 ISSN 2664-9969

15. Bettemir, Ö. H., Sonmez, R. (2015). Hybrid Genetic Algorithm with Simulated Annealing for Resource-Constrained Project Scheduling. *Journal of Management in Engineering, 31 (5)*. doi: https://doi.org/10.1061/(asce)me.1943-5479.0000323
16. *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)* (2021). Project Management Institute, Inc.
17. Ahmed, R., Shaheen, S., Philbin, S. P. (2022). The role of big data analytics and decision-making in achieving project success. *Journal of Engineering and Technology Management, 65,* 101697. doi: https://doi.org/10.1016/j.jengtecman.2022.101697

Iryna Blyznyukova, *Postgraduate Student, Department of Computer Science and System Analysis, Cherkasy State Technological University, Cherkasy, Ukraine, ORCID: https://orcid.org/0000-0001-7486-7983*

✉***Pavlo Teslenko,*** *PhD, Associate Professor, Department of Artificial Intelligence and Data Analysis, Odesa Polytechnic National University, Odesa, Ukraine, e-mail: teslenko@op.edu.ua, ORCID: https://orcid.org/0000-0001-6564-6185*

✉*Corresponding author*

UDC 004.[4+896]
DOI: 10.15587/2706-5448.2023.277933

Dmytro Ostrovka,
Vasyl Teslyuk

DEVELOPMENT OF A METHOD FOR CHANGING THE SURFACE PROPERTIES OF A THREE-DIMENSIONAL USER AVATAR

The object of study of this research paper is the processes of changing the properties of three-dimensional surfaces of a user avatar in real time. In the course of this work, the research addressed the limitations of existing solutions for synthesizing three-dimensional user avatars, particularly in terms of realism and personalization on mobile devices. Furthermore, the study tackled the challenge of efficiently adjusting color attributes without compromising the underlying texture information, ultimately enhancing user experience across various applications such as gaming, virtual reality, and social media platforms. A method consisting of three key components is proposed: pre-designed 3D models, multi-layer texturing, and software and hardware implementation. The multilayer texturing approach includes different texture maps, such as diffuse and occlusion maps, which contributes to the smooth integration of texture attributes and the overall realism of 3D avatars. The real-time change of surface properties is achieved by mixing the diffusion map with other texture maps using the Metal hardware accelerator, allowing users to efficiently adjust the color attributes of their 3D avatars while preserving the underlying texture information. The paper presents a software algorithm that uses the SceneKit game engine and the Metal framework for rendering 3D avatars on iOS devices. The result of the developed method and tool is a mobile application for the iOS platform that allows users to modify a digital 3D avatar by changing the model's colors. The paper presents the results of testing the proposed methods, means and developed application and compares them with existing solutions in the industry. The developed method can be implemented in areas such as gaming, virtual reality, video conferencing, and social media platforms, offering greater personalization and a more immersive user experience.

Keywords: *digital face, game engine, three-dimensional face modelling, digital avatar, semi-realistic avatar.*

Received date: 12.03.2023
Accepted date: 25.04.2023
Published date: 29.04.2023

How to cite
Ostrovka, D., Teslyuk, V. (2023). Development of a method for changing the surface properties of a three-dimensional user avatar. Technology Audit and Production Reserves, 2 (2 (70)), 10–15. doi: https://doi.org/10.15587/2706-5448.2023.277933

1. Introduction

In recent years, 3D avatar synthesis has attracted considerable attention from many software developers due to its wide range of applications and the growing demand for realistic digital representations of human faces. Thanks to the rapid development of computer vision, 3D graphics and machine learning technologies, 3D face avatar creation has become an important research area with the potential to revolutionize industries such as gaming, virtual or augmented reality, teleconferencing and social media platforms. The ability to create personalized and expressive 3D face avatars not only provides a more immersive user experience, but also creates a sense of presence in the virtual environment, which is crucial for effective communication. In addition, the development of efficient and accurate methods for generating 3D avatars can contribute to advances in other areas such as facial analysis, emotion recognition, and biometric authentication. The search for new methods and algorithms for creating 3D face avatars remains an active and important topic of modern research, shaping the future of digital human images and interaction.

