of joint mapping, motion retargeting is described in Section VII. Experimental results are presented in Section VIII. Finally, the paper has been concluded in Section IX.

II. RELATED WORK

In the computer graphics literature extensive research has been done on 3D character animation. Character animation techniques can be broadly divided into an extraction of the curve-skeleton, skeleton animation, motion capturing and retargeting.

Extraction of Curve-Skeleton: Curve-skeleton of an object provides an intuitive abstraction of geometrical and topological shape. The curve-skeleton is useful in shape analysis [6], 3D character animation [7], morphing and shape registration [8], object matching and shape retrieval [9] and surface reconstruction [10].

Curve skeleton extraction algorithms, properties and its application have explained in [11]. Methods of extracting a curve-skeleton from the mesh can be classified into two main categories: Geometric methods and Volumetric methods. Geometric methods worked directly on point data sets or polygon meshes for extracting a curve-skeleton. The popular approach for geometric extraction is the Voronoi diagram to calculate the skeleton from mesh vertices [12]. Day *et al.* [13] used a medial geodesic function (MGF) by extracting edges and face of Voronoi diagram. The disadvantage of such methods need a pruning and thinning algorithms to obtain a curve skeleton due to medial surface, rather than medial axis. Reeb-graph-based methods are also gaining attention to compute curve skeletons from mesh models [12]. Most Reeb-graph methods require to define the explicit boundary conditions by the user. Volumetric methods are also widely used for extracting a curve-skeleton from the mesh. Volumetric methods require a voxelized representation of the input model [14]. Wang *et al.* [15] proposed a method for computing smooth curve-skeleton of mesh by using iterative least squares optimization. Volumetric methods are not always robust to all shapes:- due to additional processing for translation of the model surfaces. The accurate extraction of skeleton from mesh is of great importance in skeleton driven animation. The appropriate extracted skeleton including the proper position of joints with connected bones.

Skeleton-driven Animation: Skeleton-driven animation

techniques are frequently used to animate 3D characters. In character rigging skeleton is binded to mesh. When the skeleton is moved the character animated automatically. Llya and Jovan [1] proposed an automatic rigging method attach a generic skeleton to 3D characters for animation. They fitted template skeleton to mesh. However, this method depends upon complex mathematics computation while embedding a skeleton for character animation. Similarly, an automatic rigging of multi-component characters based on animated skeleton is proposed by [16]. They decomposed character according to motion skeleton. This method taken an input of mesh model and kinematic skeleton with associated motion. Nevertheless, the method assumed a standard format of the skeleton.

Motion Capturing and Retargeting: A detailed overview of human motion capturing, motion analysis and retargeting has been explained in [7], [17]. The different motion captured data formats are explained in [18]. In motion capturing system the optical motion capture (OMC) is one of the most popular system. However, the devices of OMC are very expensive and often need an expert knowledge for installing and operating the system. To capture real-time human motion Kinect Xbox device attracted much attention of the researchers [19]. The Kinect capturing motion can be used in different domains such as scanning of 3D human body [20], real-time recognition of human pose [21] and human motion reconstruction [22].

Sequences of an object are derived to reuse the motion of an existing object called motion retargeting. Michael Gleicher [23] is a pioneer of introducing a motion retargeting technology in character animation. Motion Retargeting through node-node corresponding (Joint-mapping) between source and target skeleton are proposed in [24]. They described an intermediate skeleton which consumed extra storage space to store middle bone motion data.

In short, a significant amount of research has proposed different methods for extracting skeleton from the mesh [4], [7], [8]. However, most of recent research has focused only to draw the line-skeleton in a given model. In contrast, proposed approach aim to obtain an accurate curve-skeleton which describe joint position.

III. OVERVIEW

Human triangular mesh is an input to contract the mesh into an approximate zero-volume. The contract process represents the skeletal shape of the mesh. The skeletal shape of the mesh is converted into a 1D curve skeleton by applying edge collapse. Original connectivity and topology of the mesh are retained during contraction and edge collapse operation. The extracted skeleton is refined through skeleton-mesh mapping. Each node of the curve skeleton is moved to the centre of its corresponding local mesh region (Π_k). The extracted nodes of the curve-skeleton is the core part of the mesh. By using these nodes an automatic joint-based skeleton is generated. The movement's position of human joints is captured by using the Kinect sensor. To implement the skeleton animation and enrich the movement of resulting skeleton Kinect captured motion and CMU motion are re-targeted. The workflow of 3D human mesh into

skeleton animation through motion retargeting is presented in Fig. 1.

