Chapter 7
Markerless 3D Virtual Glasses Try-On System

Mariofanna Milanova and Fatima Aldaeif

Abstract This paper presents the implementation of a markerless mobile augmented reality application called a virtual eye glasses try-on system. The system first detects and tracks human face and eyes. Then, the system overlays the 3D virtual glasses over the face in real time. This system helps the consumer to select any style of glasses available on the virtual space saving both time and effort when shopping online. A method based on local-invariant descriptors is implemented to extract image feature points for eyes detection and tracking. A new approach for camera pose estimation is proposed to augment real images with virtual graphics. Experiments are conducted using Haar cascade and speeded up robust features (SURF) cascade. The system is optimized and adapted for a mobile architecture.

7.1 Introduction

Augmented reality (AR) makes use of computer vision and computer graphics techniques to merge virtual content into the real world. Ronald Azuma first outlined the features that an universal AR system should possess: (1) combination of real and virtual world; (2) interactivity and (3) three-dimensional representation of objects [1].

The implementation of AR in physical advertisement has grown rapidly and is being implemented in various industries such as automobile, food, game, engineering and many more. AR applications have the capability to add virtual clothing or apparel onto consumers' reflection, which they seem to "wear". Examples of this approach are "virtual dressing rooms" and "virtual mirrors" of brand names selling accessories such as sunglasses, jewelry or watches. The purpose of these applications is to enrich

M. Milanova (✉) · F. Aldaeif
University of Arkansas at Little Rock, Little Rock, AR, USA
e-mail: mgmilanova@ualr.edu

F. Aldaeif
e-mail: fxaldaeif@ualr.edu

R. Kountchev et al. (eds.), *New Approaches for Multidimensional Signal Processing*,
Smart Innovation, Systems and Technologies 216,
https://doi.org/10.1007/978-981-33-4676-5_7

customer shopping experiences, both in real world and online. Shoppers are able to share their choices, or "likes" through social media, and are often able to make their final purchase directly through the AR interface.

In order to strengthen competitiveness, professionals should focus on quality in the form of improving usability and design of the AR system. Using a mobile AR application, the users can get many snapshots with multiple virtual glasses at the same time allowing them to compare different glasses and designs.

7.1.1 Motivation

To meet consumers' demands, a useful virtual try-on system should be both efficient and effective. Currently, there are three main categories of eyeglasses virtual try-on techniques. The first one is adding 2D glasses image onto user's 2D facial image. This technique can only deal with frontal view of the face and cannot provide dynamic feedback for user's action. Another technique is to pre-reconstruct a 3D model of a user's head/face using the user's images and then to fit 3D glasses model to this pre-reconstructed 3D head/face model. The advantage of this method is its good fitting result. However, the reconstruction of human face is challenging and requires manual editing to have relisting model. This technique is not working in real time. The third approach is to superimpose virtual glasses 3D model onto a live face image sequence.

The real face image sequence with virtual glasses looks like a mirror using any mobile camera. In this technique, the fitting quality depends on the accuracy of eye detection/tracking algorithm and on the accuracy of a head pose estimation algorithm. The problem with this method is when the user rotates his/her head, and some parts of the virtual glasses are occluded.

7.1.2 Related Work

Huang et al. proposed a cascaded AdaBoost classifier to detect the eyes and then the glasses image fits to the eye area using affine transforms [2]. This method is simple but can only handle a front view. Human-centric design of glasses based on 3D glasses model is presented in [3]. The tracking procedure of the head movement in this system needs a 3D scan of user's head and after that the user has to wait 5–10 min until a training is done. The drawback of these methods is that the process of reconstructing a realistic 3D head model is challenging and time-consuming.

There are few commercial virtual glasses try-on systems already in use. Ray-Ban has developed a virtual glasses try-on system. Their system requires a frontal snapshot of the user, on which several features points need to be marked manually. The existing systems required manual initialization in order to obtain several pairs

of 2D–3D correspondences between the reference image and a corresponding 3D model.

