TRƯỜNG ĐẠI HỌC SÀI GÒN
TẠP CHÍ KHOA HỌC
ĐẠI HỌC SÀI GÒN
ISSN 1859 - 3208
Số 89, Tập 02/2023

SAIGON UNIVERSITY
SCIENTIFIC JOURNAL
OF SAIGON UNIVERSITY
No. 89, Vol 02/2023

Email: tcdhsg@sgu.edu.vn ; Website: http://sj.sgu.edu.vn/

APPLICATION OF MEDIAPIPE FRAMEWORK IN FACE SWAPPING

Ứng dụng nền tảng MediaPipe trong việc ghép khuôn mặt

Trương Đông Nam
Trường ĐH Công nghệ Đồng Nai

ABSTRACT

This paper presents a face swapping application using Google's MediaPipe framework. The application identifies facial landmarks using MediaPipe Face Mesh solution, then applies triangulation and image warping techniques to overlay the source face onto the target face. The result is an image/video with the source face but the position and movement similar to the target face. The advantage of this method is that it processes directly on video, allowing real-time face matching.

Keywords: *face swap, face landmark detection, computer vision, image processing, MediaPipe.*

TÓM TẮT

Bài báo trình bày một ứng dụng ghép khuôn mặt sử dụng nền tảng MediaPipe của Google. Ứng dụng này xác định các điểm mốc trên khuôn mặt (facial landmarks) bằng MediaPipe Face Mesh, sau đó áp dụng kỹ thuật tạo lưới tam giác (triangulation) và làm cong (warp) ảnh để ghép khuôn mặt nguồn lên khuôn mặt đích. Kết quả là một ảnh/video có khuôn mặt nguồn nhưng vị trí và cử động giống khuôn mặt đích. Ưu điểm của phương pháp này là xử lý trực tiếp trên video, cho phép ghép khuôn mặt theo thời gian thực.

Từ khóa: *face swap, face landmark detection, computer vision, image processing, MediaPipe.*

I. Introduction

The face is one of the most important biological for identification and communication between people. The ability to change faces openning up many interesting applications in virtual reality, entertainment, and even security [1]. Zollhöfer et al. [1] provides a comprehensive overview of state-of-the-art techniques in 3D facial reconstruction, tracking, and application from simple 2D images. The authors cover many related techniques such as facial landmark recognition, triangulation, and face matching, which provide an important theoretical foundation for facial modification and transformation applications.

Recently, face swapping has become increasingly popular with the development of recognition technologies [2]. Nirkin et al. [2] focuses on face matching in natural, uncontrolled environments, as opposed to laboratory conditions. The authors propose new face matching techniques that can work well with photos/videos with complex environments, different viewpoints, and diverse facial expressions. This research involves the application of

Corresponding author: truongdongnam@dntu.edu.vn

live face matching on video in the paper, as videos are often recorded in natural environments under uncontrolled conditions. Therefore, the techniques proposed in research [2] can serve the application of face matching on video directly.

However, most current methods only apply to still images or have high latency, not allowing direct processing on video [3]. Agarwal et al. [3] focuses on optimization performance of face matching algorithms to be able to run directly on mobile devices such as smartphones or tablets. The authors propose optimization techniques that reduce computational complexity and effectively use limited memory and system resources on mobile devices. Although not directly related to video processing, research [3] provides useful knowledge and methods to optimize the performance of face matching applications directly on video when deployed on devices like a cell phone. This research contributes to the article's mention of application deployment aspects on real devices.

This paper presents a solution for direct video face swapping based on the MediaPipe face landmark recognition solution which is fast, accurate, and capable of running directly on end devices such as smartphones. As a result, users can overlay their faces on videos with natural movement and directly without re-recording.

In the next section, we will discuss detail the facial landmark identification solution and the face swapping technique used in this study.

