

RESEARCH

Open Access



Augmented reality virtual glasses try-on technology based on iOS platform

Boping Zhang

Abstract

With the development of e-commerce, network virtual try-on, as a new online shopping mode, fills the gap that the goods cannot be tried on in traditional online shopping. In the work, we discussed augmented reality virtual glasses try-on technology on iOS platform to achieve optimal purchase of online glasses, improving try-on speed of virtual glasses, user senses of reality, and immersion. Face information was collected by the input device-monocular camera. After face detection by SVM classifier, the local face features were extracted by robust SIFT. Combined with SDM, the feature points were iteratively solved to obtain more accurate feature point alignment model. Through the head pose estimation, the virtual model was accurately superimposed on the human face, thus realizing the try-on of virtual glasses. The above research was applied in iOS glasses try-on APP system to design the try-on system of augmented reality virtual glasses on iOS mobile platform. It is proved that the method can achieve accurate identification of face features and quick try-on of virtual glasses.

Keywords: Virtual try-on, Virtual glasses, Augmented reality, Computer vision, Pose estimation, iOS

1 Introduction

Network virtual try-on is a new way of online shopping. With the development of e-commerce, it broadens the external propaganda channels of merchants to enhance the interaction between consumers and merchants. Virtual try-on fills the gap that the goods cannot be tried on in traditional online shopping. As an important part of network virtual try-on, virtual glasses try-on technology has become a key research issue in this field recently [1–4]. During virtual glasses try-on process, consumers can select their favorite glasses by comparing the actual wearing effects of different glasses in the online shopping. The research key of virtual glasses try-on system is the rapid achievement of experiential online shopping.

AR (augmented reality) calculates the position and angle of camera image in real time while adding corresponding images. The virtual world scene is superimposed on a screen in real world for real-time interaction [5]. Using computer technology, AR simulates physical information (vision, sound, taste, touch, etc.) that is difficult to experience within certain time

and space of real world. After superimposition of physical information, the virtual information is perceived by human senses in real world, thus achieving sensory experience beyond reality [6].

Based on AR principle, virtual glasses try-on technology achieves optimal purchase of user online glasses and quick try-on of virtual glasses, improving the senses of reality and immersion. Monocular camera is used as the input device to discuss try-on technology of AR glasses on iOS platform. Face information is collected by monocular camera. After face detection by SVM (support vector machine) classifier, the local features of faces are extracted by robust SIFT (scale-invariant feature transform). Combined with SDM (supervised descent method), the feature points were iteratively solved to obtain more accurate feature point alignment model. Through the head pose estimation, the virtual glasses model was accurately superimposed on the human face, thus realizing the try-on of virtual glasses. The above research is applied in iOS glasses try-on APP system to design the try-on system of AR glasses on iOS mobile platform. It is proved that the method can achieve accurate identification of face features and quick try-on of virtual glasses.

Correspondence: bopingzhang@yeah.net

School of Information Engineer, Xuchang University, Xuchang 461000, Henan, China

2 Research status of network virtual try-on technology

Glasses try-on system was first applied in the USA. Glasses companies such as Camirror, Smart Look, Ipoint Kisok, and Xview pioneered the online try-on function [7]. Users freely feel the wearing effect, enhancing the online shopping experience. Recently, online try-on function is explored by domestic and foreign glasses sellers, such as Meijing [8], Kede [9] and Biyao [10].

Virtual glasses try-on system involves computer vision, augmented reality, and image processing technology. Recently, research hotspots are speed, experience, and immersion of try-on. At present, research results can be divided into four categories, namely 2D image superposition, 3D glasses superimposed on 2D face images, 3D face modeling, and AR technology based on video stream [11–14].

Huang [15] introduced virtual optician system based on vision, which detects user's face before locating user's eyes. Three points are selected from face and glasses images. Two corresponding isosceles triangles are formed for affine transformation, thus estimating the pose and scale of face in real time. This method realizes real-time head motion tracking. However, the glasses model easily produces unrealistic deformation, affecting the realism of the glasses.

AR technology is also applied in the virtual glasses try-on system. Cheng et al. [16] selected a monocular CCD (charge-coupled device) camera as the input sensor to propose AR technology design based on the interaction of marker and face features. Virtual glasses try-on system is established based on Android mobile platform, achieving good results. During virtual try-on process, we use 2D image overlay or 3D modeling approach. There are still different defects although all kinds of virtual glasses try-on techniques have certain advantages. The superposition of 2D images is unsatisfactory in the sense of reality. Besides, the 3D modeling takes too long to meet the real-time requirements of online shopping.

In-depth research is required to realize accurate tracking and matching. These problems can be solved by AR-based glasses try-on technology to a large extent, thus providing new ideas for virtual try-on technology.

3 Methods of face recognition

It is necessary to integrate virtual objects into real environment for the application of AR technology in virtual glasses try-on system, wherein face recognition is the precondition for virtual glasses try-on system. During try-on process, it is necessary to detect the face in each frame of the video. However, the problems of posture, illumination, and occlusion can increase the omission and false ratios of face detection. The real time of detection is an important indicator of system performance to enhance people's experience senses.

General face recognition process consists of face detection, tracking, feature extraction, dimension reduction, and matching recognition (see Fig. 1) [17].

In Fig. 1, face detection is the first step to realize face recognition. Its purpose is to automatically find face region in an input image. If there is a face area, the specific location and range of face needs to be located. Face detection is divided into image-based and video-based detection. If the input is a still image, each image is detected; if the input is a video, face detection is performed throughout the video sequence.

Feature extraction is based on face detection, and the input is the detected face image. Common features are LBP (local binary patterns), HOG (histogram of oriented gradient), Gabor, etc. HOG [18] describes the edge features. Due to insensitiveness to illumination changes and small displacements, it describes the overall and local information of human face. LBP [19] shows the local texture changes of an image, with brightness invariance. Gabo feature [20] captures the local structural content of spatial position, direction selectivity, and spatial frequency. It is suitable for description of human faces.

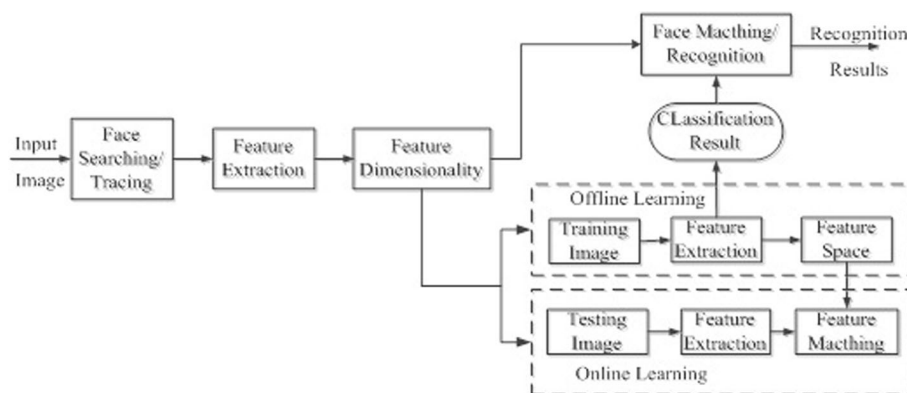


Fig. 1 Face recognition process

Feature dimension reduction is described as follows. Face feature is generally high-dimensional feature vector. Face recognition of high-dimensional feature vector increases time and space complexity. Besides, it is difficult to effectively judge the description ability of high-dimensional face features. The high-dimensional face feature vector can be projected to the low-dimensional subspace. The low-dimensional subspace information can complete face feature identification. After feature extraction, the original features are recombined to reduce vector dimension of face feature.

After the previous links, we compare the existing targets in face database and the faces to be identified based on certain matching strategy, making final decision. Matching recognition can be represented by offline learning and online matching models.

3.1 SVM-based face detection

Face detection is the premise of virtual glasses try-on technology. Recently, scholars proposed face detection methods, such as neural network, SVM (support vector machine), HMM (hidden Markov model), and AdaBoost. In the work, the classic SVM algorithm is used for face detection. SVM algorithm is a machine learning method based on statistical theory. Figure 2 shows the network structure of SVM [21]. SVM algorithm can be regarded as a three-layer feedforward neural network with a hidden layer. Firstly, the input vector is mapped from low-dimensional input space to the high-dimensional feature space by nonlinear mapping. After that, the optimal hyperplane with the largest interval is constructed in the high-dimensional feature space.

It is denoted that the input vector of SVM $x = (x_1, x_2, \dots, x_n)$. Equation (1) shows the network output of output layer based on x .

$$y(x) = \text{sgn}\left(\sum_{i=1}^{N_{\text{train}}} y_i \partial_i^* K(x^i, x) + b^*\right) \quad (1)$$

wherein the inner product $K(x^{(i)}, x)$ is a kernel function satisfying the Mercer condition. Common kernel functions consist of polynomial, Gauss, and Sigmoid kernel

functions. The Gaussian kernel function $K(x, z) = e^{-\frac{\|x-z\|^2}{2\sigma^2}}$, and σ is the width function.

Optimization problem of quadratic function (Eq. (2)) is solved to obtain the optimal parameter vector $\partial^* = (\partial_1^*, \partial_2^*, \dots, \partial_{N_{\text{train}}}^*)^T$ in discriminant function.

$$\begin{aligned} \min & \frac{1}{2} \left(\sum_{i=1}^{N_{\text{train}}} \sum_{j=1}^{N_{\text{train}}} \partial_i \partial_j y^i y^j K(x^i, x^j) - \sum_{i=1}^{N_{\text{train}}} \partial_i \right) \quad (2) \\ \text{s.t.} & \sum_{i=1}^{N_{\text{train}}} \partial_i y^i \quad i = 1, 2, \dots, N_{\text{train}} \\ & 0 \leq \partial_i \leq C \end{aligned}$$

The training sample x^i corresponding to $\partial^i > 0$ is used as a support vector. The optimization parameter b^* can be calculated by Eq. (3).

$$b^* = \frac{1}{N_{\text{sv}}} \sum_{i \in \text{SV}} \left(y^i - \sum_{j \in \text{SV}} \partial_j^* K(x^i, x^j) \right) \quad (3)$$

SVM classifier is used to determine whether the detected image is a human face. If it is not human face, then the image is discarded. If it is, then the image is retained to output the detection result. Figure 3 shows the detection process.

3.2 Face recognition based on SIFT

After face detection, face features are extracted for face recognition, providing conditions for face alignment. In the work, the robust SIFT algorithm is used for local feature extraction [22]. The algorithm finds feature points in different scale spaces. It is irrelevant to rotation, scale, and brightness changes. Besides, the algorithm has certain stability to noise, affine transformation, and angle change.

