Indonesian Journal of Electrical Engineering and Computer Science
Vol. 21, No. 1, January 2021, pp. 457~464
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v21.i1. pp457-464

3D face creation via 2D images within blender virtual environment

Ali Salim Rasheed, Rasool Hasan Finjan, Ahmed Abdulsahib Hashim, Mustafa Murtdha Al-Saeedi
Department of Media Technology Engineering, College of Engineering, University of Information Technology and Communications, Baghdad, Iraq

Article Info

Article history:

Received May 22, 2020
Revised Jul 25, 2020
Accepted Aug 14, 2020

Keywords:

Blender
Laplacian filter
Python
Virtual environment
Virtual reality

ABSTRACT

Animation and virtual reality movie-making technologies are still witnessing significant progress to this day. Building and stimulating virtual characters inside these applications is a goal. Build a 3D face via using some special tools inside the virtual world is the most important part of identifying a 3D animation. Keen Tools Face Builder add-on for Blender. Interested in creating a 3D face of a famous figure, artist or the general public by adopting several 2D images added to the virtual blinder software environment. The main problem facing these tools is that they deal with high-resolution and sharpness pictures because some images that contain blurring, the result is to build a 3D face model that contains design distortions and non- clearly. in this proposed paper, build a data set for 2D pictures of a specific character (actor), at a resolution of 1920 x 1080 pixels. These images were caught by the camera, different in sharpness and blurring (four types of blurry). Using the "Laplacian Filter algorithm" and OpenCV Library with Python language, to isolate blurry from sharpness 2D images. Sharpness images used to build a 3D face model that gave real and similar results to the character in the pictures.

Corresponding Author:

Ali Salim Rasheed
Department of Media Technology Engineering
College of Engineering
University of Information Technology and Communications
Baghdad, Iraq
Email: ali.rasheed@uoitc.edu.iq

1. INTRODUCTION

The fast advancement in virtual reality (VR) systems requires the provision in virtual reality (VR) systems requires the provision of a high-definition 3D environment that simulates the real world and provides users of these systems high interaction and immersion performance [1]. Virtual reality systems (VR) is at present popular technology utilized in various areas such as video games, cinema, medical, military training [2]. Blender is a 3D virtual environment (VE) and powerful open-source software for made 3D graphic animations and cinema movies. The most important features of this software, it performed on engineering transformations for animation, Shading, Lighting, Rendering, production of digital films and 3D video game [3].

Blender has added, over time impressive touches in the Digital Cinema and animation movies industry. These films have achieved great success on social media networking and also in cinemas through millions of views, follow-up and win international awards. These movies have aroused the interest of animation developers and those interested in this field. "Spring" is the latest short film produced by Blender Animation Studio.

Blinder is very compatible with virtual reality (VR) technologies. As its virtual 3D viewport environment and its tools, as shown in Figure 1. Provide a large space in the manufacture of virtual worlds as well as the creation of characters, 3D faces are a reflection of real people present in our real world. the main aim to design a virtual environment(VE), its interconnect the virtual world to the real by using artificial inelegance or IoT technologies [4]. virtual environment(VE) empowering designers to virtually implement the thoughts and afterward view it in a real-world, then enable the designers to display their 3D visualizations on a 2D pc screen [5]. The basic approach to making these applications (Blender) is having a virtual environment (VE) designed with specific programming languages (python) simulated with 3D animation toolbox [6].