II. Facial landmark identification solution

To determine the face position on an image/video, we use the MediaPipe Face Mesh solution. This is a machine learning model trained to detect and track 468 different facial landmarks [4]. This is the official document describing the MediaPipe Face Mesh solution published by Google's MediaPipe development team. MediaPipe Face Mesh is a machine learning model trained to detect and track 468 different landmarks on the human face from input photos or videos. This model calculates the exact location of cheekbones, eyes, nose, mouth, chin, and facial contour landmarks. The document details the neural network architecture, training method and performance of the MediaPipe Face Mesh model. At the same time, instructions are also given on how to integrate and use this solution in real applications. Google has published the first and most reputable source of documentation [4] about the MediaPipe Face Mesh solution - an important component used to identify facial landmarks in the face matching application directly on video of this article newspaper. This document helps the article obtain accurate and complete information about the key technology used.

Specifically, the model receives an image or video frame as input. After noise removal and alignment, the system outputs the coordinates (x,y) of 468 facial landmarks. These landmarks cover detailed parts of the eyes, nose, mouth, jaw, chin, and face outline as presented in Figure 1 and Figure 2.

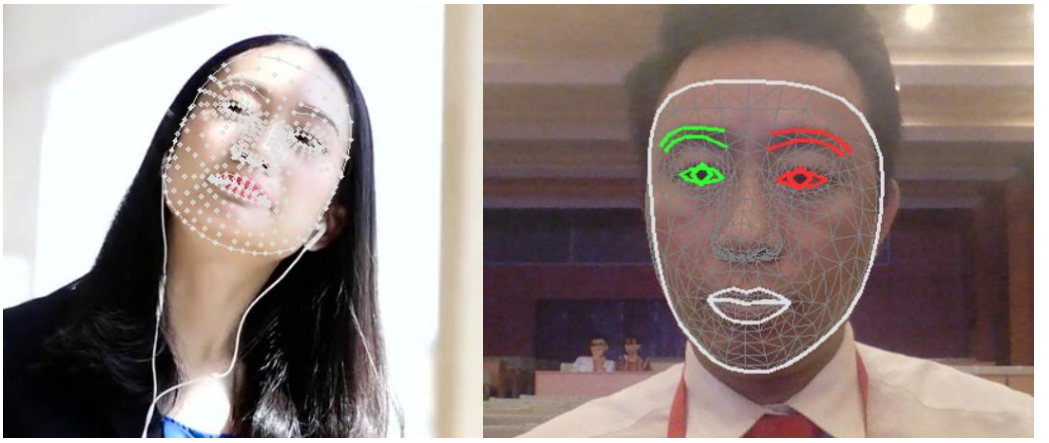


Figure 1. Facial features points

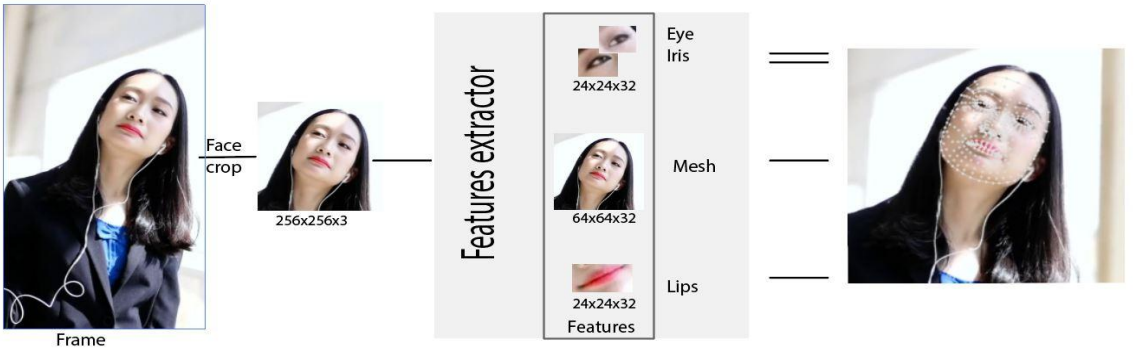


Figure 2. Attention mesh model

The advantage of Face Mesh is its fast operation, with 22ms on CPU or 1−2ms on GPU, thus it can be applied for direct video processing. Additionally, its accuracy is very high, with an average error of less than 1% compared to manually marked landmarks [5]. In a blog posted by Samuel Proell about using MediaPipe to recognize facial landmarks. In the article, the author evaluates and compares the accuracy of the MediaPipe Face Mesh solution with hand-marked facial feature points on