3.2.1 Basic principle of SIFT algorithm

In the process of feature construction by SIFT algorithm, it is necessary to deal with multiple details, achieving faster operation and higher positioning accuracy. Figure 4 shows flow block diagram of SIFT algorithm [21]. The generation process of local feature is described as follows [22]:

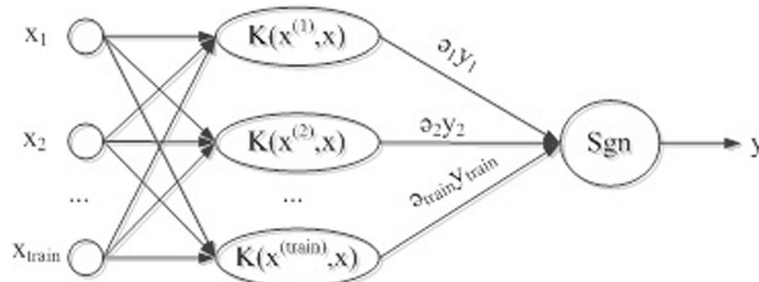


Fig. 2 SVM network structure

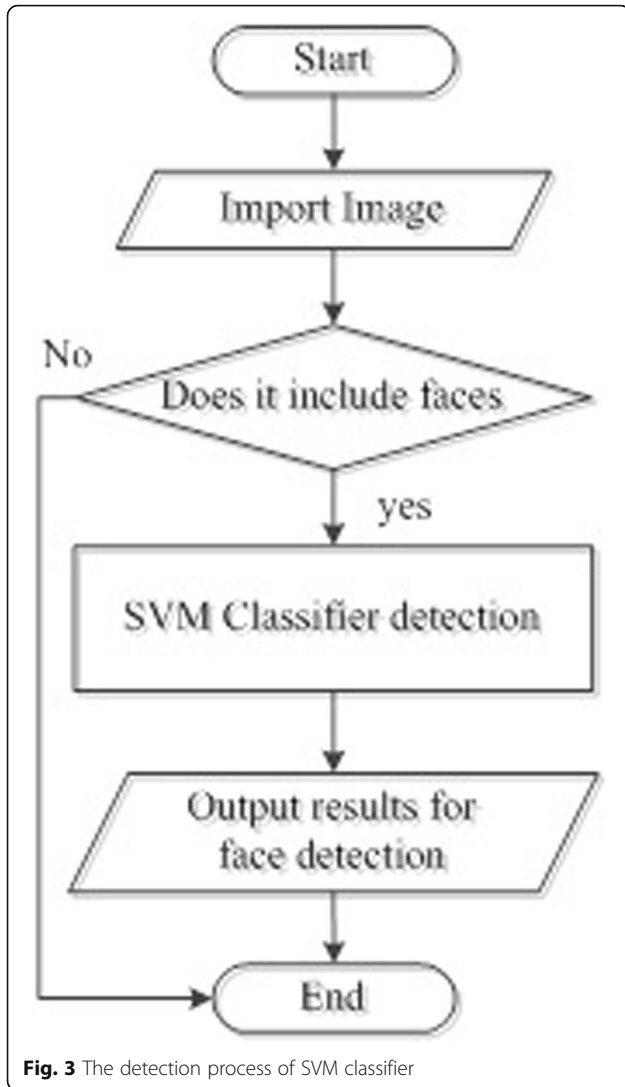


Fig. 3 The detection process of SVM classifier

① Detect extreme points

Gaussian differential functions are used for image search on all scales, thus identifying potential fixed points.

② Position key points

The scale on candidate position of model is confirmed. The stability degree determines the selection of key points.

③ Determine the direction of key points

Using the gradient direction histogram, each key point is assigned a direction with the highest gradient value to determine the main direction of key point.

④ Describe the key points

The local gradients of image are calculated and represented by a kind of symbol.

3.2.2 Key point matching

3.2.2.1 Scale space Scale space introduces a scale parameter into image matching model. The continuously variable scale parameter is used to obtain the scale space sequence. After that, the main contour of scale space is

taken as the feature vector to extract the edge features [23]. The larger scale leads to the more blurred image. Therefore, scale space can simulate the formation process of target on the retina of the human eye.

Scale space of image can be expressed as Eq. (4).

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{4}$$

In Eq. (4), $G(x, y, \sigma)$ is the Gaussian function, $I(x, y)$ the original image, and $*$ the convolution operation.

3.2.2.2 Establishing Gaussian pyramid

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-((x-d/2)^2+(y-b/2)^2)/2\sigma^2} \tag{5}$$

In Eq. (5), d and b are the dimensions of Gaussian template, (x, y) is the pixel location, and σ the scale space factor.

Gaussian pyramid is established according to Eq. (5), including Gaussian blur and down-sampling (see Fig. 5). It is observed that the pyramids with different sizes constitute tower model from bottom to top. The original image is used for the first layer, the new image obtained by down-sampling for the second layer. There are n layers in each tower. The number of layers can be calculated by Eq. (6).

$$n = \log_2\{\min(p, q)\} - d \quad d \in [0, \log_2\{\min(p, q)\}] \tag{6}$$

In Eq. (6), p and q are the sizes of the original image and d is the logarithm of minimum dimension of tower top image.

3.2.2.3 Gaussian difference pyramid After scale normalization of maxima and minima of the Gaussian Laplace function $\sigma^2 \nabla^2 G$, we obtain the most stable image features using other feature extraction functions. The Gaussian difference function is approximated to the Gaussian Laplace function $\sigma^2 \nabla^2 G$ after scale normalization. The relationship is described as follows:

$$\frac{\partial G}{\partial \sigma} = \sigma^2 \nabla^2 G \tag{7}$$

Differential is approximately replaced by the difference:

$$\sigma^2 \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \tag{8}$$

Therefore,

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k-1)\sigma^2 \nabla^2 G \tag{9}$$

In Eq. (9), $k-1$ is a constant.

In Fig. 6, the red line is the DoG operator curve; the blue line the Gauss-Laplacian curve. In extreme detection

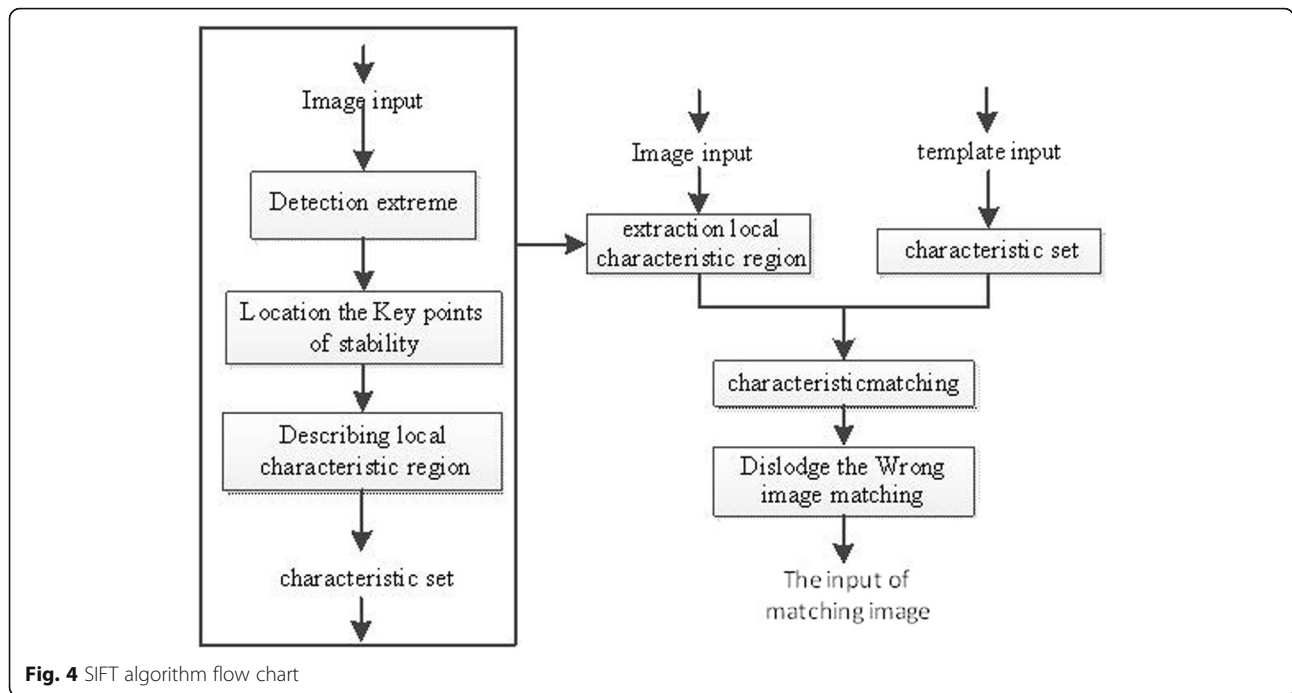


Fig. 4 SIFT algorithm flow chart

method, the Laplacian operator is replaced by the DoG operator [24] (see Eq. (10)).

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (10)$$

3.2.2.4 Spatial extreme detection In Gaussian difference space, local extreme points constitute the key points. When searching for key points, we compare the