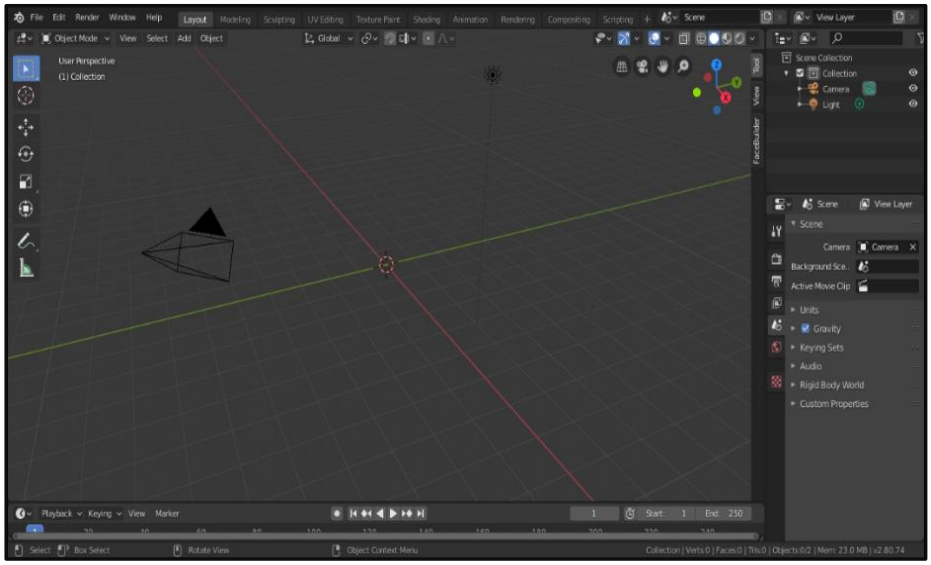

Figure 1. Blender virtual 3d viewport environment, general tools

Blinder Software is an integrated work environment that converts designers' ideas into virtual models programmable by programmers [7]. Building a 3D model like faces inside the virtual environment (VE) requires a certain effort. What's more, this exertion is predictable according to the methods used in the software [8].

In this paper, the goal will be building the 3D model of the human face in the 3D virtual environment (VE) of Blinder Software via 2D images of celebrities, artists, or the public. These faces consider are the main part of creating virtual characters to be used in the animation movies industry or virtual reality (VR) simulation systems. This duty is done through the use of Keen Tools Face Builder to building 3D models of human faces and heads using a couple of photographs. The main aim creation of the 3D face from several of 2D photos. is used should be high resolution, non- blurring and high sharpness. In order to obtain highly sharp images that give advanced results during the use of these tools, we suggested using OpenCV applications, Python language with “Laplacian operator”.

The technological advancement of digital cameras provides high quality captured images with very big data [9]. But the problem is still in the user, it may be unprofessional or not experienced in taking pictures and therefore we get high-resolution images but they contain blur and incorrect angle of photography. The human eyes cannot accurately distinguish image blurring as well as the degree of clarity through the use of digital cameras. It is the main problem we face when 3D faces create within Blender software by using of Keen Tools Face Builder.

Since the official website of Keen Tools Face Builder said (With Face Builder you don't need to be an experienced 3D modeler to create a quality 3D model with clean topology). but At the same time, the official website did not mention that the image should be high-definition, blur-free, sharp, and with a frame size of 1920 x 1080 pixels, what's more, these prerequisites are hard to accomplish by novices or the individuals who don't have an exact computerized camera. It is a real problem that was encountered while designing a 3D face inside Blinder software by using Keen Tools face builder. And we, in the paper, will work to solve this problem programmatically by using OpenCV technologies, “Laplacian filter” with Python language.

2. GENERAL STAGES PIPELINE

The way to create 3D human face inside Blinder is the same way to computer graphic pipeline used in animation creation and virtual reality (VR) systems build. The pipeline theory means the output from before stage is fed as the input of the next stage. These systems use to preparing of information to upgrade the 3D face model rendering [10], as shown in Figure 2.

