

Virtual Ornaments and Fabric Try-on Reality Application

¹G.Rajaram, ²B.Anandavenkatesan

¹Student, ²Assistant Professor

¹Embedded systems and Technology, SRM University, Chennai, India.

²Electronics and communication engineering, SRM University, Chennai, India.

¹rajaramgy@gmail.com, ²anandavenkatesan@gmail.com

Abstract— A Real time application, virtual ornaments and dressing room, which is augmented reality the compiling of layered system-generated information to real-time stream data. The proposed image processing design flow for Virtual Fitting Room (VFR) applications, targeting both personal computer and Beaglebone board. This proposed human friendly interface is implemented by a three-stage algorithm: detection and sizing of the user's body, detection of reference points based on face detection and augmented reality markers, and superimposition of the clothing over the user's image. Compared to other existing VFR systems, key difference is the lack of any proprietary hardware components or peripherals. The Proposed VFR is software based and designed to be universally compatible as long as the device has a camera.

Index Terms— Augmented reality, face detection, virtual fitting room (VFR), virtual reality, human friendly interfaces

I. INTRODUCTION

Today, online/offline sales are increasing at a rapid rate. Fast forward to 2014, powerful multicore processors, Gigabytes of memory, lots of I/O, graphics, and networking options have transformed embedded systems into a very different animal. Virtualization is being employed in embedded systems as a means to consolidate complex architectures [1]. Nevertheless, one area online sales traditionally struggled is fashion items and clothing. It is estimated that majority of the consumers don't buy clothing online and also in offline sales, the consumer wants to get try-on the models, because they don't want to take any risk with the sizes. In addition, in online sales, a large percentage of the purchased items are returned and in offline sales, they need to be wait for a long time in queue to use trial room, especially in function/festival occasion. This brings an additional financial burden to retail companies. BeagleBone is a very popular computer about the size of a deck of cards. These so-called "tinyware" devices contain USB ports, onboard Ethernet, and audio/visual output capabilities. Therefore, the objective of this work is to develop a virtual fitting room (VFR) application that can run on embedded device that has a camera and network connection. This VFR application can enhance the way customers shop offline and help them to choose the correct type and size of the clothing/ornament item. The proposed algorithm is designed to be computationally efficient and it can be used with Personal Computer and Beagle Bone, improving the way users shop offline for new clothes.

In the next section, we first discuss the existing approaches for virtual fitting room applications. Section III presents the detection and sizing of the user's body. In Section IV, we present a face detection method and augmented reality markers for determining the reference points.

II. BACKGROUND

Several commercial products exist for VFR implementation. Styku [3] presents a body scanner that creates a complete 3D model of the user. This 3D model is then used in other webpages to try the clothing items on. The model can be rotated, it can match any size and it even uses a color map to analyze the fit. The body scanning is implemented using Microsoft's Kinect and Asus' Xtion devices. A VFR implementation by JCPteen [4] gets an image of the user and using adobe flash player displays the clothing items. At the beginning, it shows a shadow on the screen where users have to fit themselves and after that the cloth is displayed. In this system if the user is moving, the item won't follow or track him. Zugara [5] offers a VFR that is similar to the JCPteens since the items don't move once they are displayed. It is based on the augmented reality concept.

The key difference in our approach is the lack of any proprietary hardware components or peripherals. Proposed VFR is software based (C++/Java) and designed to be universally compatible as long as the device has a camera. Additionally, proposed algorithm can track and resize the clothing according to user's spatial position.

In order to create the app, we have developed a human-friendly interface [11][12][13] which is defined as an interactive computing system providing the user an easier way to communicate with the machines. In particular, this can be achieved through touch-screen operations and gestures similar to what people naturally feel with their five senses. Creating intuitive interfaces with a few buttons that illustrate the basic functionality to the user is paramount for the wider acceptance of the virtual reality applications. This was one of the key objectives of this study.

III. DETECTING AND SIZING THE BODY

First step of the proposed VFR method is the acquisition of the shape of the body to get reference points. Reference points are then used to determine where to display the clothes. In order to obtain the body shape, we applied several techniques: i) Filtering with thresholding, Canny edge detection, K-means, and ii) Motion detection or skeleton detection wherein multiple frames were analysed for any movement. However, the results were unreliable and not good enough to obtain reference points for displaying cloths.