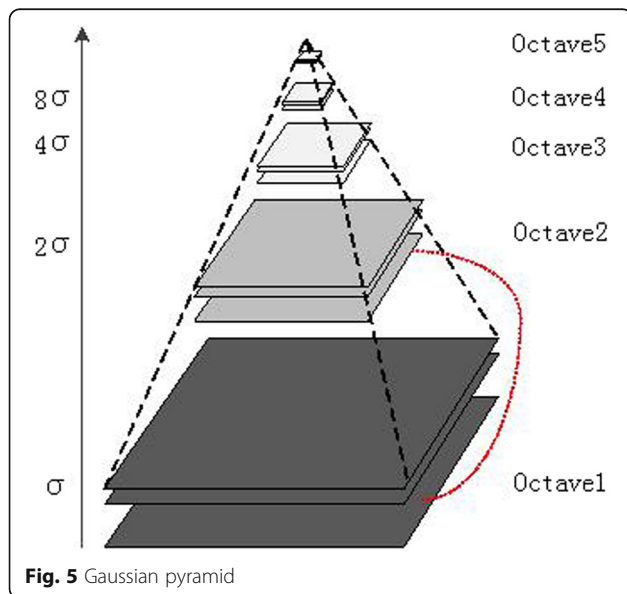


Fig. 5 Gaussian pyramid

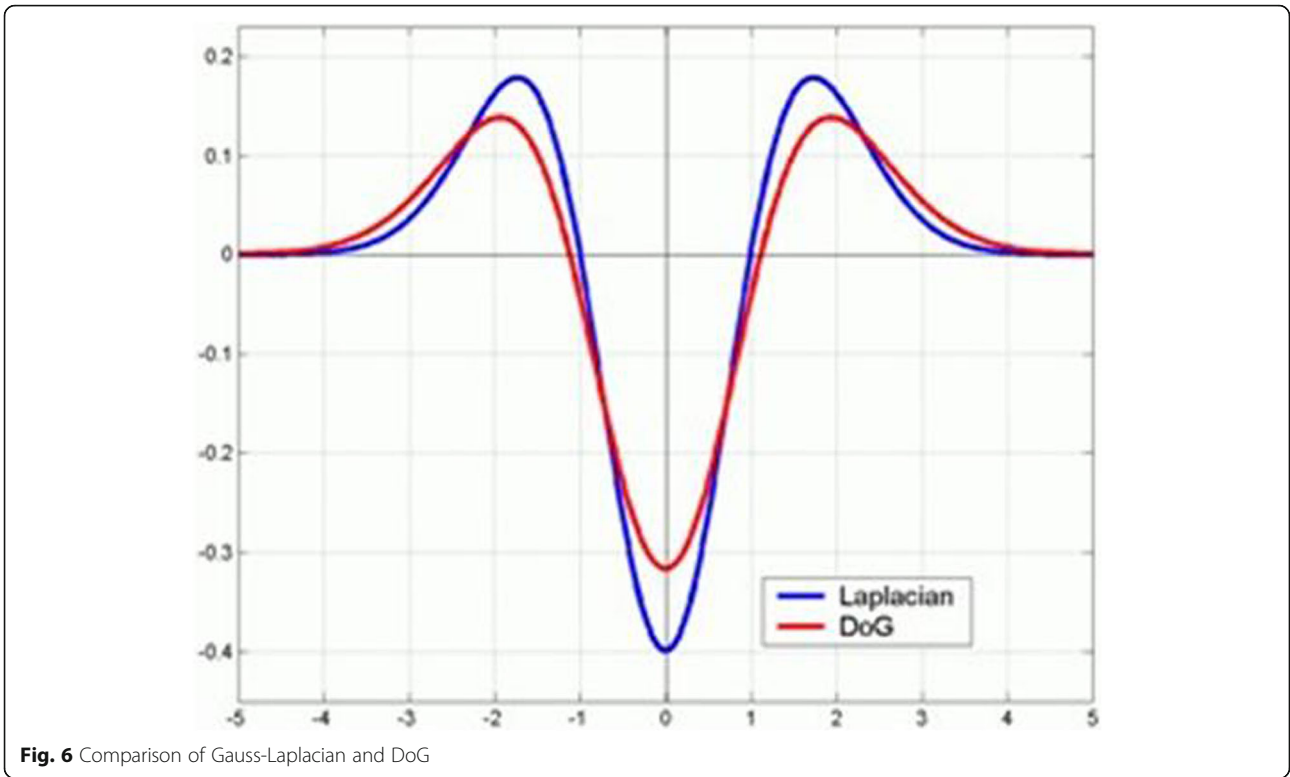
images between two adjacent layers in the same group. After that, each pixel point is compared with all the adjacent points to judge whether it is large or small (see Fig. 6). The red intermediate detection point is compared with 26 points in the surrounding, upper, and lower scale spaces to detect extreme points.

In the calculation, the Gaussian difference image is the difference between the adjacent upper and lower images in each group of the Gaussian pyramid (see Fig. 7).

3.2.2.5 Spatial extreme detection In Gaussian difference space, local extreme points constitute the key points. When searching for key points, we compare the images between two adjacent layers in the same group. After that, each pixel point is compared with all the adjacent points to judge whether it is large or small (see Fig. 8). The red intermediate detection point is compared with 26 points in the surrounding, upper, and lower scale spaces to detect extreme points.

If there are N extreme points in each group, then we need $N + 2$ -layer DoG pyramid and $N + 3$ -layer Gaussian pyramid (see Fig. 8). Due to edge response, the extreme points generated in this case are not all stable.

3.2.2.6 Key point matching At first, the key point is characterized by position, scale, and direction. To maintain the invariance of perspective and illumination changes, the key point should be described by a set of vectors. Then, the descriptor consists of key points and other contributive pixels. Besides, the independent characteristic



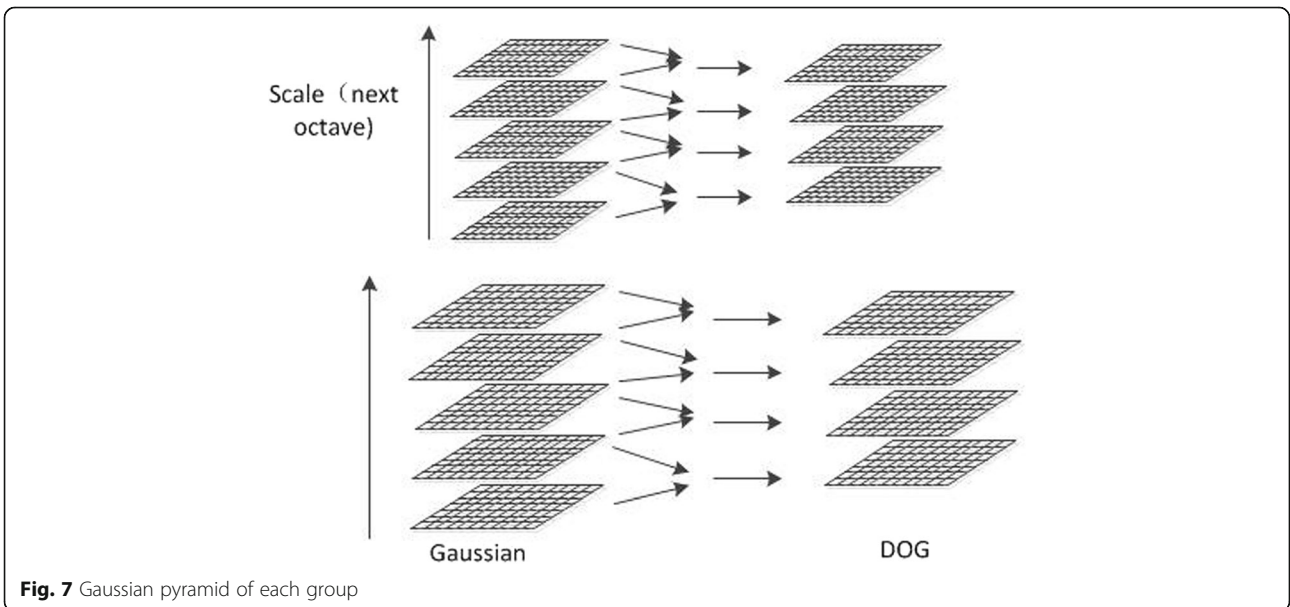
of descriptor is guaranteed to improve the probability of correct matching of feature points.

The gradient value of key point is calculated. The gradient value and direction are determined by Eq. (11).

$$m(x, y) = \sqrt{(N(x + 1, y) - N(x - 1, y))^2 + (N(x, y + 1) - N(x, y - 1))^2}$$

$$\theta(x, y) = \alpha \tan 2 \left(\frac{N(x, y + 1) - N(x, y - 1)}{N(x + 1, y) - N(x - 1, y)} \right) \quad (11)$$

In Eq. (11), N represents the scale space value of key point. Gradient histogram statistics. The gradient and direction of pixels in the neighborhood are represented by histogram. The direction ranges from 0 to 360°. There is a



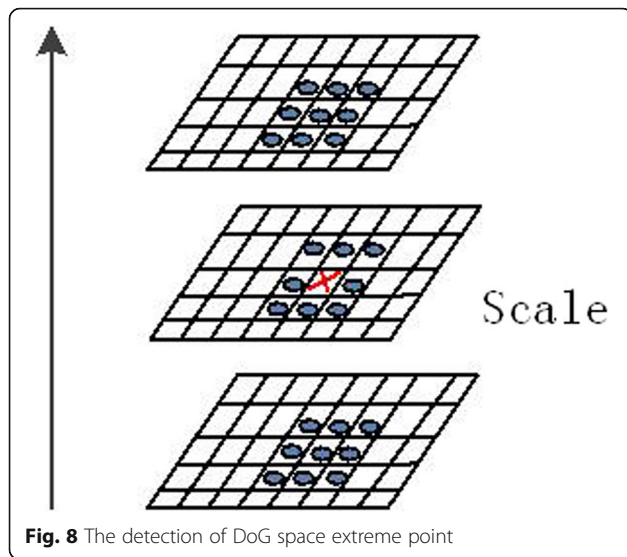


Fig. 8 The detection of DoG space extreme point

square column for every 10°, forming 36 columns [25] (see Fig. 9). In feature point field, the peak represents the gradient direction. The histogram of maximum is the main direction of key point. Meanwhile, the histogram with peak value greater than 80% of main direction is selected for auxiliary direction to improve the matching robustness.

After successful matching of key points, the entire algorithm is not over yet. This is because substantial mismatched points appear in the matching process. These mismatched points are eliminated by Ransac method in SIFT matching algorithm [26].

3.2.3 Face recognition experiment

To evaluate the algorithm, the experiment is conducted based on face infrared database provided by Terravic Research Corporation. There are a total of 20 infrared image sequences with head rotation, glasses, hats, and light-illuminated pictures. Three pairs of images are selected from each face, with a total of 60 pairs. Figure 10 shows the selected 120 images. In the work, the classic

SIFT matching algorithm is used as the initial matching method to manually determine matching accuracy and mismatch rate of each group. In other words, the matching performance is described by accuracy and error degrees. Accuracy is defined by the ratio of the number of correct matches in total number. Error degree is the ratio of the number difference (between key and matched points) in the total number of key points.

These 120 samples are conducted with abstract matching contrast according to the variables including head rotation angle, illumination transformation, glasses, and hat wearing. Meanwhile, other variables remain the same. Figures 11, 12, 13, and 14 show the matching results, respectively:

1. Matching results when head rotation angle changes
2. Matching results when wearing glasses
3. Matching results when wearing a hat
4. Matching results when light and shade change

The experimental data are shown in Table 1. The experimental image and Table 1 show:

① SIFT matching performance is more easily affected by wearing glasses than head rotation angle, light illumination, darkness, and wearing hat.

② In the case of the same number of matches, the success rate of SIFT matching is higher than that of the Harris matching method [27].

The overall trend of results can be well presented although there are inevitable errors due to the finiteness of experimental samples.

3.3 Face alignment

Face alignment is the positioning of face feature points. After face image detection, the SIFT algorithm automatically positions the contour points of the eyebrows, eyes, nose, and mouth. In the try-on process of AR glasses, the eyes are positioned to estimate the head posture. The pose estimation is applied to the tracking registration subsystem of glasses, thus producing perspective

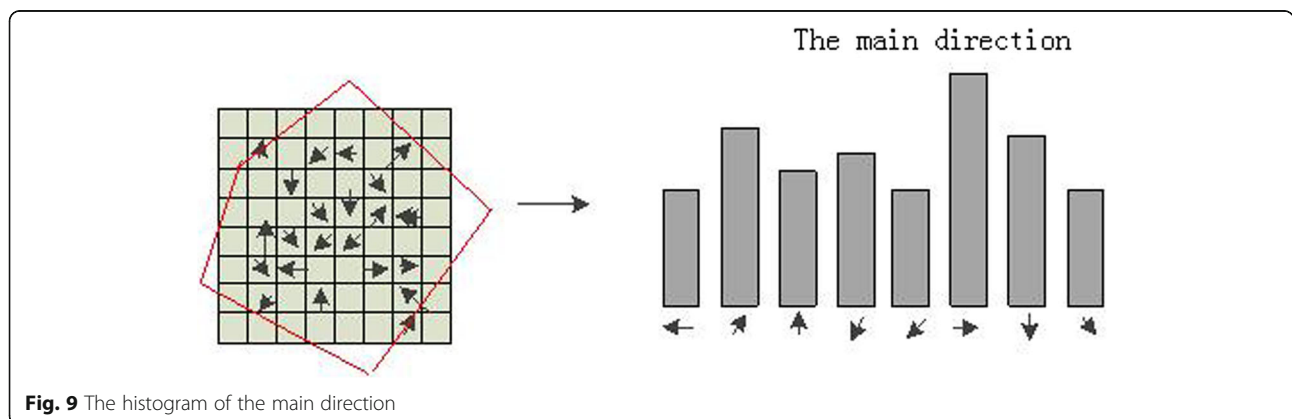


Fig. 9 The histogram of the main direction



Fig. 10 Sample sequence set

transformation. However, the pose estimation is easily affected by the positioning of face feature points, resulting in estimation error. The feature points are accurately positioned to achieve good effect of head pose estimation.