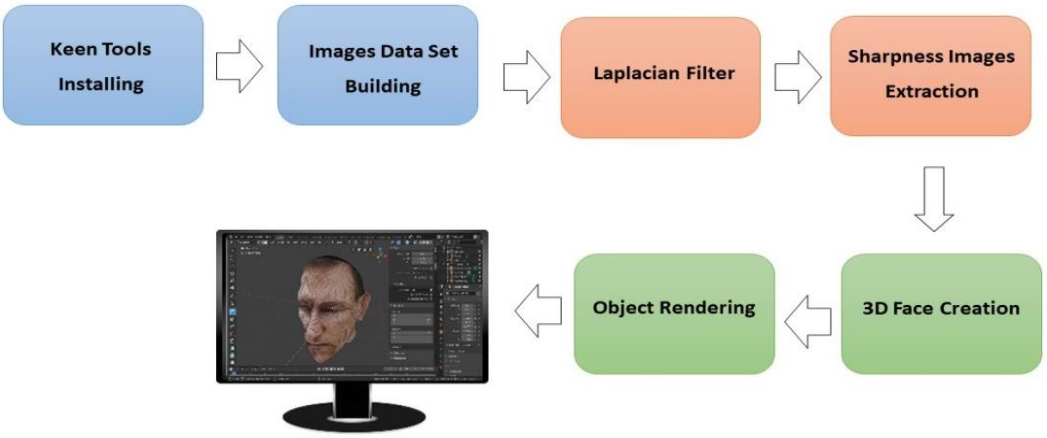

Figure 2. General stages pipeline for 3d face creation

2.1. Keen tools installing

To construct a 3D character that necessities time and exertion may take numerous years. This in itself is a futility that can be put resources into creating different activities in a similar field. Contingent upon the present general setting, the client is allowed to intelligently creator a virtual environment (VE) [11]. For this reason, the use of tools available to implement in Blinder software provides this exertion and furthermore gives another open door for inventiveness in the liveliness business and computer-generated reality (VR) applications. That there are circumstances that you don't have the opportunity to fabricate everything without any preparation from scratch. The keen Tools Face builder is a tool for 3D modeling faces and heads based on specific people or actors, developed add-on for Blender software. It is easy to utilize and gives quick and astounding outcomes. Which can be installed from the Keen Tools Face Builder website. Keen Tools give a 3D head model that can be utilized geometry transformation, sculpting, rendering, animation, as shown in Figure 3.

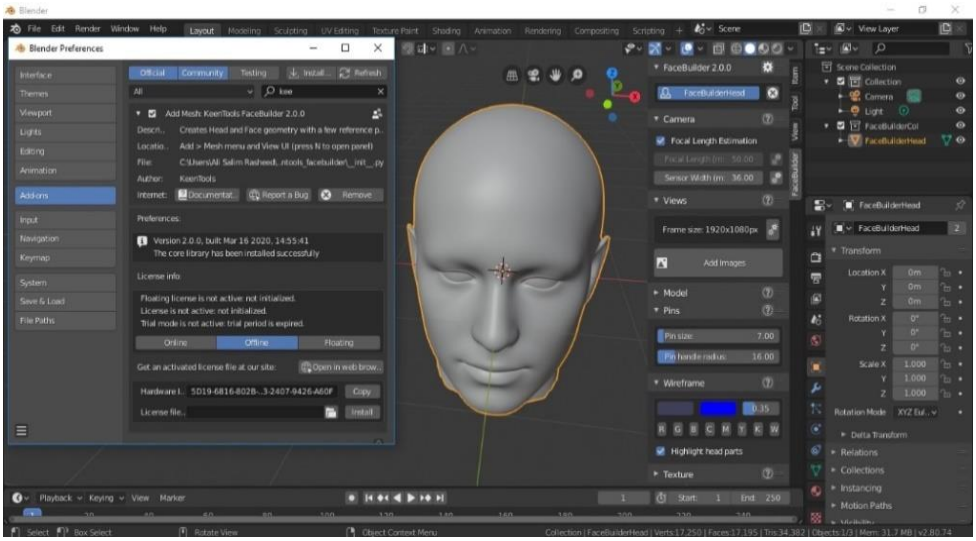

Figure 3. Keen tools face builder installation in blender

2.2. Images data set building

The process of building 3D faces of a human being using 2D images within the Blinder virtual environment (VE) requires several basic conditions that must be met in these images to achieve a high level of performance and an ideal result. That is conditions and fundamental variables from it ought to be accessible in Keen Tools Face builder. Which should be high-definition pictures, non- blurring, with high sharpness and with dimensions of frame size 1920 x 1080 pixels.

