Chapter 16
Identifying People Wearing Masks in a 3D-Scene

Wenfeng Wang, Jingjing Zhang, Xiwei Liu, Bin Hu, and Zerui Meng

Abstract Now people are facing the pandemic COVID-19 and have to wear masks. This brings a problem in face recognition—occlusion problem and particularly, identifying people wearing masks in 3D-scenes is a great challenge. This study aims to develop a system for tackling this challenge. The 3D-scene is constructed with the 2D-3D coordinate transformation. For the convenience of the fusion between the virtual scene and real scene, a 3D model is achieved by Sketchup Pro. The faces and masks data are explored from the video and occluded faces recognition is achieved with the convolutional neural network.

16.1 Introduction

In the past year, the whole world has been affected by the pandemic COVID-19. Under the influences of the epidemic, people will choose to wear masks before they go out. Although wearing a mask is an efficient way to prevent the epidemic in daily life, it also brings a challenge to face recognition. For example, when checking tickets at a train station, there is dilemma—it will affect tickets inspection if people wear masks, but it is difficult for people to take off their masks to face the virus. Especially, identifying people wearing masks in 3D-scenes can be a great challenge [1, 2].

Traditionally, a 3D-scene can be constructed as follows. First, taking a picture of the scene through a sensor and then, obtaining multiple photos and shooting from as many angles as possible when conditions permit. Finally, processing these images to obtain three-dimensional video images. In recent years, with the development of

W. Wang (✉) · J. Zhang · X. Liu · Z. Meng
Shanghai Institute of Technology, Shanghai 201418, China
e-mail: wangwenfeng@nimte.ac.cn

W. Wang
Interscience Institute of Management and Technology, Bhubaneswar 752054, India

B. Hu
Changsha Normal University, Changsha 410111, China

R. Kountchev et al. (eds.), *New Approaches for Multidimensional Signal Processing*,
Smart Innovation, Systems and Technologies 270,
https://doi.org/10.1007/978-981-16-8558-3_16

Internet technology, the efficiency of building the 3D visualization was improved [3]. At the same time, 3D models have the advantage of displaying huge amounts of information, which can be utilized to identify people wearing masks in 3D-scenes.

Objectives of this study are (1) construct a 3D-scene and manually build a 3D model (2) fusion the 3D data and the 3D model to obtain the 3D visualization scene, and (3), develop a system to recognize the occluded faces in the 3D-scene [4]. In Sect. 16.2, we mainly describe our platform which was constructed in the school office. Then, in Sect. 16.3, we show the 2D face recognition and collect frames, respectively, accordingly whether wear a mask or not. In Sect. 16.3, we will carry out the 3D face recognition in the real scene and finally solve the occluded faces recognition in the real 3D-scene.

16.2 Construction of the Platform

The actual equipment installation of the experimental platform: a five-way ceiling panoramic view, two face capture cameras, one set of VR equipment, computer workstations, laptops, temperature measurement integrated prototypes, and switch sockets. The installation is as follows:

(1) The ceiling panorama is installed on the indoor roof, and the grooves are wired;
(2) VR demonstration in the entrance area, it is easy to demonstrate the area;
(3) Face capture camera is connected to system but not fixed for the convenience of subsequent development.
(4) Computer as our workstation; booting can be normal use;
(5) Integrated temperature measurement prototype, connected to system, temperature measurement as in normal use.

The hardware situation f indoor can be described as follows:

(1) Set up a complete temperature measurement development environment indoors for further development;
(2) Ceiling design can make more space for the interior to be used and prevents a lot of accidental damage;
(3) HTC Vive Pro2.0 VR headset is used for VR equipment, which allows users to use VR in a certain area and feel the VR effect;
(4) Face capture camera, normal operation; HR-IPC2143 intelligent face recognition gun-type network camera is used in face capture machine, which can provide high face recognition accuracy with low power consumption;
(5) Computer workstation for normal use; DELL 5540 mobile workstation was used;
(6) Using an eight inches dual visual temperature measuring living face recognition machine.

Fig. 16.1 Original data: saved face images of every frames

16.3 Identifying People Wearing Masks

16.3.1 Collecting and Classifying Faces Data

At first, we just experiment on a video to recognize whether people wear a mask or not. And we have saved face images of every frame as shown in Fig. 16.1.