At present, there are many face alignment algorithms. SDM is a method of finding function approximation proposed by Zhu et al [28] by calculating the average face, and local features around each feature point are extracted to form feature vector descriptor. The offset between average and real face is calculated to obtain the step size and motion direction for iteration. The current face feature points are converged to the optimal position by repeated iterations.

Figure 15 shows the SDM-based face alignment process. The face alignment process is described as follows.

3.3.1 Image normalization

The image is normalized to achieve face alignment, thus improving the efficiency of training. The face image to be trained is manually labeled with feature points. After reasonable translation, rotation, and scaling transformation, the image is aligned to the first sample. The sample size is unified to arrange the original data information with confused, reducing interference other than shape factors. Finally, the calculated average face is placed on the sample as the estimated face. The average is aligned with the original face image in the center.

It is denoted that x^* is the optimal solution in face feature point location, x^0 the initialization feature point, $d(x) \in R_{n \times 1}$ the coordinates of n feature points in the image, and h the nonlinear feature extraction function near each feature point. If the SIFT features of 128 dimensions are extracted from each feature point, then $h(d(x)) \in R_{128n \times 1}$. The SIFT feature extracted at x^* can be expressed as $\varnothing^* = h(d(x^*))$. Then, the face feature point alignment is converted into the operation of solving Δx , which minimizes Eq. (12).

$$f(x_0 + \Delta x) = \|h(d(x_0 + \Delta x))\|_2^2 \tag{12}$$

The step size Δx is calculated based on the SDM algorithm.

$$x_k = x_k + \Delta x_k \tag{13}$$

If R_k and b_k are the paths of each iteration, then Eq. (11) can converge the feature point from the initial value x^0 to x^* .

$$x_{k+1} = x_{k-1} + R_{k-1}\varnothing_{k-1} + b_{k-1} \tag{14}$$

During training process, $\{d^i\}$ is the set of face images, $\{d^i\}$ the set of manually labeled feature points, and x_0 the feature point of each image. Face feature point location is transformed into a linear regression problem. For the problem, the input feature is the SIFT feature \varnothing_0^i at x_0 ;

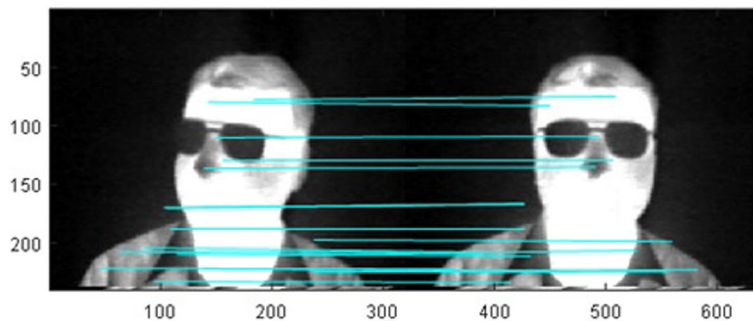


Fig. 11 Matching results when head rotation angle changes

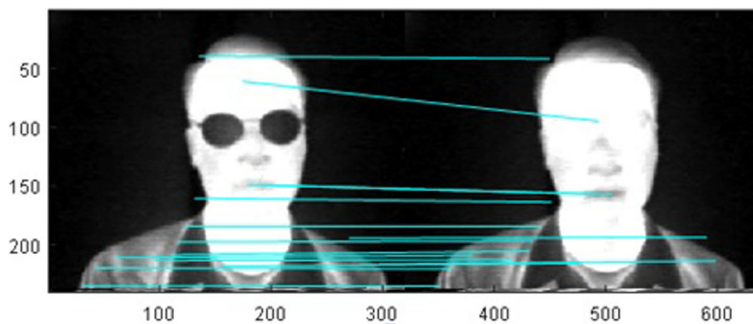


Fig. 12 Matching results when wearing glasses

the result the iteration step size $\Delta x_*^i = x_*^i + \Delta x_0^i$ from x_0 to x_* ; and the objective function Eq. (15).

$$\operatorname{argmin}_{R_0 b_0} \sum_{d^i} \sum_{x^i} \|\Delta x_*^i - R_0 \varnothing_*^i - b_0\|_2^2 \quad (15)$$

In this way, R_0 and b_0 from the training set are iterated to obtain R_k and R_k . The two parameters are used for the test phase to achieve the alignment of test images.

3.3.2 Local feature extraction of SIFT algorithm

In the work, the principal component analysis is used to reduce the dimension of image [29], the impact of non-critical dimensions, and the amount of data, thus improving the efficiency. After the dimension reduction, the local feature points are extracted from the face image. To improve the alignment accuracy of feature points, the robust SIFT algorithm is applied for local feature extraction. Section 3.2.2 introduces the extraction process in detail.

3.3.3 SDM algorithm alignment result

Training samples are selected from IBUG and LFW face databases. The former contains 132 face images. Each image is labeled with 71 face feature points, which are saved in pts file. The latter consists of the sets of test and training samples, wherein, the set of test sample contains 206 face images. Each image is labeled with 71 face feature points, which are saved in pts file. The set of training

sample contains 803 face images. Each image is labeled with 68 face feature points. Figures 16 and 17 show frontal and lateral face alignment results, respectively.

3.4 Face pose estimation

Based on computer vision, the pose of object refers to its orientation and position relative to the camera. The pose can be changed by moving the camera or object. Geometric model of camera imaging determines the relationship between 3D geometric position of certain point on head surface and corresponding point of image. These geometric model parameters are camera parameters. In most cases, these parameters are obtained by experiments. This process is called labeling [27, 29]. Camera labeling determines the geometric and optical properties, 3D position, and direction of camera relative to certain world coordinate system.

The idea of face pose estimation is described as follows. Firstly, we find the projection relationship between 2D coordinates on face image and 3D coordinates of corresponding points on 3D face model. Then, the motion coordinates of camera are calculated to estimate head posture.

A 3D rigid object has two movements relative to the camera:

- ① Translation movement

The camera is moved from current spatial position (X, Y, Z) to new spatial position (X', Y', Z') , which is called

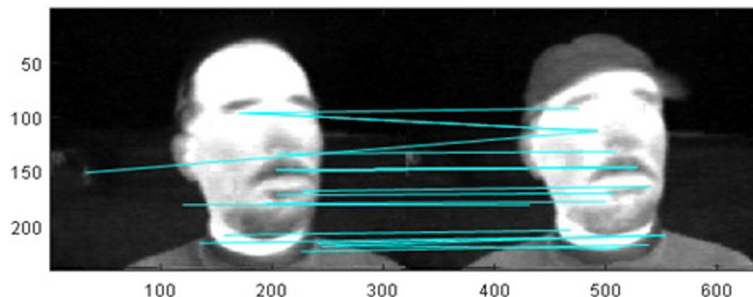


Fig. 13 Matching results when wearing a hat



Fig. 14 Matching results when light and shade change

translation. Translation vector is expressed as $\tau = (X' - X, Y' - Y, Z' - Z)$.

② Rotary movement

If the camera is rotated around the XYZ axis, the rotation has six degrees of freedom. Therefore, pose estimation of 3D object means finding six numbers (three for translation and three for rotation).

3.4.1 Feature point labelling

The 2D coordinates of N points are determined to calculate 3D coordinates of points, thus obtaining 3D pose of object in an image.

To determine the 2D coordinates of N points, we select the points with rigid body invariance, such as the nose tip, corners of eyes, and mouth. In the work, there are six points including the nose tip, chin, left, and right corners of eyes and mouth.

SFM (Surrey Face Model) is used as general 3D face model to obtain 3D coordinates corresponding to selected 2D coordinates [30]. By manual labeling, we obtain the 3D coordinates (x, y, z) of six points for pose estimation. These points are called world coordinates in some arbitrary reference/coordinate system.

3.4.2 Camera labeling

After determining world coordinates, the camera is registered to obtain the camera matrix, namely focal length of camera, optical center, and radial distortion parameters of image. Therefore, camera labeling is required. In the work, the camera is labeled by Yang and Patras [31] to obtain the camera matrix.

3.4.3 Feature point mapping

Figure 18 shows the world, camera, and image coordinate systems. In Fig. 18, O is the center of camera, c the optical center of 2D image plane, P the point in world coordinate system, and P' the projection of P on image plane. P' can be determined according to the projection of the P point.

It is denoted that the world coordinate of P is (U, V, W) . Besides, the known parameters are the rotation matrix R

Table 1 Match result analysis table

Variate	Number of matches	Total number of key points	Match ratio	False match rate
Head rotation	18	158	0.129	0.871
Wearing glasses	15	167	0.099	0.901
Wearing a hat	21	106	0.247	0.753
Light and shade change	45	281	0.191	0.809

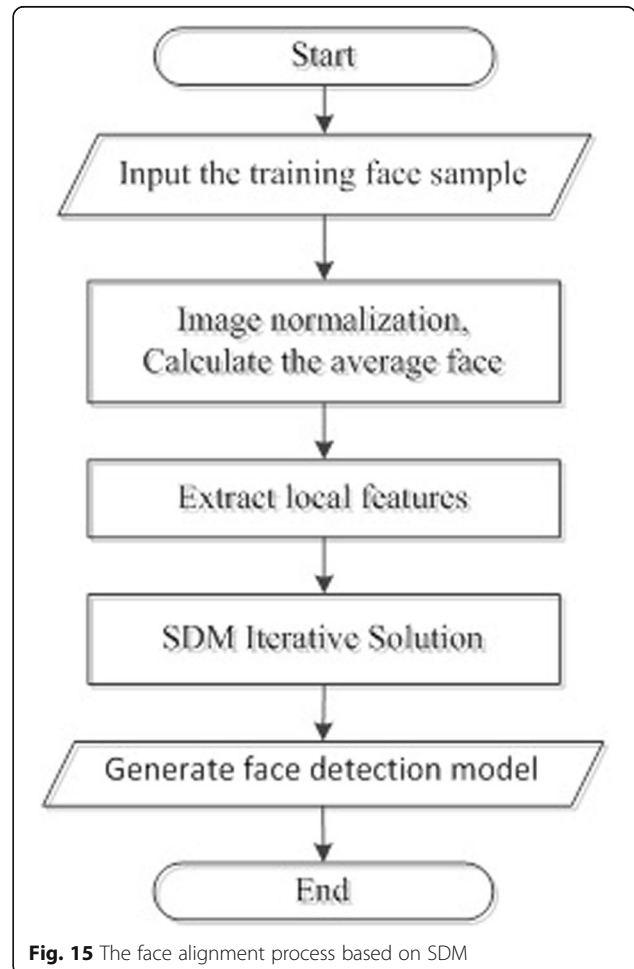


Fig. 15 The face alignment process based on SDM

(matrix 3×3) and translation vector τ (vector 3×1) from camera to world coordinate. It is possible to determine position $O(X, Y, Z)$ of P in camera coordinate system.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \tau \Rightarrow \begin{bmatrix} x \\ y \\ z \end{bmatrix} = [R|\tau] \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (16)$$

Equation (16) is expanded as follows:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & \tau_x \\ r_{10} & r_{11} & r_{12} & \tau_y \\ r_{20} & r_{21} & r_{22} & \tau_z \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ l \end{bmatrix} \quad (17)$$

If plenty of points are mapped to (X, Y, Z) and (U, V, W) , the above problem is transformed into a system of linear equations with unknown (τ_x, τ_y, τ_z) . Then, the system of linear equations can be solved.