Therefore, we introduced a new detection methodology based on locating the face of the user, adjusting a reference point at his/her neck and displaying the clothes based on that point. In addition, another point of reference can be obtained by using an Augmented Reality (AR) marker. Details of this algorithm are explained in Section IV.

deltax	deltay	8-code	corresp 4-code
0	1	0	0
-1	1	1	
-1	0	2	1
-1	-1	3	
0	-1	4	2
1	-1	5	
1	0	6	3
1	1	7	

Figure 1 - Freeman's codification

For obtaining the size of the user, we follow a similar automated body feature extraction technique as shown in [14]. The idea is to set up the user in front of the camera and hold him at the beginning at a certain predetermined distance. The algorithm extracts points on the shoulders and the belly. Measuring the distance between these points and knowing the distance from the user to the camera, the size of the user can be obtained. When the image (video frame) is acquired, a canny edge detection filter is applied to obtain only the silhouette of the body. Canny edge detection is really susceptible to noise that is present in unprocessed data; therefore it uses a filter where the raw image is convolved with a Gaussian filter. After convolution, four filters are applied to detect horizontal, vertical and diagonal edges in the processed image. Morphological functions are also applied to obtain a closed silhouette. Finally, an 8-points Freeman chain code, shown in Figure 1 is applied to assign a direction to each pixel.

We can choose to apply 8 or 4 chain code, then the following formula can be used:

$$z = 4 * (\text{deltax} + 2) + (\text{deltay} + 2) \quad (1)$$

which gives the sequence corresponding to rows 1-8 in the preceding table: $z = \{11, 7, 6, 5, 9, 13, 14, 15\}$. These values can be used as indices into the table, improving the speed of computing the chain code. Each variation between consecutive numbers represents a variation of 45° so if the difference of direction between consecutive points is measured and if the change is more than two (90°) then a feature point is detected and marked in the image.

$$ek = |dj + 1 - dj| = 2 \quad (2)$$

This is the same than saying that the absolute difference between two points is bigger than 2 as Eq. (2) States.

Finally the distance between them is measured in the image and related to the distance from user to the camera to obtain the size.

IV. REFERENCE POINTS AND CLOTH DISPLAY

A reference point is required to determine where the user is located. Using face detection, the neck (point of reference) is easily detected and the cloths can be fitted automatically for the user. On the other hand, by using an AR marker (point of reference), a user will have more freedom to choose how the cloths fit on her/him. Both reference points can be obtained and displayed by using the OpenCV (Open Source Computer Vision) library [15]. OpenCV is a library of programming functions for real time computer vision applications. This library provides a comprehensive computer vision infrastructure and thereby allows users to work at a higher abstraction layer. Additionally, library functions are optimized for fast and efficient processing.

Face Detection

In order to detect faces, we use the Haar-like features [16][21] that are digital image features used in object recognition. Other face detection approaches in the literature include methods using OpenCV [17], rectangular features [18] as well as improvements to make the algorithms faster for hardware implementation [19][20].

The Haar-like features are so called because they are computed similar to the coefficients in Haar wavelet transforms. A set of these features can be used to encode the contrasts exhibited by a human face and their spatial relationships. A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in these regions and calculates the difference between them. This difference is then used to categorize subsections of an image by comparing it to a learned threshold that separates non-objects from objects. Since a Haar-like feature is only a classifier, a large number of Haar-like features are necessary to describe an object with sufficient accuracy. The database to obtain a strong learner for the object that we want to detect and where the Haar-like features are organized is called classifier

cascade.

In the case of proposed VFR algorithm, the encoded contrasts are the contrasts of the human face as well as their spatial relationships. This classifier needs to be trained with hundreds of samples of a particular object, which will represent the positive examples. Negative examples are also trained with samples that are not considered as the object to detect. All the samples must have the same size (for example 20x20). The classifier must be easily resizable to be able to detect the object with different size in the image. Therefore, the matching procedure has to be executed on several scales.

OpenCV uses an xml file that contains all of the characteristics to detect the face. This xml file is read by the function "cvHaarDetectObjects" and it is compared with a region of interest of the input image and the classifier returns 1 if the object is detected, 0 otherwise. If every simple classifier is positive, the cascade classifier is positive, otherwise it is negative. In other words, the face detection is made with a sum of these detected samples in a predefined position. Once the face is detected a rectangle is plotted around the face and the location of a reference point is chosen based on the supposition that the neck is placed at the middle of the head, half of the rectangle's height. Also it is taken that it measures approximately a third of the head height, hence a third of the rectangle height. Note that only one face will be detected and it will be the one closest to the camera.