The world is witnessing daily a large increase in the number of digital images with the increase in digital cameras and the advancement of its technologies rapidly [12], with the widespread use of digital cameras, many users take poor quality digital photos due to a lack of experience. It is the fundamental thought in the obtaining of a blurry photo [13]. In addition to the process of capturing 2D images using these digital devices, varying from the highest blur and less clear to the least blur and the highest sharpness and clarity. Which relies on the person's association with cutting edge photography.

Accordingly, Image Data Set was built containing 2D images of the same accuracy, but they differ in clarity and blurring, according to the following:

a) Use the free Image data set, from Open Visual FX Blog which contains seven images of the Actor – Tim Delano at 1920 x 1080 pixel resolution, non- blurring and high sharpness.
b) By using the ACDSee Photo Studio Ultimate application, four different blur operations ((a) Radial blur [14], (b) Spread blur [15], (c) Smart blur [16], (d) Gaussian blur [17]), are applied to each of the original 7 images, which were explained before, and thus we have a new images data set consisting of 35 different images in blurry, sharpness and clarity photos, as shown in Figure 4.

Add a self-assertive number to a dataset of images. It will be promptly tried utilizing OpenCV applications with the Python language.

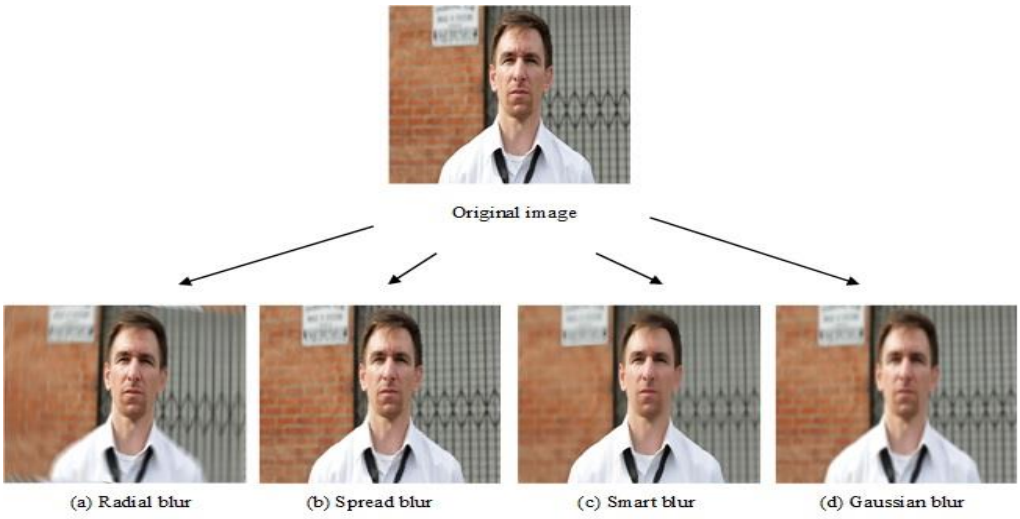

Figure 4. Images data set contains original images (sharpness) and different blurry images

3. LAPLACIAN FILTER, OPENCV WITH PYTHON

a) the 2D images that were created in the data set that was described and detailed in the past paragraph, are increasingly similar to a lot of irregular pictures that exist inside a camera of different sharpness and blurring. Captured by a user or amateur. It will be utilized to build the 3D face of animation inside the Blinder virtual environment (VE), which is used to develop the virtual reality (VR) applications. As per the accompanying: \\1) OpenCV is a library of programming capacities created by Intel and now supported by Willow Garage [18]. It consists of a series of C functions and a few C++ classes [19].
b) Laplacian operator is a sharpening filter. After applying a Laplacian filter to the image, a new image is obtained highlighting edges and other discontinuities [20]. It is the most widely used known second-order derivative filter [21] of isotropic nature. What's more, the field wherein this filter was applied for edge recognition by featuring edges and other discontinuities in particular image [22, 23]. What's more, for an element of picture f(x,y) of 2D, the computation of the second-order derivative is represented by:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

When this calculation is made, the edge will be detected where is a spatial change in intensity pixel values is strong rapidly changed.

