CGI2013 manuscript No.
(will be inserted by the editor)

Cloth and Avatar Preparation for Virtual Fitting Rooms

Preprocessing approach for real-time virtual fitting room with physics simulation

Umut Gultepe · Ugur Gudukbay

Abstract This paper presents a novel approach to adjusting a virtual cloth and avatar with respect to a specific user for a virtual fitting room framework. Proposed method scales the cloth and the avatar accordingly with the subjects body dimensions, also prepares the physics simulation and collision detection environment, with a total preprocessing time of one second, eliminating the requirement for long preprocessing times to achieve a realistic fitting experience.

Keywords Virtual Fitting Room · Depth Sensor · Character Animation

1 Introduction

Apparel industry is one of the biggest on the planet, estimated to be $2,560 trillion worldwide in 2010 [14]. It is also the second biggest and fastest growing e-commerce sector [15]. One of the most time-consuming stages of apparel shopping is trying the apparels on, which is not even possible in online stores. With the advances in the augmented reality technologies, virtual fitting rooms are slowly taking their places in both real and virtual stores [16, 17] to imrpove the quality of apparel trying experience while also making it faster.

Advanced virtual fitting rooms show the apparel items either on the video of the user or on a virtual avatar, both scaled to reflect the user's body characteristics [18]. Some of them also employ physics based garment simulation for a better fitting experience[17].

This paper presents a new approach to adjusting a virtual clothing item and a virtual avatar for a specific subject on the fly. Constraining the overall preprocessing time to 1 second, the clothing piece and the avatar is adjusted automatically and prepared for a physics powered virtual fitting room framework.

2 Previous Works

Virtual fitting rooms have been a research subject for more than a decade. Protopsaltou [9] has developed an internet based approach for virtual fitting rooms, although it was not real time and required marker based motion capture systems for animation. Zhang [10] used a multi-camera system utilizing SFS [11] techniques to build a real time intelligent fitting room.

Advances in time-of-flight technology made depth sensors available at consumer-level prices with better performance. This prompted a wave of research based on depth sensors in various fields, such as Rehabilitation [8], indoor modeling [5] and medicine [6]. Another topic which attracted significant attention from both researchers and companies is real-time virtual fitting rooms. Giovanni [7] developed a virtual try-on system utilizing a calibrated set of Kinect and an HD Camera, while comparing the two state of the art depth sensing SDKs- OpenNI[4] and Kinect for Windows SDK[3].

One problem every researcher encountered during their studies with depth sensors is the feeble quality and noisiness of the depth stream. This problem is analyzed in depth by Khoshelham [12] and concluded that standard deviation reaches 2cm in a measuring distance of 3m. Matyunin [13] attempted to improve the quality by filtering with additional information from the attached RGB camera.

Bilkent University
Ankara

Bilkent University
Ankara

3 Methodology

Objective is to acquire a set of simulation parameters from a human test-subject for a pre-modeled clothing mesh, which is to be displayed on a virtual avatar reflecting the body characteristics of the aforementioned subject. The set of parameters for the simulation include the body height and width, also the radii for the collision spheres which have their centers coinciding with the joints of the virtual avatar's skeleton. Body width and height are then utilized to estimate the body size of the user, collision spheres are used in the dressing room simulation, to collide with the cloth particles.

3.1 Depth Map Optimization

The state-of the art time-of-flight cameras still provide low resolution and quality output compared to current advanced RGB systems. The quality of the input depth map is a crucial factor on the overall performance of the system, therefore we first wish to improve the quality of the depth map by applying canonical image optimization methods.

In this approach, we assume utilization of a time-of-flight camera running with a middleware which also provides a subject map, which has the same size as and denotes the origin of the pixels of the depth map, either belonging to a subject or to the background.

Let us take the input depth map D as a MxN matrix. Initially, the user pixels from D are extracted by a pixel-by-pixel comparison with the input user map. We are only interested in the one subject and D_1 represents the depth pixels of him, whereas U_1 is the bit map of the subject. Also, the non-subject pixels are set to the mean value of the user pixels, to set the matrix properly for the subsequent optimizations.

$$D_1 = (D-(D\times U_1))\times 1/n \times \sum_{i=0}^{n}((D\times U_1)_i + d\times U_1) \quad (1)$$

The subject depth map is now prepared to be processed with Gaussian filtering, to normalize and improve the quality.

$$D_G = D_1 * G \quad (2)$$

Gaussian filtering completes the optimization of the input depth map. The overall algorithm is given in Algorithm 1.