In turn, this work aims to solve the problems associated with rendering synthesized 3D models, paying attention to correct visualization, especially on mobile platforms. Mobile devices have become one of the most common platforms for using 3D avatars, given their wide availability and accessibility. Therefore, it is important to ensure that rendering methods are optimized for these devices, which will allow for the creation of high-quality and visually appealing avatars while maintaining the efficiency of computer resources. Existing research in the field of 3D avatar synthesis, although it has made significant progress, is usually tested on PCs or laptops [1], whose hardware resources are very different from mobile devices.

By addressing rendering issues, including high color accuracy, this study aims to improve the overall quality and consistency of 3D avatars on mobile platforms. New methods are explored and optimized to improve the rendering process, taking into account factors such as lighting, shading and texture mapping, which have a direct impact on the perceived realism of avatars. In addition, the integration of advanced rendering techniques with the latest advances in computer vision is being explored to develop a robust and efficient pipeline specifically adapted for mobile devices.

In addition to the above tasks, this paper also addresses the problem of efficient real-time color modification of 3D face avatars. Current research in this area is mainly focused on creating 3D avatars the first time, without considering the subsequent modification of these avatars. As a result, when the initial texture synthesis is inaccurate or does not meet users' expectations, they are left without the ability to change or improve the appearance of the avatar.

To solve this problem, this paper proposes a new method that allows users to make changes to the color attributes of their 3D avatars, such as skin tone, hair and eye or lip color, in real time. The developed method incorporates advanced rendering approaches using the Metal framework and the SceneKit game engine, which allows users to interactively improve and personalize their avatars even after the initial synthesis is complete. This gives users the ability to correct potential inaccuracies in texture synthesis and better match the avatar's appearance to their preferences or purpose.

By addressing the problem of efficient runtime color modification, this work extends the capabilities of existing 3D face avatar generation methods and gives users more control over their digital images. This focus not only addresses the limitations of existing approaches, but also paves the way for more universal and user-centric solutions for creating and modifying 3D face avatars by leveraging the capabilities of the Metal framework and the SceneKit game engine.

Existing research on highly realistic 3D reconstruction of faces and characters has made significant contributions to the field, including the reproduction of skin tone and eye color [2–5]. Despite significant progress and the use of artificial neural networks for reconstruction, these approaches sometimes fail to cover the full range of skin tones and eye color, potentially leading to biased representations.

At the same time, studies on reconstructing semi-realistic and cartoon characters, such as [6–8], have also faced problems in preserving the diversity of skin tones, eyes, and other facial elements. These methods, although effective for converting 2D images into 3D models and avatars, often face difficult lighting conditions or unconventional artistic styles. Work [1] also describes the problems of detecting the correct eye color in the input data due to the small number of pixels and, as a result, generating a model with the wrong shade.

A number of research papers study the issue of coloring using neural networks, including three-dimensional face models [9, 10]. Although these papers focus on model synthesis and subsequent coloring based on an achromatic input photo, the described approach can also be applied to modify the color of a chromatic model. In [9], a number of potential limitations of this approach are noted, such as sensitivity to occlusions and problems with identifying facial makeup. Inaccurate coloring can occur if the face is partially occluded or if there are accessories such as glasses or a hat.

In addition to possible color mismatches between the synthesized model and the input data, existing solutions in the industry usually do not provide for correction or modification of the results. The method proposed in this paper addresses these limitations by allowing users to edit skin tone, eye and lip color in real time using GPU acceleration, providing a more accurate and diverse image of the user's avatar, as well as flexibility in user customisation.

Therefore, *the aim of the research* is to develop a method and means for changing the color properties of 3D face avatars in real time.