a test dataset. The results show that MediaPipe Face Mesh achieves very high accuracy, with an average error of just under 1% compared to manual marking. The article also deeply analyzes the advantages of MediaPipe such as the ability to run on many different platforms (web, mobile, desktop), operate fast at 40-60fps, and support multi-face recognition. This makes MediaPipe Face Mesh an optimal choice for many face-related applications such as face matching, task tracking, AR photo filtering, etc. Citation [5] provides data and evaluation the fact about the high accuracy of MediaPipe Face Mesh helps highlight the advantages and suitability of choosing this technology for the application of live face swapping on video in the article.

Therefore, the MediaPipe Face Mesh is an ideal choice for face swapping tasks.

III. Triangulation and image warping technique

To overlay the source face onto the

target face, we use the triangulation and image warping technique. This includes the following steps:

Step 1: Triangulation

- This process aims to divide the image into a set of interconnected triangles.

Typically, the vertices of these triangles are placed at important points on the image, such as outlines, corners, or key points. A common method for creating triangulation is using the Delaunay Triangulation algorithm [6] as illustrated in Figure 3.

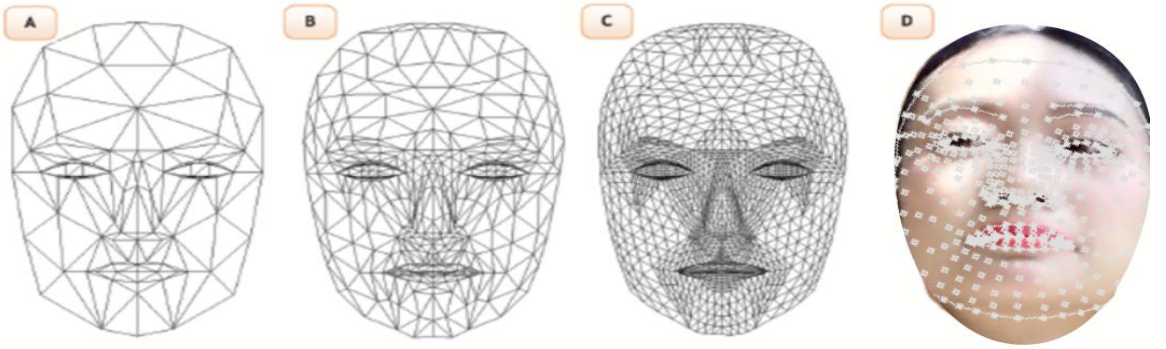


Figure 3. Triangulation of face area

Shewchuk [6] presents in detail the principles of the Delaunay Triangulation algorithm, along with algorithm design and optimization techniques to create high-quality triangular meshes that meet the criteria for triangle shape, side size, uniformity, etc.

- The research also describes a practical implementation of the Delaunay Triangulation algorithm in the program Triangle - a high-quality and efficient two-dimensional triangle mesh generator.
- The reputable study [6] is an important reference on the Delaunay Triangulation algorithm - a technique used in the article to create a triangular mesh covering the face area, which serves as the foundation for the process face transplant.

