CG Fire Simulation in Films and Games: A Comparative Study

Jiakang Lu

School of Computer Science and Engineering, Hebei GEO University, Shijiazhaung, Hebei, China

Kangkangjialu@gmail.com

Abstract. Fire shows up constantly in film visual effects and in video games, which makes it one of the most widely used - and hardest to get right - effects in computer graphics. This survey covers fire simulation work published between 2006 and 2026, picking out 54 papers that we think are broadly representative of the field. We sort these methods into six families: physics-based simulation, particle systems, procedural and data-driven approaches, neural style transfer, neural reconstruction, and AI-based video generation. Within each family we review the main techniques and weigh them against one another on physical fidelity, computational cost, temporal consistency, visual realism, and how much control an artist actually ends up with. We close by summarizing where the field seems to be heading and flagging the problems that still need to be solved.

Keywords: Fire Simulation, Fluid Simulation, Particle Systems, Neural Rendering, AI Video Generation, Computer-Generated Imagery.

1. Introduction

Of all the effects a CG artist is asked to produce, fire is probably the hardest to make convincing. At the physical level it is a translucent, self-emitting fluid that couples combustion chemistry, turbulent flow, radiative heat transfer, and soot transport together, and because its behavior is nonlinear over many scales, realism and controllability are constantly at odds. The tension shows up everywhere - in a film explosion, a burning building, or a flame-fueled character, and on the game side in a torch, a flame trap, or a skill effect. Fire also behaves differently from smoke or water: it generates its own energy through chemical reaction, which couples motion with heat release, buoyancy, and fuel consumption, and its wide dynamic range ties luminance, shape, and motion together in ways that are hard to separate.

Over the past two decades the field has moved through several generations: procedural perturbation [1], then particle systems [2], then physics-based solving [3][4][5], and most recently neural reconstruction and generative synthesis [6][7][8]. Whatever the generation, four problems keep resurfacing. First, full physics is expensive and often not worth its cost [5]. Second, the behavior is multi-scale and high in dynamic range, so an adaptive treatment is really required [9]. Third, rendering a translucent, self-emitting fluid - especially in real time - remains genuinely hard [10][11]. And fourth, physics by itself offers no artistic direction, so we still need explicit control [12][13]. We use these four as the yardstick against which each method category is judged below.

The two audiences we have in mind pull in opposite directions. Offline film work will put up with hours or even days of solve time to buy fidelity and artistic control [14], whereas a real-time game treats frame rate, memory, and power as hard limits [15][16][17]. Earlier surveys [18][19][20] helped shape how we organized this paper, but none of them compared methods under the combined constraints of film and games at once. So we took 54 representative works from 2006 to 2026, grouped them into six paradigms, and compared them head to head on physical accuracy, computational cost, controllability, and visual realism, with the full comparison in Section 8.

2. Physics-Based Combustion and Fluid Simulation

2.1. Pure Eulerian Grid Solvers

Nielsen et al. [5] bring radiative heating, detailed fuel chemistry, and validated soot formation and oxidation models together, turning combustion from a visual approximation into a physically interpretable process where energy, species, and emission are solved at the same time. The Combustion 2 solver Weta built for Avengers: Endgame [3] adds oxidant, heat, and product scalar fields and uses the heat channel as a volumetric mask, which gives artists a convenient handle for controlling explosions and flames (Figure 1). A related line from Nielsen et al. [9] makes smoke simulation in Bifrost spatially adaptive, using collocated velocities and a matrix-free adaptive multigrid projection so that solver cells concentrate where the flow actually needs them.



Figure 1. Explosion rendering in *Avengers: Endgame* with and without the heat channel of the Combustion 2 solver [3].

2.2. Hybrid Particle-Grid and Multiphysics Frameworks

Loki [4] appears to be the first production framework to systematize shared rules and bidirectional multiphysics coupling among solids, fluids, and secondary particles - a solid can push a fluid, and the fluid's pressure can in turn deform the solid. Its combustion pipeline, seen in Avatar: The Way of Water [14], tracks the flame front with SDFs and uses real fuel chemistry to produce torches, fire tornadoes, flamethrowers, and large explosions (Figure 2). Kala et al. [21] handle solid combustion with fire, smoke, and ash inside a thermomechanical, hybrid incompressible MPM; because the flame-front speeds are art-directable, wood and cloth will burn but also crack and collapse (Figure 3). Flynn et al. [22] couple a FLIP/APIC bubble solve with a volumetric water field to get art-directable underwater explosions with mushrooming two-phase surfaces.



Figure 2. Combustion shots in Avatar: The Way of Water built on the Loki multiphysics framework [14].