The main hypothesis of the research is a method for changing the surface properties of a 3D image of a user's avatar, which, by mixing texture maps of images and focusing algorithms on a low-level hardware accelerator, provides an extension of the functions of processing 3D surfaces in real time.

Practical value of the work – the developed means provide effective modification of the surface properties of a three-dimensional user avatar in real time, which will allow the creation of new and improvement of existing mobile applications in the fields of gaming, virtual reality, video conferencing and social media platforms.

2. Materials and Methods

The object of research is the processes of changing the properties of three-dimensional surfaces of a user's avatar in real time. To effectively conduct the investigation, various tools and technologies were employed. A pre-developed 3D model, created in Blender, served as the foundation for the avatar. Blender, a 3D modeling software, was utilized for this purpose. To achieve realistic and dynamic changes in the avatar's appearance, sophisticated texture layering and mixing techniques were implemented.

For the development and testing of the application, xCode, Apple's integrated development environment (IDE), was utilized. Swift, a powerful programming language tailored for iOS development, was chosen for the project's implementation. Metal, a low-level hardware-accelerated graphics API for Apple devices, was employed to render the 3D graphics efficiently. Additionally, SceneKit, a high-level 3D graphics framework, facilitated the integration of 3D assets and enabled the manipulation of 3D objects within the application.

This research incorporated performance testing and optimization to ensure efficient GPU and energy usage. xCode's built-in profiling tools and Instruments were utilized for this purpose. These tools enabled the monitoring, analysis, and optimization of the application's resource usage, thereby guaranteeing optimal performance in terms of GPU and energy consumption.

ISSN 2664-9969

3. Results and Discussion

3.1. Development of a method for changing the surface properties of a three-dimensional user avatar. In this work, a method for changing 3D surfaces has been developed, which consists of the following components: a developed 3D model, generated textures, and software and hardware implementation. A schematic representation of the method is shown in Fig. 1.

The pre-designed 3D model represents the shape of the user's face and includes mixed shapes and/or geometric morphs to enable interaction with the model in AR/VR tools [11], as well as to provide modification of the model's geometry [12].

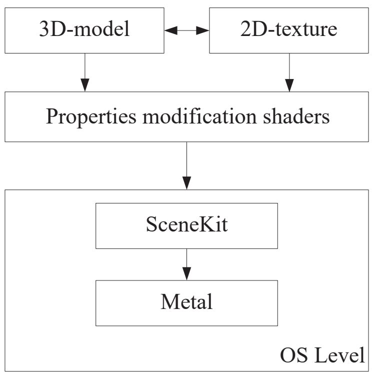


Fig. 1. Components of the 3D surface modification method

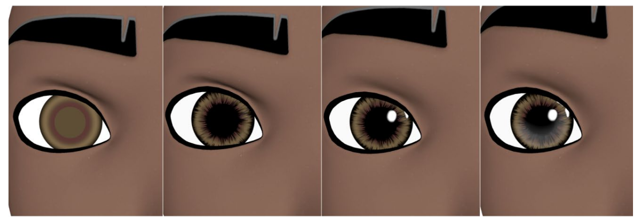
As for textures, a multi-layered texturing approach is applied, which includes a diverse set of texture maps such as diffuse maps, occlusion maps, etc. This multi-layered strategy ensures a seamless integration of different texture attributes, which contributes to the overall realism and visual appeal of the resulting 3D facial avatars.

On the software side, the method involves real-time modification of the diffuse texture, which is then blended with other texture layers using the Metal hardware accelerator. This real-time manipulation allows users to efficiently customize the color attributes of their 3D avatars, while maintaining the underlying texture information. Thanks to the use of the Metal framework, the rendering process remains resource-efficient and optimized for mobile platforms.