Then, we automatically saved faces that were without masks and with masks separately after recognition. As shown in Fig. 16.2, we can see some images especially with paper over the face which didn't save well from some relative images.

We measured the accuracy of the corresponding 2D faces with and without masks as shown in Table 16.1. It mainly reads from our prerecorded video which includes 1321 frames.

16.3.2 2D-3D Face Recognition with Mask

First of all, the camera needs to be calibrated before positioning [4]. In this case, we only need to use the camera's internal parameters and distortion parameters. We can get two-dimensional coordinates through camera identification, and then define a world coordinate system. The three-dimensional coordinates of the target point are defined, so we get the coordination of the camera [5]. Because of the relative position, we can get the coordination of the target point which is relative to the camera. Then the coordinate of the target point in the world coordinate system can be obtained by Euler Angle transformation and TF transformation. Because coordinate translation means matrix addition and subtraction, coordinate rotation means matrix multiplication. The advantage of homogeneous coordinates is through adding a dimension and expressing the addition multiplication in a formula.

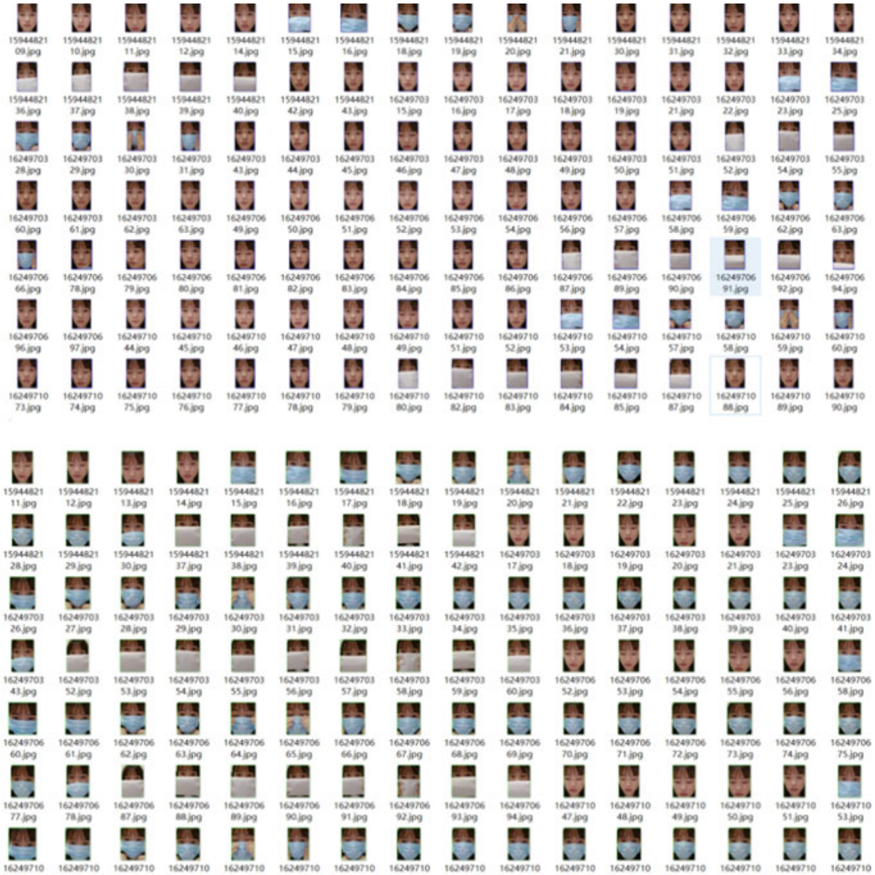

Fig. 16.2 Classified data: automatically saved images with and without masks

Table1 Accuracy of faces data classification

	Wearing mask	No wearing masks
Accuracy	0.906782247	0.973189369
Frames	720	601

$$\begin{pmatrix} \mathrm{x} \\ y \\ z \end{pmatrix} \sim \begin{bmatrix} fx & 0 & cx \\ 0 & fy & cy \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (16.1)$$

The above formula is for coordinate transformation, $[R|t]$ is an augmented matrix which includes rotation and translation.

Euler angles define the order of rotation of objects, and the degrees they have rotated about an axis. A lot of people tend to ignore the rotation order, and a lot of books call it a rule, which can be interpreted as a rule of the rotation order of Euler angles [6]. (α, β, γ) in different rotation order will have different results, firstly rotate α about the X-axis, or rotate β about the Y axis, the final result is different. There are many rules for Euler Angle, such as Z-X-Y, X-Y-Z, X-Y-X, and Z-X-Y, which have many permutations and combinations.