In our approach, the Laplace filter used to construct a blurred image classifier to isolate blurred images from non-blurred images by using a suitable threshold value, in which the value under the threshold is selected it considered as blurred 2D image and up which it would be considered as a non-blurred image. This can be calculated using a convolution filter. Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernel that can approximate the second derivatives in the definition of the Laplacian. Two commonly used small kernels are shown in Figure 5.

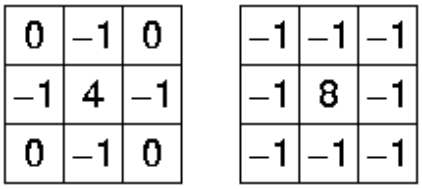

Figure 5. Two commonly used discrete approximations to the Laplacian filter. (Note, we have defined the Laplacian using a negative peak because this is more common; however, it is equally valid to use the opposite sign convention)

Using one of these kernels, the Laplacian can be calculated using standard convolution methods.

Because these kernels are approximating a second derivative measurement on the image, they are very sensitive to noise. To counter this, the image is often Gaussian smoothed before applying the Laplacian filter. This pre-processing step reduces the high frequency noise components prior to the differentiation step.

In fact, since the convolution operation is associative, we can convolve the Gaussian smoothing filter with the Laplacian filter first of all, and then convolve this hybrid filter with the image to achieve the required result. Doing things this way has two advantages:

Since both the Gaussian and the Laplacian kernels are usually much smaller than the image, this method usually requires far fewer arithmetic operations. The LoG (`Laplacian of Gaussian') kernel can be precalculated in advance so only one convolution needs to be performed at run-time on the image.

The 2-D LoG function centered on zero and with Gaussian standard deviation σ has the form:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$

4. SHARPNESS IMAGES EXTRACTION

Image pattern recognition depended on Feature extraction that is not an easy way [24]. After applying the Laplacian filter algorithm using the OpenCV with Python languages. Blurry images are separated from sharpening, high-resolution (1920 x 1080 pixel) images to be ready to use in the process of building the 3D face of a character within the Blinder virtual environment (VE) using Keen Tools Face builder.

5. 3D FACE CREATION

Virtual reality (VR) is a computer-based interactive stimulation technology that attracts a specific virtual environment (VE) used for design and simulation purposes [25]. Add seven 2D images with different expressions and perspectives to Keen tools Face builder within Blender software for 3D human face build steps implementation, as shown in Figure 6.

These seven photos will be geometrically prepared and fit on the 3D human face model. Each of these images that were previously added to Blinder is sequentially selected. Each time choosing a particular 2D image, wireframe panel will show up in blue. Using the pins provided by these tools, the wireframe panel will be adjusted and fixed to the boundaries and features of the different faces. Where these pins play a major role in the process of determining the facial features of the 3d face of character, as shown in Figure 7.