To change the eye color of the 3D avatar, a separate texture model was developed, which also involves software interaction with the Metal framework. In contrast to the approach described above, this solution allows mixing colors and generating gradient effects, providing a more natural look than traditional monochrome implementations. The developed method can also include texture layers to increase the visual authenticity of the eye's reflection. In the course of the work, a mirror reflection effect was added to the eye texture model, which allows to convey the depth and emphasize the material properties of the 3D eye model during its rendering, as shown in Fig. 2, *c*. The formula for describing the gradient filling of the cornea consists of three components and is mathematically represented as a continuous function *Gradient*(p), where p is the position of the gradient, which varies from 0 to 1:

$$Gradient(p) = (1-p)\cdot C_1 + p\cdot C_2,$$
$$\text{for } 0 <= p < 0.5,$$
$$Gradient(p) = (2-2\cdot p)\cdot C_2 + (2\cdot p - 1)\cdot C_3,$$
$$\text{for } 0.5 <= p <= 1,$$

where C_1, C_2 and C_3 are RGB color vectors representing the gradient colors, and *Gradient*(p) is a vector of RGB colors at gradient position p. The *Gradient*(p) function is implemented in a shader program that runs on the graphics processor to create a continuous and smooth gradient across the surface of the 3D eye model. The values of the p parameters can be determined by the coordinates of a previously created texture map.



a *b* *c* *d*

Fig. 2. An example of texture layering during eye rendering: *a* – without textures; *b* – mixing the cornea texture; *c* and *d* – mixing the texture with a glare

Thus, the proposed method for modifying 3D surfaces consists of a synergistic combination of pre-designed 3D models, multi-layer texturing, and real-time software modification. This method allows to expand the modification possibilities, while maintaining the realism and authenticity of 3D avatars, demonstrating the potential for more informed and user-oriented solutions in the field of 3D avatar generation and modification.

3.2. Development of a means for changing surface properties of a three-dimensional image of a user's avatar. To achieve efficient and visually appealing results, the approach developed in this work to render models for changing facial and eye color on the iOS mobile platform uses the SceneKit game engine and the Metal framework. The algorithm of the proposed tool is shown in Fig. 3.

The algorithm shown in Fig. 3 describes the following steps for changing the properties of three-dimensional avatar surfaces:

Step 1. Initialize the SceneKit game engine, which provides high-level 3D graphics capabilities, and the Metal framework for low-level GPU management that optimizes performance on iOS devices.

Step 2. Uploading a pre-designed 3D face model consisting of mixed shapes and/or geometric morphs [11, 12] to reflect the user's unique facial features.

Step 3. Loading shaders for modifying facial properties. This step involves setting up and adding all the necessary texture layers to the scene.

Step 4. Loading the shaders for modifying the eye properties of the avatar model. As in the previous step, at this stage, textures are blended, but a different shader is used exclusively to modify the eye colors.

Step 5. Rendering the entire scene, including models, textures, and shaders, using the SceneKit game engine.

Step 6. Manage the scene at a high level using the built-in methods of the SceneKit game engine and Metal to modify shader parameters.

Examples of application of the developed method and tool are shown in Fig. 4 and Fig. 5. It is worth noting that Fig. 4 shows various modifications of eye colors using the texture blending model along with the generation of gradient color effects. This was achieved by using the

Metal framework, which allows to dynamically adjust the color and effectively reproduce these changes.