Figure 3. Solid combustion with fire, smoke, and ash simulated by the hybrid thermomechanical MPM solver [21].

2.3. Specialized Combustion Frameworks for Specific Scenarios

FLAMEFORGE [23] voxelizes wooden structures and uses an SDF to model char-layer insulation, so it captures the staged burning where a charred surface gradually slows the fire's progress. Fire-X [24] treats ignition and extinction within one unified scheme based on multi-species thermodynamics and stoichiometric heat release; it runs on the GPU via WGSL and uses a hybrid SPH-grid representation, which makes interactive firefighting possible. Hädrich et al. [25] push mesoscale wildfires up from single trees to whole ecosystems, with detailed 3D tree models whose drying, ignition, and flaming spread across terrain (Figure 4). Pirk et al. [26] support interactive wood combustion across multiple botanical trees using connected particles and polygon burning surfaces. Tran [27] documents a Houdini Pyro workflow used for the torches, braziers, and fire traps of a short

film, a good example of how the physics-based paradigm gets packaged into commercial tools that small teams can actually use.



Figure 4. Mesoscale wildfire spread simulated with detailed tree models [25].

3. Particle System Methods

3.1. GPU Particle System Frameworks

Drone [2] lays out a fully GPU-resident, non-parametric particle design that folds N-body gravitation, flocking, and environmental response into a single emission-and-dynamics pipeline. Rødal et al. [28] were among the first to bring physically based fire to the GPU in real time, using a lightweight combustion model where each particle carries its own temperature and fuel. Zeler and Rohleder [29], who built Techland's system, pack maximum data density into the emission-to-rendering path (Figure 5) so that a shipped game can hold thousands of concurrent effects within its memory and bandwidth budget. Yin et al. [30] steer real-time particle flames with simplex noise, texture mapping, and collision, and in their view the shader is the main lever an artist has.



Figure 5. GPU particle-based fire and smoke effects for a modern video game [29].

3.2. GPU/CUDA-Accelerated Burning Object Simulation

Amarasinghe and Parberry [31] burn polygonal models in real time on CUDA by expanding heat-conduction boundaries and displacing vertices, so the burning front sweeps across the mesh surface and gradually eats away the geometry. Wei and Huang [32] parallelize a grid-free Lagrangian particle flame, including chemical composition, on CUDA and report about a 2.5x speedup from data-parallel combustion terms like ignition temperature and fuel mixing.

3.3. Fire-Flake/Sprite Animation and Visual Enhancement

Kim and Lee [33] synthesize fire-sprite animation straight from real fire footage, extracting features and adding artificial buoyancy and fire-flake textures (Figure 6), which effectively turns captured video into a game-ready sprite database. In later work the same authors [34] drop lookup-based reproduction in favor of a predictor trained on chaotic-advection data to approximate where embers will travel. Fei [35] tackles the ordering of multi-layer flames with a deep stripping algorithm plus texture mapping, and reports 98% accuracy when sorting semi-transparent layers with no visible artifacts.

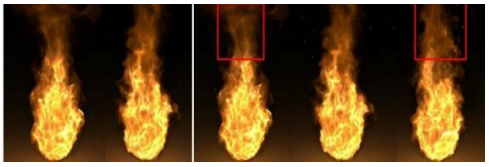


Figure 6. Fire-ball sprite animation synthesized from real fire video: top, input video frames; bottom, synthesized sprites [33].

3.4. Engine Ecosystems and Performance Optimization

He [36] puts together a physically based GPGPU pipeline in Unity that folds fluid dynamics, volumetric radiance, and volume rendering into a single compute-and-shader module, so artists can tune it without leaving the engine. Hoikkala [37] works out a Blender particle-fire workflow combined with motion-captured characters, and puts the emphasis on how fire composites with character animation. Zhang [15] optimizes UE4 particles through GPU emitters, fewer material instructions, and LOD, which reduces per-emitter cost inside a fixed frame budget. Shulha et al. [16] argue for mobile particle simplification - fewer particles, texture-replaced geometry, shader optimization - and back it up with measurements of the quality-versus-cost trade-off on portable devices. LeBlanc [17], looking at Unity 6, ranks overdraw reduction, geometry simplification, and culling, and concludes that rendering efficiency is often what decides whether an effect is usable in production.

4. Procedural Generation and Data-Driven Methods

4.1. Procedural Volumetric Fire

Fuller et al. [1] combine curve-based free-form deformation, hardware-accelerated volume rendering, and improved Perlin noise in what is now a classic architecture: procedural detail carries the small scale while particles supply the macro structure, and fire profile textures remain artist-customizable. Bannister and Nowrouzezahrai [38] optimize latent fractal parameters under image supervision through differentiable rendering, broadening IFS fractal synthesis to color, nonlinear, and multi-fractal forms (Figure 7).