Algorithm 1: Depth Map Optimization

```
Input: Raw Depth and Subject Stream From TOF
       Camera
Output: Optimized Subject Map
$depth_{sum} = 0$ ;
$n_{user} = 0$;
for $i$ from 0 to $d_width$ do
    for $j$ from 0 to $d_height$ do
        if $U(i,j)$ then
            $depth_{sum} = depth_{sum} + D(i,j)$;
            $n_{user} + = 1$;
$depth_{average} = depth_{sum}/n_{user}$ ;
for $i$ from 0 to $d_width$ do
    for $j$ from 0 to $d_height$ do
        if not $U(i,j)$ then
            $D(i,j) = depth_{average}$;
for $i$ from 0 to $d_width$ do
    for $j$ from 0 to $d_height$ do
        if $U(i,j)$ then
            $D(i,j) = D(i-m : i+m, j-n :$
            $j+m) * Gaussian(m,n,e)$;
return $D$
```

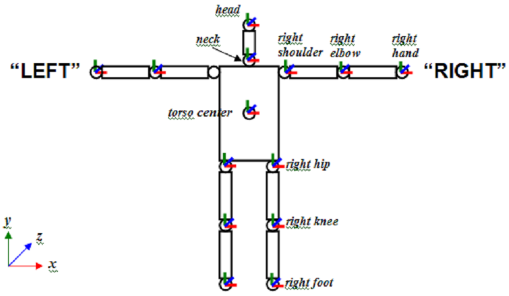

Fig. 1 Human Joints provided by NITE [1]

3.2 Body Measurement

By now, we have an optimized depth map, which is ready for performing key body dimension measurements. The key dimensions are handled in two groups, the collision sphere radii, which are used for the collision detection in the simulation; and height and width parameters, which are used to determine to size of the apparel.

3.2.1 Collision Sphere Radii

The simulation framework utilizes collision spheres and capsules instead of arbitrary geometries. This constraint allows the simulation to run in real time with exceptionally high frame rates by simplifying the algorithms. There are a total of 15 joint locations provided by the middleware as seen in Figure 1, which are key points for collision sphere placement. For this reason, the framework also utilizes 15 collision spheres and the capsules formed by pairs.

For a realistic simulation, collision spheres must be as large as possible without intersecting the skin mesh of the avatar. The optimal sphere fitting algorithm to satisfy this requirement is the straight forward approach to start with an infinitely small circle and expand it discretely until it intersects with the body contour:

1. Take vector J_i which represents the coordinates of the i^{th} joint. Initialize the radius of the sphere by setting it to the z-distance with the overlapping point in the depth map.

$$r_i^z = J_i^z - D^z(J_i^x, J_i^y) \tag{3}$$

2. Start with an infinitely small line segment parallel with x-axis. Expand it until it intersects with the body contour. Take the x-distance between the intersection point and the joint location. Repeat the same process with a line segment parallel with y-axis. Take the bigger radius. While expanding the segment, stop expanding and discard the corresponding result if the end of depth map is reach.

$$r_i^{x,y} = max(\| \pm J_i^{x,y} \mp D^{x,y}(J_i^{y,x}, J_i^z)\|) \tag{4}$$

3. Take the minimum of three-axis differences, as there should be no intersection with the body contour and the shape must be a sphere.

$$r_i = min(r_i^{x,y,z}) \tag{5}$$

The process pseudocode is given in Algorithm 2.

Algorithm 2: Sphere Fitting Algorithm

```
Input: Optimized Depth Stream From Kinect
Output: Collision Sphere radii for each joint
foreach joint do
    $p = pos_{J_m}$;
    $r_z = \sqrt{P_z^2 - D_z(P_x, P_y)^2}$ for $i$ from $P_x$ to 0 do
        if $D(i, P_y)$ equals $P_z$ then
            $r_x^- = i$;
            break;
    for $i$ from $P_x$ to $depth_width$ do
        if $D(i, P_y)$ equals $P_z$ then
            $r_x^+ = i$;
            break;
    for $j$ from $P_y$ to 0 do
        if $D(P_x, j)$ equals $P_z$ then
            $r_y^- = j$;
            break;
    for $j$ from $P_y$ to $depth_height$ do
        if $D(P_x, j)$ equals $P_z$ then
            $r_y^+ = j$;
            break;
    $r_m = min(r_z, r_x^-, r_x^+, r_y^-, r_y^+)$
return $(r_0, r_1 \dots r_n)$
```

Distance	**Width**	**Height**	**Measure Source**
Head	1w (1)	1h (2)	Depth Map+Joint Location
Body Height	-	7 (3)	Depth Map
Hip Height	-	4 (4)	Joint Location
Elbow-Fingertip	-	2 (5)	Depth Map+Joint Location
Wrist to Fingertip	-	1 (6)	Depth Map+Joint Location
Shoulder Width	3 (7)	-	Depth Map+Joint Location
Hip Width	- (8)	-	Depth Map
Torso Height	-	- (9)	Joint Location

Table 1 Human Body Proportions [2]. Numbers in parenthesis represent the lines on Figure 2.

