Photorealistic Rendering and Modification of Motion Blur for the Simulation of Video Data for Structural Dynamics

ANDRE GREEN, MOISES FELIPE SILVA, ALLISON DAVIS, ALEXANDER KARPOWICZ, ANDREW SOMBORGER, ALESSANDRO CATTANEO and DAVID MASCAREÑAS

Affiliation: Los Alamos National Laboratory, LANL TA-03 Bldg. 1400 Casa Grande Drive, Los Alamos N.M.

ABSTRACT

In recent years imager-based approaches to structural dynamics measurements have gained increasing interest. Imager-based approaches to measurement have a number of attractive properties including being able to monitor large areas at a relatively high spatial resolution with a relatively small number of imagers. The stand-off monitoring capabilities of imagers are also highly attractive for many structural dynamics and structural health monitoring applications, particularly those in dangerous, inaccessible, extreme, and high temperature environments. For their advantages though, a number of challenges must be addressed when using imager-based techniques for measuring the deformation and motion of structures. There are issues of lighting conditions that can potentially vary during the measurement period, due to movement of the sun or lights in a facility being turned on and off. In addition, video measurement of dynamic structures also requires making a number of choices associated with the measurement setup including location of the imagers, focal lengths of lenses, number-of-pixels, lens characteristics, framerates, shutter speeds, ISO, aperture settings, planes of focus, and depth-of-field. There can potentially be effects such as specular reflections or shadows which might complicate downstream data processing. The large number of parameters associated with imager measurements make them very flexible, but it also can mean that setting up imager measurements can take substantial time even under controlled laboratory conditions.

The complication associated with imager measurements suggests that high-fidelity, photorealistic rendering tools that are capable of capturing the interactions between structural dynamics, light transport, and the measurement process at the imaging plane are needed. To date, substantial work has been done by the computer graphics community to develop photorealistic rendering tools which are becoming increasingly accessible. However, in structural dynamics we often care about sub-pixel motion and it is not understood whether current techniques for modeling effects such as motion blur have sufficient fidelity for structural dynamics. Furthermore, photorealistic video renders can take a large amount of time to complete, and given current techniques do not allow for simple changes such as frame-rate/shutter speed in post processing without redoing the entire render. In this work we present an approach based on digital coded exposures for forming frames of dynamic scenes, that allows for the framerate and shutter speed to be changed in post-processing. The proposed approach is inspired by the physical operation of conventional cameras. In addition, the digital coded exposure proposed approach suggests theoretical alternatives to data capture that could potentially be used to control motion blur properties on a per-pixel basis.

INTRODUCTION

In recent years video-based approaches to structural dynamics and structural health monitoring have become of high interest on account of their remote nature, high spatial resolution and relatively low installation costs. However, to deploy these systems on large scale infrastructure it is desirable to develop tools to simulate the behavior of these systems in order to facilitate their design and to predict their performance before they are deployed. Rendering tools have been developed by the computer graphics community for decades [1], however it is not clear at this time, what level of fidelity these tools have for cases of interest to photogrammetry for structural dynamics such as sub-pixel motion and highly dynamic scenes.

Motion blur is often an afterthought when developing renders of objects in motion as well as computer vision processing systems. Conventional rendering assumes a dynamic 3D scene consists of a sequence of static 3D scenes. However, it was recently shown that visually plausible motion blur has enough degrees of freedom associated with it, that adversarial examples can be developed for deep learning systems that have motion-blur like properties [2]. This is a particularly important concern for tasks such as object detection, tracking and velocity estimation that often come up in video-based structural dynamics and structural health monitoring applications. The finding that motion-blur-like effects in images can be used to generate adversarial examples indicates that principled and physically accurate techniques for incorporating motion blur phenomena into the frame formation process should be developed and should be understood on a quantitative level.