Marker Detection

An augmented reality marker is used to display (superimpose) the cloths over the users' image. In order to detect the marker and obtain a reference, an algorithm with seven stages has been used.

- Divide image in regions: When the image is received, it is divided into regions of 40x40 pixels and there are horizontal and vertical scan lines every 5pixels.
- Detect edges in regions: For each region, a Gaussian derivative filter is used to detect the black/white edges. The filter used is $[-3 \ -5 \ 0 \ 5 \ 3]^T \cdot A$. Once an edge is detected a Sobel operator is instead of "don't." The serial comma is preferred: "A, B, and C" instead of "A, B and C."
- Find Segments: A Random Sample Consensus grouper algorithm is then used to create the line segments in the regions. This algorithm groups series of points that can be fitted into lines. First, it arbitrarily takes 2 points of the same region with the same orientation, then the close points that have a compatible orientation are added to the line segment. Finally, lines containing at least 4 segments are considered as a detected line.
- Extend lines along edges: So far only pixels on scan lines were scanned. Now, the lines are extended pixel by pixel until a detected corner or until there is no edges. The green lines displayed on Figure 5 represent the extension of the previous red lines from one corner to another corner.
- Keep lines with corner: The lines with at least one corner are kept.
- Find markers - Chains of 3 or 4 lines are kept. A chain of line is found when the end of one line corresponds to the beginning of the second one. Cloth masks are needed to determine which pixels should be displaying the clothes and which ones not. Each cloth should have a mask. A stabilization method is used since face detection provides a rectangle for
- identification that moves quickly in position and size. *Identify markers:* The last step is to identify the inside of the marker to check that it is our marker. This step only checks that the marker has a black center and is white on the inside of the boundaries.
- The result of all these steps is combined. Here our marker is detected and we have the desired reference point.

Keeping the marker in a position near the belt even using one with the AR marker allows obtaining the reference point and can properly place clothes. This also would be comfortable to the user because they will have free hands.

Cloth Display

Cloth masks are needed to determine which pixels should be displaying the clothes and which ones not. Each cloth should have a mask; there is an example of a mask in Figure 2. A stabilization method is used since face detection provides a rectangle for identification that moves quickly in position and size.



Figure 2 - Cloth and its mask

V. VFR IMPLEMENTATION AND INTERFACE

We developed the VFR in PC system level using C++ language. For universal compatibility across different platforms, we developed a Java Applet [22], which presents an ideal solution to enable every customer to be able to run the Virtual Fitting Room. The Java Applet targeting into beagle bone for ornaments try-on. Java provides easy functions to create a graphical user interface to select size, different cloths and adjust the position of the clothes. On the other side, the OpenCV code needs to be adapted from C++ to Java [25]. In order to be able to execute the algorithm in Java, JavaCV is used which is a java wrapper for OpenCV library [23][24]. JavaCV includes the commonly used libraries by researchers in the field of computer vision, such as OpenCV, FFmpeg, libdc1394, PGR, FlyCapture, OpenKinect, videoInput, and AR-ToolKitPlus. Hence, the same functions that had been used in C++ can be now used in Java. JavaCV also comes with hardware-accelerated displays, easy-to-use methods to execute code in parallel on multiple cores, user-friendly geometric and color calibration of cameras and projectors, and detection and matching of feature points. The final look of the VFR java applet and the user interface can be seen in Figure 3.

Note: Figure 3, 3(a), 3(b) shown the result image of VFR app and Ornaments try-on app (PC system level) and Java applet design for ornaments (Beaglebone output) try-on.