In this paper, we propose the developed mixed reality system for virtual glasses try-on which enables the user to view different virtual models fitted on his/her face in real time. Virtual glasses are automatically scaled and oriented to the size and orientations of the user's face as far as the user is within the camera's view. To enhance the efficiency, we tested different algorithms for eyes detection/tracing. We compared the results using Haar cascade and SURF cascade for eyes detection and tracking using database of videos of faces in challenging uncontrolled conditions ("YouTube" database).

The rest of the paper is organized as follows: Sect. 7.2 presents an overview of our virtual mirror glasses try-on system. Section 7.3 presents faces detection methods. In Sect. 7.4, a pose estimation method for building the AR system is presented. Section 7.5 presents experimental results, and Sect. 7.6 is the conclusion.

7.2 Proposed System

The flowchart of the developed mobile AR try-on system is shown in Fig. 7.1.

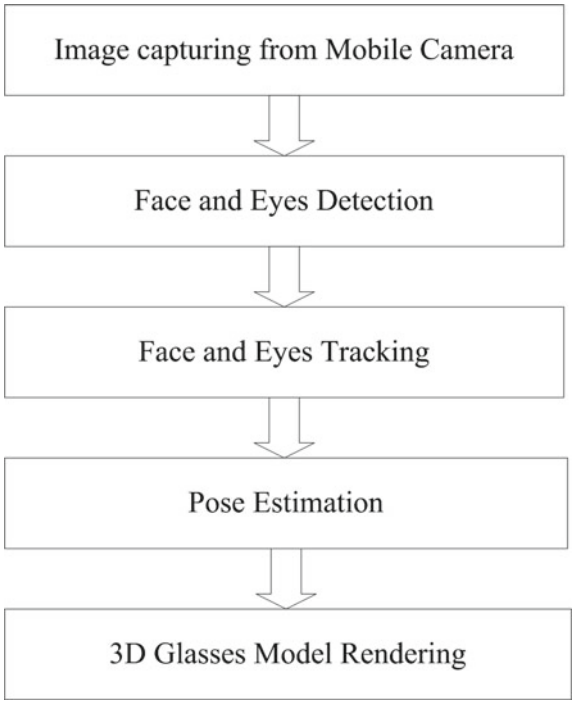

Fig. 7.1 Try-on AR system flowchart

7.3 Face Detection

The field of face detection has made significant progress in the past decade. This minor task for human beings is very challenging for computers. The difficulties associated with face detection are coming from lighting conditions, scale variations, location, orientation, pose, facial expression, occlusions, etc. In [4], a survey of recent advances in face detection is presented. In general, face detection techniques can be divided into four main categories: knowledge-based methods, feature-based approach, template matching methods and appearance-based methods. The appearance-based methods had been showing superior performance to others. Appearance-based methods learn face models from a set of representative training face images to perform detection. There are two key issues in this process, including what features to extract and which learning algorithm to apply. In this section, two techniques for face detection are presented: Haar cascade and SURF cascade.

7.3.1 Detecting user's Face and Eyes with Viola–Jones Method

The Viola–Jones (VJ) face detector contains three main ideas that make it possible for real-time object detection: the integral image for efficient Haar feature extraction, the boosting algorithm for ensemble weak classifiers and the attentional cascade structure for fast negative rejection [5] (see Fig. 7.2).

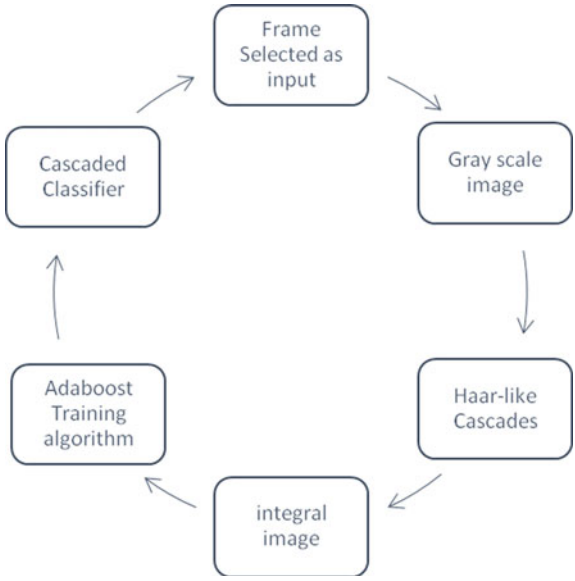

Fig. 7.2 Viola–Jones algorithm

Integral Image

Haar features depend on similar aspects for mankind face: as it is known, all mankind have the similar eyes location, size and intensity value characteristics. The integral image $T(u, v)$ at the location u, v contains the sum of the pixels above and to the left of u', v' defined as follows:

$$T(u, \upsilon) = \sum_{u' \leq u, \upsilon' \leq \upsilon} I(u', \upsilon') \tag{7.1}$$

The image can be computed in one pass over the original image using the following pair of relations:

$$S(u, v) = S(u, v - 1) + T(u, v) \tag{7.2}$$

$$T(u, v) = T(u - 1, v) + S(u, v) \tag{7.3}$$

where $S(u, v)$ is the cumulative row sum, $S(u, 1) = 0$ and $T(1, v) = 0$.

Rectangular features can be calculated in steady time, which gives them huge fast features over their more complex relatives. Averages of group of pixels were found to determine patterns like these, as shown in Fig. 7.3.

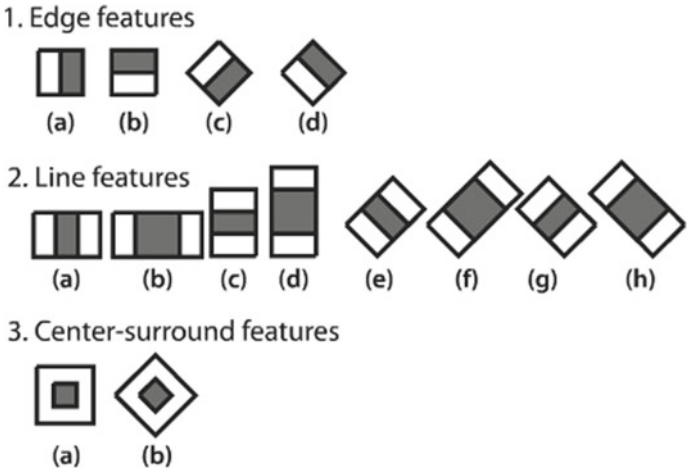

Fig. 7.3 Set of Haar-like features

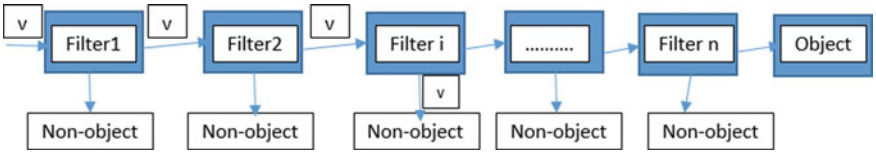

Fig. 7.4 Haar cascade

AdaBoost Training Algorithm
Using a face recognition dataset which includes a set of positive (faces and eyes) and negative (non-faces and eyes) sample images, AdaBoost algorithm will train the classification function.

Haar Cascade
As shown in Fig. 7.4, the filter will refuse non-face windows and will let face windows past to the next layer of the cascade. If an object is considered as "face," in all the filters layers, then it will be the final result.

The Viola–Jones framework has several limitations. First, the feature poll of Haar-like features is very high which is usually in hundreds of thousands level for a typical 20×20 detection template. This directs to extremely large feature search space for weak classifier learning. Second, Haar features have difficulty to handle variations due to pose and illumination. Some researchers replace Haar feature with local binary patterns (LBP), which are more robust to illumination variations. Third, the attentional cascade is trained based on two conflicted criteria: false-positive rate (FPR) f_i and hit rate for detection rate d_i true-positive rate (TPR). The VJ framework suggests min TPR $= 0.995$ and max FPR $= 0.5$ during the training procedure. To reach FPR $= 0.5$ is easy in early stages, but TPR is not converged simultaneously.

