A Comparative Study of EmberGen and Blender in Fire Explosion Simulations

Arya Luthfi Mahadika [1]*, Ema Utami [2]
Department of Master Informatics [1], [2]
University of AMIKOM Yogyakarta
Yogyakarta, Indonesia
arya.dika@students.amikom.ac.id[1], ema.u@amikom.ac.id[2]

***Abstract*— The advancement of visual effects (VFX) technology has intensified the need for efficient fire explosion simulations across film, gaming, and real-time applications. This study investigates and compares the performance of two prominent simulation tools—EmberGen and Blender—by focusing on processing time efficiency and simulation quality. The research specifically evaluates five critical simulation aspects: fire particle generation, smoke behavior, turbulence effects, light dispersion, and final rendering (finishing). A total of five professional VFX artists conducted five separate tests using each software, generating a comprehensive dataset for analysis. Results show that EmberGen achieves a 29.91% overall improvement in simulation speed compared to Blender, with significant gains in fire particle generation (38.5%), smoke simulation (42.3%), turbulence effects (15.7%), light dispersion (8.9%), and finishing (11.6%). These findings indicate that EmberGen is highly effective for real-time or rapid-turnaround projects, while Blender remains advantageous for detailed, high-fidelity simulations in cinematic contexts. The study concludes that software selection should be driven by project-specific demands, where EmberGen supports time-sensitive production workflows and Blender offers greater artistic control. This research underscores the critical need for aligning simulation tools with both creative goals and production efficiency, contributing to decision-making in VFX, animation pipelines, and educational training environments within the information systems and digital content domains.**

***Keywords*— EmberGen, Blender, Fire Explosion Simulation, Visual Effects**

I. Introduction

Visual effects (VFX) have become a critical component in modern filmmaking, enabling the creation of scenes that defy physical and natural constraints [1], [2]. From their origins in the pioneering techniques of Georges Méliès such as stop motion and matte painting VFX have evolved into sophisticated digital tools that enhance storytelling, audience immersion, and production efficiency [3]. The integration of VFX not only improves the visual quality of a film but also plays a strategic role in reducing risk and cost during the filming of dangerous or complex scenes, such as explosions or natural disasters [1], [4]. As a result, VFX are now widely used in both blockbuster productions and independent short films, serving as key visual and commercial assets.

With the growing reliance on VFX, professionals increasingly turn to digital tools capable of delivering high-quality simulations under tight production deadlines. Two prominent software platforms used for simulating fire explosions are EmberGen and Blender. EmberGen is known for its real-time GPU-based simulation capabilities, offering immediate visual feedback that allows artists to iterate quickly and meet demanding schedules [5]. EmberGen, a VFX program developed by JangaFX, enables real-time simulation and playback of fire, smoke, explosions, and other effects [6]. In contrast, Blender is a powerful, open-source 3D creation suite with extensive functionality, including a fluid simulation engine capable of producing highly detailed and customizable fire and smoke effects [7], [8]. However, Blender's complexity and longer rendering times may pose challenges for time-sensitive projects focused specifically on explosion simulations [8].

Despite the popularity of both tools, there is limited empirical research comparing their performance in fire explosion simulation workflows, particularly in the context of computational efficiency, rendering quality, and usability. Existing studies tend to focus on the technical features of each tool individually, without offering a structured, side-by-side performance analysis that can guide practical software selection in professional VFX environments. This lack of comparative analysis presents a research gap [8], [9], [10], [11], especially for practitioners and educators seeking optimal tools for animation, visual storytelling, or interactive media production.

Several previous studies have utilized for fire simulation through a combination of particle systems and mesh emitters, where a sphere serves as the particle emitter to generate orange spheres that represent flames [7], [12], this process involves animating the size of the spheres over time to mimic the natural flicker and behavior of fire, while a lattice is employed to shape the emitted particles and enhance the visual complexity of the flames. Additionally, shaders are applied to create realistic coloration and lighting effects, and the Node Editor is leveraged to integrate advanced techniques such as Vector Blur, which simulates motion blur, and Ramp Nodes, which adjust color gradients to enhance the fire's luminosity, resulting in a visually compelling and dynamic fire simulation [7], [13]. In previous research, there has never been an experiment using EmberGen software; therefore, the author aims to conduct a study using this new method with EmberGen software.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i2.2335, Copyright ©2025
Submitted : April 23, 2025, Revised : May 10, 2025, Accepted : May 14, 2025, Published : May 26, 2025

This study aims to address that gap by evaluating and comparing the performance of EmberGen and Blender in generating realistic fire explosion simulations in “I Draw It” short movie. The comparison is based on three primary dimensions: rendering quality, ease of use, and computational efficiency. By examining these aspects, the study contributes practical insights for VFX professionals, educators, and digital content creators in selecting software that aligns with both creative goals and production demands [2], [14]. The research also offers a basis for future investigations into hybrid workflows that combine the strengths of both tools for optimal results in various production contexts.

