A Real-time Virtual Dressing System with RGB-D Camera

Mingliang Chen[1], Weiyao Lin[1*], Bing Zhou[2]

[1]Department of Electronic Engineering, Shanghai Jiao Tong University, China ([*]Corresponding author)
[2]School of Information Engineering, Zhengzhou University, China

***Abstract*—As RGB-D cameras become cheaper commodities, many human-computer interaction applications are facilitated, such as human-machine interface (HMI) and virtual reality (VR). In this paper, we focus on virtual dressing, which is a newly heated application of human-computer interaction, and develop a novel framework for real-time virtual dressing. We made three major contributions in our framework: 1) We introduce an effective algorithm to pre-process and improve the skeleton data from an RGB-D camera; 2) We develop a strategy to measure the skeleton motion parameters (scaling, translation and rotation) and apply a two-step deforming scheme specific to create dress deformation results. 3) We introduce hand-integration module to integrate the clothing models and user arms, so as to guarantee user arms being properly displayed in the visual dressing result. Experimental results show that our proposed virtual dressing system achieves satisfactory visual dressing results.[1]**

I. Introduction and Related Work

As RGB-D cameras becomes more easily achievable in recent years, they have been applied in many multimedia -related applications. Specifically, due to the depth information provided by RGB-D cameras, many human-computer interaction applications are facilitated, such as gesture and posture recognition in 3D and virtual reality [1].

Virtual dressing is a newly heated application of human-computer interaction [2]. Basically, virtual dressing aims to create a result where a user takes on a 'visual' clothing. For example, in Fig. 1a, when a user in the left figure stands in front of a camera, the visual dressing system should create a result as he/she visually wares a computer-created 3D coat model (cf. the right figure in Fig. 1a). Although virtual dressing has promising applications, it is still challenging due to the high requirement on dressing results. More specifically, there exists two major challenges in visual dressing: 1) skeleton extraction & skeleton motion estimation; 2) properly integration of real-scene objects (e.g., user arms) and visual clothing models.

(a) (b) (c)

Fig. 1 (a) An illustration of a virtual dressing system. (b) The result when directly using an existing skeleton extraction method [1] to create visual dressing result. (c) The result when the integration of real-scene object and visual clothing model is not properly considered. (The unsatisfactory results are highlighted by white circles).

[1] This paper is supported in part by NSFC grants 61471235 and 61379079.

First, accurate skeleton extraction & skeleton motion estimation is important to make the computer-created clothing model fit user body. With the appearance of RGB-D cameras, many advanced skeleton extraction methods have been proposed [3-5] which create satisfactory skeleton extraction results. However, since most of them focus on the overall accuracy of the entire skeleton, they still have limitations when applied in visual dressing systems. For example, in Fig. 1b, although the left-arm skeleton from an existing skeleton extraction method [5] is correctly located in a user's left arm, the visual dressing result is not satisfactory (i.e., the left arm of the coat deviated from the use's real left arm, as indicated by the white circles), since the extracted skeleton is not perfectly aligned with the user's real body point.

Second, selecting suitable deformation methods to deform a 3D clothing model is also important. Basically, there are two major types of deformation methods. One is skeleton-driven deformation methods [6-7] which deform a 3D model based on the change of skeleton. The other type is exterior cage-driven deformation methods [8-9] which deform a 3D model by measuring the changes of exterior cages. In our paper, we focus on skeleton-driven methods since they are more coherent with our skeleton extraction results. However, the existing skeleton-driven methods still have their limitations. For example, the Linear Blending Skinning (LBS) method cannot avoid the collapse artifacts in the vicinity of bending region of the model, while the Dual Quaternion Skinning (DQS) methods [6] has high computation complexity and has trouble with model scaling.

Third, it is also non-trivial to properly integrating real-scene objects and visual clothing models. If this issue is not properly considered, the real body part maybe unsuitably occluded by the visual clothing model (cf. Fig. 1c). In order to solve this problem, some Collision detection methods [10] can be applied which determines the relative location among objects can derives potential collisions. However, these methods are time-consuming which is not suitable for real-time visual dressing applications.