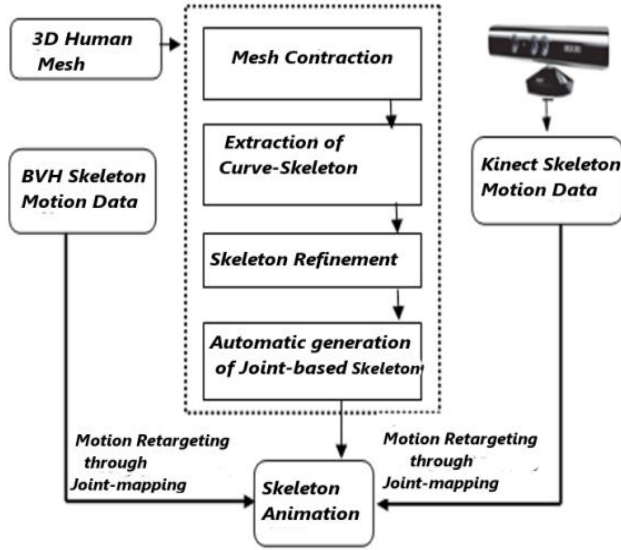


Fig. 1. Workflow of the approach.

IV. CURVE-SKELETON EXTRACTION

Several algorithms have been proposed to extract the curve-skeleton from the mesh. Therefore, the skeletonization in our work is based on the geometric approach due to its robust and noise insensitivity. The extracted curve-skeleton through geometric approach should be homotopic to the original mesh. The extraction procedure does not require voxelization. It also provides a better result with easy handling. Geometric contraction methods are commonly based on Laplacian-based mesh smoothing. The important work in the subsequent section is to apply a well-defined filter on the mesh vertices.

A. Laplacian Mesh Smoothing

The contraction process preserves the global shape of mesh model by anchoring points through an implicit Laplacian smoothing. A human mesh $G=(V, F, E)$ which has vertices V face F and edges E where $v=[v_1^T, v_2^T, v_3^T, \dots, v_n^T]^T$ are the position of the vertices. The Laplacian smoothing operator (1) is used for the surface smoothing of the mesh which is described in [25]. The smoothing operator L is $n \times n$ a square matrix. The operator L is applied to n vertices in the vector V as a filter. The term LV approximates curvature flow normals, so calculating the discrete Laplace equation: $LV' = 0$ which removes normal components of the vertices and contracts the mesh geometry into a new set of vertex positions V' (2).

$$L_{ij} = \begin{cases} w_{ij} = \cot \alpha_{ij} + \cot \beta_{ij} & \text{if } (i, j) \in E \\ \sum_k (i, k) \hat{E}^{-w_{ik}} & \text{if } i=j \\ 0 & \end{cases} \quad (1)$$

where: E is the set of edges of the mesh and the α_{ij} and β_{ij} are the opposite angles corresponding to the edges (i, j) .

Weights w_{ij} can be chosen as cotangent functions. Mesh contraction is performed by iteratively solved the linear system of vertex positions by using (2).

$$\begin{bmatrix} W_L L \\ W_H \end{bmatrix} V' = \begin{bmatrix} 0 \\ W_H V \end{bmatrix} \quad (2)$$

where L is $n \times n$ the Laplacian matrix and V' is a contracted vertex $^W L$ and $^W H$ are the diagonal weighting matrices that are used to balance constraints of the contraction and attraction process. The problem of balancing the constraints is solved through minimizing the quadratic energy function that is proposed in [26]. The iteration process updated after each iteration automatically. The iterative contraction process works as follows :

- 1) Solve $\begin{bmatrix} W_L^t L^t \\ W_H^t \end{bmatrix} V^{t+1} = \begin{bmatrix} 0 \\ W_H^t V^t \end{bmatrix}$
 - 2) Update $W_L^{t+1} = s_L W_L^t$ and $W_H^{t+1} = W_H^t \sqrt{A_i^0 / A_i^t}$ where s_L is the Laplacian contraction Scalar (for increasing contraction weight in each iteration). A_i^0 and A_i^t are the original and current areas of adjacent faces for the vertex i respectively, and ' t ' represent the iteration number.
 - 3) Computing the new Laplacian operator L^{t+1} by using (1).
- In our experiment use $W_L = 3.0$, $W_H = 1.0$ and $s_L = 2.0$, which contracted the mesh models.