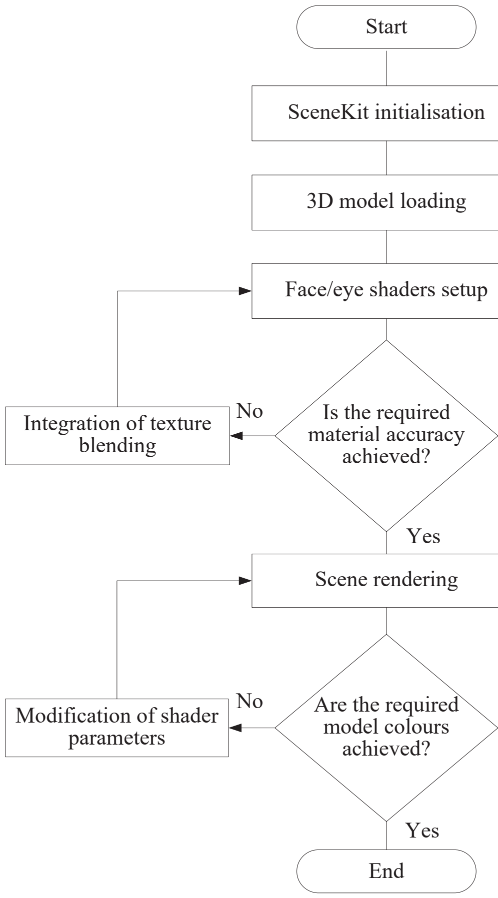


Fig. 3. Algorithm of the method of changing the properties of a three-dimensional avatar

Fig. 4. An example of changing the eye color of a model in real time

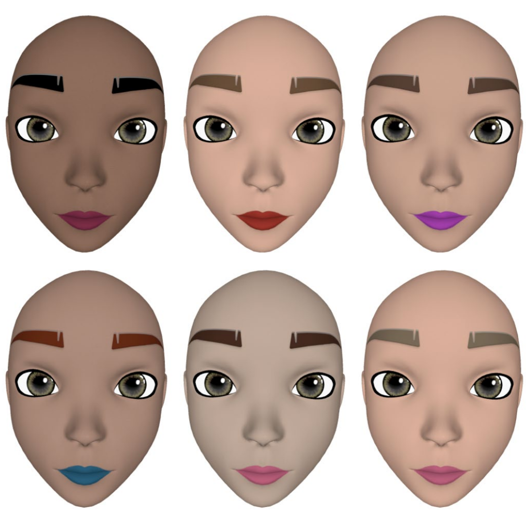


Fig. 5. An example of changing the properties of face elements

Fig. 5 shows the flexibility of the developed method, which allows changing individual parts of the model, such as eyebrows, lip color and skin color.

The modifications depicted occur in real time, while maintaining a frame rate of 60 fps, demonstrating the effectiveness of the joint operation of the Metal framework and the SceneKit game engine.

The proposed method can also be used to specify any area of the model for real-time modifications, along with the use of texture blending techniques. As a result, users can easily customize specific areas of the model, resulting in a higher level of personalization and a more immersive user experience.

In addition, the use of texture blending methodology within the proposed approach allows for the integration of multiple texture layers, resulting in a more realistic and detailed representation of the model. This feature demonstrates the advanced capabilities of the Metal framework in solving complex rendering tasks while optimising performance on iOS devices. Thus, the proposed method serves as a powerful tool for creating and modifying 3D avatars that meets a wide range of applications and user requirements with high flexibility and efficiency.

In the course of this work, performance testing was carried out on three different mobile devices: iPhone X under iOS 15.6.1, iPhone 12 Pro under iOS 16.1.1, and iPad Pro 11' (1st generation) under iOS 16.3. The results showed stable performance: all devices reached a stable 60 frames per second with consistent modification of the avatar color attributes, as shown in Fig. 6, which displays the system window for profiling and configuring iOS applications in the xCode environment. It is worth noting that up to 10 frames were lost during the rapid change of eye color on the iPhone 12 Pro device.

This may indicate that there is room for improvement in the shader for changing the properties of the eye model, which can be a computationally challenging task. In addition, the performance of the iPhone 12 Pro could be affected by the less stable version of iOS 16. Nevertheless, the results on the iPhone X and iPad Pro demonstrate the stability and effectiveness of the proposed method for modifying the color attributes of 3D avatars on mobile devices.

It is important to note that during the performance testing, it was found that all devices had a high energy impact, and the GPU had high load values because the rendering method takes place on it, as shown in Fig. 7, which shows the energy profile of the developed application when testing the proposed method on an iPad 11'.

