A Novel Motion Control Workflow for Capturing and Reproducing Realistic Lens Flares

Vincent Maurer
vincentmaurerlt@gmail.com

Technical Directing,
Filmakademie Baden-Württemberg, Animationsinstitut

Lens flares are a common optical artifact in photography and filmmaking, caused by reflections and scattering within a camera system. While they can degrade image quality, lens flares are sometimes harnessed artistically to enhance visual appeal. In contemporary film and animation productions, like "Dune" or "The Lego Movie"[4], the characteristics of vintage lenses serve as creative tools. Existing techniques predominantly employ moving 2D sprites, computationally intensive ray tracing or hybrid techniques[2]. However, these methods often can't capture the full complexity of real-world lens flares. The final year 3D animated film project, "The Deep Above"[1], necessitated the recreation of a specific anamorphic lens flare. Traditional compositing plugins failed to achieve the director's vision. Consequently, a novel workflow was proposed for capturing and reproducing these effects.

Several factors, like lens design, incoming light angle, aperture, focus, size and colour of the light source, contribute to the appearance of lens flares. To limit the amount of variables to capture, the research focused on three factors: angle of incoming light, lens model and aperture.

The flares are captured by moving a small point of light across the field of view of a camera, creating a regular grid of point flares. A focused white LED light source with an adjustable aperture was placed in a blacked-out room. The light was pointed directly at the camera, which was mounted to an industrial motion control robot. [2] A script for the MIMIC for Maya Plugin [3], a tool to create animations for motion control robots, was developed. It rotates the camera around its nodal point in a line-by-line serpentine pattern. The dataset has a density of 128 rows and 72 columns normalized to the field-of-view of the lens, combined with 1.75x overscan to record light sources that are not visible in the frustum, but still produce flares. Each lens was captured at four different apertures, from wide open (usually about f1.4) to four stops closed, while adjusting ISO and exposure time to compensate for the loss of light. For further processing, the footage was denoised, the exposure values were visually matched and the shots exported as EXR sequences in ACEScg colour space. The dataset includes a variety of lenses - vintage and current, spherical, anamorphic, and zoom - to capture diverse flare scenarios.

Initially, a compact Nuke gizmo selected appropriate lens flare images and employed a dissolve operation to blend the four closest images. This resulted in high-quality still imagery but lacked performance and exhibited blending artifacts(Figure 1.1). Inspired by previous work [3], the lens flare images were stored in a sprite sheet. Screen space UVs were used to crop out the four adjacent tiles around the selected position. The images were then blended by distance.

Based on this method, a quick prototype for the Unreal Engine (Figure 1.3 right) was developed. It showed more complex flare characteristics than the default Unreal implementation (Figure 1.3 left), which is implemented as a post-process filter, but only creates one flare, limiting flexibility. To create animatable point light flares with fewer artifacts, multiple interpolation techniques for generating smoother in-between frames were evaluated. Using Nuke's Kronos node, two samples of each line in the grid were interpolated along the x-axis, and the outputs were then used to interpolate the y-axis. This led to sharper images and smoother interpolation. The challenges of this optical flow based technique are morphing artifacts due to the overlapping moving transparent layers in a lens flare.

In addition to exploring purely image-based techniques, an implementation of the machine learning based RIFE[1] network in Nuke was tested. This network, built for retiming, estimates optical flow and interpolates images similar to the Kronos node. It results in sharper images but shows similar morphing artifacts. Using retiming tools to interpolate in a 2D grid of images has limitations due to their one-dimensional nature. To address this, Nuke's machine learning framework, Copycat, was trained to interpolate between four images. Although the results showed less clarity than those produced by traditional retiming networks (Figure 1.6), this approach presents an intriguing avenue for further research.

