International Journal of Advance Research in Science and Engineering
Vol. No.10, Issue No. 04, April 2021
www.ijarse.com
IJARSE
ISSN 2319 - 8354

Fashion Data Processing using Machine Learning

[1]Babu U S, [1]Venkateshwaran S, [1]Lakshmanan R, [1]Saran V,

[2]Dr. N.B. Mahesh Kumar

*Associate Professor,

[12]Department of Information Technology, KSR Institute of Engineering and Technology

#E-mail: usbabu300@gmail.com

Abstract

In this paper, a new framework for Fashion design and try-on simulation based on ACGPN is presented. Users can not only gain realistic try-on experience, but also design the Dress all on their own. The design approach of this system consists of three major parts. First, collect relevant information from the camera and identify the position of the clothes. Second, process the retrieved data and modulate the color of the clothes with folds and shadows remained. Third, place built-in or user-designed pictures onto the clothes and simulate their deformation while the user moves arbitrarily. In comparison with existing virtual clothes fitting systems, our system provides the flexibility of designing customized pictures on the Dress with realistic virtual try-on simulation in real-time.

I. INTRODUCTION

In the past, customers could only decide to buy clothes or not on 2D photos of garments when shopping online. Customers could not know whether the clothes fit them well or whether the clothes look good on them until they got it. This may substantially reduce the willingness of customers to purchase apparel online. Therefore, online sales of Apparel & Accessories, a product type for which customers' desire to feel and try on items before making a purchase, was traditionally regarded as a deterrent to online shopping. However, as some improvements such as free returns and innovative visualization tools have been used, online sales of Apparel & Accessories have gradually become a success. In recent years, the Internet has emerged as a compelling channel for sale of apparel. Online Apparel & Accessories sales exceeded $34 billion in 2017, and are expected to reach $73 billion in 2021.

One of the most well -known visualization tools may be virtual try-on systems (or virtual fitting room). Through these systems, customers may have more realistic feel for the details of the garments. Currently, a variety of different kinds of virtual try-on systems is presented, which will be discussed later in Section II. All of them could simulate what we look like as if we are wearing the clothes to some extent. Nevertheless, they all face the same problem that the results are not real enough. Some systems may not fit clothes on users well, and some may not react to users' motion in real time. In order to solve these problems, we present, Try-On System.

Different from the existing methods, it comes up with some brand-new ideas. It not only retrieves users'body information, but also records many useful clothes information, such as the lightness of the cloth. Through the combination of these data, it therefore can vividly simulate a real Dress with folds, shadows and deformable pictures on it. The virtual try-on results will also response to user's body motion in real-time.

The remaining part of this paper is organized as follow: related work is discussed in Section II. In Section III we present the system architecture. Then, in Section IV, we introduce the core techniques of it. The results are shown in Section V. Finally, Section VI concludes.

II. RELATED WORKS

In this section, we will discuss some of the recent works and implementation methods on virtual try-on systems (VTS). Zugara first introduced the Webcam Social Shopper, a web cam-based VTS. Using marker detection techniques, it detects the marker on a paper to adjust the size and position of the clothes image. JC Penny partnered with Seventeen.com also presented a VTS by attaching the clothes images on a pre-set silhouette on the screen. User can try-on clothes by placing her/himself in the right position. Both of the above two methods require only the camera on a computer to realize a VTS. However, in these cases clothes cannot change its position with user motion. In addition, clothes attached on the screen are 2-D images, so the users cannot fit the clothes in 3-D augmented reality.

Some designs allow virtual clothes to fit on the user when one is moving. Pereira et al. presented a webcam VTS based-on upper body detection. Garcia et al. also proposed a VTS using head detection to fit the clothes. Later version of the Webcam Social Shopper developed by Zugara also adopted head tracking to better place the virtual clothes on the user. Though body tracking techniques enable placing the clothes according to user's position in real time, the lack of 2-D information makes it difficult to create a virtual fitting system with 2-D augmented reality effect. These VTS design can only place a 2-D image on the user's body.