In this paper, we develop a novel framework for real-time virtual dressing. The major contributions of our framework is summarized in the following.

(a) We develop an effective algorithm to pre-process and improve the skeleton results from the existing skeleton extraction results. The improved skeleton results can obviously enhance the fitting accuracy between real-scene body and visual clothing model.

(b) We develop a strategy to measure the skeleton motion parameters (scaling, translation and rotation) and apply a two-step deforming scheme specific to create dress deformation

results. The proposed two-step deformation method can suitably combine the advantages of LBS and DQS deformation methods and creates satisfying clothing deformation results in real-time.

(c) We introduce a simple but effective arm-integration module to integrate the clothing models and user arms. With this module, user arms are guaranteed to be properly displayed in the visual dressing result.

The rest of this paper is organized as follows. Section II describes the framework of the virtual dressing system. Section III, IV, V describe details of the above three contributions. Section VI shows the experimental results and Section VII concludes the paper.

II. Framework

The framework of our system is shown in Fig. 2. As shown in the figure, the framework can be divided into four main modules. First, we extract color frame, depth frame and body frame from a RGB-D camera, where the body frame can be extracted by the existing skeleton extraction methods [5]. Then, a skeleton refinement is applied to correct the location of body joints in the original body frame, such as the elbows, wrists, and hands. With the improved skeleton data, we estimate the motion of each joint through kinematic chain of body skeleton. In this step, we only constrain the motion of key bones in a body to simplify the estimation. After the estimation of the motion parameters (scaling, translation and rotation), we introduce a two-step deformation method to deform the clothing model, aiming to make the layout of clothes fit with the pose of a user. In the meantime, we extract arm regions and present them in front of clothes models. Moreover, we adapt our algorithm under different circumstances, such as long-sleeves and short-sleeves clothes. This can improve the fitting experience when the user tries different kinds of clothes.

In our framework, the modules of 'skeleton refinement', 'bone motion estimation & model deformation', and 'hands & model integration' are the key parts. In the following, we will focus on discussing this modules.

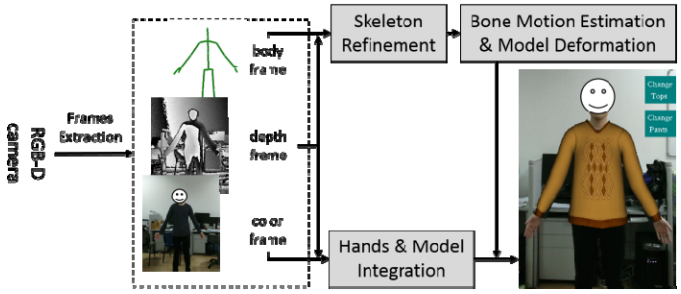


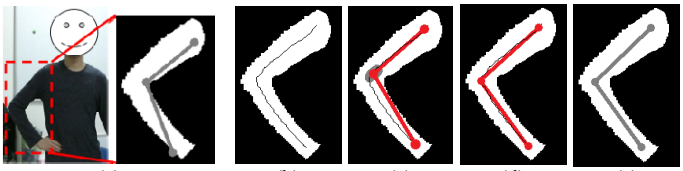
Fig. 2 The framework of the virtual dressing system.

III. Skeleton Refinement

In this section, we discuss an effective approach to improve the skeleton data extracted from the original skeleton extraction results [1]. In order to ease the discussion, we use arm as an example to illustrate the details in this method. Note that this method can be applied in other body points.

A. Arm Skeleton Extraction

RGB-D cameras provide depth frame, from which we can get the distance between the point and the camera. According to the depth-axis information of skeleton data extracted from the RGB-D camera, the depth frame can be easily binarized by thresholds to carve out arm region. After getting the binary image, we use morphological filtering method to alleviate noise interference. The right figure in Fig. 3a illustrates the binarized arm region result for the user in the left figure in Fig. 3a, where the region in white indicates the extracted arm region.