SURF Cascade
The SURF-based representation is on the basis of the "speeded up robust features," which has the properties of scale invariance, rotation invariance and computationally efficiency. SURF features do not only detect points of interest, but also propose a method for creating local-invariant descriptors. These descriptors are working under a variety of disturbing conditions like changes in scale, rotation, illumination, view-points or nose. This invariance is an important criterion for mobile systems where the environment conditions are unpredictable.

In our study, we have implemented an alternative SURF cascade [6]. The proposed approach contains four ingredients: (a) SURF features for local patch description, (b) logistic regression-based weak classifier for each patch, (c) boosting ensemble of weak classifiers for each stage and (d) area under ROC curve (AUC)-based cascade learning algorithm.

(a) SURF features for local patch description.

Features: SURF

- 2×2 cell of patch.
- Each cell is an eight-dimensional vector.
  - Sum of dx, |dx| when d$y \geq 0$.
  - Sum of dx, |dx| when d$y < 0$.
  - Sum of dy, |dy| when d$x \geq 0$.
  - Sum of dy, |dy| when d$x < 0$.
- Total is $2 \times 2 \times 8 = 32$ dim feature vector.
- Eight-channel integral images.

Feature Pool.

- In a 40×40 face detection template.
- Slide the patch (x, y, w, h) with fixed step = four pixels.
- Each cell at least 8×8 pixels, w or h at least 16 pixels.
- With 1:1, 1:2, 2:3… aspect ratio (w/h).
- Totally 396 local SURF patches.

(b) Logistic regression-based weak classifier for each patch

Given SURF feature x over local patch, logistic regression defines a probability model (7.4):

$$P(y = \pm 1|x, w) = \frac{1}{1 + \exp(-y(w^T x + b)} \tag{7.4}$$

where $y = 1$ for face samples, $y = -1$ for non-face samples, w is a weight vector for the model, and b is a bias term.

(c) Boosting ensemble of weak classifiers for each stage

Suppose there are N training samples and K possible local patches represented by d-dimensional (=32) SURF feature x, each stage is a boosting learning procedure with logistic regression as weak classifier. Given weak classifiers $h_t(x)$, the strong classifier is (7.5)

$$H^T(x) = \frac{1}{T} \sum_{t=1}^{T} h_t(x) \tag{7.5}$$

In the t-th boosting round, K logistic regression models are built for each local patch in parallel from the boosting sampled.

(d) Area under ROC curve (AUC)-based cascade learning algorithm

Each model $h_k(x)$ is tested in combination with model of previous $t-1$ rounds. Each tested model will produce an AUC score $J(H^{t-1}(x) + h_k(x))$. In the final step, the model which produces the highest AUC score is selected (7.6):

$$H^t(x) = \arg_{H^k, k=1,K} \max J(H^k = H^{t-1}(x) + h_k(x)) \tag{7.6}$$

7.4 Pose Estimation for AR Tracking

There are two subsets of camera parameters that are used to determine the relationship between coordinate systems: intrinsic and extrinsic parameters (Fig. 7.5).

The intrinsic parameters are those related to the internal geometry of a physical camera. There are (1) the focal length, (2) the location of the image center in pixel space and (3) the pixel size in the horizontal and vertical directions.

The link between image coordinates (x_{im}, y_{im}), in pixels, with the respective coordinates (x, y) in the camera coordinate system is

