



STUDENT SESSION

APPLICATION OF VIRTUAL REALITY WITH PRODUCTION ROBOTICS

Nikola Jović*,
[0009-0004-3856-8838]

Miodrag Živković,
[0000-0002-4351-068X]

Nebojša Bačanin Džakula,
[0000-0002-2062-924X]

Aleksandar Petrović,
[0000-0003-3324-3909]

Luka Jovanović
[0000-0001-9402-7391]

Singidunum University,
Belgrade, Serbia

Abstract:

Today's television and production companies increasingly use modern technology to create and broadcast their content more efficiently and economically. This includes the use of VR (virtual reality) and AR (augmented reality), which have become one of the inevitable elements in production. In addition to enabling savings in equipping studios, these technologies also provide the possibility of maximum use of space, which enables an additional form of saving funds, especially in situations where more studios are needed. In television houses, Virtual Reality and Augmented Reality are often combined with production robotics, which is increasingly present, both in television and in production houses. Robots have become an indispensable part of production, providing a faster, more accurate, and economical solution for various tasks, from setting up the scenography to controlling the cameras themselves. This combination of modern technology and robotics allows television and production companies to improve their productivity and creativity, providing viewers with interesting and modern projects. In addition, this combination of modern technologies enables the creation of high-quality content with reduced operating costs. This paper describes the camera lens calibration process, with an explanation of the individual parts of the process to help you better understand the calibration process itself and avoid potential calibration problems.

Keywords:

Production robotics, VR/AR, FreeD protocol, Unreal Engine.

INTRODUCTION

Prior to commencing, it is important to familiarize yourself with a few concepts from manufacturing robotics: pan - moving the camera horizontally (left-right), tilt - moving the camera vertically (up-down), and zoom - enlargement.

Robot and virtual reality communication take place using the FreeD [1] protocol developed by BBC R&D. Today, it is supported by many other sound names in the field of production robotics. The main role of the FreeD protocol is to track robot positions (pan, tilt, zoom) and send them to the virtual studio, in this case, Unreal Engine 5 will be used for this. Unreal Engine is a popular tool for creating virtual reality, augmented reality, and video games [2]. It is written in C++ programming language [3].

Correspondence:

Nikola Jović

e-mail:

nikola.jovic.23@singimail.rs



It appeared for the first time in 1998 [4]. It is known for its high-quality graphics and scalability, as well as cross-platform support. Also, a potential problem that can arise when creating a virtual studio will be shown. Such systems are becoming more common every day in large and small television and production houses due to the large savings that can be achieved. In the past, for one show, it was necessary to provide a suitable space and several cameramen, while with today's technology, it is possible to record several shows in one room with only one cameraman.

The integration of robots into virtual reality becomes a real challenge due to the need for precise coordination between physical robots and visual representation. If there is a lack of synchronization between the robot and virtual reality, there will be a problem with the output image, so there may be deviations in the position of some virtual objects. One of the leading problems [5] in the combination of robotics with virtual reality is the so-called lens breathing. That problem arises because of the physical characteristics of the society itself on the lenses. That problem can be solved by integrating advanced sensors for determining the distortion of society as well as its actual distance about means for calibrating society. In addition to the breathing lens in virtual reality, one of the main problems is the positioning of the physical object in the virtual one. If there is a problem with precisely tracking the position of the physical robot, objects in the virtual world may be displaced or distorted concerning real space. Furthermore, the delay in updating data between the real and virtual worlds can cause not synchronized and delayed responses, which further leads to a loss of user experience and not-so-applicable technology. In further work, the techniques used to solve these problems will be presented.

As already explained, lens breathing is one of the leading problems in the combination of production robotics and virtual reality, some of the methods for solving this problem are: the use of external lens calibration tools, the use of Unreal Engine filters to reduce the effect, use of lens calibration algorithms

Unreal Engine allows users to calibrate the camera using simple tools and a plug-in in the editor itself. This camera calibration process generates the data needed to precisely align the virtual cameras with the physical camera's position in space and model the distortion of the physical camera's lens. This designed approach makes it easier for users to just use it without the need for professional staff.