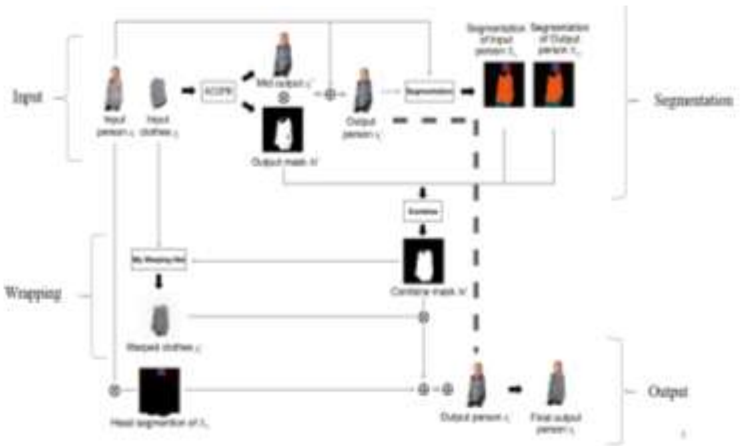
As body modelling techniques keep improving, more virtual try-on systems using robotic modelling or 2-D scanning model are proposed. Fits.me applied the robotic science to create a VTS on the website. Once the user inputs a series of measurements of the body, the try-on results for clothes of different sizes on a robotic mannequin are shown. Instead of creating a real robotic mannequin for virtual try-on, both Styku and Body metrics adopt 2-D body scanning techniques using Microsoft Kinect or ASUS Xtion. Thescanner creates a 2D virtual model of the user and allows one to analyse the fitness of the try-on clothes results through a color map. Using sophisticated robotic models or accurately measured scanning models produces realistic try-on results. As a trade-off of the accuracy, those systems cannot present try-on results with body motions in real-time.

With a depth and motion capture device, such as Microsoft Kinect or ASUS Xtion, virtual try-on systems that enable real-time clothes fitting can be designed more easily. In addition, various computer graphic algorithms for clothes modelling have been proposed in the past. Coming both the motion sensing and garment modelling techniques, Swivel, AR-Door, Fitnect, and Fitting Reality all presented their different implementations of VTS. Pachoulakis et al. reviewed some of the recent examples of virtual fitting room and their implementation concepts in. Those try-on systems first capture the user's body motion and then superimposed 2-D clothes models on the user's image. For this reason, although the systems enable virtual try-on in real-time, sometimes the virtual clothes images may not precisely fit the user. Besides, the only clothes that a user can try-on are limited to the garment model database. It is difficult for a user without computer graphic experience to design one's own wearing clothes.

In this paper, without using a depth and motion capture device, we utilize ACGPN board and its camera module, combining both image processing and ACGPN hardware implementation techniques to design try-on. This system enables real-time virtual try-on with all the folds and shadows on the shirt vividly preserved. Instead of superimposing garment models on the user image, try-on precisely modulate the user's clothes image to present the effect of virtual try-on. In addition, user can easily design pictures to put on the shirt by one's own drawing. The picture can also move and deform as the Dress is pulled or folded. Try-on is a new solution for realistic virtual try-on and Dress design in real-time.

III. SYSTEM ARCHITECTURE

In this section, we are going to introduce the design architecture of our system. The bellow is the data flow diagram, which provides a simple view of how our system functions. Diagram is the system block diagram, in which the relationships between modules are shown. These modules can be roughly classified into four parts: Image Retrieval, Memory Control, Image Processing and Detection, and Input/output Interface. A brief introduction of each part is as follow:



A. Image Retrieving

A 5-Mega Pixel Camera is used to capture images. Together with modules *CCD Capture* and *RawtoRGB*, we could get the RGB values of all pixels, which will be the data of the following process.