Figure 7. Learnable fractal flame art synthesized from image supervision [38].

4.2. Volume Rendering Enhancement and Real-Time Volumetric Techniques

Fraboni et al. [10] get unbiased refraction - heat haze, shock waves, caustics - out of null-scattering volume rendering, computing how light bends through hot air instead of approximating the effect. Ignis [11] simulates and renders fire at more than 90 frames per second at VR resolutions, leaning on approximate occlusion and adaptive dithering (Figure 8). Rašeta et al. [39] render volumetric explosions in the browser in real time on WebGPU with ray marching and volume textures.



Figure 8. VR fire fluid simulation and rendering with Ignis [11].

4.3. Data-Driven Super-Resolution Synthesis

Sato et al. [40] precompute a database of high-resolution 2D velocity fields and recombine them at runtime - roughly 0.34 seconds per step - to enrich low-resolution fire. The idea leans on the statistical self-similarity of turbulence, and it reads today as an early data-driven precursor to the generative methods in Sections 6 and 7.

4.4. Target-Driven Controllable Fire Simulation

Mizutani et al. [12] steer CFD-based fire through target-driven force coefficients and interactive force-field editing, so a user can sketch a preferred flow direction and have the simulation follow it. Cui et al. [41] fold geometric, image, and temperature targets into a single temperature-field formulation with adaptive control forces, which makes shape and temperature goals the same control primitive. Cui et al. [42] build VR fireworks from hand-drawn sketches using sketch retrieval and Gaussian-curvature point sampling (Figure 10). Leja et al. [13][43] reshape flames into custom forms-letters, for instance - by running gradient optimization over differentiable combustion, and they verified the result against a real stage flame projector (Figure 9). Hladký and Durikovič [44] model flame turbulence with differential equations that can separate, flicker, and re-merge, offering a lightweight middle ground between full CFD and bare particles.

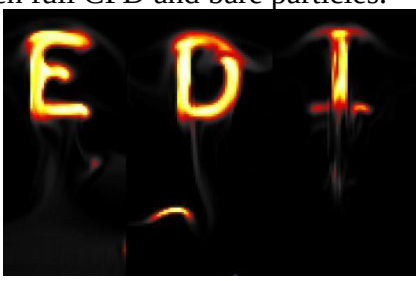


Figure 9. Shaping flames into custom forms with differentiable physics simulation [13].

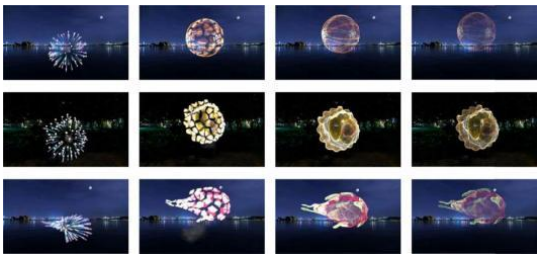


Figure 10. Sketch-based shape-constrained fireworks simulation in VR [42].

5. Neural Style Transfer and Semantic Synthesis

5.1. Transport-Based Neural Style Transfer

Kim et al. [45] propose transport-based neural style transfer, or TNST, which optimizes an incompressible, curl-free potential field so that smoke simulations up to 16 million voxels can be stylized without divergence artifacts. Aurand et al. [46] push the same idea to production-ready efficiency, transferring an artist's chosen style onto volumetric simulations fast enough that it works inside a real film schedule.

5.2. Film Industry Application: Pixar's Elemental

Kanyuk et al. [47] use voxel-based NST to paint an illustrated fire style onto simulated flames, which is how the fire characters of Elemental were made (Figure 11): the simulator still drives the coarse character motion, and the neural pass adds the fine stylistic surface. Hoffman et al. [48] walk through how the character pipeline balances physical fire properties against character performance across shape, color, and motion budgets.

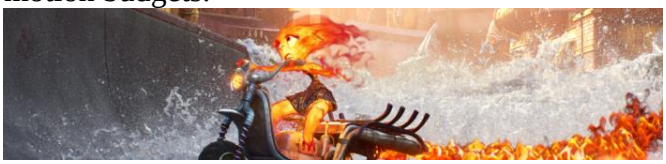


Figure 11. The fire character of Pixar's Elemental produced with volumetric neural style transfer[47].

5.3. Lightweight Stylization

VNCA [49] does away with per-frame optimization entirely, relying on neural-cellular-automata transition rules instead; after a single training pass the style unfolds deterministically over time, and it runs in real time.