RENDERING FOR LIGHT TRANSPORT SIMULATION OF VIBRATION

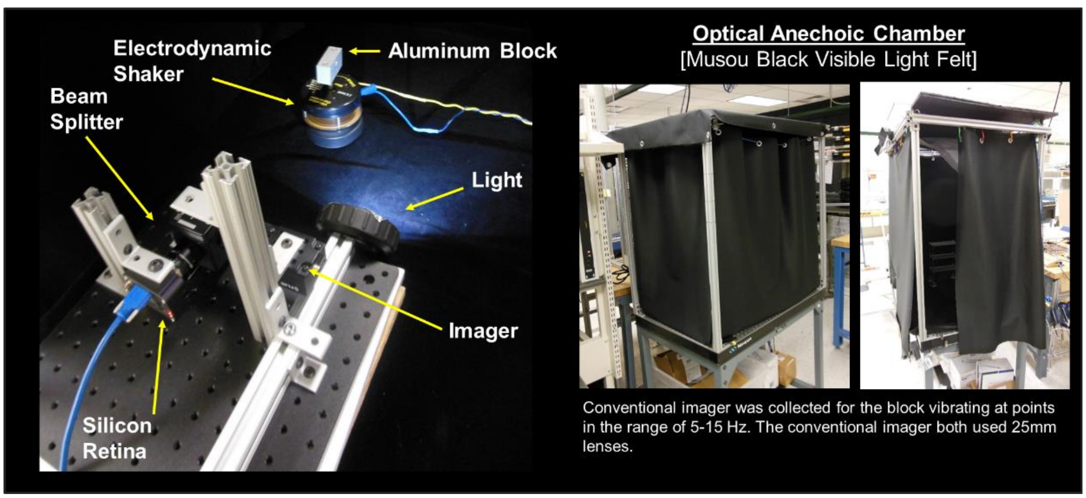

Figure 1 Details of the experimental setup (left) inside the optical-anechoic chamber shrouded in Musou black visible light felt (right).

To establish a baseline by which to measure the feasibility of digital simulation of moving structures at a fine scale, we created an optical-anechoic chamber shrouded in Musou black visible light felt (Figure 1). The felt served to isolate the optical-mechanical assembly from outside light and changes in lighting. Inside the chamber, a FLIR GS3 high-speed imager with a 25mm lens shared a beam-splitter with a DVS Silicon Retina to view an aluminum block. The targeted block was vibrated up and down via an electrodynamic shaker, and illuminated by battery-powered LED. An aluminum block was chosen because of its textural similarity to real physical structures of interest.We used three different rendering engines to generate videos to compare against the real data captured by the FLIR imager. The appearance of static frames is expected to be extremely similar, insofar as the same texture for the block was directly taken from a real frame from the FLIR imager and each renderer directly displayed the texture rather than being independently lit using the lighting supported by that renderer.

The primary facets of interest for each rendering engine are the sub-pixel rendering and motion-blur. In the real world, the face of the aluminum block is assumed to be a continuous signal which is rasterized, whereas in the simulations this same rasterized signal (as captured by the FLIR imager) is used to texture the face of the virtual block, which is then re-rasterized. Likewise, on the physical camera there is an associated exposure time over which light is collected and during which the block may move. Both of these factors are expected to influence the frequency content of the time-series associated with each pixel of the real and simulated videos.

In Figure 2, we compare the time-series data from a pixel of the real data (on left) with three different renderers: the Blender Cycles raytracing engine, the Blender Eevee real-time rendering engine [3], and a from-scratch implementation of Peter Shirley's C++ renderer [4].

The Blender Cycles engine is a physically-based path tracer, whereas Eevee is a rasterization-based rendering engine and is consequently capable of generating renders very quickly, but at the cost of reduced physical accuracy. Although the Blender Cycles engine supports motion blur to a limited extent, it appears to be a simple averaging of multiple renders. In contrast, Peter Shirley's ray-tracing adds an additional layer of realism in the form of random ray sampling through time allowing the user to render approximately continuous motion, contingent on the density of temporal sampling.

The block was vibrated at 11Hz in all cases, and the principle spike of each FFT approximately matches this value. The additional spikes, particularly the spike at 23Hz that appears more prominently in the simulations may be on account of the signal 'noise' for each time-series only being derived from angularly close parts of the block, rather than for example electronic noise which may affect the real imager. Although the Blender Cycles renderer and the C++ renderer use a similar number of samples when ray-tracing, the Blender render's smoother signal may be a result of importance sampling (as opposed to random sampling as was present in the implementation of Shirley's C++ renderer).