The obtained results highlight the importance of optimization and efficient resource management in order for the developed method to remain effective and efficient on mobile devices, especially in terms of energy consumption. In order to reduce the energy impact of the proposed method, it may be necessary to implement energy-saving techniques such as dynamic power management or GPU-accelerated algorithms to optimize GPU performance and minimize power consumption. Reducing the number of polygons in the synthesized model and using smaller texture maps can also be a potential method of optimizing the power consumption of the device, but it should be noted that a significant reduction in these elements may lead to a decrease in the quality of the rendered model.

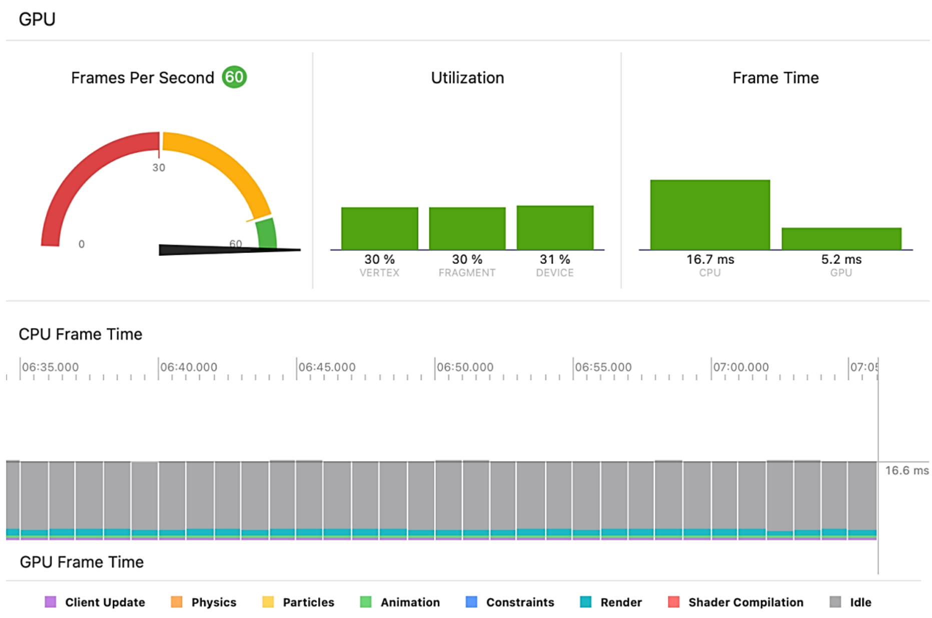


Fig. 6. GPU performance during testing of the developed models on iOS devices

Fig. 7. Indicators of device power consumption during testing of the developed models on iOS devices

3.3. Discussion of the research results. Compared to solutions for highly realistic 3D reconstruction of faces and characters [2–5], the method developed in this paper offers advantages in terms of color accuracy, visual diversity and user-friendliness, as it allows to expand the range of skin tones and eye color of a 3D avatar and avoid biased images.

The methods developed in the course of this work involve interaction with pre-designed 3D models that are stylistically similar to solutions for creating semi-realistic and cartoon avatars [6–8]. Since these solutions usually use approaches to create 2D textures of the user's face similar to those used in highly realistic reconstructions, the original avatar model may have visual inaccuracies, such as incorrect eye color [1] or other facial elements. In contrast, the developed method solves these problems by allowing users to edit skin tone, eye, lip, and eyebrow color in real time using GPU acceleration. This allows for a more accurate and varied representation of the user's avatar, as well as greater flexibility in user customization. Real-time editing capabilities provide users with a more engaging and interactive experience, allowing them to create personalized solutions that match their unique preferences.

A widely used solution in the scientific literature for simulating synthetic avatars is game engines based on OpenGL or DirectX [13]. However, the use of these tools is not the most optimal choice for mobile platforms [14] because their operating systems, such as iOS, have their own hardware accelerator. That is why an important component of the developed method is the use of the built-in Metal framework and the SceneKit game engine, which is the most efficient approach to rendering for iOS mobile applications [14]. This ensures smooth and fast interaction of user avatar modification on devices, keeping the system productive and optimized in terms of the use of mobile device resources.