6. Neural Reconstruction and Radiance Field Methods

6.1. Neural Radiance Field (NeRF) Methods

Li et al. [6] reconstruct 3D flames by combining an improved NeRF with radiative-transfer modeling, so physical constraints enforce radiative consistency rather than just view consistency. FlameRF [50] builds on tensorial radiance fields (TensoRF) to reconstruct turbulent flames quickly over time, and notably appears in a combustion-science venue.

6.2. 3D Gaussian Splatting (3DGS) Methods

Gaussians on Fire [7] stores lifespan and a linear velocity inside each 3D Gaussian and matches dense optical flow, which lets it reconstruct dynamic flames from as few as three views, with no per-frame registration (Figure 12). GauSmoke [51] guides its Gaussian representation with hybrid physical constraints. FieryGS [52] couples multimodal material reasoning, volumetric combustion simulation, and a unified renderer tightly together so it can synthesize fires that ignite, spread, and extinguish in real scenes (Figure 13). SmokeSeer [53] and SmokeSVD [54] take a different direction, recovering clean scenes and single-view smoke volumes that have been degraded by fire by-products, for use in post-production editing.



Figure 12. High-frequency temporal reconstruction of flames with 3D Gaussian splatting, Gaussians on Fire [7].



Figure 13. In-the-wild fire synthesis with physics-integrated Gaussian splatting, FieryGS [52].

7. AI Video Generation Methods

T2VFire [8] is a training-free text-to-video framework: it injects retrieval-augmented fire knowledge - how fire develops, how smoke is shaped, how different materials burn - into GPT-4o prompts, and comes out with controllable fire videos that respect commonsense behavior. It also stays agnostic to whatever diffusion model sits underneath (Figure 14).



Figure 14. Fire scene video frames generated by T2VFire, a training-free text-to-video framework for open-world fire generation [8].

8. Summary Comparison and Trend Analysis

8.1. Comprehensive Comparison

Table 1 compares the six method categories along technical, engineering, and usability dimensions based on the discussion in Sections 2-7.

Table 1. Comparison of the six method categories along technical, engineering, and usability dimensions.

Method Category	Core Representation	Physical Fidelity	Temporal Consistency	Visual Realism	Computational Cost	Data / Training	Controllability	Primary Scenario (Refs.)
Physics-based methods	Grids; FLIP/APIC; MPM; SPH	High – detailed combustion chemistry and soot formation	High – physics-driven, temporally coherent	High – validated, film-grade visuals	Very high; GPU solvers (Fire-X) reduce cost	None – solved from first principles; no training data	Medium – heat-channel and force masks for art direction	Offline film [3] [14] [22] [25] [21] [4] [23] [9] [5] [26] [27][24]
Particle systems	GPU particles; sprites	Low–Medium – stylized, artist-tuned motion	Medium	Medium	Low – runs in real time on consumer GPUs	None – empirical emission rules	High – direct control over emitters, forces, and sprites	Real-time games; VR [31] [2] [35] [36] [37] [33] [34] [17] [28] [16] [32] [30] [29] [15]
Procedural and data-driven methods	Noise/FFD; precomputed data	Low – stylized, not physics-based	Medium	Medium–High	Low – interactive and real-time capable (e.g., Ignis)	Low – precomputed example data [40]	High – procedural parameters and target shapes	Games; VR; web [38] [41] [42] [10] [1] [44] [43] [13] [12] [39] [40] [11]
Neural style transfer	Simulation + style field	Medium	High	High	Medium–High; lightweight styles run near real time (VNCA)	Medium – pre-trained style networks and reference clips	High – style control by reference example	Film (e.g., Elemental); stylized games [46] [48] [47] [45] [49]
Neural reconstruction	NeRF; 3D Gaussians	Medium	Medium–High	High	Medium; 3DGS enables moderate real-time rendering	High – multi-view video captures or reference renders	High	Film post-production; asset synthesis [53] [54] [6] [50] [7] [52] [51]
AI video generation	Diffusion + RAG	Low	Low	High	Low – offline generation only	High – large-scale pre-trained text-to-video models	Low – prompt-based control	Concept preview [8]

None of this is settled once and for all; each category buys its strengths at the price of a weakness. Physics simulation (Section 2) is still the most dependable route to high fidelity for film, helped out by neural reconstruction (Section 6) and NST (Section 5). On the game side, particle systems (Section 3) and procedural volumes (Section 4) carry most real-time work, and optimization [15][16][17] often decides whether an effect actually ships. Two trade-offs keep coming up. A method that is more physical usually needs more control interfaces, and the more control we add, the more physical consistency we tend to lose. Temporal consistency is nearly free for simulation, but reconstruction pays for it; visual realism, by contrast, is easiest for learned methods and hardest for plain particles.