3.2.2 Height and Width Parameters

The width and height of the subject is important for determining the proper actual size for the cloth. However, a straightforward estimation of body height and shoulder width is prone to errors due to the noise and quality of the incoming depth map. In order to minimize the error factor, a larger of the subject's body are measured, to be used in width-height estimation later. This set of dimensions are listed in Table 1. For the sake of relative representation, head width and head height are taken as unit width and height respectively. The Measure Source column in the table denotes the source for the estimation of respective parameter:

– Joint location represents that the measurement will take the input subject joint locations as the reference.
– Depth map represents the measurement will instead perform measurements based on the pixel distribution in the optimized subject depth map.

They are often used together for better performance. Please note that some of these dimensions are not standard enough to be used as relative references, i.e. hip width, these parameters do not effect others in estimation process, and vica versa.

Measurements will be performed in real-world space rather than projective space as a real size estimation is crucial for determining the appropriate cloth size. Although the latter would be sufficient enough for simulation purposes only, when there is no concern of real-life apparel fitting. After the acquisition of the required scaling parameters for the subject, whole cloth and avatar

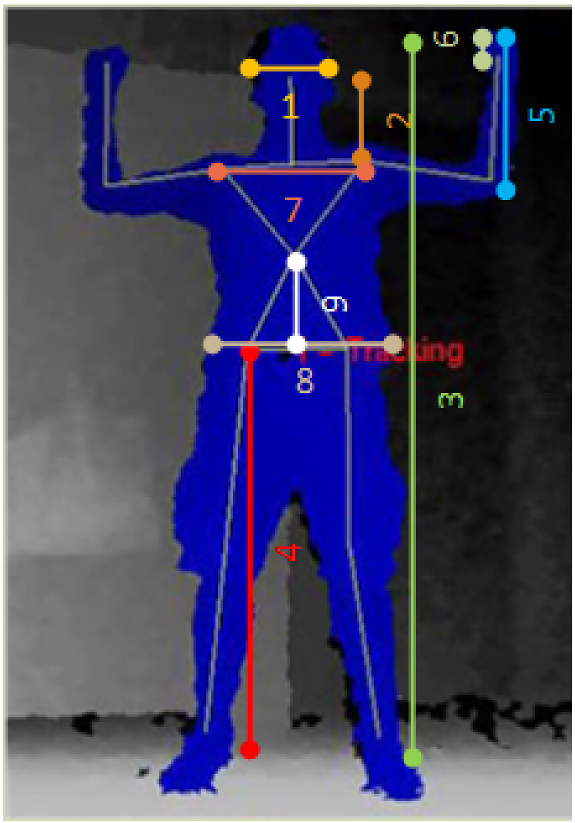

Fig. 2 Proportions on the Body

should be scaled in three dimensions uniformly, in contrast with a segmented scaling. This decision is based on the scope of this work, as it is a standard-sized apparel fitting application, without extensive customization. Following the measurements, the body width and shoulder height are estimated as following:

1. Take the primary dimension (either body height or shoulder width) P_i^0. This process will be repeated for width (W) and height (H).
2. Using the remaining measurements in the set, estimate the primary dimension P_i as P_i^j using proportion R_i^j from 1.

$$W, H_i^j = W, H_j \times R_i^j \qquad (6)$$

3. Find the optimized primary dimension as the mean of all estimations:

$$W, H_i = 1/(n+1) \times \sum_{j=0}^{n} W, H_i^j \qquad (7)$$

3.3 Temporal Optimization

By now, we have acquired the required body dimensions and collision sphere parameters for a realistic simulation.

Algorithm 3: Body Dimension Estimation

```
$t_{proportion} = import(Table1)$ ;
$t_{primary} = w_{shoulder}, h_{body}$ ;
$ct = cloth_{type}$;
$width_{main} = t_{proportion}.width(ct)$;
$width_{sum} = 0$;
$count_{effector} = 0$;
foreach $width$ in $t_{proportion}$ do
    $w_i = measure(p_i)$;
    $w_i^j = w \times t_{proportion}.ratio(p_i, parameter_{main})$;
    $width_{sum} = width_{sum} + w_i^j$;
    $count_e ffector + +$;
$width_{weighted} = width_{sum}/count_e ffector$
$x_s = width_{weighted}/width_{cloth}$;
$height_{main} = t_{proportion}.height(ct)$;
$height_{sum} = 0$;
$count_{effector} = 0$;
foreach $height$ in $t_{proportion}$ do
    $h_i = measure(p_i)$;
    $h_i^j = h \times t_{proportion}.ratio(p_i, parameter_{main})$;
    $height_{sum} = height_{sum} + h_i^j$;
    $count_e ffector + +$;
$height_{weighted} = height_{sum}/count_e ffector$
$y_s = height_{weighted}/height_{cloth}$;
return $(x_s, y_s)$
```

Yet, the measurements are performed on a filtered version of a depth sensor with high error rates. In order to overcome the noise and overall depth-sense faults, the prior measurements are repeated for the duration of 1 second, which corresponds to 30 frames of input depth map. A considerable different approach here would be to employ the temporal optimization on the depth map instead of the measured parameters. We realized the results suffer due to the motions of the subject, as most subjects failed to keep their exact form for one seconds.