The iteration process converges when the volume of the mesh model is approximately equal to zero. The contracted model is not 1D skeleton of that mesh. The contracted mesh as seen in Fig. 2(B).

B. Extraction of Curve-Skeleton and Refinement

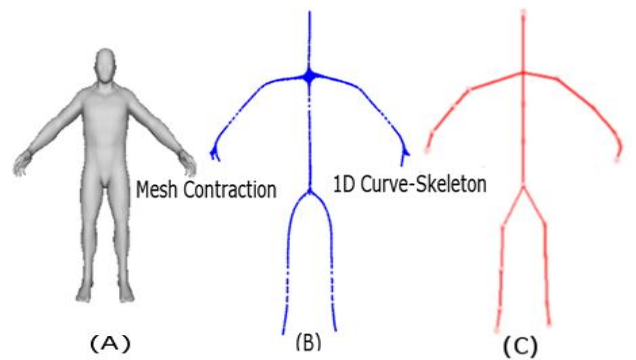


Fig. 2. Mesh contraction and extraction of 1D curve-skeleton.

The human model in Fig. 2(B) is a 1D shape of the contract mesh of Fig. 2(A) mesh. The connectivity of the contracted mesh is still with the original mesh. To convert the contracted mesh into a 1D curve-skeleton, an edge-collapse operation has been applied. The 1D curve-skeleton is computed through the edge collapse by using the proposed method of [27]. The edges have been collapsed with minimum Euclidean length to its midpoint, and triangles incident to the edge are removed. This process is repeated, until all triangles have been collapsed and computed 1D curve skeleton will

appear as in Fig. 2(C). The extracted curve-skeleton refined through mesh-skeleton mapping. Skeleton mesh mapping is relocated skeleton nodes to the centre of its corresponding mesh segment.

V. AUTOMATIC GENERATION OF JOINT-BASED SKELETON

The process of generating animated skeleton of the mesh based on the derived curve-skeleton is described in section. The human character's skeleton is a pose-able framework of bones and joints. These bones are connected to moveable joints which are arranged in a tree data structure. To generate a joint-based skeleton a script (code) is executed. This script calculates root joint, child joints, joint offsets, makes hierarchy of the joints. Fig. 3(left) illustrates the Hip (root joint), child joints and the graph of skeleton hierarchy.

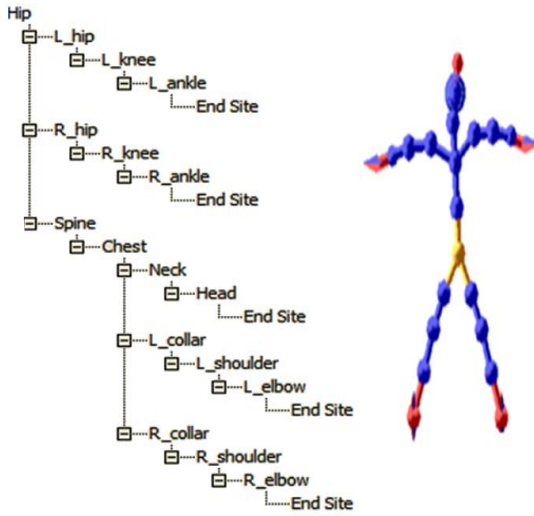


Fig. 3. Animated skeleton with its joints hierarchy.

In kinematic skeleton, joint offsets represent the positions and directions of the child joints to its parent joint. In automatic joint-based skeleton, the offset of the left hip (L_Hip) joint by using derived nodes of the left Hip (x_2, y_2, z_2) from its root, Hip (x_1, y_1, z_1) is computed as:

$$L_Hip_{(offset)} = (x_1 - x_2, y_1 - y_2, z_1 - z_2)$$

The offsets of all other joints in joint-based skeleton are calculated by using (3).

$$J_{(offset)} = (x_{i+1} - x_i, y_{i+1} - y_i, z_{i+1} - z_i) \quad (3)$$

Here (J_{offset}) represent the joints offset, ($x_{i+1}, y_{i+1}, z_{i+1}$) is the child segment and (x_i, y_i, z_i) is the parent node of that child segment. In the articulated skeleton root joint has no parent joint so, hip joint is zero offset. This skeleton is composed of different joints such as, Head, Neck, Chest, Left Shoulder, Left elbow, Left wrist, Right Shoulder, Right elbow, Right wrist, Spine, Hip, Left hip, Left knee, Left ankle, Right hip, Right knee and Right ankle. The length of child segment of its relative parent joint has been calculated by using (4).