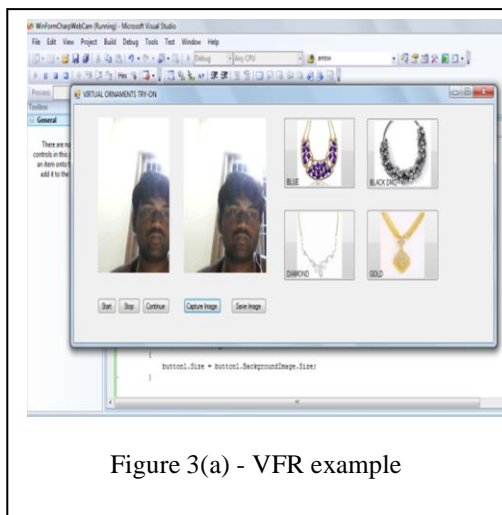


Figure 3(a) - VFR example

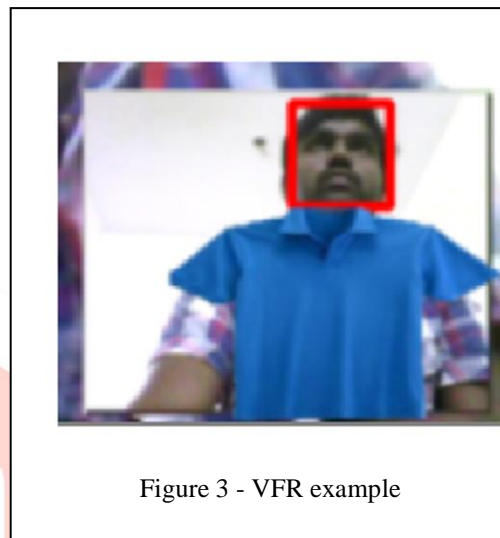


Figure 3 - VFR example

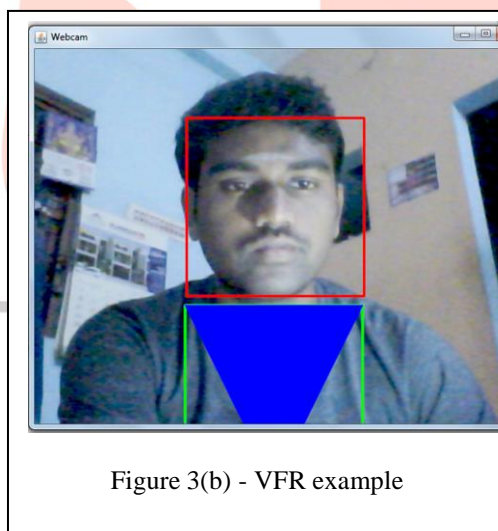


Figure 3(b) - VFR example

VI. BEAGLEBONE BASED APP DEVELOPMENT

The app has been developed using Eclipse and Java. The same has been developed using C++, which is efficient in embedded beagle bone board.

Once the VFR app is started, the user can choose between different options. First he has to select the size, between XSmall, by default, and XLarge. After that he selects the clothing that he wants to try. The app detects the face of the user and it displays the cloth using as reference a rectangle that is drawn around the face of the user (Red rectangle in Figure 4). The size of this rectangle depends on the user's face: if the user is close to the camera the rectangle will be bigger, on the other hand if he is farther it will be smaller. By using this rectangle and references, the clothes that are going to appear in the screen are scaled by the app. The maximum distance that the user can reach with his arm sometimes isn't enough to obtain a clear view so here the app offers two possibilities. One is leaving the phone in a fixed position and make himself fit into the screen and see how the clothes fit on him or use the button to switch to the back camera and let someone else hold the phone and take a picture of the user to be able to check how the cloth looks in him.

The clothes that are displayed on the screen follow and track the user similar to desktop computer implementation shown in Section V. In order to calculate the position where the image has to be displayed, the measurement from the face rectangle and the image width and height has been used. The image is displayed setting the origin, reference point, at the top-left corner. X coordinate is obtained by acquiring the X center of the rectangle and subtracting half of the image's width. Y coordinate is obtained starting from the bottom point of the rectangle and adding one third of the image's height. This is represented in Figure4.

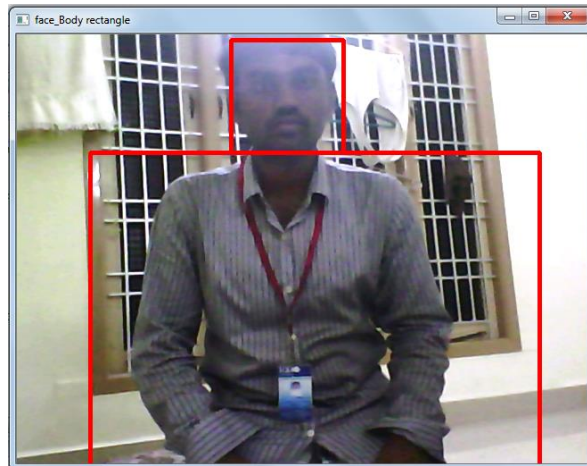


Figure 4

The equations applied are:

$$posX = Rect.centerX - imagewidth / 2 + clothes[index].offsetY$$

$$posY = Rect.bottom + clothes[index].offsetY$$

As expected, the camera follows the user in real time. In order to implement the face detection, the Android's face detection API is used, released in the Android 4.0 SDK, which can detect the face in the previous frame and specify facial features as the position of the eyes or the mouth. To use this, a FaceDetectionListener has been implemented and it returns an array containing all the faces that have been detected with their characteristics.