8.2. Hybrid and Fusion Trends

The biggest shift we noticed in the recent literature is the rise of hybrid and fusion approaches that join physics-based simulation with learning-based methods. The two sides genuinely complement each other: the solver brings physical fidelity, numerical stability, and temporal coherence, while the learned part chips in high-frequency detail, style control, and semantic flexibility that a pure solver could not easily supply.

When we sorted the work from 2006 to 2026, four arrangements kept showing up. In the first, physics is embedded inside a neural representation: FieryGS [52] locks 3D Gaussian fire onto a volumetric combustion solver, and GauSmoke [50] leans on hybrid physical priors to keep the Gaussian reconstruction honest. In the second, data-driven detail is layered on top of a coarse simulation: Sato et al. [40] precompute high-resolution velocity fields and replay them, while Kim and Lee [34] go a step further and learn ember motion from chaotic-advection data rather than merely storing it. The third arrangement uses differentiable physics for direct artistic control, best shown by Leja et al. [13][43], who reshape a flame by running gradient descent against a target image. The fourth lets a large language model act as the reasoning layer: T2VFire [8] feeds fire-domain knowledge into diffusion prompts, and FieryGS reads scene materials straight from language input.

Two things struck us in particular. The physics solver is increasingly treated as a set of constraints rather than as a finished solution: it guarantees stability, conservation, and temporal coherence, and leaves the detail, style, and semantics to the learned component. Just as telling, the direction of information flow between the two sides is now something the designer chooses rather than a pipeline that is fixed in advance. That flexibility is why we expect hybrid solvers to set the agenda in fire simulation research for some time.

8.3. Future Trends

If we had to list the directions the field is moving in, five stand out. The line between real time and high fidelity is blurring, thanks to things like Fire-X's GPU acceleration and Ignis solving at VR frame rates. Reconstruction and generation have joined pure simulation as legitimate ways to obtain data. Controllability is turning into one of the main battlegrounds - whether control comes from a target [41], a sketch [42], a differentiable objective [13], a style example (NST), or a text prompt (T2VFire). Physical constraints are becoming the quality guarantee behind neural methods. And scientific computing is drifting back toward entertainment [50][6], so DNS and LES data are likely to supervise neural fire models. The likely upshot is that next-generation film tools will center on hybrid solvers pairing a physics core with a learned appearance model, and authoring will move away from sliders and toward natural language.

8.4. Conclusion

We set out to survey fire simulation for CG films and games, working through 54 representative papers from 2006 to 2026 and organizing them into six method categories. Where each category lands is fairly clear. Physics-based methods (Section 2) give the strongest fidelity for offline film, but their cost rules out real time. Particle systems (Section 3) and procedural or data-driven methods (Section 4) satisfy real-time demands, though they sacrifice physical accuracy to do it. Neural style transfer (Section 5) supplies controllable artistic expression, and neural reconstruction (Section 6) plus AI video generation (Section 7) sidestep numerical solving altogether, trading it for view capture or a text prompt. Since no single approach wins on every axis, the practical answer is to pick one method- or a hybrid - based on the scenario, the compute budget, and how much control the artist actually needs.

References

[1] Fuller, Alfred R., Hari Krishnan, Karim Mahrous, Bernd Hamann, and Kenneth I. Joy. 2007. "Real-Time Procedural Volumetric Fire." In Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games (I3D '07), 101–108. ACM. https://doi.org/10.1145/1230100.1230119.

[2] Drone, Shannon. 2007. "Real-Time Particle Systems on the GPU in Dynamic Environments." In Advanced Real-Time Rendering in 3D Graphics and Games Course, SIGGRAPH 2007, Chapter 7: 80–96. ACM.

[3] Aguilera, Gerardo, and John Johansson. 2019. "Avengers: Endgame, A New Approach for Combustion Simulations." In Proceedings of SIGGRAPH '19 Talks. ACM. https://doi.org/10.1145/3306307.3328203.

[4] Lesser, Steve, Alexey Stomakhin, Gilles Daviet, Joel Wretborn, John Edholm, Noh-hoon Lee, Eston Schweickart, Xiao Zhai, Sean Flynn, and Andrew Moffat. 2022. "Loki: A Unified Multiphysics Simulation Framework for Production." ACM Transactions on Graphics 41(4), Article 50: 1–20. https://doi.org/10.1145/3528223.3530058.

[5] Nielsen, Michael B., Morten Bojsen-Hansen, Konstantinos Stamatelos, and Robert Bridson. 2022. "Physics-Based Combustion Simulation." ACM Transactions on Graphics 41(5), Article 176: 1–21. https://doi.org/10.1145/3526213.