$$L_i = \sqrt{(X_{j_{i+1}} - X_{j_i})^2 + (Y_{j_{i+1}} - Y_{j_i})^2 + (Z_{j_{i+1}} - Z_{j_i})^2} \quad (4)$$

Here (L_i) is the length of the segment, ($X_{i+1}, Y_{i+1}, Z_{i+1}$) is child offset and (x_i, y_i, z_i) indicating the parent joint of that child. J_i denotes the parent joint of child J_{i+1} joint. The derived 1D curve skeleton of Fig. 2(C) converts into joint-based skeleton automatically. Fig. 3 (right) presents the animated skeleton by using an extracted curve skeleton.

Joint Rotation Adjustment

In the skeleton generation framework, information about the joints, such as its location, direction and rotation around its parent joint have been calculated. The structure of the automatic joint-based skeleton follows BVH skeleton structure. So, adjustment in some bones has been made by using rotational angle limits of BVH skeleton. The adjustment of bones using rotational angles (α, β and γ) of joints along the axis (X, Y and Z) has been made by (5).

$$R' = R(\gamma).R(\beta)R(\alpha) \quad (5)$$

The position of joint segment has calculated from local translation and rotational information of that joint. The joint offsets are demonstrated the translation information of that segment. The rotational information carried out by motion of skeleton joints.

VI. THE KINECT MOTION CAPTURING

To animate a character need a motion of skeleton joints. Motion capturing and reusing of captured motion have been significantly used in computer vision applications. The Kinect has attracted and opened new possibilities of researches in motion capture. In this work first, capture the original coordinated data of human body joints by using Kinect. The skeleton of the body is determined by the Kinect color image and infrared depth data streaming. The Kinect depth sensor is capturing position data of the human body skeleton. OpenNI and NITE libraries are used to record the original human skeleton joint coordinate data from Kinect. The different frames of the Kinect get different skeleton position. First, we determined the fixed bones skeleton from Kinect captured frames because the BVH file has fixed bone skeleton. The length of bones skeleton between joints is calculated as:

$$l_{(p_1, p_2)} = |p_1 p_2| = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (6)$$

Here $p_1 = (x_1, y_1, z_1)$ and $p_2 = (x_2, y_2, z_2)$ are coordinates of two joints and $l(p_1, p_2)$ represent the bone length of the skeleton. The intension of this part is to convert first frame data of Kinect into a fixed bone skeleton. To achieve this by transforming frame data to fit the skeleton. The Kinect captured position of the joint coordinate first transformed at origin(0,0,0). Hip is considered the root of the skeleton L_Hip, R_Hip and Spine joints are the child of Hip(Root). Let the original frames of Hip and Spine

coordinate are $p_0=(x_0, y_0, z_0)$ and $p_1=(x_1, y_1, z_1)$. The transformed coordinate points of Hip and Spine joints are $p'_0=(x'_0, y'_0, z'_0)$ and $p'_1=(x'_1, y'_1, z'_1)$. To calculate the transformed data of p'_0 and p'_1 by using (7).

$$\begin{aligned} x'_1 &= \sqrt{|p'_0 p'_1| / |p_0 p_1|} (x_1 - x_2) + x'_0 \\ y'_1 &= \sqrt{|p'_0 p'_1| / |p_0 p_1|} (y_1 - y_2) + y'_0 \\ z'_1 &= \sqrt{|p'_0 p'_1| / |p_0 p_1|} (z_1 - z_2) + z'_0 \end{aligned} \quad (7)$$

Here $p'_0 p'_1$ is the bone length of fixed skeleton.

Similarly, calculate every joint of the fixed skeleton from first to last frame of Kinect captured data. Three different actions of captured motion to its corresponding skeleton as shown in Fig. 4.



Fig. 4. Three different actions of Kinect captured skeleton.