Finally, the performance of the application has been analyzed using the efficient tool. In order to measure the performance of the tool, the method profiling has been started. This tracks certain metrics about a method, such as number of calls, execution time, and time spent executing the method.

Based on this profiling, face detection algorithm takes only 0.1% exclusive time and 13.7% of inclusive time. This is due to the fact that its child has to display the clothes depending upon the position of the face. The method for drawing (i.e., displaying) clothes shows an exclusive time of 0.2% and 29.7% of inclusive time. Therefore, it can be seen that the most expensive processes, in terms of computation time, are the ones related with image displaying.

VII. CONCLUSION

In this work, a virtual fitting room application for beagle bone and PC system was implemented successfully. The main objective to obtain a real time, platform independent application was achieved. Users are able to select sizes from XS to XL and choose between different cameras on the device to implement the VFR. In addition, the algorithm can track and scale the clothing according to user's position and movement. This application can be used by retail companies for increasing their online/offline presence.

VIII. ACKNOWLEDGMENT

This project required genuinely efforts. Nevertheless, it would not have been realizable without the assistance and help from many individuals and organizations. Various tutorials and resources given in the website for Beaglebone and openCV have made initial works easier. We would also like to acknowledge with much appreciation the crucial role of the project coordinator, Mr.V.Natarajan as well as the panels especially in the project presentation that has improved some features by their comments and tips. Last but not least, my thanks and appreciations also go to my colleagues who have willingly helped me out with their abilities.

REFERENCES

- [1] IBM Core metrics Benchmark Reports, Available at: <http://www-01.ibm.com/software/marketing-solutions/benchmark-reports/index-2011.html>.
- [2] IBM Press release, November 29, 2011. Available at: <http://www-03.ibm.com/press/us/en/pressrelease/36113.w>
- [3] Skytu, <http://www.styku.com/business/>
- [4] JCPteen, <http://www.seventeen.com/fashion/virtual-dressing-room>
- [5] Zugara, <http://zugara.com/>
- [6] Swivel, <http://www.facecake.com/swivel/index2.html>.
- [7] RayBan, <http://www.ray-ban.com/usa/science/virtual-mirror>
- [8] Divalicious, <http://www.divaliciousapp.com>
- [9] Topshop, <http://ar-door.com/dopolnennaya-realnost/?lang=en>

- [10] Microsoft Kinect, <http://www.microsoft.com/en-us/kinectforwindows/>
- [11] M. Popa, "Hand gesture recognition based on accelerometer sensors", International Conference on Networked Computing and Advanced Information Management, pp. 115-120, June 2011.
- [12] J. Liu, "A new Reading Interface Design for Senior Citizens", Instrumentation, Measurement, Computer, Communication and Control, pp. 349-352, October 2011.
- [13] N. Kubota, D. Kouda, and S. Kamijima, "Human-Friendly Interface Based on Visual Attention for Multiple Mobile Robots", Automation Congress World, pp. 1-6, July 2006.
- [14] Y. Lin, Mao-Jiun and J. Wang, "Automated body feature extraction from 2D images", Expert Systems with Applications, vol. 38, no. 3, pp. 2585-2591, 2011.
- [15] Intel Inc., "Open Computer Vision Library". <http://opencv.org/>
- [16] Haar-like feature: <http://opencv.willowgarage.com/wiki/FaceDetection>
- [17] M. Zuo, G. Zeng, and X. Tu, "Research and improvement of face detection algorithm based on the OpenCV", International Conference on Information Science and Engineering (ICISE), pp. 1413-1416, December 2010.
- [18] D. Lee, "A face detection and recognition system based on rectangular feature orientation", International conference on System Science and Engineering(ICSSE), pp. 495-499, July 2010
- [19] L. Acasandrei, and A. Barriga, "Accelerating Viola-Jones face detection for embedded and SoC environments",
- [20] S. Rigos, "A hardware acceleration unit for face detection", Mediterranean Conference on Embedded Computing (MECO), pp. 17-21, June 2012.
- [21] C. Chai, and Y. Wang, "Face detection based on extended Haar-like features", International Conference on Mechanical and Electronics Engineering (ICMEE), pp.442-445, August 2010.
- [22] Java documentation: <http://download.oracle.com/javase/6/docs/api>