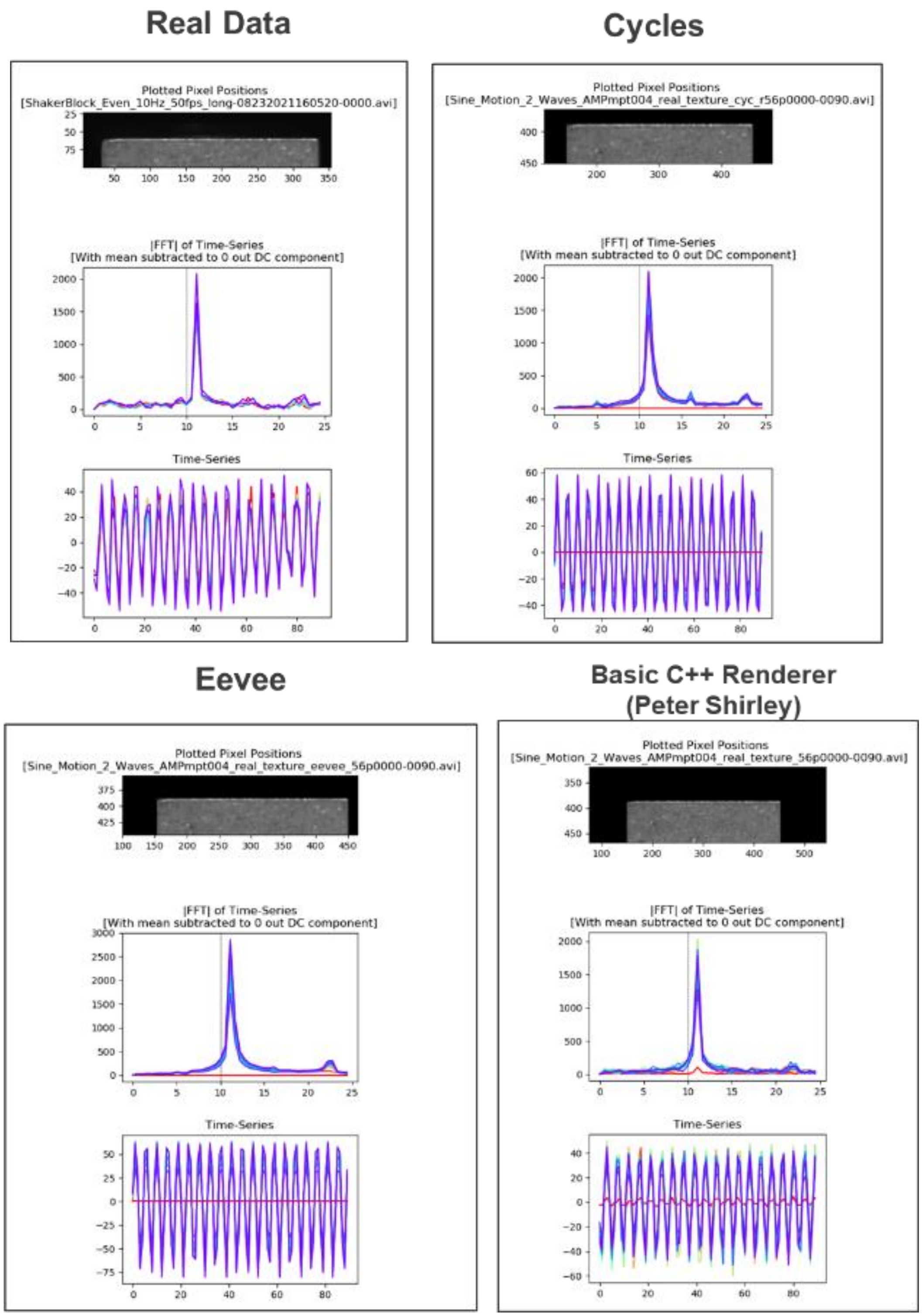

Figure 2 The blocks in all of the above simulations oscillated at 11Hz. The aluminum block object was made the same resolution and size across all renders (300 pixels, rendered as a plane with no lighting, purely referencing the texture). This size was chosen to match the size of the block as it appears in real-world data from the FLIR imager (on left). No post-processing scaling was applied. Different colors indication different pixels.