In the next formula about $x = r\cos\phi$, $y = r\sin\phi$, $x' = r\cos(\theta + \phi)$, and $y' = r\sin(\theta + \phi)$, putting x' and y' into the x and y, we can get the matrix form:

$$\begin{bmatrix} \mathrm{x}' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \mathrm{x} \\ y \end{bmatrix} \tag{16.2}$$

We're going to scale it up, so the final form is as follows.

Rotating about the X-axis:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{16.3}$$

Rotating about the Y-axis:

$$R_{\mathrm{y}}(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{16.4}$$

Rotating about the Z-axis:

$$R_{\mathrm{z}}(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{16.5}$$

We need to find the rotation matrices of each axis, and multiply them in order to get the whole rotation matrix. Since the rotation matrix is left-multiplied, rotation matrix [7] is $R{=}R\mathrm{x}RyRz$ for the Y-X-Z Euler angle. If it's a Z-Y-X regular Euler angle, the corresponding combined rotation matrix is $R{=}R\mathrm{x}RyRz$. R is a 3×3 matrix, the rotation matrix for the entire transformation of coordinates. We can figure out the angle by using the inverse trigonometric function:

$$\theta_{\mathrm{x}} = \arctan\frac{r_{32}}{r_{33}},\ \theta_{\mathrm{y}} = \arctan\frac{-r_{31}}{\sqrt{r_{32}^2 + r_{33}^2}},\ \theta_{\mathrm{z}} = \arctan\frac{r_{21}}{r_{11}} \tag{16.6}$$

16.3.3 2D Alignment of the Fundamental Ratio

It is easy to verify the relative translation between frames in the scene, which can be obtained as follows:

$$E_{\mathrm{i,j}} \sim [t_{ij}]_X R_{i,j} \sim E'_{c(i),c(j)} \tag{16.7}$$

where $\sim$ means multiplication which is equal to non-zero scale factor, $[]_x$ is the symbol of skew symmetric matrix and represents the cross product. Therefore, when the camera calibration matrices K and K' of sequence V and V' are the same, the corresponding uncalibrated matrix $F_{\mathrm{i},j}$ with $F'_{c(i),c(j)}$ should be equal and can be used for video synchronization [8]. $R_{i,j}$ is the rotation matrix.

But in the more common case, K and K' are constants, but they are different in sequences.

In the case of a simplified camera model, such as unit aspect ratio and zero deviation, it can be verified that:

$$F^{2\times2} \sim \begin{bmatrix} \in_{1\mathrm{st}}\ t^s_{i,j} r^t_1 & \in_{1\mathrm{st}}\ t^s_{i,j} r^t_2 \\ \in_{2\mathrm{st}}\ t^s_{i,j} r^t_1 & \in_{2\mathrm{st}}\ t^s_{i,j} r^t_2 \end{bmatrix} \tag{16.8}$$

where $\in_{\mathrm{rst}}$ for r, s, t = 1,2, …,3 means permutation tensor, r_i means columns of $R_{i,j}$. t_j means camera translation. It is worth noting that $F^{2\times2}$ is the state of observation, its elements are $F_{i,j}$. The ratios have nothing to do with the internal parameters of the camera, and only reflect the self-motion of the camera. In this article, we call these fundamental ratios [9]. Therefore, we can extract an independent four-dimensional feature V_{f}:

$$V_{\mathrm{f}} = sig(F_{11})[F_{11}, F_{12}, F_{21}, F_{22}]/||F^{2\times2}||_F \tag{16.9}$$

among them $\|\cdot\|_F$ is Frobenius norm. When two cameras are associated with a similar transformation matrix H_{s}, we can prove that two cameras have the same motion trajectory. Under this premise, they can be aligned. Proportional ambiguity H_{s} is defined as $H\begin{pmatrix} 0.8I & 0 \\ 0 & 1 \end{pmatrix}$. The position and posture of the camera are related by the proportional ambiguity, but there exists noise pollution. We calculate the camera pair which corresponds to each different position.