Firstly, the six points on 3D model are manually labeled to derive their world coordinates (U, V, W) . Equation (18) is used to determine 2D coordinates (X, Y) of six points in image coordinate system.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = S \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ Z \end{bmatrix} \quad (18)$$

where f_x and f_y are the focal lengths in the x and y directions, (c_x, c_y) is the optical center, and S the unknown scaling factor. If P in 3D is connected to O , then P' where light intersects image plane is the same image connecting all points in the center of the camera produced by P along the ray.

Equation (18) is converted to the following form:

$$S \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & \tau_x \\ r_{10} & r_{11} & r_{12} & \tau_y \\ r_{20} & r_{21} & r_{22} & \tau_z \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ l \end{bmatrix} \quad (19)$$

The image and world coordinates are known in the work. Therefore, Eqs. (18) and (19) are transformed into the following form:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = S \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{00} & r_{01} & r_{02} & \tau_x \\ r_{10} & r_{11} & r_{12} & \tau_y \\ r_{20} & r_{21} & r_{22} & \tau_z \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ l \end{bmatrix} \quad (20)$$

If the correct poses R and τ are known, then the 2D position of 3D facial point on image can be predicted by projecting the 3D point onto the image (see Eq. (20)). The 2D facial feature points are known. Pose estimation can be performed by calculating the distance between the projected 3D point and 2D facial feature. If the pose is correctly estimated, the 3D points projected onto image plane will almost coincide with the 2D facial features. Otherwise, the re-projection error can be measured. The least square method is used to calculate the sum of squares of the distance between the projected 3D and 2D facial feature points.

3.5 Tracking registration system

Tracking registration technology is the process of aligning computer-generated virtual objects with scenes in the real world. At present, there are two tracking registration techniques. The first superimposes certain point of face feature with a point of virtual glasses based on the face feature point tracking method [32]. The second is based on the geometric transformation relation tracking method. Face geometry and virtual glasses model are conducted with affine transformation. Virtual glasses model moves with the movement of human head, making corresponding perspective changes and realizing 3D try-on effect [33]. For the first technique, the virtual glasses cannot be changed with the movement of user head, causing poor user experience. The second technique has good tracking effect. The virtual glasses will be distorted with overlarge head corner. Combined with the two methods, the glasses model is conducted with perspective transformation using six degrees of freedom obtained by pose estimation in Section 3.3. After face superposition, accurate tracking is realized through better stereoscopic changes.

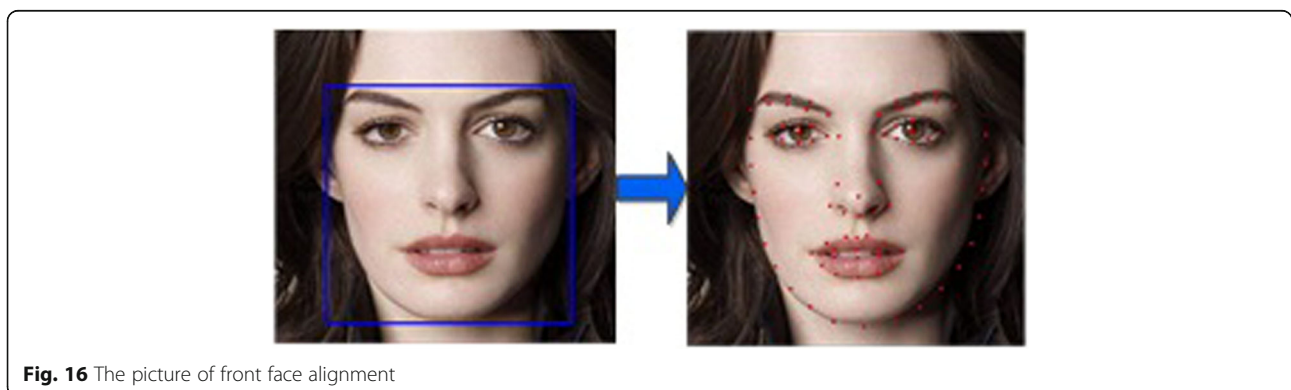


Fig. 16 The picture of front face alignment

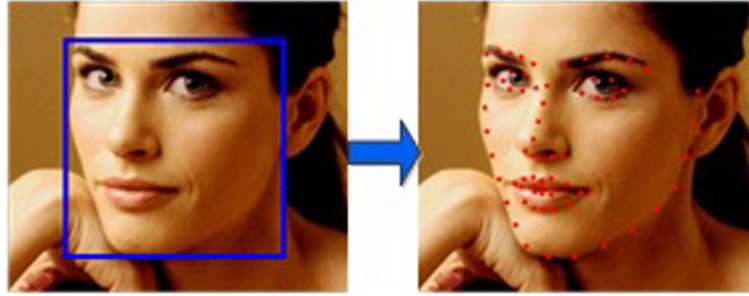


Fig. 17 The picture of side face alignment

3.5.1 Affine transformation method of glasses try-on

In Fig. 19, the center between two corners of the eye is calculated according to the distance between them. An isosceles right triangle ABC is defined [34]. The coordinates of the triangle are $A(a_1, a_2)$, $B(b_1, b_2)$, and $C(c_1, c_2)$. If the threshold is determined by experiment ahead of time, the coordinates of C are as follows.

$$C(c_1, c_2) = (b_1 - b_2 + a_2, b_1 + b_2 - a_2) \tag{21}$$

During try-on process, the glasses model is matched to the eye of user using the affine transformation Eq. (22).

$$x' = ax + by + c \quad y' = dx + ey + f \tag{22}$$

In the glasses model, the vertices of isosceles right triangle are priori, with the coordinates of (x_1, y_1) , (x_2, y_2) , and (x_3, y_3) . The vertices of isosceles right triangle on user face (x'_1, y'_1) , (x'_2, y'_2) , and (x'_3, y'_3) can be detected in motion. The affine transformation parameter $h = (a, b, c, d, e, f)^T$.

$$\begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_1 & y_1 & 1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_2 & y_2 & 1 \\ x_3 & y_3 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} \tag{23}$$

Equation (23) is abbreviated as $P = Ah$. Finally, the affine transformation parameter h ($h = (A^T A)^{-1}$) is calculated by least square method. If h is applied to the isosceles right triangle, then the image of glasses will be projected onto the right position of the face.

3.5.2 Perspective transformation method of glasses try-on

Affine transformation can realize the tracking of 3D model. The tracked glasses are prone to deformation because the affine transformation has the characteristics of flatness and parallelism based on three-point transformation [35]. The six degrees of freedom are obtained from head pose estimation. After perspective transformation, the glasses are superimposed with eye feature points to achieve real-time tracking effect. When the head moves, the space model of glasses should conform to human visual law, with certain deformation. It is realized by perspective transformation [36] (Fig. 20).

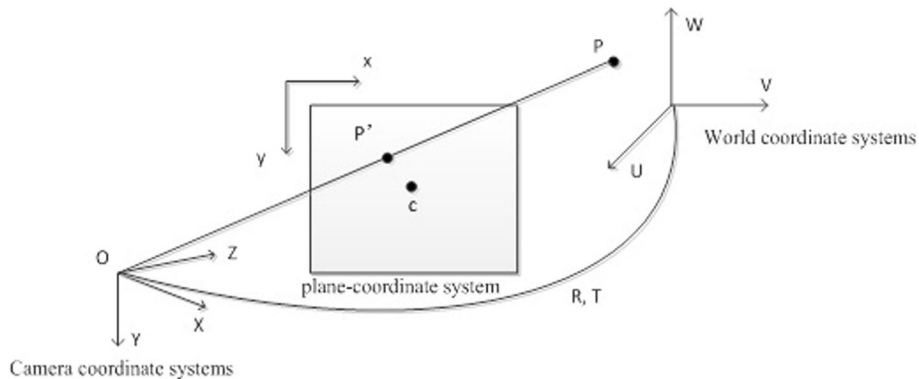


Fig. 18 Three coordinate systems

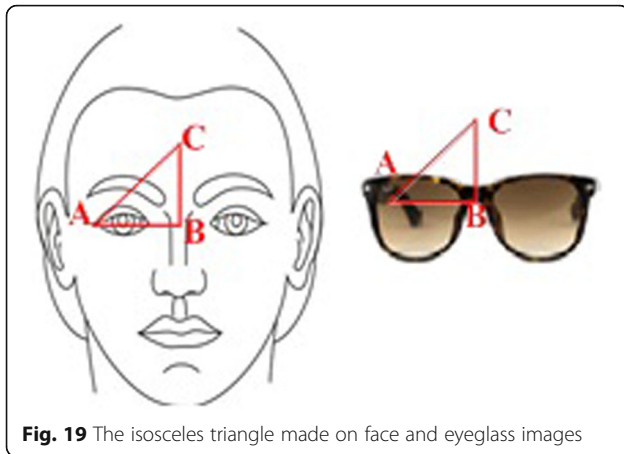


Fig. 19 The isosceles triangle made on face and eyeglass images

3.6 Virtual model generation system

In the work, the 3D glasses model is built in 3ds max and exported to 3DS format file. The 3DS data file cannot display the 3D model in OpenGL in real time. Firstly, the 3DS model is conducted with analytic operation. Only by transferring the operational data to the OpenGL function can we draw the virtual glasses model in this case [37].

3.7 Virtual and real synthesis system

To achieve the perfect combination of virtual glasses and realistic scenes, virtual glasses must be positioned to the exact position in the real world at first. This process is achieved by integrating markers with natural features. Figure 21 shows the overall structure of fused glasses try-on subsystem [38].