[6] Li, Yiting, Jun Chang, and Yi Huang. 2024. "3D Flame Reconstruction via Improved NeRF and Radiative Transfer Modeling." Optica Open Preprint. https://doi.org/10.1364/opticaopen.26412871.

[7] Nazarenus, Jakob, Sören Pirk, Reinhard Koch, Dominik Michels, Wojtek Pałubicki, Simin Kou, and Fang-Lue Zhang. 2025. "Gaussians on Fire: High-Frequency Reconstruction of Flames." arXiv: 2511.22459.

[8] Zheng, Hongtao, and Xinyan Huang. 2026. "Photorealistic Fire Scene Video Generation via Multimodal Large Language Model and Pre-Trained Video Diffusion Model." Computational Visual Media 12(3): 841–848. https://doi.org/10.26599/CVM.2025.9450511.

[9] Nielsen, Michael B., Konstantinos Stamatelos, Morten Bojsen-Hansen, Duncan Brinsmead, Yannick Pomerleau, Marcus Nordenstam, and Robert Bridson. 2018. "A Collocated Spatially Adaptive Approach to Smoke Simulation in Bifrost." In Proceedings of SIGGRAPH '18 Talks. ACM. https://doi.org/10.1145/3214745.3214749.

[10] Fraboni, Basile, Tsz Kin Chan, Thibault Vergne, and Jakub Jeziorski. 2023. "Can You See the Heat? A Null-Scattering Approach for Refractive Volume Rendering." In SIGGRAPH '23 Talks. ACM. https://doi.org/10.1145/3587421.3595427.

[11] Shenton, Charlie. 2023. "Ignis: Eulerian Fluid Simulation and Rendering at VR Frame Rates." In SIGGRAPH '23 Talks. ACM. https://doi.org/10.1145/3610542.3626120.

[12] Mizutani, Keisuke, Yoshinori Dobashi, and Tsuyoshi Yamamoto. 2016. "Interactive Control of Fire Simulation Based on Computational Fluid Dynamics." In Proceedings of the 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016), 242–247. https://doi.org/10.5220/0005746902400245.

[13] Leja, Laura, Oskars Teikmanis, and Kārlis Freivalds. 2024. "Shaping Flames with Differentiable Physics Simulations." In Machine Learning and the Physical Sciences Workshop at NeurIPS 2024.

[14] Edholm, John, Alexey Stomakhin, Rahul Deshprabhu, David Caeiro Cebrian, Florian Hu, and Caitlin Pope. 2023. "Fire and Explosions in Avatar: The Way of Water." In SIGGRAPH '23 Talks. ACM. https://doi.org/10.1145/3587421.3595406.

[15] Zhang, Juan. 2020. "Implementation and Optimization of Particle Effects Based on Unreal Engine 4." Journal of Physics: Conference Series 1575: 012187. https://doi.org/10.1088/1742-6596/1575/1/012187.

[16] Shulha, Maksym, Dmytro Matvieiev, Oleksii Nazarov, and Nataliia Nazarova. 2023. "Researched Methods for Simplifying and Optimizing Particles for Portable Gaming Devices." Bionics of Intelligence 1(99): 88–98. https://doi.org/10.30837/bi.2023.1(99).13.

[17] LeBlanc, Jeremy. 2025. "Impact of Optimization on Visual Effects Tools in Unity 6: A Performance and Fidelity Study." Bachelor's Thesis, JAMK University of Applied Sciences.

[18] Guo, Yixuan. 2025. "Fire Simulation in CG: A Survey." In Proceedings of the 2025 International Conference on EMIS, 129–139.

[19] Mahadika, Arya Luthfi, and Ema Utami. 2025. "A Comparative Study of EmberGen and Blender in Fire Explosion Simulations." Jurnal SISFOKOM (Sistem Informasi dan Komputer) 14(02): 216–221. https://doi.org/10.32736/sisfokom.v14i2.2335.

[20] Schlüter, Markus, Tom Kwasnitschka, Armin Berstetter, and Jens Karstens. 2025. "Rendering Large Volume Datasets in Unreal Engine 5: A Survey." arXiv:2504.07485.

[21] Kala, Victoria, Jingyu Chen, David Hyde, Alexey Stomakhin, and Joseph Teran. 2024. "A Thermomechanical Hybrid Incompressible Material Point Method." ACM Transactions on Graphics 43(4): 1–10.

[22] Flynn, Sean, Alexey Stomakhin, Joel Wretborn, and Daniel White. 2024. "Art Directable Underwater Explosion Simulation." In SIGGRAPH Talks '24. ACM. https://doi.org/10.1145/3641233.3664314.