$$x = -(x_{\text{im}} - o_x)s_x$$

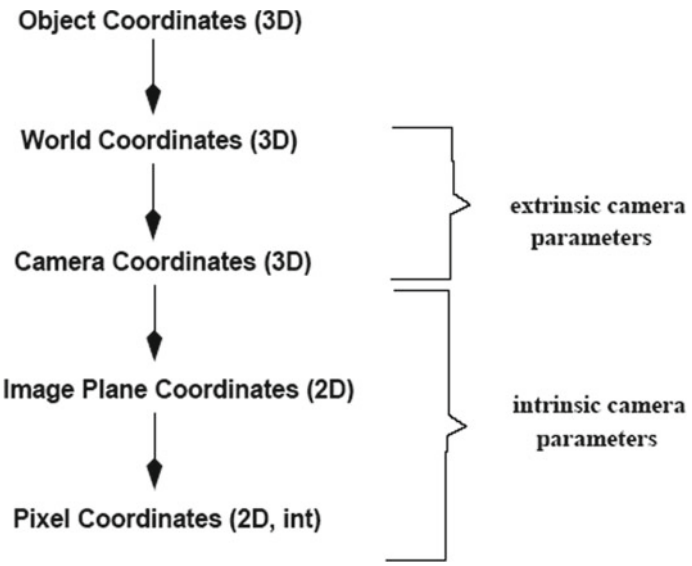

Fig. 7.5: 3D–2D coordinates transformation

$$y = -(y_{\text{im}} - o_y)s_y \tag{7.7}$$

where (o_x, o_y) define the pixel coordinates of the principal point, and (s_x, s_y) define the size of the pixels (in millimeters), in horizontal and vertical directions.

The extrinsic parameters are concerned with external properties of the camera, such as position and orientation information. These parameters are (1) the 3×3 rotation matrix R that brings the coordinate system of the object and camera coordinate system on top of one another and (2) the 3D translation vector T describing the relative positions of the origins of the two coordinate systems.

In other words, if we have a point P_{w} in world coordinates, then the same point in the camera coordinates p_{c} would be

$$p_{\text{c}} = RP_{\text{w}} + T \tag{7.8}$$

Using perspective projection and Eqs. (7.7) and (7.8), we get

$$\begin{aligned} -(x_{\text{im}} - o_x)s_x &= f\frac{R_1^T(P_w - T)}{R_3^T(P_w - T)} \\ -(y_{\text{im}} - o_y)s_y &= f\frac{R_2^T(P_w - T)}{R_3^T(P_w - T)} \end{aligned} \tag{7.9}$$

where R_i, $i = 1, 2, 3$, denotes the 3D vector formed by the i-th row of the matrix R.

Inserting the equations in homogeneous matrix form the projection, we get

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = M_{\text{int}} M_{\text{ext}} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \tag{7.10}$$

where $x_1/x_3 = x_{\text{im}}$, $x_2/x_3 = y_{\text{im}}$, M_{int} is intrinsic matrix, and M_{ext} is extrinsic matrix.

In the rendering step, Blender software is used to design the 3D glasses model, which is exported as an OpenGL ES code using tools like mtl2opengl.

7.5 Experimental Results

Comparison of HAAR and SURF in eye detection from a video on android is as follows.

7.5.1 Dataset Collection

We used the 5749 names of subjects included in the LFW dataset to search YouTube for videos of these same individuals [7]. The top six results for each query were downloaded. We minimized the number of duplicate videos by considering two videos' names with edit distance less than three to be duplicates. Downloaded videos were then split to frames at 24 fps. We detected the eyes in these videos using the Haar (Viola–Jones) face detector. Automatic screening was performed to eliminate detections of less than 48 consecutive frames, where detections were considered consecutive if the Euclidean distance between their detected centers was less than ten pixels. This process ensures that the videos contain stable detections and are long enough to provide useful information for various recognition algorithms. Finally, the remaining videos were manually verified to ensure that (a) the videos are correctly labeled by subject, (b) are not semi-static, still-image slide shows and (c) no identical videos are included in the database.

The screening process reduced the original set of videos from the 18,899 originally downloaded (3345 individuals) to 3425 videos of 1595 subjects. An average of 2.15 videos is available for each subject. The shortest clip duration is 48 frames, the longest clip is 6070 frames, and the average length of a video clip is 181.3 frames.

All video frames are encoded using several well-established, face image descriptors. Specifically, we consider the face detector output in each frame. The bounding box around the face is expanded by 2.2 of its original size and cropped from the frame. The result is then resized to standard dimensions of 200 × 200 pixels. We then crop the image again, leaving 100 × 100 pixels centered on the face. Images are then converted into grayscale. The eyes are detected from these images of faces (Fig. 7.6).

Fig. 7.6 Images from the YouTube database

Table 7.1 .

