This is the accepted version of a paper presented at 12th International Conference on Image Processing Theory, Tools and Applications, IPTA 2023, Paris, 16 October through 19 October 2023.

Citation for the original published paper:

Goswami, P., Johansson, H., Cheddad, A. (2023)
Animated lightning bolt generation using machine learning
In: 12th International Conference on Image Processing Theory, Tools and Applications, IPTA 2023 Institute of Electrical and Electronics Engineers (IEEE)
https://doi.org/10.1109/IPTA59101.2023.10320085

N.B. When citing this work, cite the original published paper.

©2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:bth-25819
Animated lightning bolt generation using machine learning

Prashant Goswami  
Department of Computer Science  
Blekinge Institute of Technology  
Karlskrona, Sweden  
prashant.goswami@bth.se

Henrik Johansson  
Department of Computer Science  
Blekinge Institute of Technology  
Karlskrona, Sweden

Abbas Cheddad  
Department of Computer Science  
Blekinge Institute of Technology  
Karlskrona, Sweden  
abbas.cheddad@bth.se

Abstract—In this paper, we investigate the possibility of leveraging the predictive power of machine learning to generate animated lightning bolts in the image space efficiently. To this end, we selected state-of-the-art machine learning architectures based on Generative Adversarial Network (GAN) and trained them on the commonly available videos. We demonstrate that visually convincing animations are achievable even when employing a limited dataset. The visual realism of the generated sequences of lightning bolts is assessed by conducting a user study on the participants.

Index Terms—lightning, animation, GAN, machine learning

I. INTRODUCTION

Generating realistic lightning bolts in overcast outdoor settings can enhance the feel of rough weather in computer graphics (CG) and virtual reality (VR). Purely procedural methods can generate high-quality static bolts but are limited in producing convincing animations. Physics-based methods, on the other hand, entail capturing complicated environmental conditions in a three-dimensional space, making it prohibitively expensive for interactive applications in terms of computations and memory usage. Moreover, for applications like computer games or simulators, a high level of physical accuracy is often not desired so long as the perceived appearance is acceptable.

In this paper, we propose to take the benefit of machine learning capabilities to generate efficient lightning animations in the image space. We utilize the power of GAN [11] to generate a sequence of spatially and temporally coherent lightning frames starting from an initial provided image. The advantage of the image space method is that thunderbolts could be added to the rendered scene in a simple and inexpensive postprocessing step. Our findings suggested that GAN can obtain patterns close to the ground truth with a relatively small input dataset. We also performed a user study to ascertain if the produced lightning bolts were perceived as real by human participants.

II. RELATED WORK

Different techniques have been tried in CG to produce lightning effects. Kruszweski [2] reproduces lightning using probability theory and binary trees to generate a skeleton of the lightning bolt. Potential fields are used in the air to create lightning shapes in [6]. Static lightning patterns are generated on the GPU by generating a mesh between two points and randomly displacing the vertices of the mesh [5]. Bryan et al. [3] introduce a method to generate lightning bolts in a 3D virtual environment using cellular automata, generating a skeleton with streakers.

Video generation with the help of machine learning is becoming a more popular area of research [16]. Logacheva et al. [13] train a mixture of static landscape images and animations to build a new model of landscape videos by extending StyleGAN. Machine learning has been shown to produce promising results for efficient cloud animation [14], [17]. However, to the best of our knowledge, no published work to create lightning bolts using machine learning exists. This paper bridges the aforesaid gap by exploiting the properties of suitable machine learning methods to generate animations.

III. METHOD

GAN is an unsupervised machine learning system where two learning models compete in an adversarial process. GAN applies a generative model $G$ and a discriminative model $D$, which compete in this adversarial process to beat each other. During training, $D$ tries to maximize the chance of correctly labeling the generated data and the training data. During training, $G$ attempts to minimize the chance of $D$ correctly labeling the generated data as a part of the training data. In this work, we have chosen GAN due to its interesting property of generating new image and video data with similar properties as the training set. Several GAN-based machine learning architectures have recently been proposed to generate video from a provided image, for instance, dynamic time-lapses.