Figure 6. Add 2D images to keen tools face builder within blender software

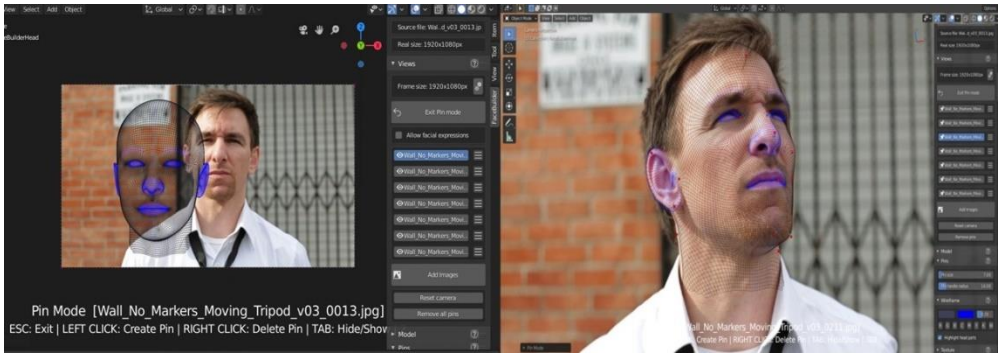

Figure 7. Wireframe panel to adjust on the 2D face

6. OBJECT RENDERING

Making renderable 3D models from actual objects in the real world is a problem and a big challenge [26]. A 3D object rendering can be enhanced which is performed using a pipeline concept. These engineering adjustments of the wireframe panel on the faces in the seven different images are the expressions and angles of a particular character using pins, which is one of the main features available in this tool, which requires trying more than once to increase skills and mastery. In the Render stage, the result is translated to the 3D face. The model of the 3D face caused by the Render is a group of pixels affected by the engineering calculations in the computer graphic as well as the degrees of illumination and shading [27]. This model can be used to produce videos, animation, and virtual reality (VR) applications, as shown in Figure 8.

Figure 8. 3d face model after render stage

7. CONCLUSION

Building a 3D model of faces within the Blinder virtual environment (VE) using Keen Tools Face builder requires several 2D images of a specific character. These pictures should be non- blurring with high sharpness and accuracy. This is a major problem, that 3D face model designers may encounter using these tools, especially that the human eye cannot achieve this distinction with high accuracy because the photographer may be amateur or unprofessional. After that built data set for 2D images of different blurring a (four types of blurry) and sharpness, and give the data set a random index number. Supposed it was a group of pictures taken from an individual utilizing a computerized camera. In this paper proposed using the Laplacian Filter algorithm with OpenCV applications using the Python programming language, to get rid of 2D images with blurring and the survival of images with high sharpness, which achieved high and real results in isolating and neglecting blurry images and the use of high sharpness images in building models for 3D faces. Performing engineering operations on faces in the selected 2D images using a wireframe panel with pins, a 3D face model was produced.

REFERENCES

[1] M. Raees and S. Ullah, "DOP: Discover Objects and Paths, a model for automated navigation and selection in virtual environments," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 27, no. 4, pp. 2784-2797, 2019.

[2] H. Izwan Rahmat, S. Ahmad, and M. Ismail, "Collaborative virtual reality application for interior design," *Indones. J. Electr. Eng. Comput. Sci. (IJEECS)*, vol. 16, no. 1, p. 500, 2019, doi: 10.11591/ijeecs.v16.i1.pp500-507.

[3] F. Alecu, "Blender Institute–the Institute for Open 3D Projects," *Open Source Sci. J.*, vol. 2, no. 1, pp. 36-45, 2010.

[4] K. Tabassum, "Using wireless and mobile technologies to enhance teaching and learning strategies," *Indones. J. Electr. Eng. Comput. Sci (IJEECS).*, vol. 17, no. 3, pp. 1555-1561, 2020, doi: 10.11591/ijeecs.v17.i3.pp1555-1561.

[5] S. Nasir, M. N. Zahid, T. A. Khan, K. Kadir, and S. Khan, "Augmented reality an economical solution for engineers and designers," *Indones. J. Electr. Eng. Comput. Sci. (IJEECS)*, vol. 17, no. 2, pp. 833-844, 2019, doi: 10.11591/ijeecs.v17.i2.pp834-844.

[6] Ö. BAŞER and E. I. Konukseven, "7-DOF Haptic device and interface design," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 21, no. 2, pp. 493-499, 2013.

[7] R. Gupta, R. Nawani, and V. P. Talreja, "Virtual Reality Content Creation using Unity 3D and Blender," *Int. J. Comput. Appl.*, vol. 156, no. 3, pp. 8-12, 2016.

[8] E. ERTUĞRUL, Z. Baytar, Ç. ÇATAL, and Ö. C. A. N. MURATLI, "Performance tuning for machine learning-based software development effort prediction models," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 27, no. 2, pp. 1308-1324, 2019.

[9] V. K. Balasubramanian and K. Manavalan, "Knowledge-based genetic algorithm approach to quantization table generation for the JPEG baseline algorithm," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 24, no. 3, pp. 1615-1635, 2016.

[10] H. Gümüşkaya and B. Örencik, "A parallel pipelined computer architecture for digital signal processing," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 6, no. 2, pp. 107-130, 1999.