Step 2: Calculate the corresponding transformation for each triangle

- For each triangle in the triangulation grid of the original and target image, calculate the corresponding transformation (usually a linear coordinate transformation) so that the vertices of the triangle in the original image match the corresponding vertices in the target image [7]. Hartley et al. [7] is a classic document on multi-view geometric theory in computer vision. Chapter 4 of the book describes 2D affine transforms in detail, including the mathematical foundations, properties, and calculation of affine transform matrices. The authors also mention the application of affine transform in many problems such as image calibration, image stitching, coordinate conversion, etc.
- This reference provides the article with a solid theoretical basis on the affine transformation method used to warp and merge facial triangles, helping to improve the scientific and convincing nature of the research.

Step 3: Warp image

- Apply the calculated transformation for each triangle in the original image's triangulation grid to create the target image.
- This process may include techniques such as interpolation, filtering, or resolution adjustment to ensure image quality. Based on the landmarks identified in Step 2, build a grid of triangles covering the entire face. Each triangle is defined by 3 landmark points forming the vertices. Take each triangle on the source face and the corresponding triangle on the target face. Use the affine warp equation to transform the source triangle to fit the target triangle frame.
- The equation takes three vertices of the triangle as anchor points, then calculates the transformation matrix mapping pixels from one triangle to another.
- Apply the affine transformation to all pixels inside the triangle to "warp" them to match the target triangle.
- This process is repeated for all corresponding triangle pairs to gradually merge the individual pieces into a new photo/face. The result is the source face but the size, shape, and movement match the target face.
- In short, we will apply an affine transformation to warp the source image, stitching each piece into the target face. Use the "seamless cloning" technique to join the fused fragments [8].

IV. Face swapping application deployment

To build the face swapping application, we use Python along with the OpenCV and MediaPipe libraries. The deployment steps are shown in Figure 4:

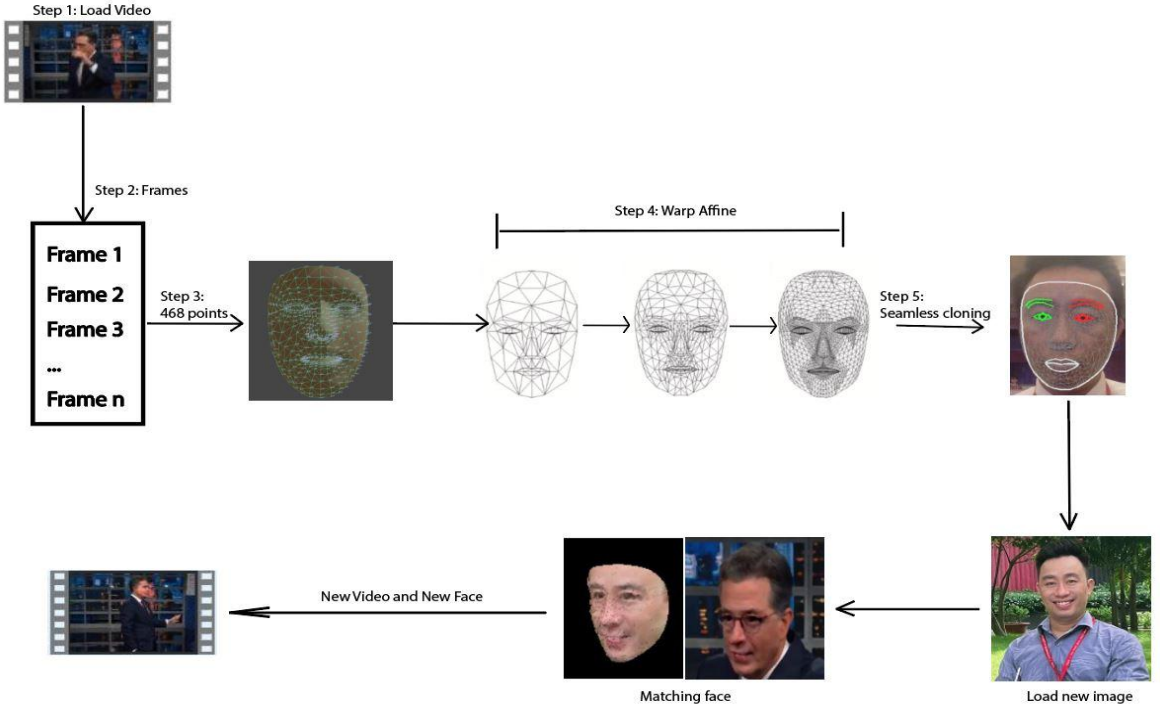


Figure 4. Face matching deployment model

Step 1: Load the input video and initialize processing objects. Extract each frame for processing.

Step 2: Use the Mediapine Face Mesh solution to detect the face and determine the coordinates of 468 landmarks on each frame.

Step 3: Build a mesh of triangles/ covers covering the entire face based on the landmarks.

Step 4: Iterate through each pair of corresponding triangles between the source and target faces. Apply affine warp to merge the small parts together.

Step 5: Combine the patches using the "seamless cloning" algorithm for smooth blending. Then display the face matching result.

Step 6: Repeat the above steps for all subsequent frames of the input video.

V. Experimental results and discussion

To evaluate the performance of the live video face swapping system, we conduct several experiments with input video datasets varying in resolution, lighting conditions, and number of faces.

Experimental setup

Test equipment includes:

• CPU: Intel Core i5-7300 2.5GHz

• GPU: NVIDIA GeForce GTX 1050

• RAM: 16GB

Input video datasets:

• Dataset 1: Video 1080p, 30fps, 2 faces, good lighting.