4 iOS system application

To verify the effectiveness of proposed system method, we develop a mobile “AR Glasses Try-on Sales System” for iOS platform. This system comprehensively uses the

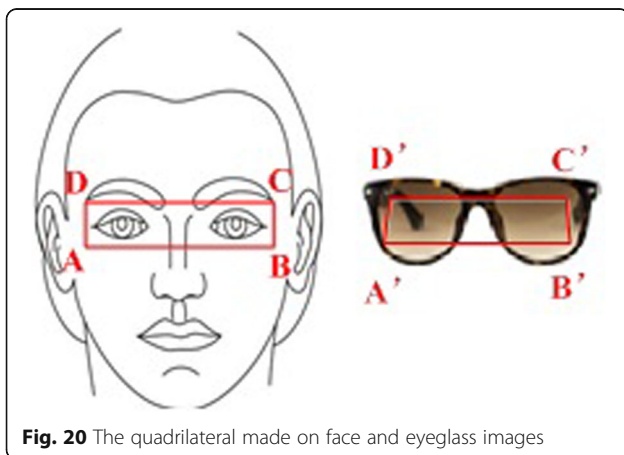


Fig. 20 The quadrilateral made on face and eyeglass images

common controls of iOS. Besides, the controls are reconstructed and optimized to improve the operating efficiency of the system. Meanwhile, most function blocks are modified to minimize the application, dependencies, and maintenance procedure of third-party framework. The system realizes the basic functions including browsing of glasses products, user registration, login, goods collection, adding to carts, user address adding, modifying, deleting, goods purchasing, integrated Alipay payment, and order management [39, 40]. In addition, it is embedded with glasses try-on, photographing, recording video, uploading, and sharing WeChat and Weibo. Meanwhile, the system has its social platform for users to browse try-on effects of others. The same glasses are quickly tried on, thus meeting the needs of vast users (Fig. 22).

Commodity browsing module is the core of the system, covering commodity browsing, screening, try-on, photographing, uploading, and sharing. The try-on and quick try-on subsystems need to call face recognition method in Part 3.

4.1 Menu module

Menu module is the framework module of the whole application. All the sub-modules are switched by a MenuViewController controller. This control contains the views of menu, home page, favorites, shopping cart, order, coupon, photo wall, setting modules, initialization, and the switching method of controllers. The menu page is not displayed in the default startup page. User calls up the menu page by clicking the upper-left icon of the default page or sliding to the right in the home page.

Menu page adopts the traditional frosted glass method. Firstly, the UIImage object is obtained by taking a screen shot and then processed by the frosted glass tool. The frosted glass UIImage is used as the background of the menu to realize the translucent frosted glass effect.

Menu view is the leftmost view of MenuViewController. The view is located at the top of the entire application. The menu can be switched out in all modules. It is achieved mainly by the table. The rows are selected by the table to trigger the effect of selected rows.

The module title in the menu is clicked to trigger the proxy event of table, thus calling the method of selecting current module for module switching. The switchable modules mainly include user, commodity, collection, shopping cart, order, coupon, photo wall, and setting.

4.2 User registration module

User registration module is used for the management of registered users. Registered users enjoy the VIP promotion activities and prices. Unregistered users enter the registration page by clicking the registration button.

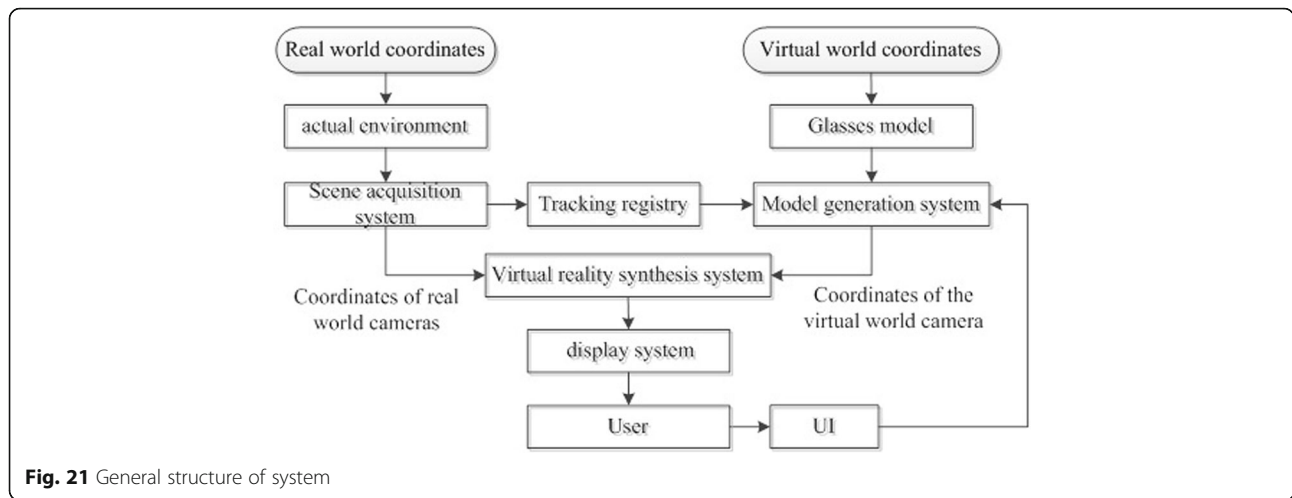


Fig. 21 General structure of system

4.3 Commodity module

Commodity module is the key of “AR glasses sales system,” including commodity browsing, selection, and adding to cart.

4.3.1 Commodity browsing

All the products are browsed in the login or non-login status. In commodity browsing, the big images of glasses are slid to browse the front of product image and leg style. The detail page can be slid to view more commodity information. More commodity information is loaded by sliding up.

The pull-up and pull-down are proxy methods based on the table and its parent class (UIScrollView).

```
(void)scrollViewDidScroll:(UIScrollView *)scrollView
```

When the height of parent container offset is greater than 20% of table height, the pull-down refresh is called. The pull-up refresh is called when the height difference

between the height of parent container and the sum of table height and offset exceeds 10% of the screen height.

4.3.2 Commodity screening

In commodity browsing, users quickly find products meeting their needs and click to select multiple items. If one item is selected, the system will feed back the number of eligible products in time. The display result button is clicked to display the screening result. Screening results can be cleared by clicking on the cross on the right of blue subtitle.

List screening is realized by modified and nested tables. The segment head of custom table is used as first-level screening title. Second-level screening catalog is achieved by nested table and custom table cells. By nesting second-level screening catalog, we obtain third-level screening catalog. The subtitle of first-level catalog is refreshed by recording the selected filter item in real time.

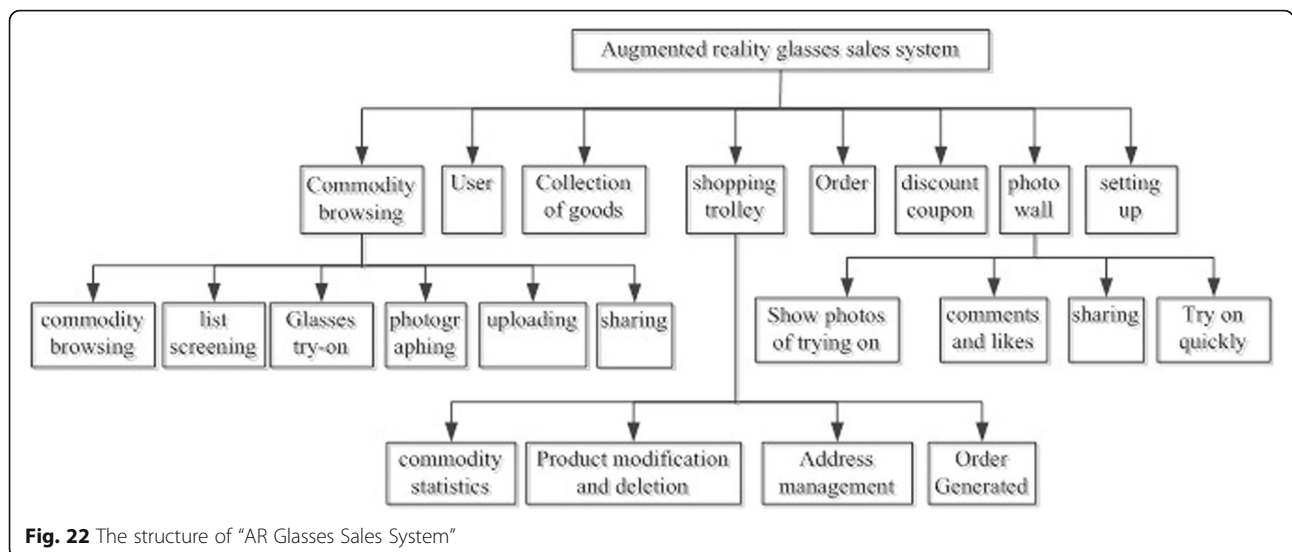


Fig. 22 The structure of “AR Glasses Sales System”

Simultaneously, the server is synchronized to get the remaining product information.

In the screening tool class, the record of third-level menu is complicated. In implementation, the third-level options are recorded for local summary, updating selected or unselected state. Full summary is performed by reusing local summary, indicating in the first-level menu.

4.3.3 Glasses try-on

In the system, the face data are captured by the camera for further processing. Firstly, user's face is located at 30–50 cm right ahead the front camera of mobile phone. The face is slightly rotated, without getting out of capture area of camera. In addition, user should not keep his/her back to the light during the try-on process because the light affects the capture effect of the camera. When the camera does not capture face data, face position will be adjusted by a prompt. As user wiggles in front of camera, the engine re-recognizes the face information. The quick try-on button is clicked to enter the quick try-on page. User can put the phone 30–50 cm in front of him. Then, his/her face appears on the screen of the mobile phone. Meanwhile, the system automatically recognizes user face data to put on glasses. The product details are viewed by

sliding the glasses or clicking on the left detail button. Figure 23 shows the try-on process in detail.

In the try-on process, the third-party openframeworks is used to modify the class library according to the requirements. Section 3 introduces face recognition, alignment, tracking registration and head pose estimation, and virtual model generation methods. Combined with these methods, the interface is packaged to increase the stability of system, reducing the dependence on third-party controls. The part embedded in openframeworks starts with the main function. Using openframeworks, the window is initialized to call the AppDelegate class of openframeworks. This class is compatible with UIKit library in iOS. The ofApp will be initialized to call engine loading model.

4.3.3.1 AFNetworking In iOS development, the NSURLConnection of XCode is competent to submit a request to a simple page of Web site, thus obtaining the response from server. However, most web pages to be visited are protected by authority. The pages cannot be visited by a simple URL. This involves the processing of Session and Cookie. Here, NSURLConnection can be used to realize access, with larger complexity and difficulty.