B. Memory Control

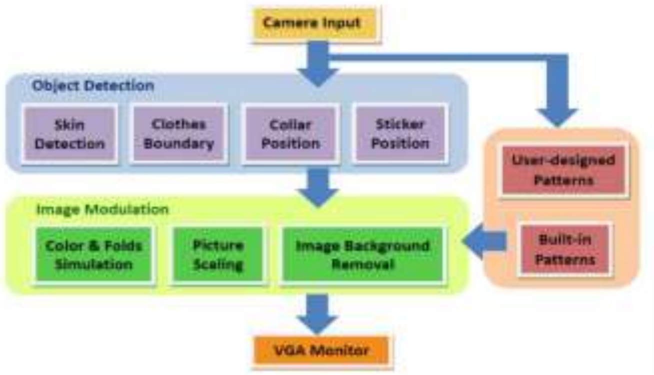
The retrieved images mentioned above are temporarily saved in SDRAM, and the built-in pictures are saved in FLASH memory. In order to achieve real-time processing results on VGA output, SSRAM is used as the buffer memory for data to be processed. Both the built -in pictures and the user-designed pictures will be buffered in SSRAM before processing.

C. Image Processing and Detection

This part can be view as three steps. First, *Clothes Detection* uses skin color to filter out the skin part and identify the boundary of the clothes. Second, after knowing the region of the clothes, *Color Adjusting* changes the clothes' color with folds and shadows retained. Third, *Pattern Adjusting* analyses the customer's motion and adjusts the picture's looks with proper deformation in real-time.

D. Input/Output Interface

Using switches and keys on the ACGPN board, users can control the systems with desired command. The final try-on simulation results will be output through *VGA controller* to the monitor. So, we will use Colab as an IDLE



IV.TECHNIQUES

A. Key Concept

For the existing wearing try-on simulation systems, one of the major problems is that how to make the clothes moves, rotates, and even folds according to our body motions. The traditional way is to modify an existing target clothes image and superimpose it onto the image of the users, such as how Fit nect and VI Podium do. The outline of the target clothes usually does not match the outline of the user body. Even if the image is scaled to the size of the body, there are still many positions that the outlines do not coincide and the clothes seem weird on the users. What's more, the folds and shadows have to be created according to the body motions, which is extremely difficult and requires lots of computation. The clothes eventually seem unreal because the shapes are mismatched and the folds and shadows are disappeared.

However, we do not need to create the folds and shadows actually. The folds and shadows already exist on the original Dress that the users are wearing. Besides, the outline of the original Dress is also perfectly matching the body of the users. Therefore, our key concept is that, instead of modifying the target Dress to fit the body motions, we combine the folds and shadows information from the original Dress with the color and pattern of the target Dress to simulate a real Dress vividly. As a result, the new Dress on the user is perfectly matching and folding according to the body motions.

B. Skin Detection

To identify the original Dress on the users, skin detection is required to filter out the skin part. According to the research of Basilio et al, the color of skin is more explicit in YCbCr color space. Therefore, we transform the color of each pixel to YCbCr color space and determine whether it is skin by the following conditions.
The thresholds are slightly modified because of the different races. The conditions are designed to work well on the Asians. In diagram, the left image is taken from a user and the skin part is detected and colored red in right figure.

C. Clothes Positioning

After the skin is detected, the region of the Dress could be identified as well. The image is scanned row by row from top to bottom. For each line, after filtering out the background and the skin part, the remaining part is treated as the Dress. However, in order to tolerate the inevitable noises and skin on the Dress, we apply the maximum sub-array algorithm. Therefore, the outline of the Dress could be precisely determined even if there is some noise or false detection of skin on the Dress. The pseudo code of the maximum sub-array algorithm is shown as following.

D. Folds and Color Changing

After identifying the region of the T -shirt successfully, we could modify the color according to the target color. As mentioned before, the original color is not simply replaced by the target color. On the contrary, the target color is mixed with the original one by the following formula. The new RGB values of each pixel are calculated by the formula, which considers both the brightness of the shirt image and the target color. Therefore, not only the Dress turns out in a new color, the original folds and shadows are also shown as well.

E. Picture Selection

We provide two ways to choose a pattern of the target Dress. The first method is choosing from our built-in pictures. Each picture is 500 pixels in height and width, and is 0.5 MB in total. Therefore, an 8MB Flash memory could store 16 pictures. In addition, since built-in pictures are stored in the Flash, reading data from flash has delay problems. Therefore, we use the SSRAM as a buffer. After copying the data in Flash to SSRAM once, we could use SSRAM instead of flash to achieve real time performance.