II. RESEARCH METHOD

Based on the research title, the author created a research flow for the stages or pipeline of the Performance Comparison of EmberGen and Blender in Fire Explosions, as shown in Figure 1 below.

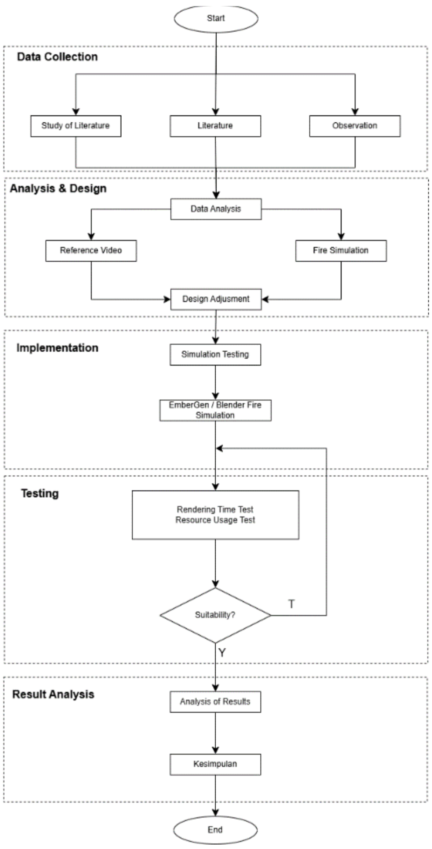

Fig. 1. Reseearch Flow of Performance Comparison Between EmberGen and Blender in Fire Explosions

This research focuses on comparing the performance of EmberGen and Blender in generating fire explosion simulations. The initial stage involves collecting literature and reference videos related to fire explosions. The literature review aims to understand the fundamental principles of fire simulation, rendering techniques, and the features offered by EmberGen and Blender. Reference videos are gathered as visual benchmarks for analyzing the realism and quality of simulations produced by both software.

The next stage is Analysis & Design, where the simulation process is planned and adapted for both software. EmberGen and Blender are tested using specific methods such as frame sampling to control simulation intervals, blending to combine visual elements, and fire particle motion analysis. Adjustments are made iteratively, with each software’s simulation results compared to the reference videos to evaluate accuracy and realism.

In the Implementation and Testing phase, the designed simulations are executed in EmberGen and Blender to assess their performance. The testing process includes quantitative analysis, such as rendering time and resource efficiency, as well as qualitative evaluation of the fire explosion visuals. The simulation results are then assessed based on parameters like realism, efficiency, and ease of use. The study concludes with an in-depth analysis of the testing results, which are used to determine which software performs better in generating fire explosion simulations.

TABLE IV. FIRE EXPLOSION SIMULATION

Reference	*Fire Visual Effect*
	

This study observes the simulation workflow to collect data on the performance and output of two simulation methods, EmberGen and Blender, in a fire explosion scenario. The collected data is analyzed to evaluate the effectiveness of each method, which serves as a reference for dynamic simulations in computer graphics visual effects. Although fire explosion simulations may seem simple, in-depth analysis reveals significant fluctuations and complexity. These simulations include key elements of realistic animation, such as physical accuracy, visual fidelity, and overall rendering performance [15].

EmberGen and Blender have unique approaches to creating fire explosion simulations. In the simulation process, EmberGen employs a voxel-based approach, enabling fast calculations with high accuracy. Its parameter control system allows users to easily modify materials, fluid dynamics, and lighting effects in fire and explosion simulations. This study compares EmberGen’s simulation performance with Blender, focusing on visual resolution, material adjustments, and rendering time efficiency [6].

Through this comparison, key aspects such as rendering time, user control flexibility, and the aesthetic appeal of the simulation results are measured to provide insights applicable to film production and real-time applications. The results show that EmberGen excels in rendering speed and ease of use, while Blender offers greater flexibility in adjusting simulation details and producing varied visual outcomes according to project needs [7], [15].