BVH Conversion

The BVH (Biovision hierarchical data) file consist of two parts: skeleton part and motion data. The skeleton part of BVH described the hierarchy and initial pose of the skeleton. The motion part contains the rotation and translation of skeleton joints. The root of BVH skeleton contains is zero. All other non-root joints has three Euler angle rotation data of joints. Then we need to calculate the rotation matrix and translation matrix of every joint relative to its parent joint. The rotation matrix of the joint along the Z-axis, X-axis and Y-axis are computed by using (5). The offset value of joints described the translation value of every joint according to its parent joints. Let the Left_knee (P_{Lknee}) joint rotation are calculated as:

$$P_{Lknee}(x, y, z) = T_{root} R_{root} T_{Lhip} R_{Lhip} T_{Lknee} R_{Lknee} P_0(x, y, z)$$

Here T_{Root} , T_{Lhip} , T_{Lknee} are the relative positions of

Root, LHip and Lknee joints from its parent joint and R_{Root} , R_{Lhip} , R_{Lknee} are the rotation of the joints.

$P_0(x, y, z)$ is the initial position i.e $[0,0,0,1]^T$. Similarly, we calculated the every joint rotation and translation relative to the parents' joint. The conversion result of of Kinect captured data into a BVH skeleton motion with different poses as shown in Fig. 7.

VII. MOTION RETARGETING AND JOINT MAPPING

The goal of motion retargeting process is to transfer motion data of the Kinect motion to the static mesh skeleton. To transfer a motion into a target skeleton (generated) a mapping strategy of joints has been used. Automatic joints mapping is preserving the correspondence between the motion skeleton and generated skeleton. The motion skeleton and proposed skeleton do not have the same number of joints. Thus, one-to-one joint correspondences of both skeletons have been established. Monzani *et al.* [24] stated that during joints mapping some joints can be ignored. The ignored joints are not a very important contribution in human character animation. In our work also ignored some futile leaf joints from the motion skeleton. Because these joints cannot locate a matching joint in the mesh skeleton. The joint mapping between the motion skeleton and proposed skeleton has illustrated in Fig. 5. The pipeline of motion retargeting as shown in Fig. 6.

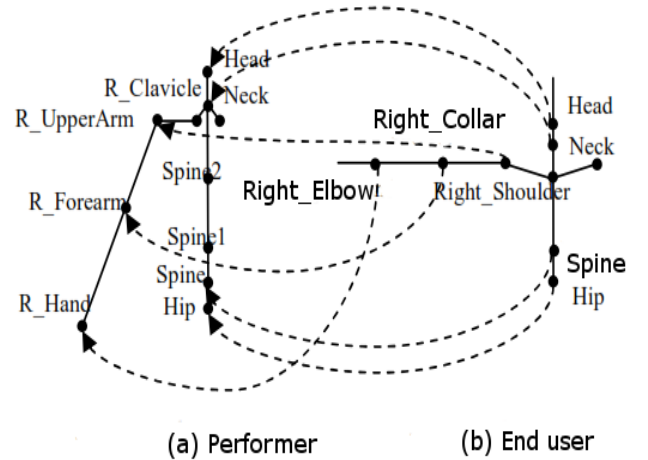


Fig. 5. Joint-to-joint correspondence between the performer(motion) and the end user(mesh) skeleton.

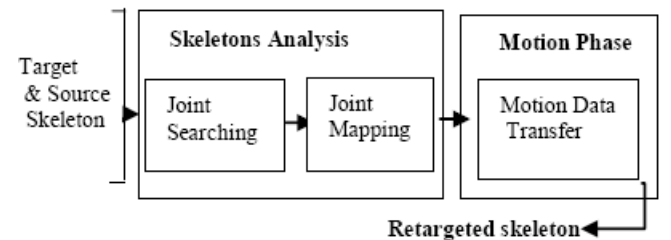


Fig. 6. Pipeline of motion transforming.

The main steps in joint mapping and motion transferring are:

- 1) A recursive depth first search algorithm has been used for analysis of both skeletons.
- 2) The total number of channels in one frame has counted.

- 3) The angular motion of the root joint and child joints have been transferred automatically to target skeleton.

VIII. EXPERIMENTAL RESULTS

To perform experiments three different human mesh models with different poses were used. The geometric details of mesh models that were used in the experiments shown in Table I. The experimental results of mesh contraction, extraction of 1D curve-skeleton and generated joint-based skeleton of a given model is shown in Fig. 7(b), Fig. 7(c) and Fig. 7(d) respectively. Results of Kinect captured motion data of human skeleton converted into a BVH skeleton with associated motion as shown in Fig. 8. The supplementary video, where we show the Kinect human captured skeleton animation and Kinect motion retargeting to mesh skeleton for animation. Although, mesh generated skeleton has used a fewer number of joints, different heights of skeleton as compared to the motion skeleton. Fig. 9 demonstrates the different poses of the generated skeleton animation by retargeting Kinect captured motion through joint mapping. Fig. 9(a) represents the Kinect motion captured skeleton and (b) is the mesh generated skeleton. The animation sequences of the resulting skeleton by applying CMU [26] motion are illustrated in Fig. 10. The results of the skeleton animation by applying different types of motion are encouraging and show that, the generated skeleton has preserved the motion of the joints, and does not require any manually defined tweaking in the skeleton.