[1]The Deep Above Making Of: https://www.therookies.co/entries/31298
[2]Light: Godox S30 + Godox SA06, Robot: Kuka KR-16)
[3]MIMIC Plugin for Maya: https://www.mimicformaya.com/

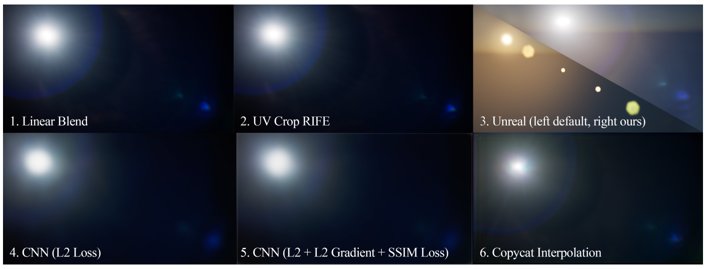

Figure 1: Comparison between different explored lens flare techniques

Instead of relying solely on interpolation, custom models for inferring the images were also explored. Their input consists of a lens flare image and the corresponding three-dimensional vector: x and y positions in normalized device coordinates (NDC) and an aperture value normalized from f1 to f22 down to the range 0-1. Training a conditional generative adversarial network (cGAN) on the dataset showed multiple issues. These include mode collapse, instability, and non-convergence.

Subsequently, training reverted to an inverse classification convolutional neural network (CNN) because it was well-suited for the task and straightforward to implement. The architecture is composed of 6 upsampling convolution layers with batch normalization and LeakyReLU. This is followed by a fully connected layer to the final resolution of 512x256. The resulting images lost high frequency detail compared to the ground truth and appeared cloudy (Figure 1.4). This might be connected to an insufficient L2 loss function, but could also be a result of misalignment in the dataset due to the capture method. To enhance detail preservation, L2 loss, L2 gradient loss, and structural similarity index (SSIM) were combined with equal influence. This exhibited sharper edges, but still failed to resolve some small details (Figure 1.5). After training, the tested networks were exported as TorchScript and loaded in as a gizmo in Nuke's Cattery Format [4] for further comparison and inference.

The innovative capture method and its resulting lens flare tools have demonstrated successful applications in various VFX and CG productions at Filmakademie. Notably, the anamorphic flares and glows have piqued the interest of our compositors due to their ease of use and instant realistic appearance, requiring minimal manual adjustments. Looking ahead, the development of an affordable, compact motion control setup to capture flare data is envisioned, along with the exploration of more advanced machine learning techniques. A portion of the dataset will also be published for further research [5]. Ultimately, a deeper understanding of lens and camera characteristics, coupled with more accurate data, will undoubtedly enhance the image quality of visual effects and animated films.

The author would like to thank the staff at Doublecheese Film for supporting us with their Motion Control System, Benjamin Völker (Zeiss) and Johanna Barbier (The Foundry), as well as the Filmakademie team: Niklas Wolff, Andreas Blind, Max Pollmann, Liliane Maurer, Fynn Aurich, Alexander Kreische, Jonas Trottnow, Simon Spielmann and Volker Helzle.

[1] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation, 2022. URL `https://arxiv.org/abs/2011.06294`.

[2] M. Hullin, E. Eisemann, H.-P. Seidel, and S. Lee. Physically-based real-time lens flare rendering. *ACM Trans. Graph.*, 30(4), jul 2011. doi: 10.1145/2010324.1965003.

[3] S. Jo, Y. Jeong, and S. Lee. Real-time nonlinear lens-flare rendering method based on look-up table. *Journal of KIISE*, 44:253–260, 03 2017. doi: 10.5626/JOK.2017.44.3.253.

[4] E. Pekkarinen and M. Balzer. Physically based lens flare rendering in "the lego movie 2". In *Proceedings of the 2019 Digital Production Symposium*. ACM, 2019. doi: 10.1145/3329715.3338881.

[4]Nuke Cattery: https://community.foundry.com/cattery
[5]Lens Flare Dataset: https://vincent-maurer.github.io/lens-flare-capture/