Images are captured from the video as shown in Figure 5

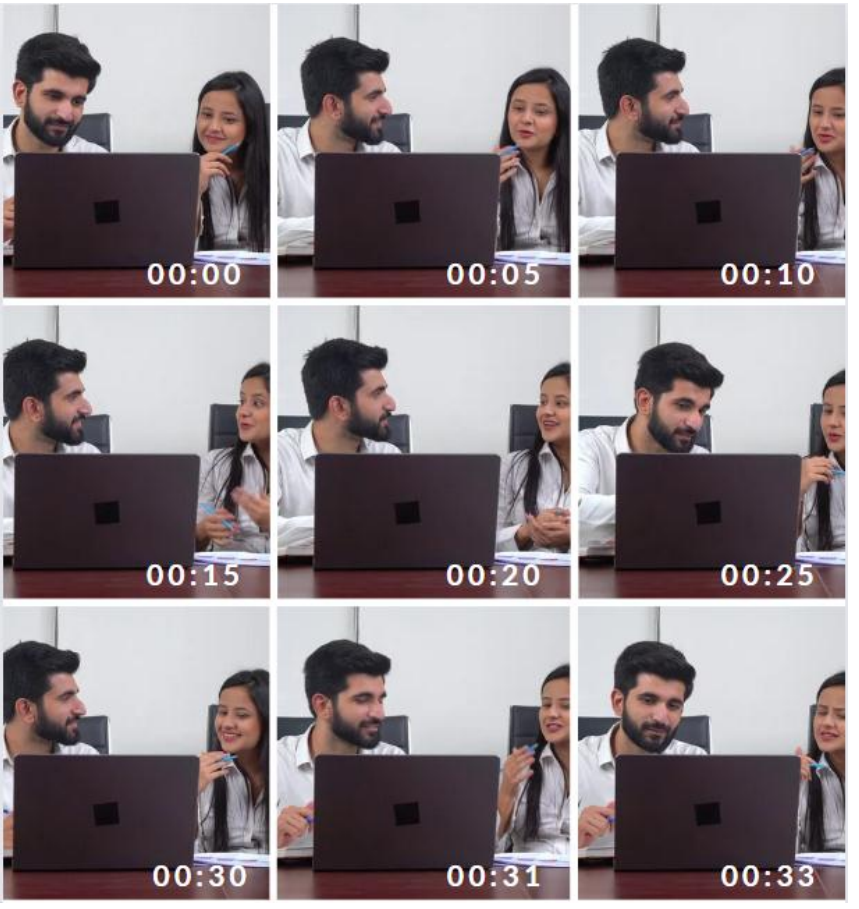


Figure 5. Dataset 1

• Dataset 2: Video 720p, 60fps, 1 face, low light

Images are captured from the video as shown in Figure 6



Figure 6. Dataset 2

• Dataset 3: 4K video, 30fps, 4 faces, medium light

Images are captured from the video as shown in Figure 7



Figure 7. Dataset 3

Results and Evaluation

Video processing speed: In general, the system achieves a fairly high processing speed of 20-25 frames per second, enough to process video in real time. Exact speed for data as illustrated in Figure 8.

• Dataset 1: 22 fps

• Dataset 2: 25 fps

• Dataset 3: 18 fps

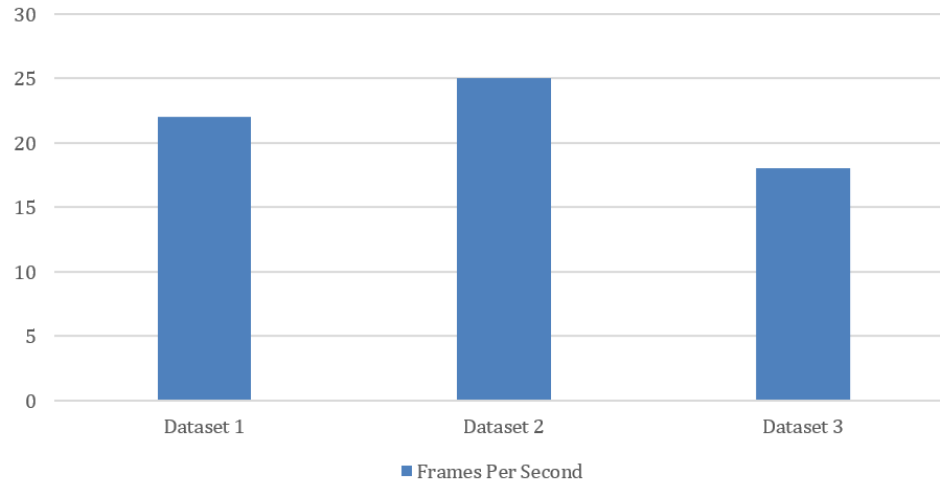


Figure 8. Video processing speed

Speeds are slower for 4K video due to greater computational load on the hardware.

Performance using GPU: To increase performance, we leveraged the NVIDIA GTX 1050 GPU for image processing and face matching calculations. The result are shown in Table 1

Table 1. Performance compared GPU and CPU

No.	Factors	GPU Performance Compared to CPU
1	Simple image transformation	Increase 2-4 times
2	Complex image operations	Increase 4-8 times
3	Feature calculation	Increase 5-10 times
4	Overall	Increase 3-6 times

The results show a fairly high performance gain:

• Increase by 4-8 times compared to using CPU [9] alone for complex image transformations and calculations.

• Increase by 5-10 times in calculating facial feature points.

• Increase by 3-6 times in overall performance.

Optimize input image size: Testing

with a 1920x1080 input image shows that using a smaller image with a size of 960x540 helps increase processing performance by about 3-4 times. Reducing to 640x360 gives a 5-7x increase but reduces the output image quality. The result are shown in Table 2

Table 2. Performance when optimizing input image size

No.	Size	Performance
1	1280×720	Increase 1.5-2 times
2	960×540	Increase 3-4 times
3	640×360	Increase 5-7 times

Face collage quality: In general, face fusion results are quite realistic and smooth thanks to the use of seamless cloning technique. However, there are still some shortcomings such as imperfect matching with areas with complex lighting, differences in skin color and texture between the source and target faces.