TABLE I: MESH MODELS

Models	Number of Vertices	Number of Faces
Model1	4450	8896
Model 2	13336	2668
Model3	5075	10146

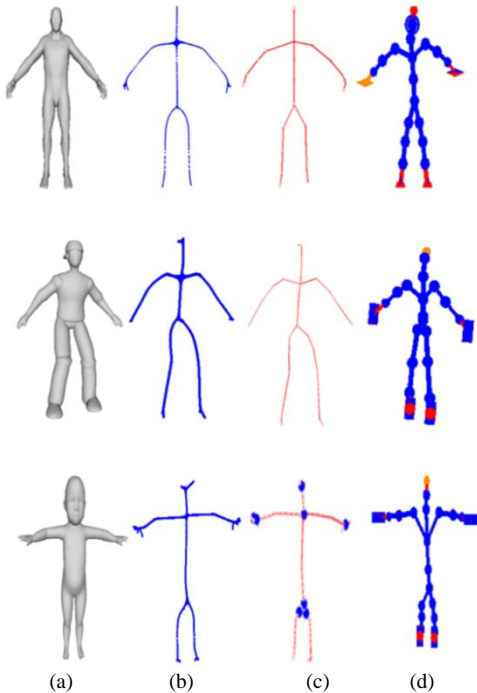


Fig. 7. (a) represents the Kinect motion captured skeleton (b) represent the results of mesh contraction, (c) is the extraction of the refined curve-skeleton and (d) is the generated skeleton of mesh models.

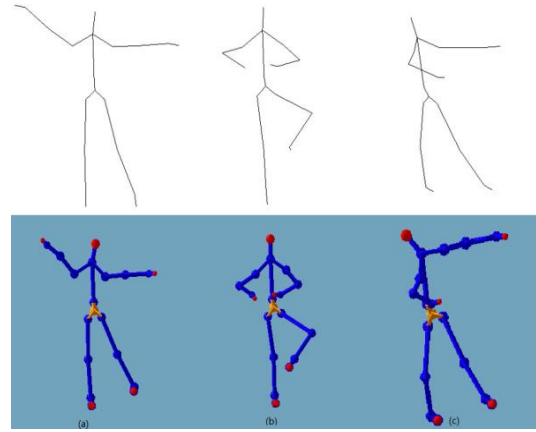


Fig. 8. Result of Kinect motion convert into BVH skeleton motion.

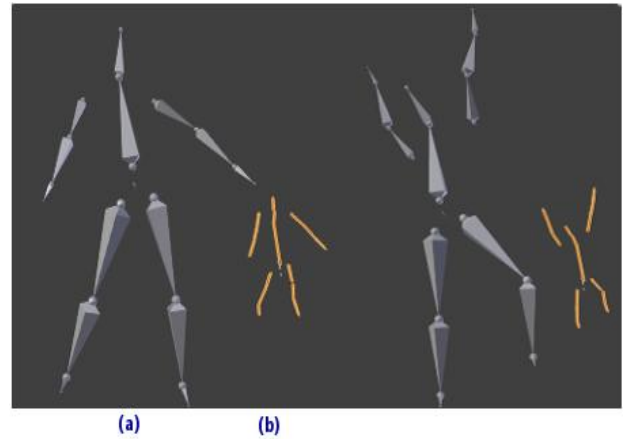


Fig. 9. (a) Kinect Motion. (b) retargeting to mesh generated skeleton.

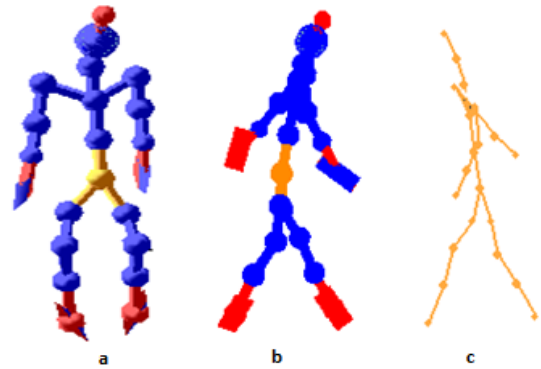


Fig. 10. Animation sequence of mesh skeleton by using motion data of [28].