The second way is to use a user -designed picture. One could arbitrarily draw its own T -shirt picture in his/her own way, and then use our camera to take a picture of the drawing. After taking the photo, the image will be processed. First, the white background will be removed so that when changing Dress color, the image will not have the white background. Second, we apply a series of complex memory manipulation on SSRAM, sharing the buffer memory resources with both flash and SDRAM, so that the users could still switch between built -in pictures and user-designed picture easily. Finally, the user could put the user-designed picture on the Dress.

F. Picture Placement

After choosing a picture, one could place the picture by two ways as well. The first method is using collar detection and hand gestures. In this mode, the picture would be place under the collar by default. We could detect the position of the collar by maximum sub-array algorithm as before. If the skin part is less than 100 pixels and the clothes width is larger than 800 pixels, the centre of skin line would be identified as the position of collar. One could also set the position of the picture by gestures. When pointing down with hands, the position of fingers can be detected by a similar method of collar detection. The system will remember the displacement between the position specified by gestures and the collar, so the picture could be placed to the desired place afterward.

In the second method, users could not only move the picture, but also scale the picture to any size simply by 3M stickers. The positions of the stickers could be detected by maximum sub-array algorithm as described before. The top left red stickers specify the position of the top left corner of the picture, and the right green sticker and bottom red sticker specify the right and bottom edge of the picture. The picture is scaled both in width and height accordingly. To overcome the delay of calculation, we use pipelines to maintain real time performance.

V. RESULTS

A. Color Changing

The color of the Dress could be changed to arbitrary color by users. In the demo pictures, Diagram, we showed how it is changed to the primary colors and their combinations. The simulated Dress has the folds and shadows as well. In addition, both hands can move freely in front of the Dress and do not affect the coloring effect at all. In Figure 13, it is shown that even when the user turns back, the Dress is coloured as well.

B. Dress Pictures

After color changing, we can choose the picture to put on the Dress from built-in pictures and user designed pictures.

1) **Built-on Pictures**: The system provides up to 16 built-in pictures for users to place on to the Dress. The pictures are automatically placed on the chest of the users. The following figures show two example pictures. Please note that even the hands are in front of the Dress, the pictures are not affected just like a real Dress. the chest by default. The following figure shows one of the designed Dress on our own.

C. Picture placement

The picture is placed on the chest by default. However, it is convenient to move the picture to the desired position. We provide two ways to move the picture: Using gestures and using stickers.

1) **Placement by Gestures:** This is the simpler way to move the position of the picture. By pointing down the target position, the system will detect and memorize the desired position. After the gesture adjustment, the picture could be placed to the any desired position at any time.

2) **Placement by Stickers:** In this method, the picture position could be conveniently indicated by three 3M stickers. The picture will be scaled and placed in the sticker-specified area. Therefore, it is easy to shrink or enlarge the picture as well, which makes the design of Dress more conveniently. In addition, because that the picture is placed in the stickers, it could move, shrinks, expands, deforms or even rotates with the body motions of the users. All the folds and shadows in the region of the picture will also retain as the picture deforms, which makes the try-on results highly realistic.

VI. CONCLUSION

In this paper, we presented Try-on, a new framework for Dress design and try-on simulation based on ACGPN implementation. Instead of modifying the existing clothes models to fit the body motions, we utilize the information from the original T -shirt image to simulate the virtual try-on results realistically. The color of the Dress could be changed arbitrarily with all the folds and shadows retained. Built -in pictures or user-designed pictures could be placed on the Dress by hand gestures or 3M stickers. When we use stickers to locate the picture, all the folds and shadows on the picture will also retained as the picture deforms with user's body motion. Combining the new design concepts with implementation techniques, try-on provides the flexibility of designing customized pictures on the Dress with highly realistic virtual try-on simulation in real-time.

ACKNOWLEDGMENT

We would like to express our gratitude to our Guide Dr. N.B. Mahesh Kumar, ASP/IT and the researcher's andrew_ng, yann lecun for their blogs on this.

REFERENCES

- Try on ACGPN https://github.com/b01902041/Deep-Virtual-Try-on-with-Clothes-Transform
- Andrew_ng https://github.com/andrewjong/SwapNet
- Yann leCun blogs http://yann.lecun.com/