AFNetworking is more suitable to process requests to Web sites, including detailed Sessions and Cookies

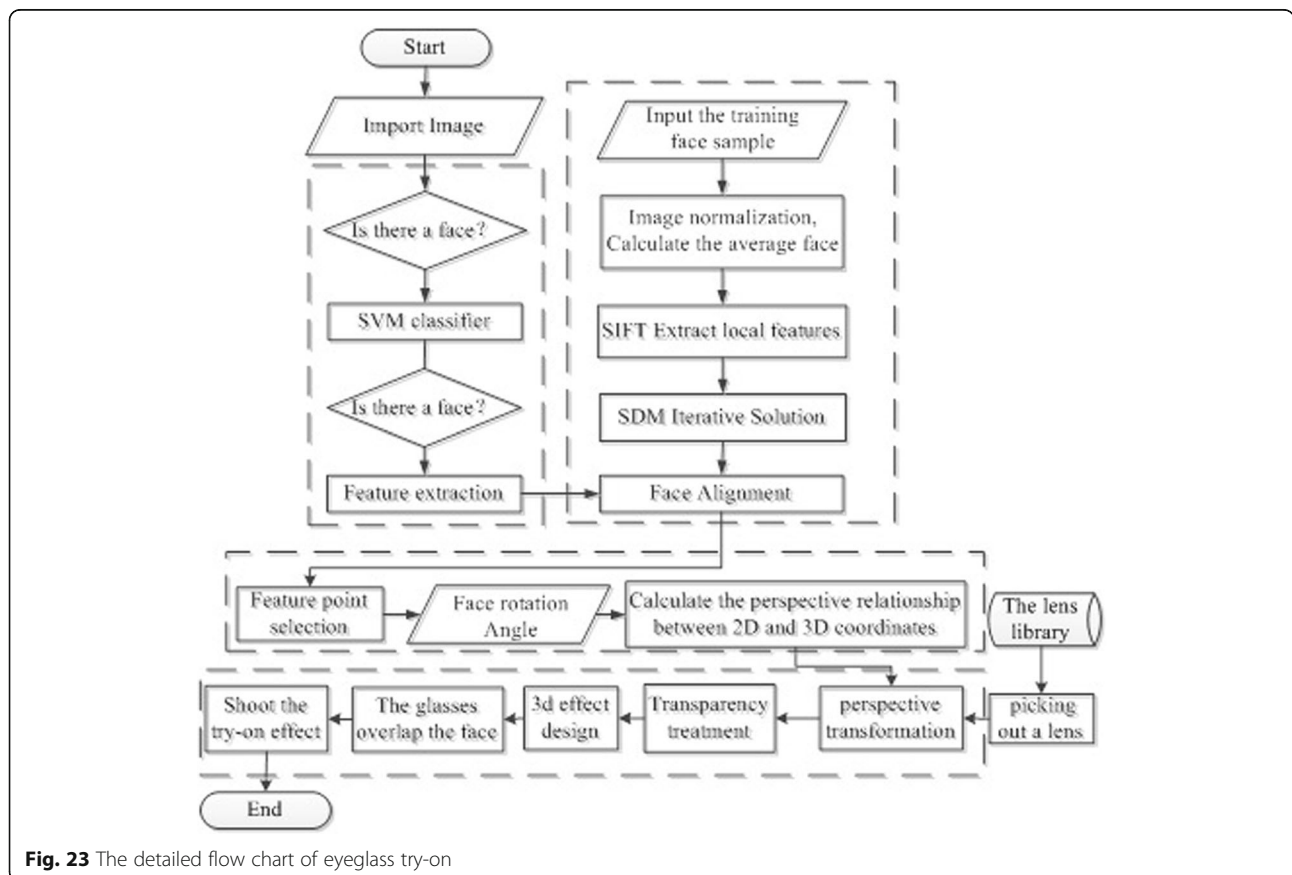


Fig. 23 The detailed flow chart of eyeglass try-on

problems. It can be used to send HTTP requests and receive HTTP responses. However, it does not cache server responses or execute the Javascript code in the HTML page. Meanwhile, AFNetworking has built-in JSON, plist, and XML file parsing for convenient application.

Some interfaces of the library are packaged to facilitate the use of AFNetworking. The packaged AFNetworking can record the operation due to disconnection request failure. After networking, the request is re-initiated.

When data needs to be requested, if get request is called, the following methods will be called:

```
-(void)GET:(NSString *)URLString parameters:(id)parameters WithBlock:(resultBlock)result;
```

If a image necessary upload pictures the following methods will be call.

```
// Upload pictures
```

```
-(void)POST:(NSString *)URLString parameters:(id)parameters WithData:(NSData *)data WithKey:(NSString *)key WithTypeO:(NSString*)pngOrMp4 WithBlock:(resultBlock)result;
```

If a video necessary upload pictures the following methods will be call.

```
// Upload video
```

```
-(void)POST:(NSString *)URLString parameters:(id)parameters WithDic:(NSDictionary *)dic WithTypeO:(NSString*)pngOrMp4 WithBlock:(resultBlock)result;
```

If post the following interface will be call.

```
// post
```

```
-(void)POST:(NSString *)URLString parameters:(id)parameters WithBlock:(resultBlock)result;
```

4.3.3.2 SDWebImage SDWebImage is a framework for third-party applications. It is used to implement asynchronous loading and caching of images. In this system, all network images are loaded using this framework. By defining interface classes, we can easily implement asynchronous loading and caching of images.

```
#import <Foundation/Foundation.h>
@interface TGImageTool : NSObject
+ (void)downloadImage:(NSString *)url placeholder:(UIImage *)place imageView:(UIImageView *)imageView;
+ (void)clear;
@end
#import "TGImageTool.h"
#import "UIImageView+WebCache.h"
@implementation TGImageTool
+ (void)downloadImage:(NSString *)url placeholder:(UIImage *)place imageView:(UIImageView *)imageView
{[imageView setImageWithURL:[NSURL URLWithString:url] placeholderImage:place options:SDWebImageLowPriority | SDWebImageRetryFailed];}
+ (void)clear
{
```

```
// 1. Clear the cached images in memory
[[SDImageCache sharedImageCache] clearMemory];
[[SDImageCache sharedImageCache] clearDisk];
// 2. Cancel all download requests
[[SDWebImageManager sharedManager] cancelAll];
}
@end
```

The image is loaded by the above tool class method. When the cache is implemented, the image will be automatically added to the cache.

The clear method of tool class is called to clear the cache.

4.3.3.3 JSONKit JSONKit is used in this system only when the order information is submitted. It transcodes the complicated parameter information to JSON strings for server application. The conversion method is described as follows.

```
NSString *new=[dic JSONString]
```

4.3.4 Adding to cart

The satisfied glasses are added to shopping cart by clicking on the “Add to Cart” button. The animation of “Add to Cart” is realized by path and combined animation in the QuartzCore library.

4.3.5 Buy a glasses immediately

User directly jumps to the page for purchasing the glasses without adding to cart. The function is realized by directly jumping to order information improvement page after summarizing commodity information.

4.3.6 Taking photos or recording videos

Users with glasses can take off their glasses after logging in. VR glasses are tried on to take photos or record videos. The try-on effects can also be watched after wearing glasses. The system provides functions of taking photos and recording videos. The photo/video button is used to switch between taking pictures and recording videos. In the work, this function is realized by modifying the engine in oepnframeworks. This system only involves the call. The photos and videos are placed in the four preview areas below, where user can click to view the details.

4.3.7 Uploading and sharing

The system provides uploading and sharing functions of photos or videos to share satisfactory try-on results and wonderful moments with friends. “Share” button is clicked to upload videos to photo wall, friend circle, Weibo, or WeChat in the server. The photos or videos are deleted by clicking the “Delete” button.

The third-party AFNetworking method is used to upload files. The files can be shared to Sina Weibo, WeChat circle, friends, and photo wall.

The sharing principle is to obtain the information of the photo or video on server side. Then, the html5 page is generated, including image, video, like, and comment. The URL is returned to the client and shared to WeChat and Sina Weibo.

Users can choose whether to share to photo wall at the same time. Sharing to photo wall is to send a request to the server. The photo or video is backed up in the table corresponding to database photo wall information. When being requested, the shared information can be obtained in the photo wall.

4.4 Collection module

In the implementation of collection module, the cells of table are reused in the home page. The data are replaced with the data of favorite list. After logging in, the favorite item is added to the collection list of personal information by clicking the gray heart button, which is convenient for next viewing. "Collect" button is clicked to cancel the collected item, removing it from the collection list.

4.5 Shopping cart module

After logging in, the satisfactory item is added to shopping cart in the try-on interface. In implementation, the custom tool class is used to record the selected state. When clicking "Select All" button, all data in the table are selected. The selected state of "Select All" button is removed to cancel certain item. Meanwhile, the sums of selected item quantities and unit prices are calculated. The head position shows the number of items. "Settle" button at the bottom of table shows the total number of items. Users can modify orders and postal addresses, while submitting orders and paying online.

4.6 Order module

After logging in, users can see their historical orders in "My Order." There are two states in the order, including pending (immediate payment) and successful payment. "Pay now" button is clicked to jump to payment interface. During order payment, it will jump to the immediate payment page of shopping module and then to Alipay.

4.7 Coupon module

The coupon module is a channel through which merchants can distribute benefits to users. After logging in, users check the coupons matching their own eligibilities. There are three types of coupons received: available, used, and expired coupons. After reading coupon usage rules, users can select whether to use the coupon in the interface of order information completion.

4.8 Photo wall module

The photo wall is a display platform provided by the system to user. It is convenient for user to browse the

try-on results of others. Based on dynamic prompt function, user quickly finds the favorite style of glasses.

User dynamic prompt function is implemented by detecting new messages. Once menu page pops up, a request is sent to the server, requesting a new unread message. If there is a new message, it will show user avatars of last dynamic message and the number of new messages; otherwise, the prompt box is not displayed.

While seeing the favorite try-on results, users can like, comment, forward, or view the same item and try it on quickly.

"View commodity" button is clicked to view the detailed information of glasses try-on results. The product information is uniquely determined according to the product ID. It is the same as quick try-on principle.

User can directly try on the same glasses worn by other users by clicking the quick try-on button. The photo wall and product data are bound in the database at the beginning. Therefore, the product can be directly found and tried on according to the product ID.

4.9 Setting module

The setting module contains "check updates," "clean up picture cache," "about us," "rating," "feedback," and "exit current user." Relatively, it is the application of native table control, which is not described here.

5 Results and discussion

Experimental environment is described as follows.