Feature extracted using	Hits	Misses	Speed (on phone) (s)
Haar–Viola–Jones	732	468	132
SURF cascade	936	264	165

7.5.2 Experimental Setup

An android application was developed to test the accuracy of the above classifiers. The dataset for this classification was used from YouTube videos frame dataset. Every second's video frame was converted into an image, and it was tested using Haar and SURF. Haar cascade classifier is already present in OpenCV for eye detection (using Viola-Jones framework). The SURF cascade however extracted 100 keypoints per image and was tested using 20 negative and 80 positive samples. The following results were obtained:

Hardware Setup: Samsung S4 Phone, Android 5.0.1, Quad-core 1.6 GHz Cortex-A15 and quad-core 1.2 GHz Cortex-A7, PowerVR SGX544MP3 GPU, 2 GB RAM (Table 7.1).

7.5.3 Conclusion

In detection of the eyes from video frames taken from YouTube videos, SURF cascade works better than Haar (Viola–Jones framework) but it takes a slightly longer time. This implementation shows Haar is 23% faster than SURF cascade in face and eye detection. SURF cascade is 22% better at eye detection in the current YouTube faces dataset as compared to Haar–Viola–Jones over the same dataset.

We implemented our application on an android operating system. We used the android SDK version 7 under Ubuntu 9.10 and as IDE: Eclipse and Android Development Tools Plug-in (ADT). Android application program is written by a developer inside a special platform called Eclipse. It emulates the different android devices and can run the application on a real android device. The application is written in Java language and can be run on different versions of android devices.

The virtual try-on algorithms were tested in a mobile AR-prototype application using android device which makes the mobile screen working as a mirror. The developed algorithm was able to calculate the central pose and display a 3D glass model over the human eyes (Fig. 7.7).

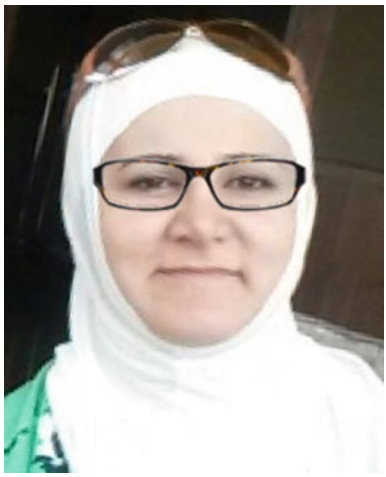

Fig. 7.7 Rendering the 3D glasses model in real time

7.6 Conclusions

In this paper, we introduced and implemented a markerless AR application: virtual try-on glasses. The most important part in an augmented reality system is the estimation of the camera poses which use the application of the homography algorithms on the extracted features. Tracking features and camera pose estimation algorithms were then tested in android phones. The SURF features and SURF cascade have proved accuracy of the detection of human eyes in uncontrolled conditions.

Acknowledgements This work was supported by the National Science Fund of Bulgaria: KP-06-H27/16 *Development of efficient methods and algorithms for tensor-based processing and analysis of multidimensional images with application in interdisciplinary areas* and NOKIA Corporate University Donation number NSN FI (85) 1198342 MCA.

References

1. Azuma R.: A survey of augmented reality. In: Teleoperators and Virtual Environments, vol. 6, pp. 355–385 (August 1997)
2. Huang, W., Hsieh, C., Yeh, J.: Vision—based virtual eyeglasses fitting system. Proc. ISCE **2013**, 45–46 (2013)
3. Huang, S., Yang, Y., Chu, C.: Human centric design personalization of 3D glasses frame in markerless augmented reality. Adv. Eng. Inform. **26**(1), 35–45 (2012)
4. Zang, C., Zhang, Z.: A survey of resent advances in face detection. In: Technical Report Microsoft Research, June 2010
5. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. Proc. CVPR, 511–518 (2001)

6. Li, J., Zhang, Y.: Learning SURF cascade for fast and accurate object detection. Proc. CVPR, 3468–3475 (2013)
7. Wolf, L., Hassner, T., Maoz, I.: Face recognition in unconstrained videos with matches background similarity. Proc. CVPR, 529–534 (2011)