(a) (b) (c) (d) (e)

Fig. 3 The graphical illustration. (a) Arm region extraction (gray points in the right figure indicate shoulder, elbow, and wrist joints from the original skeleton extraction results [1]). (b) The one-pixel-width skeleton of the arm region. (c) Update the joint points (red points indicate updated joints). (d) Update the turning point on the skeleton line. (e) The refined skeleton joints

After extracting arm regions, we apply thinning or skeletonization [11] to the binary image to extract a one-pixel-width skeleton of the arm. The resulting one-pixel-width skeleton line is shown in Fig. 3b.

B. Skeleton Points Location Refinement

In order to make arm skeleton points perfectly align with a user's real body point, skeleton points should be close to the center of an arm region. In this paper, we utilize the one-pixel-width skeleton line extracted in Section IIIA as a reference line of arm centers, and refine skeleton points by finding their nearest points on the skeleton line (cf. the red points in Fig. 3c).

However, the above process still has problem. For example, the updated elbow point is not placed on the turning point of the skeleton line when the arm bends (cf. Fig. 3c). To solve this problem, we propose an algorithm to move the joint point to the turning point. We argue that the turning point of a skeleton line should satisfy the following criteria: 1) the point is on the skeleton line; 2) the fold lines connected by the skeleton points (the red lines in Fig. 3d) should be close to one-pixel-width skeleton line (cf. black line in Fig. 3d). In this paper, we use the sum of distance between points from fold lines and the one-pixel-width skeleton line to measure their closeness. Moreover, in order to shorten the search time, the solution candidates are among a neighboring region of the updated elbow point (cf. the points in the gray region in Fig. 3c). The entire algorithm is presented in Algorithm 1.

Algorithm 1 Turning point refinement (use elbow for example)

Input: $\{J_i\}_{i=1}^{N}$ points among the vicinity of the updated elbow joint, $\{P_i\}_{i=1}^{K}$ points on the skeleton line, B_1 the previous joint of elbow, B_2 the previous joint of elbow,
Output: turning point T
Initialization: $minDist \leftarrow$ Inf.
for $i \leftarrow 1$ **to** N **do**
  $dist \leftarrow 0$
  $l \leftarrow$ the fold line connected by B_1, J_i and B_2
  for $j \leftarrow 1$ **to** K **do**
    $dist \leftarrow dist +$ distance between P_j and l
  end for
  if $dist < minDist$
    $minDist = dist$
    $T = J_i$
  end if
end for

Fig. 3e shows the revised joint points after skeleton point location refinement. Compared with the original skeleton points in Fig. 3a, the location of joint points is obviously improved.

IV. Model Deformation

After skeleton extraction, we are able to perform deformation on the clothing model to make it fit with the pose of a user. In order to properly deform a clothing model, we first estimate the motion parameters of a user, and then apply a two-step deformation to deform a clothing model.

A. Estimation of Motion Parameters

To describe the transformation of a rigid object (e.g., a bone between two points in a skeleton), we need to estimate three types of motion patterns: scaling, translation and rotation. In our framework, we estimate scaling parameters by the length change a bone in a skeleton, estimate translation parameters by the displacement of one end of the bone (in fact, only the translation of the root bone need to be estimated), and estimate rotation parameters in the axial direction and the radial direction. In this paper, we focus on discussing the deformation of coat models for a user's upper body. Thus, we estimate the above parameters via kinematic chain structure connected by the upper-body skeleton points, as shown in Fig. 4(a).

Moreover, since it is hard to estimation the rotation parameters of a bone in the radial axial direction, in order to simplify the task, we estimate rotation of the spine bone (bone0 in Fig. 4a) in the axial direction, and estimate rotation of other bones in the radial direction. Fig. 4b-c illustrate this process.