FRAME BASED DIGITAL CODED EXPOSURE

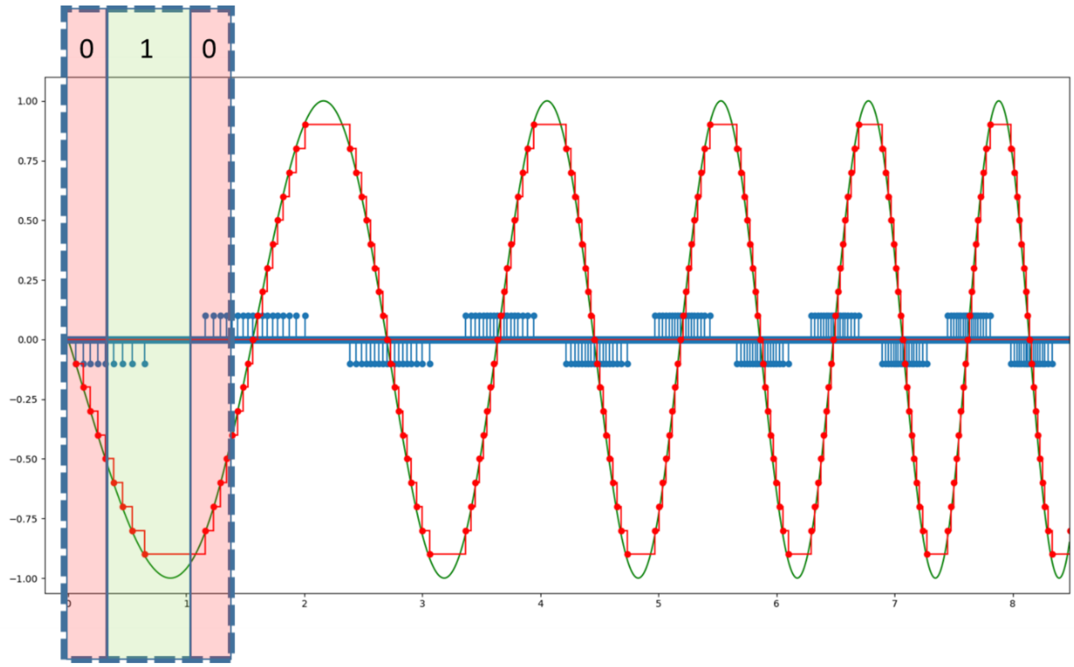

Figure 3 The region of the interpolated signal (red line) that occurs during the period over which the digital shutter is open (indicated by the green rectangle) is integrated to acquire the signal value of the frame-based coded exposure of the signal at the end-time of the green region. This image illustrates a boxcar exposure found in a conventional camera.

To account for motion blur we make use of the concept of the digital coded exposure as introduced in [5] for event-based imagers. In this work we extend the concept of the event-based digital coded exposure to conventional frames either generated in simulation by a render engine or captured using a conventional frame-based camera. Frame-based digital exposure is most simply accomplished by discarding frames corresponding to times when the digital shutter is closed, retaining those frames corresponding to times when the digital shutter is open, and summing the frames during the open periods as grouped by each with each open and shut cycle of the shutter. More generally, the base signal may be interpolated (Figure 3 shows a zero-hold interpolation as the red line) and then integrated during times when the digital shutter is open (shown in green), which allows the technique to be applied to imagers which capture pictures at irregular intervals; for example, event-driven imagers or conventional frame cameras with irregular frame rates assuming knowledge of when the images were captured is known.

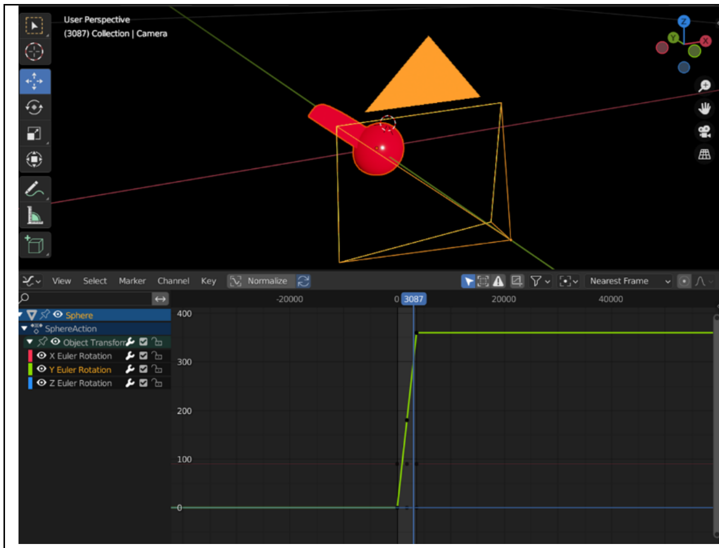

Figure 4 A video is generated with 3,600 frames at 25% of 640x480px (160 x 120px). As shown in the f-graph above, linear interpolation was used to rotate the propeller evenly.