The new architectures of GAN are constantly growing; currently, hundreds of variants are created for different purposes [18]. Based on the findings of Nuha et al. [16] and our independent research, we selected a few state-of-the-art image-to-video generating GAN architectures for our analysis. This included stochastic adversarial video prediction (SAVP) [8], dynamic time-lapse video generation (Dtvnet) [12], stochastic image-to-video synthesis (SI2VS) [15] (using conditional invertible neural networks or cINN), multi discriminator generative adversarial net (MDGAN) [7].
A. Data pre-processing

Due to a lack of publicly available datasets containing videos of lightning strikes, we selected relevant videos from the YouTube. In order to maximize the training success and the quality of the generated videos, specific requirements had to be met by each clip included in the dataset. We ensured that the videos were shot in the daytime settings with a static camera movement (with no camera flicker or tear) and that the containing lightning bolts struck the ground. Choosing a large dataset with diverse backgrounds containing thunderbolts can help enrich the outcome. After extracting frames containing lightning, further preprocessing was done to zoom and crop the video so that the area containing the lightning bolt became the focus of the clips. To this end, over 200 clips were compiled from 11 different videos.

Following the dataset selection, the original video was cut into smaller clips, each containing a lightning bolt and each clip no longer than 16 frames, due to the size limitations in the selected machine learning architectures. The second step entailed selecting the region containing the lightning bolt and reducing the clips’ height and width from the source size to $512 \times 512$. This step produced images with bolts in the focus while still keeping some regions with the clouds and ground in the frame. Finally, all the extracted content was exported into sequences of .tif images.

B. Training

After having filtered all the clips for any issues, only 106 clips remained; 94 were used for training and 12 for evaluation. The number of frames for each video clip in the training dataset varied between 6 and 16 frames for a total of 960 frames for the training part and 104 frames for the evaluation part. The training under all the selected architectures was conducted using their default procedures with some minor modifications. For instance, training with Dtvnet is a two-step process requiring an optical flow encoder to produce motion vector field images, whereas SI2VS requires a three-step training process.

C. Perceived realism

In order to evaluate the visual quality of generated lightning bolts using machine learning, qualitative data was gathered using a questionnaire. The questionnaire was released to post-graduation game programming students using online forms.

IV. RESULTS

We used a PC with Win10 OS, AMD Ryzen 9 3900 (3.8 GHz) CPU, NVIDIA GeForce RTX 3090 Suprim x 24G GPU, and 32 GB RAM for our implementation. The implementation of SAVP, Dtvnet, SI2VS, and MDGAN were acquired from their respective GitHub pages. We enabled the GPU versions of these implementations for training and validation. Of these four architectures, only SI2VS successfully produced visible lightning bolts with an acceptable quality that resembled thunderbolts. We tried to generate thunderbolts both on existing backgrounds (containing bolts supplied for learning) and using new images, see also Fig. 1 and 2. As is visible in Fig. 2 the generated outcome using SI2VS can vary significantly from the ground truth.

A. Training time

All the input and output images had a resolution of $128 \times 128$ pixels, and each animation clip consisted of 16 frames. The training for each stage of the architectures continued until no more improvements in the Fréchet video distance [9] were possible. In the case of SI2VS, the first stage was trained for 13 hours until it reached the lowest score of 0.54, the second stage was trained for a total of 6 hours and 50 minutes, while the final stage had another 4 hours of training for a total of 5 hours and 10 minutes.

Fig. 1. Lightning animation sequences produced on different backgrounds using the stochastic image-to-video synthesis (SI2VS) machine learning architecture (original image sizes are shown).

B. User study

The questionnaire received a total of 28 responses over three weeks, wherein 10 participants were female, 16 participants were male, and 2 participants chose not to disclose their gender. The collected data from the participants were anonymous and could not be linked to them individually. The participants were informed of this and the confidentiality of their data before starting the survey. Participation was voluntary, and the user could stop the experiment at any time without providing a reason.

As a part of the questionnaire, the participants were tasked with watching a total of 10 videos depicting generated lightning bolts generated using SI2VS architecture. The videos were shown one at a time, together with a question. Each video was played in a repeat mode with no native option to pause or change the playback. The original resolution of the videos was $128 \times 128$ pixels, while an upscaled resolution of $256 \times 256$ pixels was displayed to the participants. In order to minimize the risk of the perceived bolts being too short or excessively flashy, the 16 animated frames of each clip were prepended with 14 frames and appended with 16 frames, all...
of which corresponded to the static background. Each video consisted of 45 frames played at 30 per second, resulting in 1.5 seconds of video time. A few thunderbolt sequences included in the questionnaire (as videos) are shown in Fig. [1].