Operating system: iOS 9

Development tools: Xcode 6

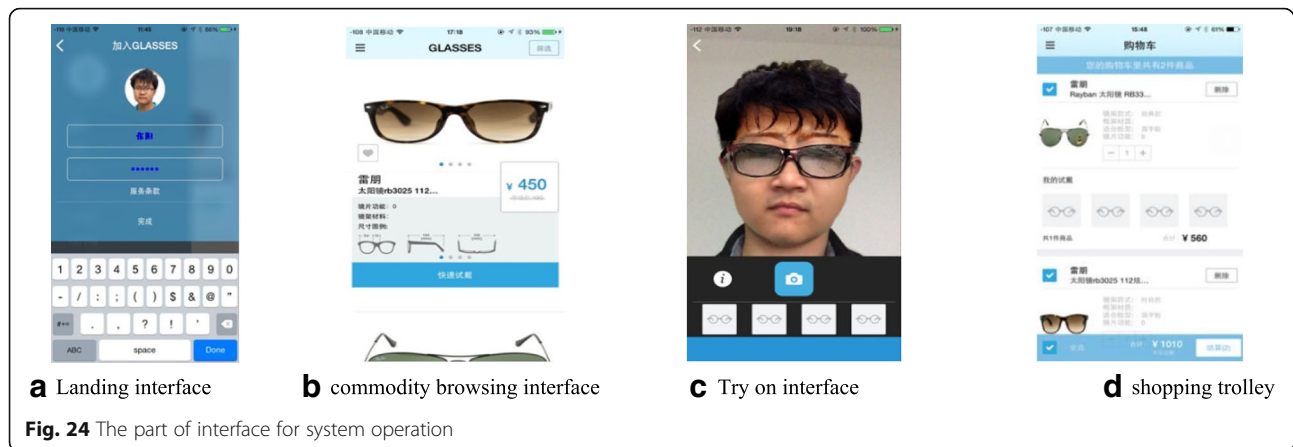
Related libraries: OpenCV, MFC

Programming language: C language, Objective-C, C++

Figure 24 shows the partial operation interface of the system.

Although a lot of jobs are done, there are still some shortcomings in the system:

1. Equipped with a try-on engine, the system has certain requirements on the performance of iPhone. The higher configuration of iPhone leads to the more accurate identification. At present, the models running smoothly are iPhone 5s and iPhone 6 and iPhone 6 Plus.
2. It is difficult for user to perform subsequent operations in the case of unstable network environment, especially for failed login.
3. In the system, the face data are captured by the camera for further processing. Firstly, user face is located at 30–50 cm right ahead the front camera of the mobile phone. The face is slightly rotated, without getting out of the capture area of the camera.
4. User should not keep his/her back to the light during the try-on process because the light affects



capture effect of the camera. When the camera does not capture face data, face position will be adjusted by a prompt. As user wiggles in front of camera, the engine re-recognizes user face information.

- At present, only the Chinese version of “AR Glasses Sales System” has been developed. There is no corresponding English version.

6 Conclusions

In the work, we discussed augmented reality virtual glasses try-on technology. Face information was collected by monocular camera. After face detection by SVM classifier, the local face features were extracted by robust SIFT. Combined with SDM, the feature points were iteratively solved to obtain more accurate feature point alignment model. Through the head pose estimation, the virtual glasses model was accurately superimposed on the human face, thus realizing the try-on of virtual glasses. This theoretical research was applied in iOS platform for the try-on of virtual glasses, thus providing best services for user selection. Experiments showed that the virtual glasses had realistic effect and high try-on speed and user satisfaction. Consequently, AR-based glasses try-on technology provided new idea for virtual try-on technology. Camera capture under complex light conditions will be further studied. App running test on iPhone 7 and above will be carried out. Multilingual versions will be developed.

Abbreviations

APP: Application; AR: Augmented reality; CCD: Charge-coupled device; HMM: Hidden Markov model; HOG: Histogram of oriented gradient; IBUG: Intelligent Behaviour Understanding Group; iOS: iPhone OS; LBP: Local binary patterns; LFW: Labeled Faces in the Wild; SDM: Supervised descent method; SFM: Surrey Face Model; SIFT: Scale-invariant feature transform; SVM: Support vector machines; VR: Virtual reality

Acknowledgements

This work is partially supported by Shanxi Province Universities Science and Technology Innovation Project (2017107) and Shanxi Province Science Foundation for Youths (201701D12111421). Thanks to the editor and reviewers.

Funding

The paper is subsidized by science and technology key project of Henan Province, China. NO.172102210462

Availability of data and materials

Data will not be shared; reason for not sharing the data and materials is that the work submitted for review is not completed. The research is still ongoing, and those data and materials are still required by my team for further investigations.

Author's contributions

BZ designed the research, analyzed the data, and wrote and edited the manuscript. The author read and approved the final manuscript.

Author's information

Boping Zhang, female, is currently an Associate Professor at the School of Information Engineering, Xuchang University, China. She received master's degree from Zhengzhou University, China, in 2006. Her current research interests include computer vision, image processing, virtual reality, and pattern recognition.

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The author declares that she has no competing interests. The author confirms that the content of the manuscript has not been published or submitted for publication elsewhere.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 15 August 2018 Accepted: 5 November 2018

Published online: 27 November 2018

References

- DITTO. <http://www.ditto.com/>
- O. Deniz, M. Castrillon, J. Lorenzo, et al., Computer vision based eyewear selector. *Journal of Zhejiang University-SCIENCE C (Computers & Electronics)* **11**(2), 79–91 (2010)
- Gongxin Xie. A transformation road for the glasses industry[J]. *China Glasses*, 2014,03:112–113
- Liu Cheng, Wang Feng, Qi Changhong, et al. A method of virtual glasses try-on based on augmented reality[J]. *Industrial Control Computer*, 2014, 27(12):66–69
- Boping Zhang. Design of mobile augmented reality game based on image recognition[J]. *EURASIP Journal on Image and Video Processing*, 2017, 20:2–20

6. Yan Lei, Yang Xiaogang, et al. Mobile augmented reality system design and application based on image recognition[J], *Journal of Image and Graphics*, 2016, 21(2):184–191
7. Niswar A, Khan I R, Farbiz F. Virtual try-on of eyeglasses using 3D model of the head[C]. *International Conference on Virtual Reality Continuum and ITS Applications in Industry*. New York: ACM; 2011:435–438.
8. Meijing[OL]. <http://www.meijing.com/tryinon.html>
9. Kede [OL]. <http://www.kede.com/frame>
10. Biyao [OL]. <http://www.biyao.com/home/index.html>
11. Li Juan, Yang Jie. Eyeglasses try-on based on improved Poisson equations. 2011 Conference on Multimedia Technology. New York: ICMT 2011. 2011; 3058–3061.
12. DU Yao,WANG Zhao-Zhong. Real-like virtual fitting for single image[J]. *Computer Systems Application*, 2015, 24(4):19–20
13. Y. Lu, W. Shi-Gang, et al., Technology of virtual eyeglasses try-on system based on face pose estimation[J]. *Chinese Optics* 8(4), 582–588 (2015)
14. Yuan M, Khan I R, Farbiz F, et al. A mixed reality virtual clothes try-on system[J]. *IEEE Transactions on Multimedia*. 2013;15(8):1958-968.
15. Huang W Y, et al. Vision-based virtual eyeglasses fitting system[C]. *IEEE, International Symposium on Consumer Electronics*. New York: IEEE. 2013;45–46
16. Wang Feng, Qi Changhong, Liu Cheng, Jiang Wei, Ni Zhou, Zou Ya. Reconstruction of 3D head model based on orthogonal images [J]. *Journal of Southeast University (Natural Science Edition)*. 2015;45(1):36-40.
17. Zhang B. *Cluster Comput.* 2017. <https://doi.org/10.1007/s10586-017-1330-5>.
18. Maatta J, Hadid A, Pietikainen M. Face spoofing detection from single images using texture and local shape analysis[J]. *IET Biometrics*, 2012, 1(1):3–10
19. Lin Y, Lv F, Zhu S, et al. Large-scale image classification: fast feature extraction and svm training[C]. *Computer Vision and Pattern Recognition (CVPR)*, 2011 IEEE Conference on. New York: IEEE; 2011:1689–1696
20. Lowe D G, Lowe D G. Distinctive image features from scale-invariant keypoints[J]. *Int. J. Comput. Vis.*, 2004, 60(2):91–110
21. Zhang Boping. Research on automatic recognition of color multi dimensional face images under variable illumination[J]. *Microelectronics & Computer*, 2017,34(5) :128–132
22. MING An-Long MA Hua-dong. Region-SIFT descriptor based correspondence between multiple cameras[J]. *CHIN ESE JOURNA L OF COMPUTERS*, 2008, 12(4):650–662
23. He Kai, Wang Xiaowen, Ge Yunfeng. Adaptive support-weight stereo matching algorithm based on SIFT descriptors[J]. *Journal of Tiajin University*, 2016, Vol.49(9):978–984
24. D.G. Lowe, Distinctive image features from scale-invariant key points. *Int. J. Comput. Vis.* 60(2), 91–110 (2004)
25. Chen Guangxi, Gong Zhenting, et al. Fast image recognition method based on locality-constrained linear coding[J]. *Computer Science*, 2016, vol. 43(5), 308–314
26. Bai Tingzhu, Hou Xibao. An improved image matching algorithm base on SIFT[J]. *Transaction of Beijing Institute of Technology*, 2013, 33(6):622–627
27. Xiong X, Tome F D L. Supervised descent method and its applications to face alignment[C].*Computer Vision and Pattern Recognition*. New York: IEEE; 2013: 532–539
28. Zhu JE, et al. Real-Time Non-rigid Shape Recovery Via Active Appearance Models for Augmented Reality (Proc. Of 9th European Conference on Computer Vision, Graz, 2006), pp. 186–197
29. Huber P, Hu G, Tena R, et al. A multiresolution 3D morphable face model and fitting framework[C]. *Visapp*. 2015
30. Zhang Z. A flexible new technique for camera calibration[J]. *IEEE Transactions on Pattern Analysis&Machine Intelligence*, 2000, 22(11):1330–1334
31. Yang H, Patras I. Sieving Regression Forest Votes for Facial Feature Detection in the Wild[C]. *New York: ICCV*; 2013:1936–1943
32. Dantone M, Gall J, Fanelli G, et al. Real-time facial feature detection using conditional regression forests[C]. *Computer Vision and Pattern Recognition*. New York: IEEE; 2012:2578–2585
33. Google Release online AR mobile games Ingress[OL]. <http://www.csdn.net/article/2012-11-16/2811943-google-launches-ingress>
34. D. Shreiner, G. Sellers, J.M. Kessenich, B.M. Licea-Kane, *OpenGL Programming Guide: The Official Guide to Learning OpenGL*, 8th edn. (Addison-Wesley Professional, United States, 2013)
35. J. Kim, S. Forsythe, Adoption of virtual try-on technology for online apparel shopping. *J. Interact. Mark.* 22, 45–59 (2008)
36. A. Merle, S. Senecal, A. St-Onge, Whether and how virtual try-on influences consumer responses to an apparel web site. *Int. J. Electron. Commer.* 16, 41–64 (2012)
37. Niswar, A; Khan, I.R.; Farbiz, F. In Virtual try-on of eyeglasses using 3d model of the head, *Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry*. New York: ACM; 2011. pp 435–438
38. Koestinger M, Wohlhart P, Roth PM, Bischof H. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. *First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011
39. Q. Zhou, Multi-layer affective computing model based on emotional psychology. *Electron. Commer. Res.* 18(1), 109–124 (2018). <https://doi.org/10.1007/s10660-017-9265-8>
40. Q. Zhou, Z. Xu, N.Y. Yen, User sentiment analysis based on social network information and its application in consumer reconstruction intention. *Comput. Hum. Behav.* (2018) <https://doi.org/10.1016/j.chb.2018.07.006>

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com