IX. CONCLUSION

This paper presented a method of an automatic generation of animated skeleton from 3D mesh. The extracted 1D curve skeleton is the simplified shape of the mesh. The nodes of the derived curve-skeleton with mesh vertex correspondence is used to generate a joint-based skeleton automatically. The joint-based skeleton satisfied quality criteria of model animation based on the skeleton. The mesh generated skeleton is animated by retargeting the Microsoft Kinect captured motion. Recording of skeleton motion of human by using Kinect has reduced the cost of motion capturing. The shape of animated skeleton is closely resembles to animator designed skeleton. The generated skeleton also has the ability for preserving the motion of joints.

Furthermore, this work is capable to transfer motion data from the source to a customize target skeleton. The resulting

skeleton of mesh is also suitable to animate by using a variety of motion data of CMU [28].

The proposed work is limited to human mesh. Considerably, more work could be done to determine the accuracy of the proposed method using other mesh models. Further work will concentrate for automatic decomposition, deformation, rigging and skinning of 3D mesh models using the generated skeleton.

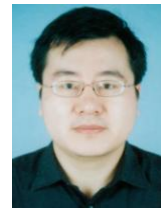
REFERENCES

- [1] B. Llaya and J. Popović, "Automatic rigging and animation of 3D characters," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, pp. 72, 2007.
- [2] J.-T. Li and G.-D. Lu, "Skeleton driven animation based on implicit skinning," *Computers & Graphics*, vol. 35, no. 5, pp. 945-954, 2011.
- [3] L.-B. Liu, K. Yin, B. Wang, and B. Guo, "Simulation and control of skeleton-driven soft body characters," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, 2013.
- [4] O. K.-C. Au, C. L. Tai, H. K. Chu, D. C. Or, and T. Y. Lee, "Skeleton extraction by mesh contraction," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 1-10, 2008.
- [5] M. Jaehwan and S. Choi, "Kinematic skeleton extraction from 3D articulated models," *Computer-Aided Design*, vol. 46, pp. 221-226, 2014.
- [6] X.-T. Li, T. W. Woon, T. S. Tan, and Z. Huang, "Decomposing polygon meshes for interactive applications," in *Proc. 2001 Symposium on Interactive 3D graphics*, pp. 35-42.
- [7] L.-Y. Yu, Q. Han, and X. Niu, "An improved contraction-base method for mesh skeleton extraction," *Multimedia Tools and Applications*, pp. 1-14, 2013.
- [8] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 104, no. 2, pp. 90-126, 2006.
- [9] S. Andrei, T. Lewiner, A. Shamir, and L. Kobbelt, "On-the-fly Curve-skeleton Computation for 3D Shapes," *Computer Graphics Forum*, vol. 26, no. 3, pp. 323-328, Blackwell Publishing Ltd, 2007.
- [10] N. D. Cornea, M. F. Demirci, D. Silver, A. Shokoufandeh, S. J. Dickinson, and P. B. Kantor, "3D object retrieval using many-to-many matching of curve skeletons," in *Proc. International Conference on Shape Modeling and Applications*, 2005, pp. 366-371.
- [11] N. D. Cornea, D. Silver, and P. Min, "Curve-skeleton properties, applications and algorithms," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 3, pp. 530-548, 2007.
- [12] W. F. Che, W. C. Ma, R. H. Liang, B. Y. Chen, and M. O. Young, "Domain connected graph: the skeleton of a closed 3D shape for animation," *The Visual Computer*, vol. 22, no. 2, pp. 117-135, 2006.
- [13] T. Julien, J. P. Vandeboorde, and M. Daoudi, "3D mesh skeleton extraction using topological and geometrical analyses," in *Proc. 14th Pacific Conference on Computer Graphics and Applications*, 2006.
- [14] S. Fang and H. Chen, "Hardware accelerated voxelization," *Computers & Graphics*, vol. 24, pp. 433-442, 2000.
- [15] W. Y. Shuen and T. Y. Lee, "Curve-skeleton extraction using iterative least squares optimization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 4, pp. 926-936, 2008.
- [16] B. Gaurav, T. Thormählen, H. P. Seidel, and C. Theobalt, "Automatically rigging multi-component characters," *Computer Graphics Forum*, vol. 31, pp. 755-764, Blackwell Publishing Ltd, 2012.
- [17] H. M. Kai, B. Y. Chen, and M. O. Young, "Motion retargeting and transition in different articulated figures," in *Proc. IEEE Ninth International Conference on Computer Aided Design and Computer Graphics*, 2005.
- [18] M. Maddock and S. Maddock, "Motion capture file formats explained," Department of Computer Science, University of Sheffield, 2001.
- [19] Microsoft Kinect. (2010). Kinect for Microsoft windows and xbox 360. [Online]. Available: <http://www.xbox.com/kinect>.
- [20] T. Jing, J. Zhou, L. Liu, Z. Pan, and H. Yan, "Scanning 3d full human bodies using kinects," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 4, pp. 643-650, 2012.
- [21] S. Jamie, T. S. A. Kipman, A. Fitzgibbon, M. Finocchio, A. B. M. Cook, and R. Moore, "R. Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116-124, 2013.
- [22] H. Sehoon, Y. Bai, and C. K. Liu, "Human motion reconstruction from force sensors," in *Proc. the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2011, pp. 129-138.