V_f has five freedom degrees: rotation $R_{\mathrm{i,j}}$ and $\mathrm{t}_{i,j}$. There are three freedom degrees, but there is a fuzzy scale. In addition, for the same basic matrix, there are four possible settings for the relative camera position and orientation. In fact, based on the proposed method has not available in some situations, such as pure translation, where the camera center is fixed (that is, there is no change in camera position) or the basic ratio [10] is calculated in a flat scene. However, as shown in this paper, similar camera self-motion will produce the same V_f. This can be used to synchronize video sequences.

In the process of calculating the basic ratio, SIFT features [11] are used to complete the correspondence between the initial frames and the frame, and the MAPSAC algorithm that minimizes reprojection is used to calculate the purely rotated planar homography and the basic matrix error of general camera motion [12]. The eigenvalue decomposition of the matrix is used to calculate the outer pole of a pure translation straight line. In the two frames, i and l may be far apart. Therefore, when there is no correspondence between the calculation of the basic matrix, the observation graph theory is used to calculate the basic matrix between two frames [13]. In other words, three views (i, j, k) with (j, k, l). The basic matrix inside is available.

Finally, in order to improve the robustness of the proposed method, we use a coarse-to-fine framework, because the coarse-level synchronization captures global features, so errors will not propagate to the rest of the regular path in the frame correspondence calculation [14].

The calibration error in the time axis model is used:

$$E(j, j') = \mathrm{dist}(j, j') + \min\{E(j, j' - 1), E(j - 1, j' - 1), E(j - 1, j')\} \tag{16.10}$$

among them, the dist $E(j, j)$, the mean square error between them is calculated.

Therefore, by a set of parameters $c(j)$ the determined synchronization calculation is as follows:

$$\mathrm{c(j)} = \arg \min_{c(1)\leq\cdots\leq c(N)} \sum_{\mathrm{j}=1}^{N} E(\mathrm{j}, c(\mathrm{j})) \tag{16.11}$$

where N means the number of input video frames, and then the dynamic program is used to solve the optimization problem which is defined in the equation.

16.4 3D Face Recognition in the Real Scene

16.4.1 Video Image Format Conversion

Many video images are usually in YUV format [15] obtained from the camera, but we only use process images in RGB format in the PC. Here we need to convert YUV image format to RGB format. But YUV and RGB are two different color decoding schemes. In YUV, Y is brightness, Chroma is represented by U and V, which are used to specify the color of pixels in the acquired image, and described the saturation and color of the image. Therefore, if a picture only has Y channel data, it can still display the complete picture, but the picture is black and white. And we can convert an image in YUV format to an image in RGB format through the following formula, as it has mentioned in Keith Jack's book [16]:

$$\begin{aligned} B &= 1.164(Y - R) + 2.018(U - 128) \\ G &= 1.164(Y - 16) - 0.813(V - 128) - 0.391(U - 128) \\ R &= 1.164(Y - 16) + 1.596(V - 128) \end{aligned} \tag{16.12}$$

Noting in the above formula, the range for RGB is [0, 255], the range for Y is [16, 235], and the range for UV is [16, 239]. If the result is out of this range, the processing is truncated.

It is the simplest and most direct way to convert YUV format into RGB format. By accessing each pixel in the image pixel by pixel, the conversion from YUV to RGB format image can be completed.

16.4.2 The 3D-Visualized Face Recognition

Through panoramic camera to obtain panoramic video, we need to built a three-dimensional model and fuse with it, and then a three-dimensional visualization scene can be established for face recognition. SketchUp is used as a design tool oriented to the creation process of design schemes for 3D architectural design. The program runs, as shown in Fig. 16.3. First, draw a floor plan of the room, which can be done using the Line Tool. Then, using the push-pull tool to build a preliminary