The current research had a number of limitations and shortcomings, which should be taken into account when the suggested solutions are put into reality or when doing additional theoretical research. One limitation is the focus on iOS devices, which may not be representative of the performance and capabilities of other platforms. Although the developed method could potentially be applied to other mobile platforms using different shader languages and game engines, the current study only considers its implementation on iOS devices. This could restrict the generalizability of the findings and their application to other devices, operating systems, or hardware configurations. Additionally, while the potential for hair change was mentioned in the work, the study does not explore the possible benefits of creating a separate shader for hair rendering. Implementing a dedicated shader for hair rendering could result in more visually natural and appealing effects, enhancing the overall experience for users. Future research could expand upon the current findings by exploring cross-platform compatibility, the implementation of the developed method on other devices and operating systems, and the development of dedicated shaders for specific avatar components, such as hair.

4. Conclusions

In the course of the work, the existing solutions for the synthesis of three-dimensional user avatars in the industry and on mobile devices were analysed. The problems of adjusting and changing the properties of the surfaces of the original models are identified. A hypothesis is presented regarding a possible solution to this problem on mobile devices using built-in methods and means to ensure high quality rendering. The purpose and main tasks of the study to test the presented hypothesis are formed, which were successfully completed in the course of the research.

A method for modifying the properties of 3D surfaces has been developed that offers advanced avatar modification capabilities while maintaining a high degree of realism and reliability of the resulting 3D face avatars.

The developed method for modifying the properties of 3D surfaces consists of three key components: a pre-designed 3D model, generated textures, and software and hardware implementation. This method is aimed at improving the editing and realism of 3D avatars, in particular, for changing the color of facial and eye elements on mobile platforms. The proposed solution involves a multilayer approach to texturing using various texture maps, such as diffuse and occlusion maps, which contributes to a smooth combination of texture attributes and the overall realism of 3D avatars.

In the course of the work, was developed a software algorithm that uses the SceneKit game engine and Metal framework for rendering 3D avatars on iOS devices. SceneKit simplifies the rendering process by providing high-level 3D graphics capabilities, while Metal offers precise control over the rendering pipeline to optimize performance. By using shaders, this approach allows for real-time programmatic changes to color attributes, while maintaining resource efficiency and minimizing the size of the application.

The result of the developed method and algorithm is a mobile application for the iOS platform, which allows the user to modify a digital 3D avatar by changing the model's colors. The developed system is scalable, and due to the use of a built-in game engine and hardware accelerator, the rendering and modification algorithm is performed in the most optimized way on a mobile platform, compared to common solutions in the industry. The system can be implemented in augmented and virtual reality solutions. The example of the developed method demonstrates potential modifications of the user's facial elements, such as changing the color of lips, eyes, eyebrows and skin. During the testing of the developed system and methods, stable frame rates (60 frames per second) were obtained, as well as high energy consumption of the device and conclusions about the potential improvement of these metrics were made.

The developed method can be applied to numerous applications in industries such as gaming, virtual reality, video conferencing, and social media platforms. By enabling users to easily customize specific areas of their 3D avatars, this approach offers increased personalization and a more immersive user experience.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

The study was self-funded.

Data availability

The manuscript has no associated data.