The collected data is analyzed in detail to calculate time and determine optimal animation parameters for each fire explosion

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i2.2335, Copyright ©2025
Submitted : April 23, 2025, Revised : May 10, 2025, Accepted : May 14, 2025, Published : May 26, 2025

simulation. This analysis includes the assessment of creation duration, motion details, and other technical aspects to ensure the simulation results achieve the desired realism. After the analysis is completed, the author proceeds with the design validation process as an essential step to ensure the results align with the research objectives and standards. This validation is conducted in collaboration with three visual effects experts who have experience and expertise in creating high-quality animation simulations. Their presence aims to identify and minimize potential errors before the final implementation, ensuring more accurate and effective simulation results.

During the testing phase, fire explosion simulations are performed using two main software programs, EmberGen and Blender. Each software is tested by three professionals, with each performing five simulations per software. These repeated tests aim to collect consistent and comprehensive data, allowing for an in-depth analysis of the effectiveness and efficiency of both methods. Throughout this process, animation supervisors actively evaluate each simulation result, providing critical feedback and necessary improvements to enhance both the visual and technical quality of each animator’s work. With this approach, the study not only produces quantitative data but also establishes a high-quality standard for each tested simulation.

III. Result And Discussion

The performance comparison process between EmberGen and Blender in fire explosion simulation begins with analyzing the visual characteristics of fire explosions through reference videos. These videos are observed from various angles to understand the natural movement of fire, including spread patterns, intensity, and dissipation. Simulations are conducted using two different software programs, EmberGen and Blender, to assess their capabilities in rendering fire explosions. The simulation setup involves modifying various parameters, such as particle density, turbulence, and temperature variations, to match the reference footage.

The next step is to apply fire simulation techniques in EmberGen and Blender. In EmberGen, fire simulation is performed procedurally in real-time using voxel-based rendering, allowing for instant adjustments and immediate feedback. On the other hand, Blender utilizes the Mantaflow solver for fluid simulation, which requires preprocessing before generating the final render. This difference impacts workflow efficiency in fire effect animation production. Adjustments are made in both software to optimize realism and responsiveness, ensuring that the simulated fire movement accurately reflects natural flame behavior.

Finally, performance evaluation is conducted by comparing rendering time, resource usage, and visual realism between EmberGen and Blender. Rendering speed is tested by analyzing frames per second (FPS) and computational load, while resource usage is measured based on CPU consumption during the simulation. Visual realism is evaluated by comparing the generated fire effects with the reference footage, assessing aspects such as flame flickering, smoke behavior, and heat distortion. The results of this comparison will determine which software is more efficient and effective for fire explosion simulation in animation and visual effects production.

Fig. 2. Fire Explosion Simulatioon in EmberGen

After each fire explosion simulation is completed, the results are presented in tables and graphs comparing the performance of EmberGen and Blender. The evaluation is based on rendering time, resource efficiency, and the visual realism produced. In this test, both software programs are given the same initial parameters, such as explosion size, light intensity, and smoke density, to ensure a fair comparison. The success of the simulation depends on each software’s ability to generate realistic fire effects efficiently.

The simulation testing process is conducted in five main stages: initial explosion simulation, flame development, smoke movement, turbulence effects, and the final stage of fire dispersion before fading [16], [17]. In EmberGen, simulations run in real-time, allowing animators to instantly see parameter adjustments without requiring a baking process. Meanwhile, Blender, with the Mantaflow solver, requires additional processing time to calculate fluid dynamics before achieving the final result [8], [12]. This difference creates a distinct workflow experience, where EmberGen offers higher responsiveness, while Blender provides more detailed control during the refinement stage.

As part of a comprehensive performance evaluation, each simulation method is rigorously tested by measuring the time required to complete various stages of the fire explosion simulation. This assessment aims to provide a detailed understanding of the efficiency and practicality of each software tool involved. The results of these tests are systematically organized into a table that outlines the duration of each simulation phase in minutes, offering a clear comparative view of performance across the different tools [7].

By closely examining this data, it becomes possible to draw meaningful conclusions about the overall effectiveness of each software solution. Specifically, the analysis focuses on three key aspects: rendering speed, which reflects how quickly the software can produce visual outputs; physical accuracy, which evaluates how realistically the explosion is simulated according to real-world physics; and ease of use, which considers the user interface and workflow efficiency for animation and visual effects production [18], [19]. This comprehensive evaluation not only highlights which software is faster, but also which one strikes the best balance between speed, realism, and usability crucial factors for professionals in the fields of animation and visual effects [7], [20].