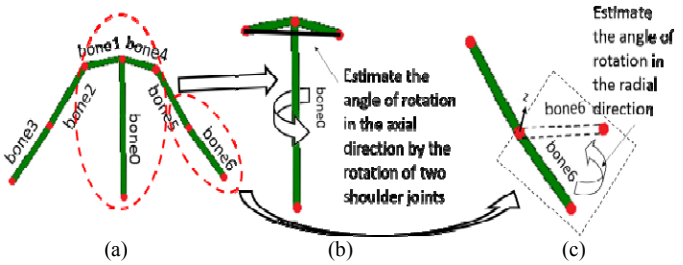


Fig. 4 (a) The kinematic chain structure of the joints of the upper body. (b)-(c): Different strategies of Estimation of the rotation for different bones: (b) Estimation for the spine bone; (c) Estimation for the other bones.

B. Two-step Deformation

Before the process of clothing model deformation, we need to assign a weight for each 3D point on the clothing model according to the kinematic chain structure of a skeleton. In this work, we use heat equilibrium algorithm [12] to find the weights. Fortunately, this set of weights between a model and a skeleton can be shared in two steps of the model deformation.

To make the clothing model fit a user's body size, the model should be scaled at first. And then, the translation and rotation is applied to the model. In this paper, we utilized LBS and DQS methods to perform clothing model deformation. However, as mentioned, both methods have their disadvantages. Therefore, we proposed a two-step deformation scheme whose first step deals with the scaling of the model and second step deals with the rigid transformation of the model (i.e. translation and rotation).

In the first step, the scaling parameter of the whole skeleton is estimated by the length ratio of a specific bone in a user's skeleton and its corresponding bone in the clothing model. In this paper, we use the spine bone (bone0 in Fig. 4a) as the key bone to derive scale variation. After we apply scaling to the clothing model, the size of the scaled cloth can be updated to have similar size as the user's overall skeleton (e.g., the left figure in Fig. 5 (b)).

However, simply scaling the entire clothing model is not enough. For example, the length of local parts in the clothing model may still be inconsistent with the user, such as arms. Therefore, we further estimate the scaling parameter by comparing the length ratio of the corresponding bones in a user's skeleton and the clothing model. Note that in this step, we only scaling local bones in the axial direction to make them coherent with other parts in the body. Fig. 5b shows an example. From the left figure to right figure in Fig. 5b, the arm of a clothing model is scaled from 0.5 unit length to 0.6 unit length. After that, LBS method is applied to deform the model based on the scaling parameters (cf. the right figure in Fig. 5b).

In the second step, we further deform the pose of a clothing model by DQS method (cf. Fig. 5c). Since LBS method has low computation complexity and good performance in scale deformation, while DQS method can properly avoid collapse artifacts around bending bones. By combing both DQS and LBS methods, clothing models can be properly deformed in real-time.

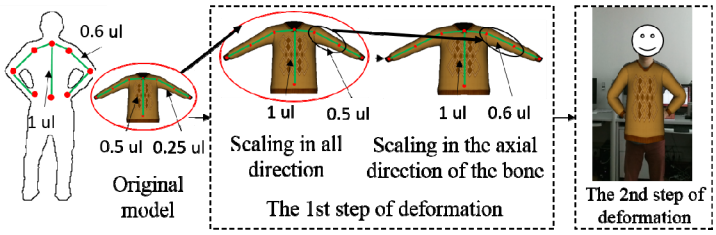


Fig. 5 The graphical illustration of the two-step deformation algorithm. (a) Body skeleton and clothing model. (b) The first deformation step which scales the clothing model according to user's body skeleton size. (c) The second deformation step which deforms the pose of the clothing model, so as to make it fit a user's pose. (ul means unit length)