Corresponding to each question on rating the perceived realism of each shown video, the participants had a choose their answer from the seven provided choices (column “Opinion” in Tab. [I]). By default, no option was pre-selected, and the participants had to make the current choice before proceeding to the following video. The total number of votes for all the questions is split into two graphs and shown in Fig. [3].

To calculate the resulting means and medians and to simplify the statistical analysis, the response options were converted to numerical values (column “Value” in Tab. [I]). This is done by assigning positive values to positive choices and negative values to negative choices, and the neutral option is given a zero.

The mean and median values corresponding to all 10 questions are given in Tab. [II]. Statistical analysis was performed to determine whether the results were statistically significant. To this end, all the responses corresponding to values 3, 2, or 1 (positive answers) in Tab. [I] were grouped into the realistic set. In contrast, those with the values -3, -2 or -1 were grouped into the unrealistic set. Since the response data did not follow a normal distribution curve, Wilcoxon-Mann-Whitney test was chosen over the t-test for the significance analysis [4]. After splitting the questionnaire data into two sets (realistic and unrealistic), the statistical significance was calculated using ranksum at the 1% statistical significance from the Python library Scipy [1], and a p-value of 0.00005 was obtained. This indicates that there is evidence of a statistically significant result, proving that the lightning bolts generated from the SI2VS model were perceived as real.

<table>
<thead>
<tr>
<th>Opinion data for all shown bolts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opinion</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>Completely realistic</td>
</tr>
<tr>
<td>Very realistic</td>
</tr>
<tr>
<td>Somewhat realistic</td>
</tr>
<tr>
<td>Can’t decide</td>
</tr>
<tr>
<td>Somewhat unrealistic</td>
</tr>
<tr>
<td>Very unrealistic</td>
</tr>
<tr>
<td>Completely unrealistic</td>
</tr>
</tbody>
</table>

TABLE I
Options given to the participants to rate the quality of each video (“Opinion”) and the corresponding numerical mapping of each option to perform statistical analysis (“Value”).

<table>
<thead>
<tr>
<th>Questionnaire answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>Mean</td>
</tr>
<tr>
<td>Median</td>
</tr>
</tbody>
</table>

TABLE II
The mean and median scores of the questionnaire answers for each question, including the overall score. The mean and median values calculated are based on the conversion shown in Tab. [I].

C. Performance

As seen in Tab. [III], the average model initializing time took 3.4 seconds; of these, 0.7 seconds was used to initialize CUDA. Subsequent initializations took 2.7 seconds, while another 2.2 seconds was added to the initialization to load CUDA for the video generation. All the various components resulted in a total of 5.6 seconds of initialization time for the SI2VS model. The average time to generate a single lightning bolt video was 0.1 seconds. This only accounts for the generation process, ignoring all preprocessing or other transformation times.

D. Limitations

One limitation we encountered was possible overfitting in some cases, as the generated data was similar to the training dataset in size. In terms of performance, our experiments suggest that it took around 0.1 seconds to generate a lightning bolt animation on any given image of the specified dimensions. This is still not real-time in terms of performance. However, machine learning could still be used for real-time applications with some level of apriori decision-making or asynchronous computation to generate thunderbolts. A general limitation when dealing with machine learning methods is the size of input training images and that of the output obtained. This is also true in our case, wherein we dealt with small image sizes
to reduce the training and generation time. However, there exists a possibility to overcome this limitation by using the output from SI2VS as input maps for a much higher resolution 3D rendering, for instance, in [10].

V. CONCLUSIONS

We have proposed to leverage the power of machine learning to generate visually realistic lightning animations in the image space efficiently. We tried four architectures of the GAN family to this end. Out of these, the stochastic image-to-video synthesis (SI2VS) method emerged as a viable candidate to generate lightning bolts of acceptable quality. The data employed for training and validation was derived from publicly available videos containing thunderbolts and was limited in size. We have shown that realistic appearing animations can be obtained with a suitable machine learning architecture even for relatively low input training data. We verified the visual quality of generated animations through a user study and performed a statistical significance analysis to generalize our findings to a larger population. Moreover, our findings expand on possible applications of machine learning outside of already established areas and known datasets. In future, the suitability of more machine learning architectures could be explored for generating more complex lightning patterns.

VI. SUPPLEMENT MATERIAL

The videos showing our approach and comparison to actual lightning animations is available here.