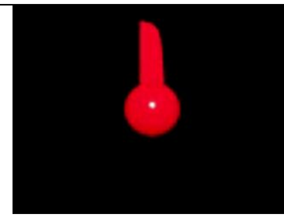

Figure 5 [160 x 120] A propeller rotates 360 degrees over 12 seconds. This clip is a naively down-sampled version of the video generated for analysis.

To demonstrate the utility of frame-based digital coded exposure, we generate a simulation of a small propeller rotating at a constant velocity in Blender. This case is of interest because the velocity of a rotating blade increases proportionally to the distance between the point on the blade and the center of hub about which the propeller rotates. Consequently, the amount of motion blur present in each pixel for a real video of a rotating propeller likewise varies with the distance of the pixel along the propeller blade to the center of the propeller hub.

For this simulation, the propeller hub was approximately 0.68” in diameter with a 1.54” blade. The physically-based-rendering (PBR) shader used for this propeller had a roughness of 0.1 to approximate the specular reflection of a plastic material. A virtual 25mm lens affixed to a camera positioned 0.125 meters in front of the propeller was used to view the scene, which was illuminated by a point light placed 3 meters in front and 1 meter above the propeller at a brightness of 1,000W.

BOXCAR EXPOSURE

The shutter function associated with conventional cameras can be approximated as a boxcar function where the exposure is uniform over the time period that the shutter is open. We now demonstrate how motion blur associated with the conventional boxcar exposure can be simulated.

