A Real Time Virtual Dressing Room Application using Kinect

F. Isikdogan and G. Kara

Abstract—We introduce a virtual dressing room application using the Microsoft Kinect sensor. We extract the user from the video stream by using depth and user label data provided by the Kinect sensor, register the cloth model with the Kinect skeletal tracking data, and detect skin to adjust the order of layers. We report an average percentage of overlap between the user and the cloth models of 83.97%.

Index Terms—Augmented Reality, Human-Computer Interaction, Kinect

Fig. 1. The user interface of the application.

1 Introduction

Trying clothes in clothing stores is usually a time-consuming activity. Moreover, it might not even be possible to try on clothes in the store, such as when ordering clothes online. Here we propose a simple virtual dressing room application to make shopping for clothing faster, easier, and more accessible.

The first problem we address in the design of our application is the accurate superimposition of the user and virtual cloth models. Detection and skeletal tracking of a user in a video stream can be implemented in several ways. For example, Kjaerside et al. [1] proposed a tag-based augmented reality dressing room, which required sticking visual tags for motion tracking. More recently, Shotton et al. [2] have developed a real-time human pose recognition system that predicts the 3D positions of body joints, using a single depth image without visual tags. In this project, we use Shotton et al.'s method and a Microsoft Kinect sensor to create a tagless, real-time augmented reality dressing room application.

Microsoft Kinect has become a popular depth image sensor in the market after its launch in 2010. Developer tools, such as the ones included in the OpenNI framework and the Microsoft Kinect SDK, ease developing applications based on the Kinect sensor. We used the Kinect SDK as it includes a robust real-time skeletal body tracker based on [2]. Our approach can be summarized as follows:

We first extract the user from the video stream by using depth and user label data provided by the Kinect sensor. Then, we register the cloth model with the Kinect skeletal tracking data. Finally, we detect skin to adjust the order of layers, as shown in a screenshot of the application in Fig. 1.

2 Methods

2.1 Extraction of the User

We extract and isolate the user from the background to create an augmented reality environment. To segment the foreground, we use the depth images and user labels that are provided by the Kinect sensor as shown in Fig. 2.

We detect skin color and bring it to the front layer to allow the user to fold arms or hold hands in front of the cloth model. We threshold the image in YCbCr color space, using the values that were found to be effective for skin-color segmentation [3], as follows:

$$\begin{aligned} Y &= 0.299R + 0.587G + 0.114B < 70 \\ 77 < C_b &= 128 - 0.169R - 0.332G + 0.5B < 127 \\ 133 < C_r &= 128 + 0.5R - 0.419G - 0.081B < 173. \end{aligned} \quad (1)$$

To prevent any background pixel from being labeled as skin, we apply the threshold only on the segmented foreground (Fig. 3).

2.2 Tracking

The skeletal tracker estimates the spatial coordinates and depth of body joints. We use nine body joints (Fig. 4) to fit a cloth model to the skeletal model of the user. We smooth the spatial coordinates over time to reduce flickers and vibrations on the joints. We compute the angle between the joints to set the angle of rotation

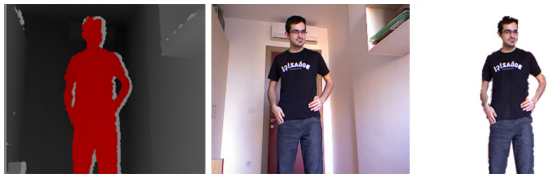

Fig. 2. Background removal: depth image (left), color image (middle), extracted user image (right).

Fig. 3. Skin color segmentation: user image (left), segmented skin colored areas (right).

of the corresponding parts of the cloth model. For example, arms of a t-shirt are rotated by

$$\theta = \text{atan2}(y_{elbow} - y_{shoulder}, x_{elbow} - x_{shoulder}). \quad (2)$$