- [23] G. Michael, "Retargeting motion to new characters," in *Proc. the 25th Annual Conference on Computer Graphics and Interactive Techniques*, 1998, pp. 33-42.
- [24] M. J. S. Bastien, P. Baerlocher, R. Boulic, and D. Thalmann, "Using an intermediate skeleton and inverse kinematics for motion retargeting," *Computer Graphics Forum*, vol. 19, no. 3, pp. 11-19, 2000.
- [25] D. Mathieu, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *Proc. 1999 the 26th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 317-324.
- [26] S. Olga and D. C. Or, "Least-squares meshes," *Shape Modeling Applications*, 2004.
- [27] T. Andrea, H. Zhang, and D. C. Or, "Curve skeleton extraction from incomplete point cloud," *ACM Transactions on Graphics*, vol. 28, no. 3, pp. 1-9, 2009.
- [28] The CMU Graphics Lab Motion Capture Database. [Online]. Available: <http://mocap.cs.cmu.edu/>



Abdul Razzaq received his M.S degree in computer science from the Islamia University of Bahawalpur, Pakistan. He received M.Sc degree in computer science from the University of Agriculture, Faisalabad, Pakistan. He is currently a PhD candidate in the Institute of Virtual Reality and Visualization, Beijing Normal University, Beijing, China. His research interests include computer graphics, 3D human motion, and computer animation.



Zhongke Wu is a full professor and PhD student supervisor in College of Information Science and Technology, Beijing Normal University (BNU), China. Prior to joining in BNU, he worked in Nanyang Technological University (NTU), Singapore, Institute National de Recherche en Informatique et en Automatique (INRIA) in France, Institute of High Performance Computing (IHPC) and Institute of Software, Chinese Academy of Sciences in China from 1995 to 2006. Prof. Wu's current research interests include computer graphics, animation, virtual reality, geometric modelling, volume graphics and medical imaging.



Mingquan Zhou is a professor and doctoral supervisor at the College of Information Science and Technology, Beijing Normal University, and he is also the dean of this college. His research interests are computer graphic and virtual reality.



Sajid Ali received the M.Sc degree in computer science from the University of Agriculture, Faisalabad, Pakistan. He is also a faculty member in the Department of Computer Science at University of Education, Lahore, Pakistan. He received his PhD degree from Institute of Virtual Reality and Visualization Technology, Beijing Normal University, Beijing, China. Dr. Ali research interests are computer graphics, 3D human motion, surveillance system and computer animation.



Khalid Iqbal received his PhD degree in applied computer technology from University of Science and Technology Beijing in 2014, the B.Sc. and MS(CS) degrees from University of the Punjab, Lahore and SZABIST Karachi respectively. He was awarded a fully funded scholarship by Chinese Scholarship Council for the entire duration of his PhD studies. He also won the excellent researcher from University of Science and Technology Beijing. He is currently an assistant professor in the Department of Computer Science, COMSATS Institute of Information Technology, Attock Campus. He has worked on Bayesian network application for privacy preserving of XML association rules and text localization in scene images. His research work has been published in several international conference proceedings and journals. His research interests include pattern recognition, machine learning and data mining. he is the recipient of the CSC scholarship and QCRI/Boeing Travel grant. He is also the member of IACSIT.