[11] D.-M. Popovici, C.-M. Bogdan, M. Polceanu, and R. Querrec, "Applying of an ontology based modeling approach to cultural heritage systems," *Adv. Electr. Comput. Eng.*, vol. 11, no. 3, pp. 105-110, 2011.

[12] R. Bansal, G. Raj, and T. Choudhury, "Blur image detection using Laplacian operator and Open-CV," in *2016 International Conference System Modeling & Advancement in Research Trends (SMART)*, pp. 63-67, 2016.

[13] H. Tong, M. Li, H. Zhang, and C. Zhang, "Blur detection for digital images using wavelet transform," in *2004 IEEE international conference on multimedia and expo (ICME)*(IEEE Cat. No. 04TH8763), vol. 1, pp. 17-20, 2004.

[14] C. B. Webster and S. J. Reeves, "Radial deblurring with ffts," in *2007 IEEE International Conference on Image Processing*, vol. 1, pp. I-101, 2007.

[15] D. A. Fish, J. Grochmalicki, and E. R. Pike, "Scanning singular-value-decomposition method for restoration of images with space-variant blur," *JOSA A*, vol. 13, no. 3, pp. 464-469, 1996.

[16] J. Zheng, K. P. Valavanis, and J. M. Gauch, "Noise removal from color images," *J. Intell. Robot. Syst.*, vol. 7, no. 3, pp. 257-285, 1993.

[17] J. Flusser, S. Farokhi, C. Höschl, T. Suk, B. Zitová, and M. Pedone, "Recognition of images degraded by Gaussian blur," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 790-806, 2015.

[18] A. K. Soe and X. Zhang, "A simple PSF parameters estimation method for the de-blurring of linear motion blurred images using wiener filter in OpenCV," in *2012 International Conference on Systems and Informatics (ICSAI2012)*, pp. 1855-1860, 2012.

[19] Y. M. Wang, Y. Li, and J. B. Zheng, "A camera calibration technique based on OpenCV," in *The 3rd International Conference on Information Sciences and Interaction Sciences*, pp. 403-406, 2010.

[20] P. Sharma, M. Diwakar, and S. Choudhary, "Application of edge detection for brain tumor detection," *Int. J. Comput. Appl.*, vol. 58, no. 16, 2012.

[21] R. Bausys, G. Kazakeviciute-Januskeviciene, F. Cavallaro, and A. Usovaite, "Algorithm Selection for Edge Detection in Satellite Images by Neutrosophic WASPAS Method," *Sustainability*, vol. 12, no. 2, p. 548, 2020.

[22] S. A. Coleman, B. W. Scotney, and S. Suganthan, "Edge detecting for range data using laplacian operators," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2814-2824, 2010.

[23] G. T. Shrivakshan and C. Chandrasekar, "A comparison of various edge detection techniques used in image processing," *Int. J. Comput. Sci.*, vol. 9, no. 5, p. 269, 2012.

[24] E. L. Hall, R. P. Kruger, S. J. Dwyer, D. L. Hall, R. W. Mclaren, and G. S. Lodwick, "A survey of preprocessing and feature extraction techniques for radiographic images," *IEEE Trans. Comput.*, vol. 100, no. 9, pp. 1032-1044, 1971.
[25] S. Shahrbanian and M. J. Simmonds, "Effects of different virtual reality environments on experimental pain rating in post-stroke individuals with and without pain in comparison to pain free healthy individuals," *Annu. Rev. Cybertherapy Telemed.*, pp. 653-659, 2008.
[26] R. Furukawa, H. Kawasaki, K. Ikeuchi, and M. Sakauchi, "Appearance Based Object Modeling using Texture Database: Acquisition Compression and Rendering.," in *Rendering Techniques*, pp. 257-266, 2002.
[27] M. Latifi-Navid, M. Bilen, E. I. Konukseven, M. DOĞAN, and A. Altun, "Fast and accurate semiautomatic haptic segmentation of brain tumor in 3D MRI images," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 24, no. 3, pp. 1397-1411, 2016.