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i2.2335, Copyright ©2025
Submitted : April 23, 2025, Revised : May 10, 2025, Accepted : May 14, 2025, Published : May 26, 2025

TABLE II. TEST RESULT OF ONE OF THE VISUAL EFFECTS ARTISTS

Test in Time								
No	***Software***	***Simulation Aspects***	***Test 1***	***Test 2***	***Test 3***	***Test 4***	***Test 5***	***Average***
1	**Blender**	Fire Particle Generation	01.28	01.20	01.27	02.22	01.30	**01.37**
		Smoke Simulation	01.27	01.54	01.54	01.52	01.24	**01.42**
		Light Dispersion	02.00	01.47	01.26	02.32	02.31	**02.03**
		Turbulence Effects	01.31	01.53	01.52	02.07	01.42	**01.49**
		Finishing	02.21	01.56	02.03	01.54	02.24	**02.07**
2	**EmberGen**	Fire Particle Generation	01.18	01.31	01.08	00.54	01.22	**01.14**
		Smoke Simulation	00.47	01.00	01.18	01.37	00.58	**01.08**
		Light Dispersion	01.02	00.50	00.50	01.03	01.32	**01.03**
		Turbulence Effects	01.15	01.19	0103	00.42	01.01	**01.04**
		Finishing	01.32	00.41	01.03	01.23	01.09	**01.09**

After recording the time measurements for each test conducted by the Visual Effects Artist, the results are compiled into a Fire Explosion Simulation test result table, using seconds (s) as the unit of measurement. This table details the time taken for each part of the fire simulation and the average processing time for two software programs: Blender and EmberGen. The testing results for Blender and EmberGen fire simulations are presented in Table 3 and Table 4 below.

TABLE III. TESTING ON BLENDER FIRE SIMULATION

Testing on Blender Fire Simulation in seconds						
	Test 1	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Average**
VFX Artist 1	320	298	310	290	315	**306.6**
VFX Artist 2	580	460	430	510	495	**495**
VFX Artist 3	260	280	300	310	200	**270**
VFX Artist 4	400	370	340	320	315	**349**
VFX Artist 5	290	270	250	260	240	**262**
Average Fire Simulation						**336.52**

Based on the data in Table 3, the fire simulation in Blender, across five test trials, has an average processing duration of 336.52 seconds, equivalent to approximately 5 minutes and 36 seconds.

TABLE IV. TESTING ON EMBERGEN FIRE SIMULATION

Testing on EmberGen Fire Simulation in seconds						
	Test 1	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Average**
VFX Artist 1	150	140	145	135	148	**143.6**
VFX Artist 2	290	270	260	280	275	**275**
VFX Artist 3	130	120	125	140	110	**125**
VFX Artist 4	220	210	200	195	180	**201**
VFX Artist 5	180	160	150	170	155	**163**

According to Table 4, the fire simulation using EmberGen across five test trials has an average processing time of 181.52 seconds. The time testing results from five Visual Effect Artists show varying outcomes for each simulation aspect, including Main Fire, Smoke Dispersion, Light Dispersion, Turbulence Effects, and Finishing. Consequently, the comparative testing between Blender and EmberGen is presented in Table 5 below.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i2.2335, Copyright ©2025
Submitted : April 23, 2025, Revised : May 10, 2025, Accepted : May 14, 2025, Published : May 26, 2025

TABLE V. COMPARISON TESTING BLENDER AND EMBERGEN

No		***Blender***	***EmberGen***
1	**VFX Artist 1**	306.6	143.6
2	**VFX Artist 2**	495	275
3	**VFX Artist 3**	270	125
4	**VFX Artist 4**	349	201
5	**VFX Artist 5**	262	163
Averages		**336.52**	**181.52**

Based on the results in Table 5, the fire simulation processing time shows a significant difference between the two software programs. EmberGen demonstrates nearly twice the efficiency of Blender in simulating fire explosions.