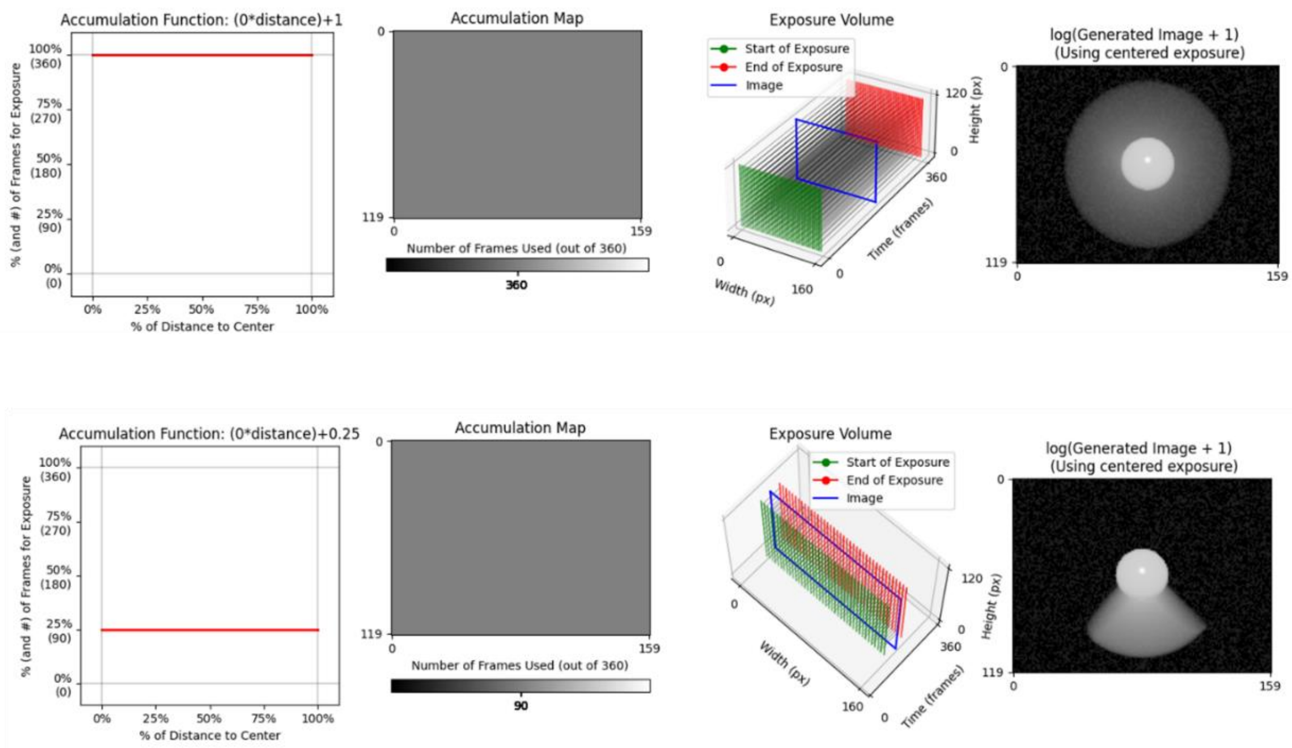

Figure 6 The exposure volume (third column) is the spatiotemporal volume from which an image is created (right-most column) over a period of time. Traditional images captured by cameras have a rectangular brick-shaped exposure volume, the length of which is the amount of time the sensor was exposed for. In the images above, the exposure time is shorter (1/4 a rotation) in the second row than in the first (1 rotation).

Figure 6 demonstrates the effect of varying the duration of the digital exposure. By setting the digital exposure to the duration of one full rotation of the propeller, the blade is completely smeared evenly into a circle. When the exposure is set to the duration corresponding to a quarter-rotation centered on the point in time where the propeller blade is pointing down, the frame-based coded exposure imager is a quarter-arc whose center is pointing down. This is directly analogous to varying the exposure duration of a real imager.

VELOCITY BASED EXPOSURE

Although uniform exposure through frame-based digital coded exposure is a useful tool for exploring the expected behavior of a traditional camera, the technique is not limited to using rectangular exposure volumes.

As mentioned previously, in the case of a rotating propeller the velocity varies across the blade; consequently, there is no uniform exposure duration for which the amount of motion blur present at each pixel of the image is the same. Therefore, we use a non-rectangular exposure volume to expose different parts of the frame for different durations.