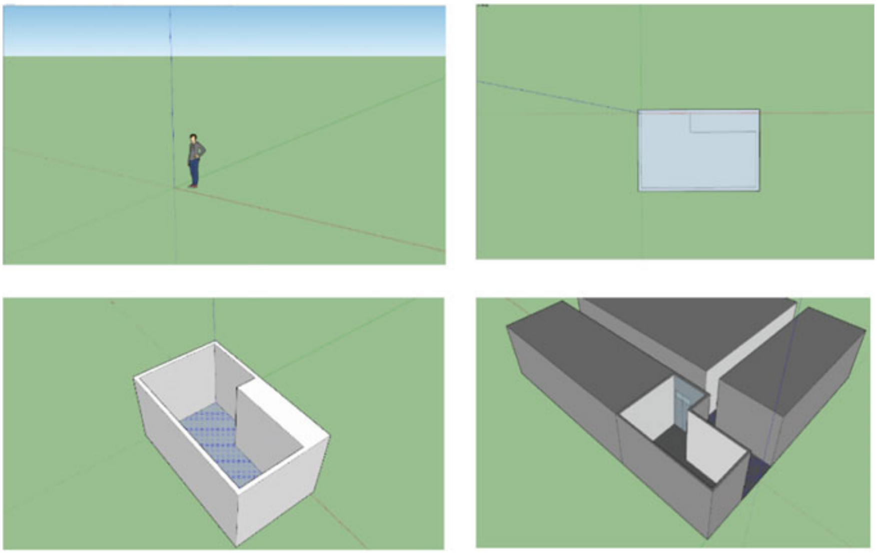

Fig. 16.3 The process for 3D-visualized face recognition

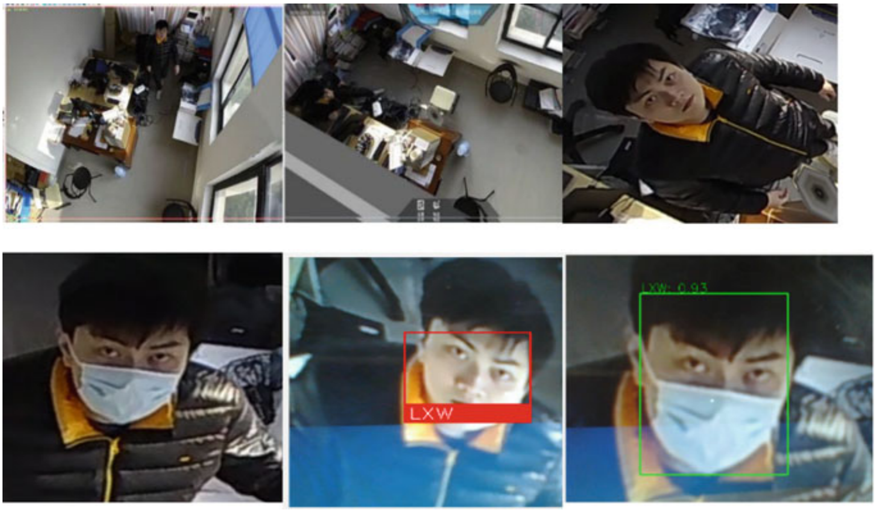

Fig. 16.4 The actually experimental results

three-dimensional model. Adding the hand-built model from SketchUp Pro to the folder where the Holographic Camera software is located to perform the 3D fusion [17, 18]. Finally, the three-dimensional model is optimized to get the following effect pictures. Meanwhile, performing 3D fusion, we must obtain images through a holographic camera as shown in Fig. 16.3. Then by fusing the image obtained by the holographic camera with the 3D model, the image after the virtual and real fusion can be obtained [19]. Based on the above three-dimensional fusion process, we can expand face recognition in the three-dimensional visualization scene. First, we obtain the face image and then achieve 3D fusion.

A homogeneous representation of camera coordinates to video image coordinates, there we use Binocular camera and we have shown the 2D-3D transformation above [20], as shown in Figs. 16.3 and 16.4.

In this way, the occluded face recognition in the 3D visualization scene is completed. From the above results, it can be seen that the establishment of a three-dimensional visualization scene has a positive effect on the recognition of occluded faces, which can assist the recognition of occluded faces. In this paper, we have collected many faces which include wearing a mask and no mask as in our face library and it helps us to further achieve 3D face recognition.

16.5 Conclusions and Perspectives

This paper mainly proposes to apply 3D technology to recognize people and use 2D scene to collect the face. And we will further statistic the accuracy of the standard face library. Through simple attempt, we solve the occluded faces problem that are difficult to recognize. During the COVID-19 pandemic, we compared contact authentication such as fingerprints, contactless face recognition authentication which has become an important tool during the May 1st Conference. The most important thing of face recognition is the biological information of the face, including facial contour, position of the nose and mouth, etc. The more feature information, the more accurate of face recognition results. However, during the epidemic, people wear masks in and out of public places, which greatly affects the accuracy of face recognition and two-dimensional occluded face recognition. After combining 3D video face recognition technology, through the recognition of face images and video, the face images in the 2D scene are obtained and we get the face library. With the help of more facial feature information, face recognition can be easier used in lots of scenes, the accuracy of face recognition can be greatly improved eventually.