V. Arm and Clothing Model Integration

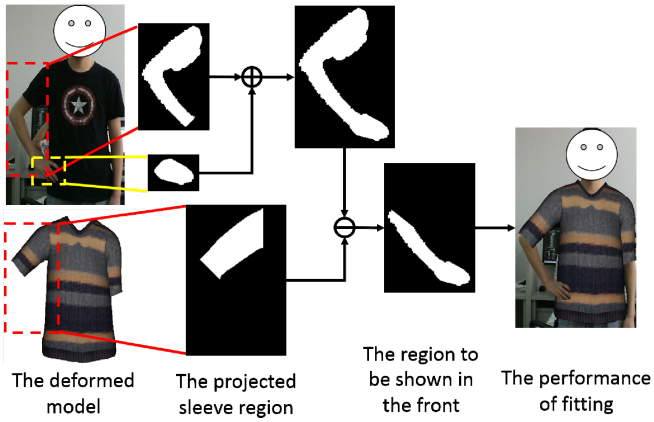


Fig. 6 The graphical illustration of the integration process.

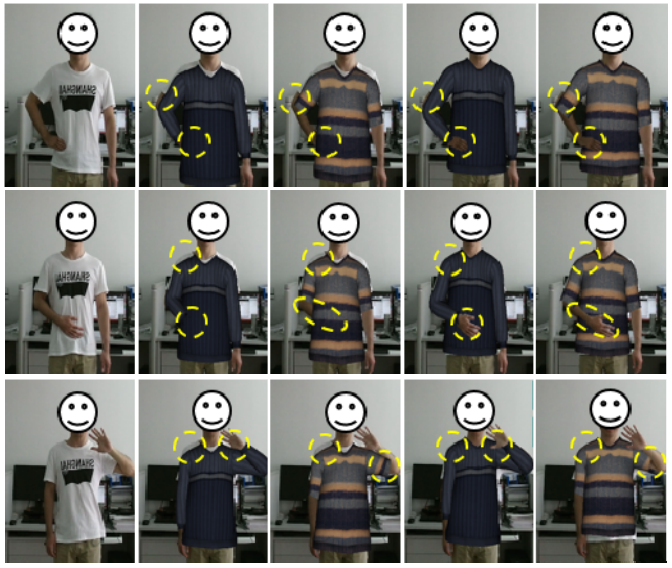
Finally, we also introduce an arm-integration module to guarantee user arms being properly displayed in the visual

dressing result. An example of the process is shown in Fig. 6.

In Fig. 6, we first apply thresholding on depth frame to carve out the whole arm, and use skin color model [13] in color frame to carve out the hand. We combine these two results to get the whole arm and hand extraction. Then, we project the sleeve part of the deformed clothing model (achieved in Section IV) onto the 2D image plane. After subtracting the projected sleeve part region from the extracted arm & hand region, we can achieve the parts that should be displayed in front of the clothing model.

VI. Experimental Results

In this section, we show experimental results of our system. The experiment is implemented on a PC equipped with 4-core CPU and 2G RAM.



(a) (b) (c)

Fig. 7 The experimental results of our work. (a) A users' action. (b) The results of "raw-skeleton+two step deform" method. (c) The results of our framework. (The yellow circles show the details under comparison.)

Some experiment results are shown in Fig. 7. In Fig. 7, visual clothing results for three postures with two different kinds of clothes are displayed. For each posture, we show the visual dressing results of the "raw-skeleton+two step deform" method (i.e., using the original skeleton data [1] and our two-step deformation to create results) and our method (which includes skeleton refinement, two step deformation, & arm integration).

From Fig. 7b, we can observe the problems of the "raw-skeleton+two step deform" method. For example, the clothing models cannot fully cover use's arm (cf. white circles in left figure of Fig. 7b), the clothing models occlude hands or arms of the user (cf. white circles in left figure of Fig. 7b).

Comparatively, in Fig. 7c, the above problems are properly solved. For example, the non-covering region of the body in the background is much less, especially on the arm. Also, our framework solves the hand integration problem and properly show hands and arms when they are supposed to be shown in the reality. Moreover, our work is adapted to different kinds of clothes, such as long-sleeves and short-sleeves. The circles in Fig. 7b-c show detailed differences of visual dressing results between the compared methods.