2. CAMERA CALIBRATION STEPS

The calibration of each new camera is very important for the whole process of connecting the real and virtual world. Every time the camera is changed, it is necessary to re-calibrate due to the very physical properties of the camera, which will be described below. Individual properties of the camera will be shown, as well as their theoretical basis for a better understanding of the camera's characteristics, which are necessary to understand for successful camera calibration.

Zoom, focus, and iris calibration plug-in lens element file to convert zoom, focus, and iris values to physical units used in Unreal Engine. Unopened input values can include absolute or normalized encoder positions that Unreal Engine receives from the outside world via LiveLink [4], so it is necessary to convert them to the appropriate units. Calibration data can depend and vary depending on focus and zoom, so it is necessary to calibrate using multiple calibration points for the best possible quality. In addition to lens calibration, it is also necessary to determine lens distortion. Distortion of society can be determined based on the Brown-Conrady [6] model, which uses five distortion parameters ($K1$, $K2$, $K3$, $P1$, $P2$). Parameters $K1, K2, K3$ represent the radial distortion coefficient, while $P1$ and $P2$ represent the tangential distortion parameter.

$$\begin{aligned} xu &= x d + (x d - x c) (K1r^2 + K2r^4 + \dots) + (P1 [r^2 + 2\{x d - x c\}^2] + 2P2 [x d - x c] [y d - y c]) (1 + P3r^2 + P4r^4 \dots) \\ yu &= y d + (y d - y c) (K1r^2 + K2r^4 + \dots) + (2P1 [x d - x c] [y d - y c] + P2 [r^2 + 2\{y d - y c\}^2]) (1 + P3r^2 + P4r^4 \dots) \end{aligned}$$

Equation 1. Brown-Conrady model.

Equation 1 shows Brown-Conrady model, the terms in the equations: yd and xd refer to the coordinates for the distorted points, yu and xu represent the point of the undistorted image, (xc, yc) is the distortion center, Kn represents the n^{th} radial distortion coefficients, pn is the coefficient for the n^{th} tangential distortion, r is the square root of $\{x_d - x_c\}^2 + \{y_d - y_c\}^2$ [7].

The focal length combined with the center of the image defines how the 3D points in the coordinate system are mapped to the 2D image. They also need to be calibrated for the exact camera model. The focal distance is represented as a 2D vector (Fy / Fx) [8]. To get a general view, the zoom distance is stored as a normal value that is divided by the height and width of the image or by resolution in pixels.



Figure 1 is represents an example of where the points needed to calculate the focal length are located on the camera. Equation 2 represents the formula for calculating the focal length.

$$f = \frac{1}{\frac{1}{o} + \frac{1}{i}}$$

Equation 2. Focal length calculation.

The nodal point of the lens is the point where the light rays converge. In Unreal Engine, it is very important to place the virtual camera on the nodal point to ensure that the virtual objects are aligned. This part of the calibration should find the point between the real and the virtual camera. The nodal point represents the place where light enters the lens without any deviations. This point is of great importance as it helps the virtual and real image to be displayed in proper perspective and alignment, an example of the nodal point is shown in Figure 2.

3. CALIBRATION PROCESS

It is also necessary to prepare a virtual environment that requires adding a connection with the camera and preparing the environment and lighting in which the calibration process itself will take place. Subsequently, it is necessary to enter the parameters of the specific lens in Unreal Engine for the calibration to be successful. The parameters of the camera mean the parameters listed above, such as focal length, lens distortion, etc. When all the parameters are entered and when the virtual and real scenes are ready, the calibration process begins.

The first step is to determine the checkerboard [9], Figure 3, distance and enter it in Unreal Engine, after that it is necessary to take pictures of the checkerboard in as many positions as possible so that the calibration is as accurate as possible, as shown in Figure 4.

Unreal Engine will use the OpenCV [10] library to determine the calibration and as a result, will return the calibration parameters that need to be saved, as shown In Figure 5.