Discussion: Experimental results show that the face matching system directly on video works quite effectively, meeting real-time processing requirements by taking advantage of the GPU and optimizing the input image size. However, there is still room for improvement in terms of speed and quality of grafting. Future development directions may focus on:

• Optimize the algorithm, further exploiting the high-performance parallel computing ability of GPU.

• Techniques to simulate details such as skin, muscles, and lighting to enhance the naturalness of the face composite effect.

• Develop solutions for face matching on many different platforms such as web, mobile, and desktop.

In summary, this study has presented an effective solution to the problem of face matching on video, contributing a new approach to this challenging field of face matching.

VI. Conclusion

The article presented a face swapping application directly on video using Google's MediaPipe Face Mesh solution. The application combines facial landmark recognition, triangulation mesh creation and image warping techniques to match the source face to the target face. The result is a photo/video with the source face but the same position and movement as the target face.

Experimental results show that the system has high processing speed, reaching 20-25 frames/second, suitable for real-time video processing. Taking advantage of NVIDIA GTX 1050 GPU increases performance by 3-6 times compared to using CPU alone. Optimizing input image size also increase performance by 5-7 times. In terms of quality, the face transplant results are quite realistic thanks to the seamless cloning technique, although there are still some limitations with complex lighting cases or differences in skin color.

The main contribution of the research

is the face swapping solution on live video, allowing users to experience the face swapping effect with natural movements without re-recording. This is one of the first solutions to successfully apply face swapping technology for video processing, opening up many potential applications in augmented reality, entertainment or video communication.

Future development directions include further optimizing the face swapping algorithm in terms of speed and naturalness of the effect using detailed simulation techniques of skin, muscles, and lighting. Besides, expanding solution deployment on many other platforms such as web, mobile or desktop will also open up many new application opportunities.

In summary, this research has provided a new and effective approach to the problem of face swapping on video - an area that still has many challenges and potential for development. The results achieved are not only meaningful in the field of image/video processing, but also open up application opportunities in many other fields such as virtual reality, entertainment, communications or security.

REFERENCES

[1] Zollhöfer, M., Thies, J., Garrido, P., Bradley, D., Beeler, T., Pérez, P., ... & Theobalt, C. (2018). "State of the Art on Monocular 3D Face Reconstruction, Tracking and Applications." Computer Graphics Forum, 37(2), 523-550.

[2] Nirkin, Y., Masi, I., Tuan, A., Hassner, T., & Medioni, G. (2018). "Face Swapping in the Wild." IEEE Transactions on Biometrics, Behavior, and Identity Science, 1(1), 14-26.

[3] Agarwal, A., Bharath, A. A., Jaiantililiken, A. (2019). "SwapIt: Face Swapping on Mobile Devices." IEEE Transactions on Image Processing, 28(10), 4770-478

[4] Google MediaPipe Team (2022). MediaPipe Face Mesh. https://github.com/google/MediaPipe/blob/master/docs/solutions/face_mesh.md

[5] Proell, S. (2022). Facial landmark detection made easy with MediaPipe. https://www.samproell.io/posts/yarppg/yarppg-face-detection-with-MediaPipe/

[6] Shewchuk, J. R. (1996). Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In Applied computational geometry towards geometric engineering (pp. 203-222). Springer, Berlin, Heidelberg.

[7] Hartley, R., & Zisserman, A. (2003). Multiple View Geometry in Computer Vision (pp. 107-119). Cambridge University Press.

[8] Pérez, P., Gangnet, M., & Blake, A. (2003). Poisson image editing. ACM Transactions on Graphics, 22(3), 313-318.

[9] Owens, J. D., Houston, M., Luebke, D., Green, S., Stone, J. E., & Phillips, J. C. (2008). GPU computing. Proceedings of the IEEE, 96(5), 879-899.

Ngày nhận bài: 27/3/2024 Ngày chấp nhận đăng: 19/5/2024