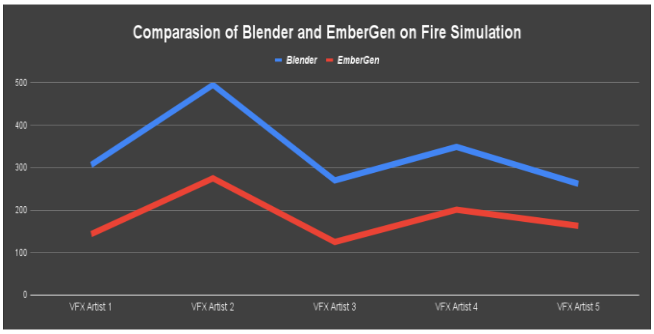

Fig. 3. Graphics Comparison of blender nnd EmberGen on Fire Simulation

Based on the result in Figure 3 above, the fire explosion simulation demonstrates that EmberGen has a higher processing speed compared to Blender. This is reflected in both the table and graph, where the average processing time for EmberGen is 181.52 seconds, whereas Blender requires 336.52 seconds. The difference in processing time between the fire explosion simulation methods using EmberGen and Blender is 155 seconds. This value is calculated based on the average time taken by each method to perform the fire explosion simulation and is considered an indicator of each software's efficiency. Furthermore, calculations were performed to determine the percentage of simulation time efficiency, with the result explained in equation 1 below:

$$Persentase = \frac{Difference\ in\ processing\ time}{\Sigma\ Total\ processing\ time} \times 100\% \qquad (1)$$

$$= \frac{155}{518.04} \times 100\% \quad = 29.91\%$$

According to equation 1, to obtain the percentage of the fire explosion simulation time rate using four idealized rules, first, find the difference in processing time (between Blender and EmberGen), second, find the total processing time, third, divide the difference in processing time by the total processing time, fourth, multiply the result of that division by 100.

Based on processing efficiency calculations, there are four key steps in comparing the performance of EmberGen and Blender in fire explosion simulations.

The first step is determining the difference in processing time between the two methods. The time difference between them is 155 seconds. The second step is calculating the total processing time by summing the times of both methods, resulting in 518.04 seconds. Next, the third step involves dividing the time difference by the total processing time and multiplying the result by 100 to obtain the efficiency percentage. From this calculation, it is determined that EmberGen is approximately 46.1% faster than Blender in terms of processing efficiency.

According to earlier studies, using Blender software to generate a fire simulation takes longer, around 16 minutes [12]. The test results indicate that EmberGen excels in both speed and ease of use, making it more suitable for projects requiring fast results, such as fire effects in games or real-time animations. However, Blender retains an advantage in its flexibility, particularly for simulations involving complex interactions, such as explosions that affect surrounding objects. The supervisor in this study observed that while EmberGen can generate realistic simulations in a shorter time, Blender is more reliable for detailed physics settings and more complex effect customizations. Nevertheless, this investigation is restricted to visual impressions and time-based performance in a controlled environment that includes five artists.

In visual effects production, choosing the right software should align with the project's specific requirements. If speed and efficiency are the main concerns, EmberGen is the preferable option. However, for projects demanding greater detail and manual control over simulations, Blender is the more suitable choice. In some scenarios, integrating both software can offer an optimal solution using EmberGen for quick simulations and Blender for refining and fine-tuning explosion effects. Ultimately, while EmberGen has demonstrated superior speed compared to Blender, the ideal choice depends on the unique demands of the visual effects production.

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i2.2335, Copyright ©2025
Submitted : April 23, 2025, Revised : May 10, 2025, Accepted : May 14, 2025, Published : May 26, 2025

IV. CONCLUSION

This study reveals a notable performance edge for EmberGen over Blender in fire explosion simulations, with EmberGen achieving a 29.91% faster overall processing time. It consistently surpassed Blender in all tested areas, particularly in smoke simulation 42.3% and fire particle generation 38.5%. These findings demonstrate EmberGen's effectiveness for real-time applications and rapid prototyping, where speed is essential. Conversely, Blender excels in flexibility and parameter control, making it suitable for high-detail simulations in cinematic visual effects.

The practical takeaway from this research is a strong recommendation for a hybrid workflow: using EmberGen for time-sensitive tasks like video game effects or pre-visualization, while turning to Blender for intricate, layered simulations that require precise artistic control. This combined approach could enhance production efficiency without compromising visual quality.

However, this research is limited to time-based performance and visual impressions from a controlled environment involving five artists. Future studies should investigate other factors such as rendering quality, user experience, GPU efficiency, and cross-platform integration. Additionally, broadening the scope to include other VFX tools like Houdini or Unreal Engine's Niagara system, as well as more complex interactive simulations, could yield valuable insights into best practices for fire simulation workflows in professional settings.

REFERENCES

[1] M. S. Hwang, M. K. Park, and H. S. Lee, "A Study on the Visual Effects Production Process for Efficient Underwater Explosion CG Visualization," *J Coast Res*, vol. 116, no. sp1, pp. 518–522, 2024, doi: 10.2112/jcr-si116-105.1.