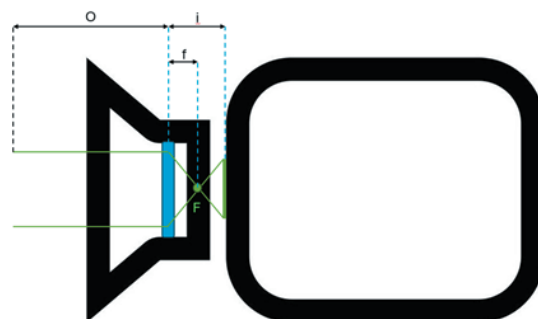


Figure 1. Focal length example.

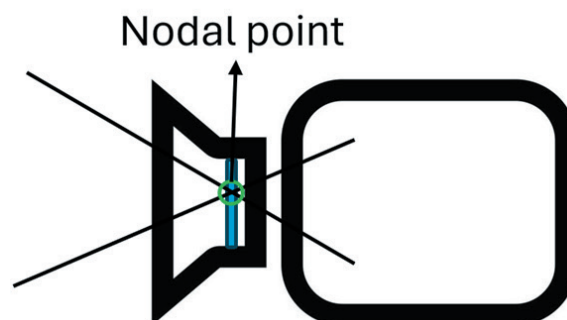


Figure 2. Nodal point example.



Figure 3. Checkerboard for lens calibration.



Figure 4. Screenshot from Unreal Engine, sample image for calibration image.

Evaluated Camera Settings		Distortion Parameters	
Focus Distance	1,500.000 cm	K1	-0.129
Aperture	22.0 F-Stop	K2	0.127
Focal Length	25.727 mm	K3	-0.027
Horizontal FOV	69.508 deg	P1	-0.009
		P2	0.006

Figure 5. Output from OpenCV library for lens calibration.

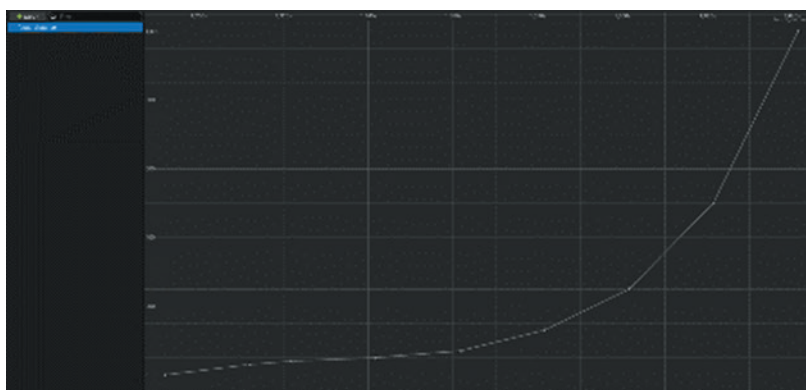


Figure 6. Focus values after calibration.

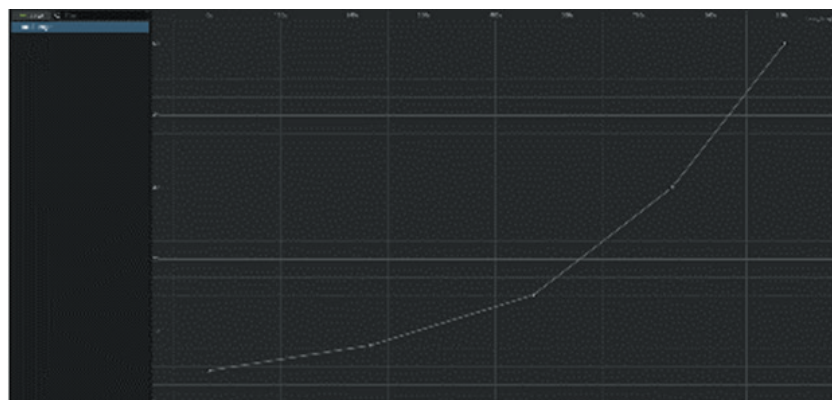


Figure 7. Zoom values after calibration.