Temporal averaging consists of collecting the specified parameters for each frame in one second and taking the mean. This step finalizes the parameters and delivers the required parameters for simulation environment creation.

Algorithm 4: Temporal Averaging

```
Input: Raw Depth Stream From Kinect
Output: Depth Stream With Patched Holes and
        Gaussian Optimization
$s =$
$2 \times 30 Arrayforxandyscalingparametersfor30frames$
;
$r = 16 \times 30 Arrayforjointradiifor30frames$;
for $i$ from 0 to $30frames$ do
    r[i]=fitSpheres();
    s[i]=optimizeScaleParameters();
$r_{final}$=avg(r);
$s_{final}$=avg(s);
```

4 Experiments

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec imperdiet tincidunt nibh quis dictum. Pellentesque eget tellus nunc. Donec semper blandit metus vel rhoncus. Nullam tincidunt enim nibh, suscipit volutpat neque. Ut eget ipsum dolor, ac semper sapien. Nunc nec rutrum nulla. Vivamus sodales auctor orci, vel bibendum leo pretium at. Fusce placerat lectus mi.

5 Conclusion

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec imperdiet tincidunt nibh quis dictum. Pellentesque eget tellus nunc. Donec semper blandit metus vel rhoncus. Nullam tincidunt enim nibh, suscipit volutpat neque. Ut eget ipsum dolor, ac semper sapien. Nunc nec rutrum nulla. Vivamus sodales auctor orci, vel bibendum leo pretium at. Fusce placerat lectus mi.

References

1. PrimeSense, NITE, PrimeSense Natural Interaction (2012)
2. Willis B, Body Proportions in Art, [Online] (2012)
3. Microsoft, Kinect for Windows, [Online] (2013)
4. OpenNI, The standard framework for 3D sensing, [Online] (2013)
5. Peter Henry and Michael Krainin and Evan Herbst and Xiaofeng Ren and Dieter Fox, RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments, The International Journal of Robotics Research , Volume 31, 647-663 (2012)
6. Gallo L, Controller-free exploration of medical image data: Experiencing the Kinect, Computer Based Medical Systems ,Volume 24, (2011)
7. Stevie Giovanni, Yeun Chul Choi, Jay Huang, Khoo Eng Tat, and KangKang Yin, Virtual Try-On Using Kinect and HD Camera, Lecture Notes in Computer Science , Volume 7660, 55-65 , (2012)
8. Yao-Jen Changa and Shu-Fang Chenb and Jun-Da Huang, A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities, Research in Developmental Disabilities, Volume 32, 25662570 (2011)
9. D. Protopsaltou, C. Luible, M. Arevalo-Poizat, N. Magnenat-Thalmann, A body and garment creation method for an Internet based virtual fitting room. Proc. Computer Graphics International 2002 (CGI '02), Springer, pp. 105-122, 2002.
10. Wei Zhang, Takashi Matsumoto, Juan Liu, Maurice Chu, and Bo Begole. An intelligent fitting room using multi-camera perception. In Proceedings of the 13th international conference on Intelligent user interfaces (IUI '08). ACM, New York, NY, USA, 60-69, 2008
11. Kong-man Cheung, Simon Baker, Takeo Kanade,Shape-From-Silhouette Across Time Part II: Applications to Human Modeling and Markerless Motion Tracking,International Journal of Computer Vision ,Volume 63, pp 225-245, 2005
12. Khoshelham, K.; Elberink, S.O. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. Sensors 2012, 12, 1437-1454.
13. Matyunin, S.; Vatolin, D.; Berdnikov, Y.; Smirnov, M.; , "Temporal filtering for depth maps generated by Kinect depth camera," 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011 , vol., no., pp.1-4, 16-18 May 2011
14. Breyer, M, 25 Shocking Fashion Industry Statistics, [Online] (2012)
15. Fredricksen C,Apparel Drives US Retail Ecommerce Sales Growth, [Online] (2012)
16. Fitnect Interactive,Fitnect, [Online] (2012)
17. Styku, LLC, [Online] (2013)
18. FaceCake Marketing Technologies Inc, [Online] (2013)