[23] Liu, Daoming, Jonathan Klein, Florian Rist, Wojciech Pałubicki, Sören Pirk, and Dominik Michels. 2025. "FLAMEFORGE: Combustion Simulation of Wooden Structures." Proceedings of the ACM on Computer Graphics and Interactive Techniques 8(4), Article 50: 1–19. https://doi.org/10.1145/3747855.

[24] Wrede, Helge, Anton R. Wagner, Sarker Miraz Mahfuz, Wojtek Pałubicki, Dominik L. Michels, and Sören Pirk. 2025. "Fire-X: Extinguishing Fire with Stoichiometric Heat Release." ACM Transactions on Graphics 44(6): 1–17. https://doi.org/10.1145/3763338.

[25] Hädrich, Torsten, Daniel T. Banuti, Wojtek Pałubicki, and Sören Pirk. 2021. "Fire in Paradise: Mesoscale Simulation of Wildfires." ACM Transactions on Graphics 40(4), Article 163: 1–15. https://doi.org/10.1145/3450626.3459954.

[26] Pirk, Sören, Michał Jarząbek, Torsten Hädrich, Dominik L. Michels, and Wojciech Pałubicki. 2017. "Interactive Wood Combustion for Botanical Tree Models." ACM Transactions on Graphics 36(4), Article 197: 1–12. https://doi.org/10.1145/3130800.3130814.

[27] Tran, Rebekah L. 2025. "Simulating and Designing Pyro FX to Communicate a Hostile and Magical Environment." Undergraduate Honors Thesis, Brigham Young University. https://scholarsarchive.byu.edu/studentpub_uht/445.

[28] Rødal, Samuel, Geir Storli, and Odd Erik Gundersen. 2006. "Physically Based Simulation and Visualization of Fire in Real-Time Using the GPU." In Theory and Practice of Computer Graphics 2006, edited by M. McDerby and L. Lever, 187–194. Eurographics.

[29] Zeler, Wojciech, and Paweł Rohleder. 2016. "Particle Effect System for the Needs of a Modern Video Game Using the GPU." Machine Graphics & Vision 25(1/4): 35–44. https://doi.org/10.22630/MGV.2016.25.1.4.

[30] Yin, Peng, Jie Li, and Junnan Liu. 2019. "Research and Simulate of Real-Time Particle Systems Based on GPU-Acceleration." In 2019 International Conference on Computer Network, Electronic and Automation (ICCNEA), 151–155. IEEE. https://doi.org/10.1109/iccnea.2019.00035.

[31] Amarasinghe, Dhanyu, and Ian Parberry. 2011. "Towards Fast, Believable Real-Time Rendering of Burning Objects in Video Games." In Proceedings of the 6th International Conference on the Foundations of Digital Games (FDG '11), 256–258. ACM. https://doi.org/10.1145/2038266.2038309.

[32] Wei, Wei, and Yanqiong Huang. 2011. "Real-Time Flame Rendering with GPU and CUDA." International Journal of Information Technology and Computer Science 3(1): 40–46. https://doi.org/10.5815/ijitcs.2011.01.06.

[33] Kim, Jong-Hyun, and Jung Lee. 2019. "Fire Sprite Animation Using Fire-Flake Texture and Artificial Motion Blur." IEEE Access 7: 110002–110011. https://doi.org/10.1109/ACCESS.2019.2934163.

[34] Kim, Jong-Hyun, and Jung Lee. 2025. "Numerical Dispersed Flow Simulation of Fire-Flake Particle Dynamics and Its Learning Representation." PeerJ Computer Science 11: e2836. https://doi.org/10.7717/peerj-cs.2836.

[35] Fei, Jingjing. 2022. "Real-Time Simulation Method of Flame Animation Based on the Deep Stripping Algorithm and Texture Mapping." Journal of Sensors 2022, Article ID 2580575. Hindawi. https://doi.org/10.1155/2022/2580575.

[36] He, Yiyang. 2017. "A Physically Based Pipeline for Real-Time Simulation and Rendering of Realistic Fire and Smoke." Master's Thesis, KTH Royal Institute of Technology.

[37] Hoikkala, Ville. 2016. "Fire Simulation for a 3D Character with Particles and Motion Capture Data in Blender." Bachelor's Thesis, Tampere University of Applied Sciences.

[38] Bannister, Jordan J., and Derek Nowrouzezahrai. 2025. "Learnable Fractal Flames: A Differentiable Approach to Image-Guided Fractal Art Synthesis." arXiv:2406.09328.

[39] Rašeta, Marta, Žiga Lesar, and Ciril Bohak. 2025. "Real-Time Volumetric Explosion Special Effects for Games." In Proceedings of the 33rd Spring Conference on Computer Graphics (WSCG). https://doi.org/10.24132/SSRN.3799902.