Figure 6 presents a diagram of how lens values for focus (focus distance) look. Figure 7 presents a diagram of how lens values for zoom (focal length) look.

4. CONCLUSION

This work represents one of the examples where today all the representation is the use of modern technology, which includes a combination of hardware (robotic) systems as well as modern software solutions. Unreal Engine, even if it was primarily created for the growth of the gaming industry, has also greatly expanded its application in broadcast agencies. The possibilities of VR/AR solutions are advancing more and more every day, both because of the potential for improving the world, and because of the cost reduction of the agencies that own it.

The paper presents one of the leading problems in the broadcast industry as well as one of the potential solutions. By improving the Unreal Engine itself in the future, the possibilities will increase and newer versions of the software will be developed, which will greatly facilitate the work of people who will work with those systems.

In addition to the applications mentioned above, virtual reality has many applications [11] in other industries as well [12]. With the constant progress and improvement of people and software solutions, virtual reality will gradually spread to other areas of everyday life.

5. REFERENCES

- [1] "GoDoc - freeD," Go Documentation, [Online]. Available: <https://pkg.go.dev/github.com/stvmyr/freeD#section-readme>. [Accessed: 15 April 2024].
- [2] J. H. Kong, K. D. Kim, and R. Y. C. Kim, "Best Practices on Improving the Virtual Reality (VR) Content Development Process with EPIC's Unreal Engine," *International Journal of Advanced Culture Technology*, vol. 9, no. 4, pp. 417-423, 2021.
- [3] T. Hilfert and M. König, "Low-cost virtual reality environment for engineering and construction," *Visualization in Engineering*, vol. 4, pp. 1-18, 2016.
- [4] Unreal Engine. (n.d.). Unreal Engine. [Online]. Available: <https://www.unrealengine.com/en-US?lang=en-US>. [Accessed: 18 April 2024].
- [5] K. Tanaka, "Zoom lens without focus-breathing phenomena," *Zoom Lenses III*, SPIE, 2001, pp. 63-67.
- [6] J. Kelcey and A. Lucieer, "Sensor correction and radiometric calibration of a 6-band multispectral imaging sensor for UAV remote sensing," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 39, pp. 393-398, 2012.
- [7] M. Blázquez, I. Colomina, A. C. F. Gauss, and P. M. de la Tecnología, "On the role of self-calibration functions in integrated sensor orientation," *Proceedings of the International Calibration and Orientation Workshop (EuroCOW)*, Castelldefels, Spain, 2010, pp. 1-7.
- [8] "Camera Lens Calibration Overview," Unreal Engine Documentation, [Online]. Available: <https://docs.unrealengine.com/4.27/en-US/Working-WithMedia/IntegratingMedia/CameraCalibration/Overview/>. [Accessed: 18 April 2024].



- [9] O. Araar, S. Bouhired, S. Moussiou, and A. Laggoune, "Towards Low-Cost Indoor Localisation Using a Multi-camera System," *Computational Science – ICCS 2019, 19th International Conference*, Faro, Portugal, June 12–14, 2019, Proceedings, Part V, pp. 137-148, 2019.
- [10] "Calculate Lens Parameters," Unreal Engine Documentation, [Online]. Available: https://docs.unrealengine.com/4.26/en-US/BlueprintAPI/LensDistortion_1/OpenCV/Calibration/CalculateLensParameters/. [Accessed: 18 April 2024].
- [11] L. Muñoz-Saavedra, L. Miró-Amarante, and M. Domínguez-Morales, "Augmented and Virtual Reality Evolution and Future Tendency," *Applied Sciences*, vol. 10, no. 1, p. 322, 2020. [Online]. Available: <https://doi.org/10.3390/app10010322>. [Accessed: 18 April 2024].
- [12] S. Choi, K. Jung, and S. D. Noh, "Virtual reality applications in manufacturing industries: Past research, present findings, and future directions," *Concurrent Engineering*, vol. 23, no. 1, pp. 40-63, 2015.