Acknowledgements This research was supported by the Shanghai High-Level Base-Building Project for Industrial Technology Innovation (1021GN204005-A06).

References

1. Brunelli, R., Poggio, T.: Face recognition. IEEE Trans. Pattern Anal. Mach. Intell. **15**(10), 1042–1052 (1993)
2. Cheng, S.C., Su, J.Y., Hsiao, K.F., et al.: Latent semantic learning with time-series cross correlation analysis for video scene detection and classification. Multimed. Tools Appl. **75**(20), 12919–12940 (2016)
3. Monteiro, J., Cardoso, J.: A cognitively-motivated framework for partial face recognition in unconstrained scenarios. Sensors **15**(1), 1903–1924 (2015)
4. Zhe, X.P., Yang, J.H., Yan, Y.X.: Research of the three-dimensional tracking and registration method based on multiobjective constraints in an AR system. Appl. Opt. **57**(32), 9625–9634 (2018)
5. Yang, Y., Xiao, F., Meng, K., et al.: Solution to render the 2D timely in the 3D visual reality. Comput. Sci. **38**(6), 279–282 (2011)
6. Wang, H., Stout, D.B., Taschereau, R., et al.: MARS: a mouse atlas registration system based on a planar x-ray projector and an optical camera. Phys. Med. Biol. **57**(19), 60–63 (2012)
7. Wang, W.: The explicit expression of axis and angle of a rotation matrix. College Math. J. **52**(1), 39–44 (2021)
8. Meng, W., Gao, Y., Lu, K., et al.: View-based discriminative probabilistic modeling for 3D object retrieval and recognition. IEEE Trans. Image Process. **22**(4), 1395–1407 (2013)
9. Mohammadzade, H., Hatzinakos, D.: Iterative closest normal point for 3D face recognition. IEEE Trans. Softw. Eng. **35**(2), 381–397 (2013)
10. Wang, Y., Pan, G., Wu, Z.: A survey of 3D face recognition. J. Comput. Aided Des. Comput. Graph. **20**(7), 819–829 (2008)
11. Shah, X.M.: Tri-view morphing. Comput. Vis. Image Underst. **96**(3), 345–366 (2004)

12. Miao, S., Zhou, Y., Wei, Z.: An efficient architecture for adaptive deblocking filter of H.264/ AVC video coding. IEEE Trans. Consum. Electron. **50**(1), 292–296 (2004)
13. Lee, J.H., Lim, K.W., Song, B.C., et al.: A fast multi-resolution block matching algorithm and its LSI architecture for low bit-rate video coding. IEEE Trans. Circuits Syst. Video Technol. **11**(12), 1289–1301 (2001)
14. Jiang, C., Ding, G., Gamal, A.E., et al.: IEEE TCCN special section editorial: machine learning and artificial intelligence for the physical layer. IEEE Trans. Cogn. Commun. Netw. **7**(1), 1–4 (2021)
15. Atitallah, A.B., Kadionik, P., Ghozzi, F., et al.: An FPGA implementation of HW/SW codesign architecture for H.263 video coding. AEU Int. J. Electron. Commun. **61**(9), 605–620 (2007)
16. Eleftheriadis, A., Jacquin, A.: Automatic face location detection and tracking for model-assisted coding of video teleconferencing sequences at low bit-rates. Signal Process. Image Commun. **7**(3), 231–248 (1995)
17. Zhou, Z., Zhou, Y., Xiao, J.J.: Review of virtual reality enhancement technology. Sci. China: Inf. Sci. **45**(2), 157–180 (2015)
18. Cao, X., Lin, W., Xiao, J., et al.: Video synchronization and its application to object transfer. Image Vis. Comput. **28**(1), 92–100 (2010)
19. Cotte, Y., Toy, M.F., Arfire, C., et al.: Realistic 3D coherent transfer function inverse filtering of complex fields. Biomed. Opt. Express **2**(8), 2216–2230 (2011)
20. Hu, L., Li, Y., Li, T., et al.: The efficiency improved scheme for secure access control of digital video distribution. Multimed. Tools Appl. **75**(20), 1–18 (2016)