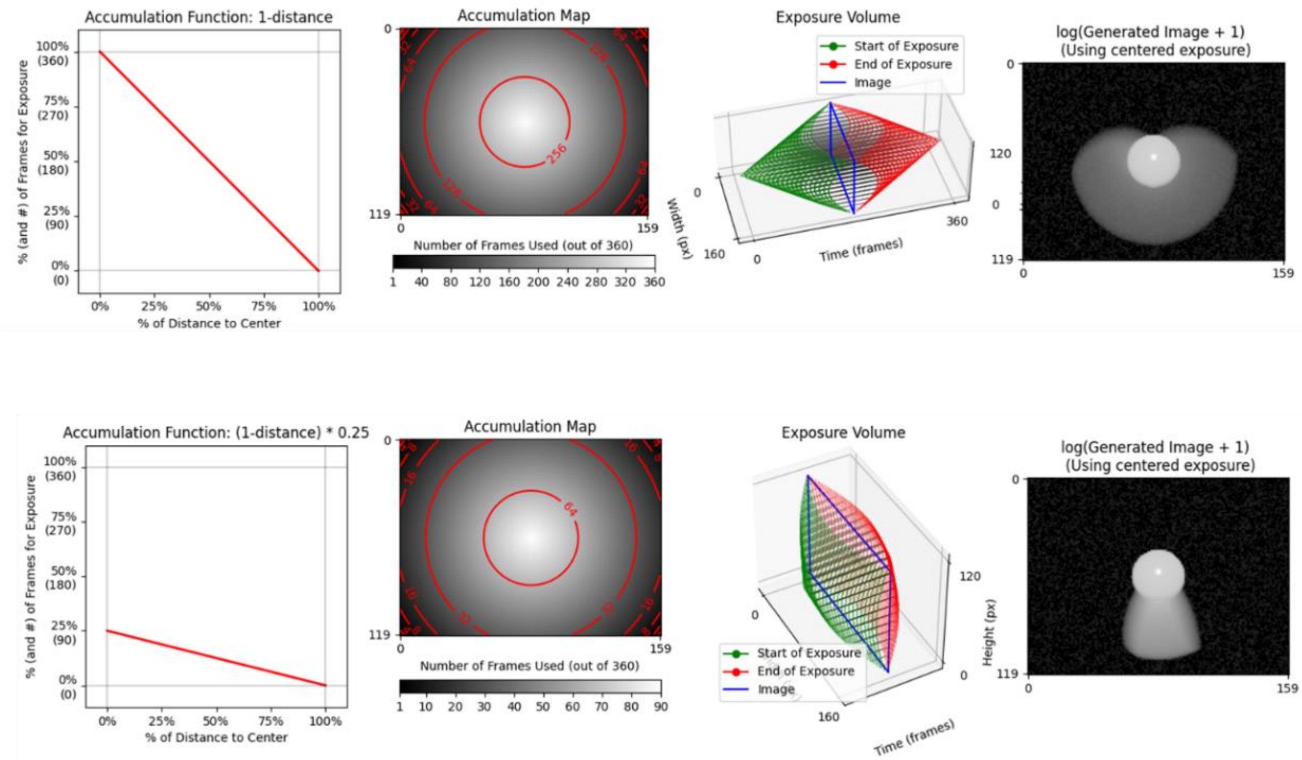

Figure 7 The exposure volume (third column) is the spatiotemporal volume from which an image is formed (right column); here, the exposure volume has been shaped such that the exposure time associated with a pixel is proportional to the velocity of the propeller as it passes through that pixel. In this way, the amount of velocity (and consequently motion-blur) is uniform across all pixels in the final image. In the images above, the exposure time is shorter (1/4 a rotation) in the second row than in the first (1 rotation).

In Figure 7, the exposure volumes have been reshaped into a back-to-back set of cones. This exposure volume ensures the same amount of motion blur is captured in each pixel by giving pixels traveling at lower velocities (near the center of the propeller) more frames of exposure, and pixels at higher velocities (near the edge of the propeller) fewer frames. This makes it possible to effectively control the motion blur across the diameter of the propeller in such a way that motion blur is effectively constant regardless of the tangential velocity of the propeller.

SHIFTING

Previous sections demonstrated the effect of ‘stretching’ the exposure volume (the arc of the propeller in the final image is broader). The exposure volume can be shifted/slid through time, which effectively changes the angular position of the propeller as captured in the image. This effect is evident in Figure 8, where the exposure volume is shifted forward in time; because the propeller rotates clockwise, the center of the blurred propeller rotates clockwise.

GENERALIZABILITY OF DIGITAL CODED EXPOSURE

Although shifting, stretching, and reshaping the coded exposure volume has utility, frame-based coded exposure can be more broadly generalized: the shape of the exposure volume can be varied through time, and the values used to determine whether a given pixel from the base video-matrix contributes to the frame may be softened from binary to floating point or even passed through functions.

Furthermore, while we have assumed previously that each pixel is summed independently along time, context-sensitive functions (e.g. convolution filters) or even distributed filters such as those used for refocusing light-field imagers may be used to mix contributions from pixels spatially.