To scale the cloth model, we make use of the distance between joints and the distance of the user from the sensor. We first define a distance based scaling factor as the ratio of the size of the cloth model when the user is one meter distant from the sensor $\mathbf{s_{model}}$ to the distance of the spine of the user to the sensor z_{spine}:

$$DS = \frac{\mathbf{s_{model}}}{z_{spine}}. \quad (3)$$

Then, we define shape based scaling factors based on the Euclidean distance between the joints. For instance, the width and height of the body and arms of a t-shirt are computed as follows:

$$\begin{aligned} H^k_{arm} &= \sqrt{(x^k_{shoulder} - x^k_{elbow})^2 + (y^k_{shoulder} - y^k_{elbow})^2} \\ W^k_{arm} &= H^k_{arm} \times \alpha \\ H_{body} &= |x_{shouldercenter} - x_{hipcenter}| \\ W_{body} &= |x^{left}_{shoulder} - x^{right}_{shoulder}| \end{aligned} \quad (4)$$

where x and y are the 2D spatial coordinates of the joints, α is a constant width to length ratio, and $k = \{left, right\}$.

Finally, we define an overall scaling function S as a weighted average of depth and shape based scaling

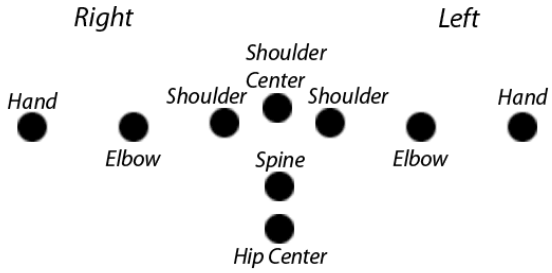

Fig. 4. Body joints that are used for registering the cloth model.

Fig. 5. A set of poses that are used in the evaluation of the performance.

factors as:

$$\begin{aligned} S_x &= \frac{w_1 W_x + w_2 DS_x}{w_1 + w_2} \\ S_y &= \frac{w_1 W_y + w_2 DS_y}{w_1 + w_2} \end{aligned} \quad (5)$$

where w_1 and w_2 are the weights for shape and depth based scaling factors. In our experiments we set $w_1 = 1$ and $w_2 = 3$.

3 Experiments

We evaluated the performance of the application on a set of 12 poses with different angles of rotation and distance from the sensor (Fig. 5). We measured the performance as the amount of overlap between the constructed cloth model and manually labeled ground truth data as

$$P = \frac{A_c \cap A_g}{A_c \cup A_g} \quad (6)$$

where A_c is the area of the constructed model and A_g is the area of the ground truth model in terms of the number of pixels.

We observed an average overlap of 83.97 or higher between the ground truth and the computed models, within a rotation range of $0-45°$. Rotations along the

TABLE 1
Experimental Results

Arm Rotation	$\mathbf{0^\circ}$	$\mathbf{45^\circ}$	$\mathbf{90^\circ}$
Performance	89.80%	90.54%	76.64%
Body Rotation	$\mathbf{-45^\circ}$	$\mathbf{0^\circ}$	$\mathbf{45^\circ}$
Performance	83.68%	90.19%	88.84%
Horizontal Rotation	$\mathbf{-45^\circ}$	$\mathbf{0^\circ}$	$\mathbf{45^\circ}$
Performance	74.88%	87.74%	77.87%
Distance From Sensor	$\mathbf{1.5m}$	$\mathbf{2m}$	$\mathbf{3m}$
Performance	86.64%	89.87%	70.98%
Average Overlap	83.97%		

vertical axis dropped the performance as the fitting is performed in only 2-dimensions. The best results were obtained when the distance from the sensor was 2 meters. The results are summarized in Table 1.

4 Conclusions and Future Work

We developed a real-time virtual dressing room application that requires no visual tags. We tested our application under different conditions. Our experiments showed that the application performs well for regular postures. The application can be further improved towards creating more realistic models by using 3D cloth models and a physics engine.

References

[1] K. Kjærside, K. J. Kortbek, H. Hedegaard, and K. Grønbæk, "Ardresscode: augmented dressing room with tag-based motion tracking and real-time clothes simulation," in *Proceedings of the Central European Multimedia and Virtual Reality Conference*, 2005.

[2] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[3] D. Chai and K. N. Ngan, "Face segmentation using skin-color map in videophone applications," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 9, no. 4, pp. 551–564, 1999.