[40] Sato, Syuhei, Takuya Morita, Yoshinori Dobashi, and Tsuyoshi Yamamoto. 2012. "A Data-Driven Approach for Synthesizing High-Resolution Animation of Fire." In Proceedings of DigiPro '12, 37–44. ACM. https://doi.org/10.1145/2370919.2370926.

[41] Cui, Qiaodong, Zhaohui Wu, Chang Xing, Zhong Zhou, and Wei Wu. 2015. "Target Temperature Driven Dynamic Flame Animation." In Eurographics 2015 Short Papers, 45–48. Eurographics Association. https://doi.org/10.2312/egsh.20151011.

[42] Cui, Xiaoyu, Ruifan Cai, Xiangjun Tang, Zhigang Deng, and Xiaogang Jin. 2020. "Sketch-Based Shape-Constrained Fireworks Simulation in Head-Mounted Virtual Reality." Computer Animation and Virtual Worlds 31(2): e1920. https://doi.org/10.1002/cav.1920.

[43] Leja, Laura, and Kārlis Freivalds. 2025. "Controlling Shape of Flame with Differentiable Physics." In Engineering for Rural Development, 666–672. Jelgava, Latvia.

[44] Hladký, Jozef, and Roman Durikovič. 2018. "Fire Simulation in 3D Computer Animation with Turbulence Dynamics Including Fire Separation and Profile Modeling." International Journal of Networking and Computing 8(2): 186–204.

[45] Kim, Byungsoo, Vinicius C. Azevedo, Markus Gross, and Barbara Solenthaler. 2019. "Transport-Based Neural Style Transfer for Smoke Simulations." ACM Transactions on Graphics 38(6), Article 188. https://doi.org/10.1145/3355089.3356560.

[46] Aurand, Joshua, Raphael Ortiz, Silvia Nauer, and Vinicius C. Azevedo. 2022. "Efficient Neural Style Transfer for Volumetric Simulations." ACM Transactions on Graphics 41(6), Article 257. https://doi.org/10.1145/3550454.3555517.

[47] Kanyuk, Paul, Vinicius C. Azevedo, Raphael Ortiz, and Jingwei Tang. 2023. "Singed Silhouettes and Feed Forward Flames: Volumetric Neural Style Transfer for Expressive Fire Simulation." In SIGGRAPH '23 Talks. ACM. https://doi.org/10.1145/3587421.3595467.

[48] Hoffman, Jonathan, Te Hu, Paul Kanyuk, Stephen Marshall, George Nguyen, Hope Schroers, and Patrick Witting. 2023. "Creating Elemental Characters: From Sparks to Fire." In SIGGRAPH '23 Talks. ACM. https://doi.org/10.1145/3587421.3595467.

[49] Wang, Dongqing, Ehsan Pajouheshgar, Yitao Xu, Tong Zhang, and Sabine Süsstrunk. 2025. "Volumetric Temporal Texture for Smoke Stylization Using Dynamic Radiance Fields." arXiv:2502.09631.

[50] Liu, Shiyu, Haiou Wang, Yuqi Huang, Kun Luo, and Jianren Fan. 2026. "FlameRF: A Fast Time-Resolved Reconstruction Technique for Turbulent Flame Using Tensorial Radiance Fields." Energy and AI. https://doi.org/10.1016/j.egyai.2026.100758.

[51] Zhang, Wenran, Yuxiang Cai, Letian Huang, Dongwei Ye, Jie Guo, and Bo Ren. 2026. "GauSmoke: Hybrid Physics-Optical Gaussian Splatting for Sparse Smoke Reconstruction." ACM Transactions on Graphics. https://doi.org/10.1145/3799902.3811148.

[52] Shen, Qianfan, Ningxiao Tao, Qiyu Dai, Tianle Chen, Minghan Qin, Yongjie Zhang, Mengyu Chu, Wenzheng Chen, and Baoquan Chen. 2026. "FieryGS: In-the-Wild Fire Synthesis with Physics-Integrated Gaussian Splatting." In Proceedings of ICLR 2026. arXiv:2605.00177.

[53] Jain, Neham, Andrew Jong, Sebastian Scherer, and Ioannis Gkioulekas. 2026. "SmokeSeer: 3D Gaussian Splatting for Smoke Removal and Scene Reconstruction." In Proceedings of the International Conference on 3D Vision (3DV).

[54] Li, Chen, Shanshan Dong, Sheng Qiu, Jianmin Han, Yibo Zhao, Zan Gao, Taku Komura, and Kemeng Huang. 2025. "SmokeSVD: Smoke Reconstruction from a Single View via Progressive Novel View Synthesis and Refinement with Diffusion Models." arXiv:2507.12156.