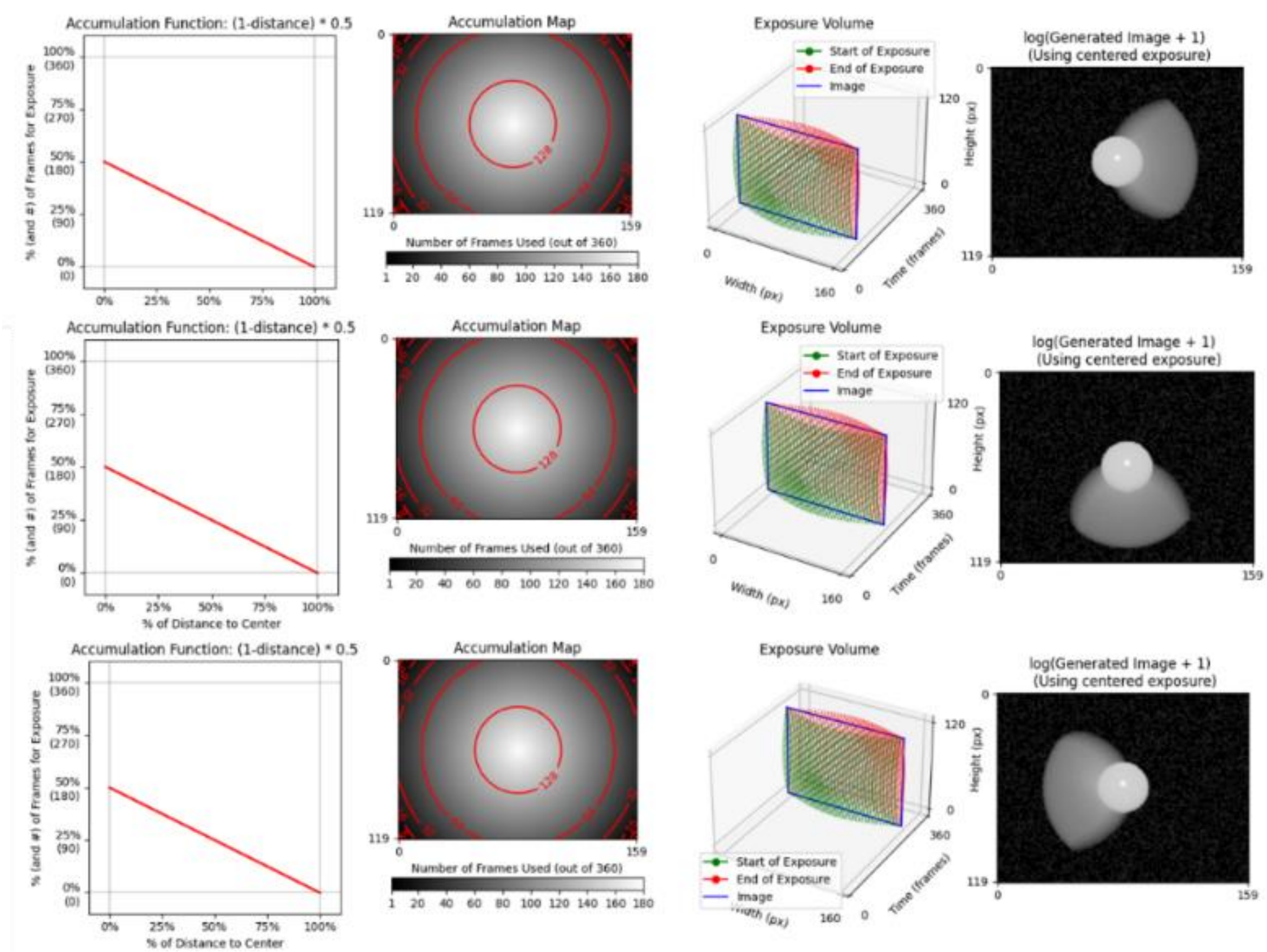

Figure 8 The exposure volume (third column) is the spatiotemporal volume from which an image is formed (right column); here, the exposure volume remains the same shape between rows, but is shifted forward in time.

CONCLUSIONS

In this work we extend the concept of the digital coded exposures for event-based imagers to conventional frame imagers and renderings of 3D scenes. We show how the digital coded exposure can simulate motion blur as well as introduce new image processing capabilities associated with the manipulation of motion blur.

REFERENCES

1. M. Pharr, W. Jakob and G. Humphreys, Physically Based Rendering: From Theory to Implementation 3rd Edition, Morgan Kaufmann, 2016.
2. Q. Guo, F. Juefei-Xu, X. Xie, L. Ma, J. Wang, B. Yu, W. Feng and Y. Liu, "Watch out! Motion is Blurring the Vision of Your Deep Neural Networks," in *NeurIPS*, 2020.
3. "Blender," [Online]. Available: https://www.blender.org/. [Accessed 17 March 2023].
4. P. Shirley, "Ray Tracing In One Weekend: The Book Series," 7 December 2020. [Online]. Available: https://raytracing.github.io/. [Accessed 9 May 2023].
5. A. Gothard, D. Jones, A. Green, M. Torrez, A. Cattaneo and D. Mascareñas, "Digital coded exposure formation of frames from event-based imagery," *Digital coded exposure formation of frames from event-based imagery,* vol. 2, no. 014005, 2022.