References

1. Wu, Q., Zhang, J., Lai, Y.-K., Zheng, J., Cai, J. (2018). Alive Caricature from 2D to 3D. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 7336–7345. doi: https://doi.org/10.1109/cvpr.2018.00766
2. Cai, H., Guo, Y., Peng, Z., Zhang, J. (2021). Landmark Detection and 3D Face Reconstruction for Caricature using a Nonlinear Parametric Model. *Graphical Models, 115,* 101103. doi: https://doi.org/10.1016/j.gmod.2021.101103
3. Egger, B., Smith, W. A. P., Tewari, A., Wuhrer, S., Zollhoefer, M., Beeler, T. et al. (2020). 3D Morphable Face Models – Past, Present, and Future. *ACM Transactions on Graphics, 39 (5),* 1–38. doi: https://doi.org/10.1145/3395208
4. Li, M., Huang, B., Tian, G. (2022). A comprehensive survey on 3D face recognition methods. *Engineering Applications of Artificial Intelligence, 110,* 104669. doi: https://doi.org/10.1016/j.engappai.2022.104669
5. Sang, S., Zhi, T., Song, G., Liu, M., Lai, C., Liu, J. et al. (2022). AgileAvatar: Stylized 3D Avatar Creation via Cascaded Domain Bridging. *SIGGRAPH Asia 2022 Conference Papers.* doi: https://doi.org/10.1145/3550469.3555402
6. Tiwari, H., Subramanian, V. K., Chen, Y.-S. (2022). Real-time self-supervised achromatic face colorization. *The Visual Computer.* doi: https://doi.org/10.1007/s00371-022-02746-1
7. Xu, S., Yang, J., Chen, D., Wen, F., Deng, Y., Jia, Y., Tong, X. (2020). Deep 3D Portrait From a Single Image. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 7707–7717. doi: https://doi.org/10.1109/cvpr42600.2020.00773
8. Yang, H., Zhu, H., Wang, Y., Huang, M., Shen, Q., Yang, R., Cao, X. (2020). FaceScape: A Large-Scale High Quality 3D Face Dataset and Detailed Riggable 3D Face Prediction. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 598–607. doi: https://doi.org/10.1109/cvpr42600.2020.00068
9. Ye, Z., Xia, M., Sun, Y., Yi, R., Yu, M., Zhang, J. et al. (2023). 3D-CariGAN: An End-to-End Solution to 3D Caricature Generation From Normal Face Photos. *IEEE Transactions on Visualization and Computer Graphics, 29 (4),* 2203–2210. doi: https://doi.org/10.1109/tvcg.2021.3126659
10. Zhang, R., Zhu, J.-Y., Isola, P., Geng, X., Lin, A. S., Yu, T., Efros, A. A. (2017). Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics, 36 (4),* 1–11. doi: https://doi.org/10.1145/3072959.3073703
11. Ostrovka, D., Teslyuk, V. (2023). Synthesis model of a three-dimensional image of a user based on the SceneKit game engine and the USDZ format for IOS. *Scientific Bulletin of UNFU, 33 (1),* 89–94. doi: https://doi.org/10.36930/40330112
12. Ostrovka, D., Teslyuk, V. (2023). A method for dynamically changing the geometry of 3D surfaces in USDZ format with further implementation in the SceneKit game engine. *Scientific Bulletin of UNFU, 33 (2).*
13. Gui, J., Zhang, Y., Li, S. (2016). Realistic 3D Facial Wrinkles Simulation Based on Tessellation. *2016 9th International Symposium on Computational Intelligence and Design (ISCID).* Hangzhou, 250–254. doi: https://doi.org/10.1109/iscid.2016.1064
14. Abdallah, Y., Abdelhamid, A., Elarif, T., Salem, A.-B. M. (2015). Comparison between OpenGL ES and metal API in medical volume visualisation. *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS).* Cairo, 156–160. doi: https://doi.org/10.1109/intelcis.2015.7397213

✉***Dmytro Ostrovka,*** *Postgraduate Student, Department of Automated Control Systems, Lviv Polytechnic National University, Lviv, Ukraine, e-mail: dmytro.v.ostrovka@lpnu.ua, ORCID: https://orcid.org/0000-0003-4818-3822*

Vasyl Teslyuk, *Doctor of Technical Science, Professor, Head of Department of Automated Control Systems, Lviv Polytechnic National University, Lviv, Ukraine, ORCID: https://orcid.org/0000-0002-5974-9310*

✉*Corresponding author*