In order to further demonstrate the effectiveness of our framework, we further perform a user test. Similar to [14], we select 4 different clothes, including long-sleeves and short-sleeves and let 10 users trying visual dressing results. We ask these users to rate their experience about the system performance within the range of 1–5, where 1 indicates very poor quality and 5 indicates very good quality. Table 1 compares the user study results.

From Table 1, we can see that our framework can create more satisfying results than the "raw-skeleton+two step deform" method, and achieves a score of 3.35, which is above the "acceptable quality" level.

Finally, Table 2 shows the average time spent on each module. According to Table 2, our system spends 31.3ms on average to process one frame. It satisfies the requirement for real-timing and can perform visual dressing over a frame rate of 20 fps (frequency per frame).

Table 1 The user study results

Clothes	1	2	3	4	Average
Original	2.3	2.5	2.7	2.1	2.4
Enhanced	3.5	3.3	3.5	3.1	3.35

Table 2 The average time spent on each module

Module	Frame extraction & pre-processing	Skeleton Refinement	Model Deformation	Arm Integration	Total
Average time (ms)	2.1	12.9	5.8	10.5	**31.3**

VII. Conclusion

In this paper, we build a real-time system for virtual dressing. We introduce: 1) an effective algorithm to improve the skeleton data from an RGB-D camera; 2) a strategy to measure the skeleton motion parameters (scaling, translation and rotation) and apply a two-step deforming scheme specific to create dress deformation results, 3) a hand-integration module to integrate the clothing models and user arms. Experimental results demonstrate the effectiveness of our framework.

References

[1] X. Guo, L. Feng and H. Liu, "Research on the Virtual Reality Simulation Engine", *IEEE Conf. on Intelligent Systems and Applications*, 2010.

[2] X. Yang and G. Chen, "Human-Computer Interaction Design in Product Design", *IEEE Conf. on Education Technology and Computer Science*, 2009.

[3] M. Ye, X. Wang, R. Yang, L. Ren and M. Pollefeys, "Accurate 3D Pose Estimation from a Single Depth Image", *ICCV*, 2011.

[4] M. Sun, P. Kohli and J. Shotton, "Conditional Regression Forests for Human Pose Estimation", *CVPR*, 2012.

[5] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman and A. Blake, "Real-Time Human Pose Recognition in Parts from a Single Depth Image", *CVPR*, 2011.

[6] L. Kavan and S. Collins, J. Zara and C. O'Sullivan, "Geometric Skinning with Approximate Dual Quaternion Blending", *ACM Trans. on Graphics,* vol. 27(4), pp. 105-127, 2008.

[7] L. Kavan and J. Zara, "Spherical blend skinning: A Real-time Deformation of Articulated Models", *ACM SIGGRAPH SI3D*, 2005.

[8] T. Ju, S. Schaefer and J. Warren, "Mean value coordinates for closed triangular meshes", *ACM Trans. on Graphics,* vol. 24(3), pp. 561-566, 2008.

[9] Y. Lipman, D. Levin and D. Cohen-Or, "Green Coordinates", *ACM Trans. on Graphics*, vol. 27(3), 2008.

[10] M. Tang, D. Manocha and R. Tong, "Fast Continuous Collision Detection Using Deforming Non-penetration Filters", *ACM SIGGRAPH SI3D*, 2010.

[11] L. Lam, S. Lee, and C. Y. Suen, "Thinning Methodologies-A Comprehensive Survey," *PAMI*, vol. 14(9), pp. 869-885, 1992.

[12] I. Baran and J. Popovic, "Automatic Rigging and Animation of 3D Characters", *ACM Trans. on Graphics*, vol. 26(3), 2007.

[13] M. Yang and N. Ahuja, "Detecting human faces in color images", *ICIP*, 1998.

[14] Z. Liu, C. Zhang and Z. Zhang, "Learning-Based Perceptual Image Quality Improvement for Video Conferencing", *ICME*, 2007.