[2] V. Ong, *Artificial intelligence in digital visual effects*. dr.ntu.edu.sg, 2021. [Online]. Available: https://dr.ntu.edu.sg/handle/10356/151632

[3] M. S. Hwang and H. Lee, "Pipeline Design for Efficient Visual Effects Production," *Journal of Multimedia Information System*, 2022, [Online]. Available: https://www.jmis.org/archive/view_article?pid=jmis-9-3-219

[4] M. Choi, J. A. Wi, T. Kim, Y. Kim, and C. H. Kim, "Learning representation of secondary effects for fire-flake animation," *IEEE Access*, 2021, [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9334976/

[5] JangaFX, "EmberGen Documentation," JangaFX. Accessed: Jan. 15, 2025. [Online]. Available: https://jangafx.com

[6] D. A. Akinkugbe, "Artifact," *RIT Digital Institutional Repository*, pp. 1–34, 2021.

[7] D. Senkic, "Dynamic Simulations in a 3D-Enviornment a comparison between Maya and Blender," 2010.

[8] V. Hoikkala, *Fire Simulation for a 3D Character with Particles and Motion Capture Data in Blender*, no. May. 2016.

[9] Z. H. Wu, Z. Zhou, and W. Wu, "Realistic fire simulation: A survey," *Proceedings - 12th International Conference on Computer-Aided Design and Computer Graphics, CAD/Graphics 2011*, pp. 333–340, 2011, doi: 10.1109/CAD/Graphics.2011.26.

[10] R. H. Lewis *et al.*, *Fire and smoke digital twin – A computational framework for modeling fire incident outcomes*, vol. 110, no. 1. Association for Computing Machinery, 2024. doi: 10.1016/j.compenvurbsys.2024.102093.

[11] A. A. Khan, M. A. Khan, K. A. Cashell, and A. Usmani, "An open-source software framework for the integrated simulation of structures in fire," *Fire Saf J*, vol. 140, no. June, p. 103896, 2023, doi: 10.1016/j.firesaf.2023.103896.

[12] "Primjena novih mogućnosti simulacije fluida, vatre i dima u programu BLENDER," 2025.

[13] H. Yang, C. Hu, G. Li, and J. Fan, "A Fire Escape Simulation System Based on the Dijkstra Algorithm.," *Comput. Syst. Sci. Eng.*, 2021, [Online]. Available: https://cdn.techscience.cn/ueditor/files/csse/TSP_CSSE-39-3/TSP_CSSE_16377/TSP_CSSE_16377.pdf

[14] J. Lever and T. Komura, "Real-time controllable fire using textured forces," *Visual Computer*, vol. 28, no. 6–8, pp. 691–700, 2012, doi: 10.1007/s00371-012-0684-1.

[15] P. Beaudoin, S. Paquet, and P. Poulin, "Realistic and controllable fire simulation," *Proceedings - Graphics Interface*, pp. 159–166, 2001.

[16] S. Rødal, G. Storli, and O. E. Gundersen, "Physically based simulation and visualization of fire in real-time using the GPU," *Theory and Practice of Computer Graphics 2006, TPCG 2006 - Eurographics UK Chapter Proceedings*, no. June, pp. 13–20, 2006.

[17] W. Jahn, G. Rein, and J. L. Torero, "The effect of model parameters on the simulation of fire dynamics," *Fire Safety Science*, pp. 1341–1352, 2008, doi: 10.3801/IAFSS.FSS.9-1341.

[18] D. Q. Nguyen, R. Fedkiw, and H. W. Jensen, "Physically based modeling and animation of fire," *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '02*, pp. 721–728, 2002, doi: 10.1145/566570.566643.

[19] M. Son, B. Kim, G. Wilensky, and S. Lee, "Still-frame simulation for fire effects of images," *Computer Graphics Forum*, vol. 32, no. 7, pp. 295–304, 2013, doi: 10.1111/cgf.12237.

[20] L. Xu, D. Mohaddes, and Y. Wang, "LLM Agent for Fire Dynamics Simulations," no. NeurIPS, 2024, [Online]. Available: http://arxiv.org/abs/2412.17146

p-ISSN 2301-7988, e-ISSN 2581-0588
DOI : 10.32736/sisfokom.v14i2.2335, Copyright ©2025
Submitted : April 23, 2025, Revised : May 10, 2025, Accepted : May 14, 2025, Published : May 26, 2025