HGCalVis: Web-Based Importance-Ordered Visualization of CERN's HGCal Particle Showers

Waad F. Alsheri
KAUST
Thuwal, Saudi Arabia
waad.alshehri@kaust.edu.sa

Matevž Tadel
University of California
San Diego, USA
CERN
Geneva, Swaziland
matevz.tadel@cern.ch

Alja M. Tadel
University of California
San Diego, USA
CERN
Geneva, Switzerland
alja.mrak.tadel@cern.ch

Marco Rovere
CERN
Geneva, Swaziland
marco.rovere@cern.ch

Markus Hadwiger
KAUST
Thuwal, Saudi Arabia
markus.hadwiger@kaust.edu.sa

Alberto Jaspe-Villanueva
KAUST
Thuwal, Saudi Arabia
alberto.jaspe@kaust.edu.sa

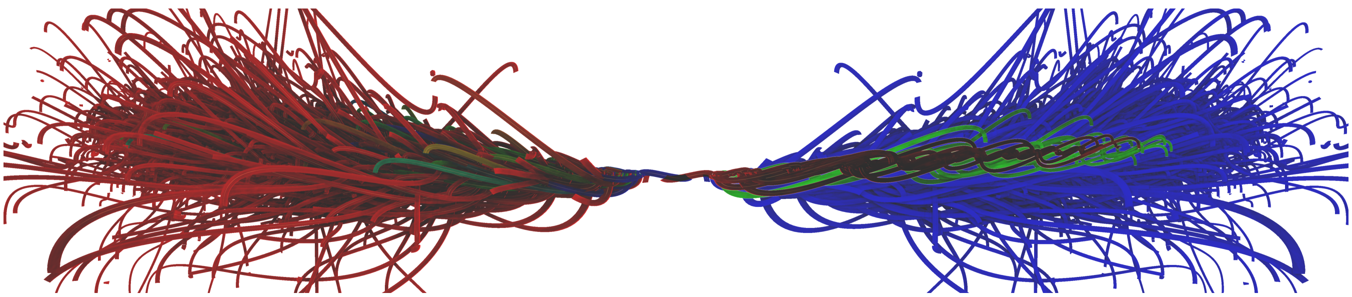

Figure 1: Two renderings of a simulated electromagnetic shower in CERN's HGCal sub-detector. Left: cylinder-shaded lines with screen-space ambient occlusion, where high- and low-energy trajectories are visually indistinguishable and the most significant tracks are occluded in the dense core. Right: the same event rendered with our Importance-Ordered Rendering (IOR), which assigns trajectories to energy-based clusters and composites them so that high-energy tracks (red) remain visible on top of the medium- (green) and low-energy (blue) context.

Abstract

The next major upgrade of the Compact Muon Solenoid (CMS) experiment at CERN's Large Hadron Collider will introduce the High Granularity Calorimeter (HGCal), a new endcap sub-detector designed to cope with the order-of-magnitude increase in collision rates expected from the High-Luminosity LHC. Validating its design relies on Monte Carlo simulations that produce highly cluttered particle shower events: cascades of thousands of trajectories confined within a small detector volume, where physically significant tracks are easily occluded by lower-energy ones. We present a web-based, interactive 3D visualization system for exploring these simulated showers. The system combines a linked 2D hierarchical schematic view, cylinder-shaded lines with screen-space ambient occlusion, multi-attribute filtering, two complementary animation modes and, as our main rendering contribution, an Importance-Ordered Rendering (IOR) algorithm. IOR partitions trajectories into energy-based clusters, renders each into its own G-buffer in a separate pass, and composites them using Gaussian-filtered importance masks so that high-energy tracks remain visible without losing the spatial context provided by low-energy tracks. The system targets restricted graphics environments such as WebGL, where most existing dense-line techniques cannot be used directly. We report a performance evaluation on four production HGCal datasets (1.9k to 22.6k tracks) on both workstation and laptop hardware, a preliminary within-subjects user study (N=19) that isolates the contribution of each main visual feature, and qualitative feedback from CERN collaborators. In the study, importance-ordered rendering increased the accuracy of identifying the highest-energy trajectory relative to unordered rendering, while energy color-coding alone did not, and the linked schematic view increased recovery of a track's daughter set.

CCS Concepts

• **Human-centered computing → Scientific visualization**; *Web-based interaction*; • **Computing methodologies → Rendering**.

Keywords

Web-based visualization, WebGL, dense line rendering, importance-driven visualization, particle physics, high-energy physics, scientific visualization, user study

Web3D '26, Doha, Qatar

ACM ISBN 979-8-4007-2810-5/26/10
https://doi.org/10.1145/3822516.3834872

ACM Reference Format:
Waad F. Alsheri, Matevž Tadel, Alja M. Tadel, Marco Rovere, Markus Hadwiger, and Alberto Jaspe-Villanueva. 2026. HGCalVis: Web-Based Importance-Ordered Visualization of CERN's HGCal Particle Showers. In *The 31th International Conference on 3D Web Technology (Web3D '26), October 13–15, 2026, Doha, Qatar*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3822516.3834872

1 Introduction

The Large Hadron Collider (LHC) at CERN is the world's largest particle accelerator, designed to study the fundamental constituents of matter by colliding proton beams at energies up to 13.6 TeV [Evans and Bryant 2008]. Among the four major LHC experiments, the Compact Muon Solenoid (CMS) [CMS Collaboration 2008] is undergoing a deep upgrade in preparation for the High-Luminosity LHC (HL-LHC) [Apollinari et al. 2017], which will increase the instantaneous luminosity, and therefore the collision rate, by roughly a factor of ten starting around 2029. To withstand the resulting radiation and pile-up, the existing CMS endcap calorimeter will be replaced by the High Granularity Calorimeter (HGCal) [CMS Collaboration 2017], a sampling calorimeter with about six million silicon and scintillator channels that resolves showers in three dimensions with unprecedented spatial and timing precision.

Because HGCal is still under construction, its physics performance and reconstruction algorithms are validated almost exclusively through Monte Carlo simulations running on top of the CMS software stack. A single simulated event consists of a *particle shower*: a hierarchical cascade in which one incoming particle interacts with the detector material and produces secondary particles, each of which may radiate further descendants. A representative HGCal shower contains between two thousand and more than twenty thousand simulated trajectories packed into a detector endcap roughly one cubic meter in volume. Visualizing such an event is a hard, density-bound rendering problem: the trajectories that matter physically (high-energy primary fragments) make up only a small fraction of all tracks, yet they are easily lost behind a forest of soft, low-energy secondaries.

CERN simulation experts, calorimeter designers, and reconstruction algorithm developers all need to inspect individual simulated events to: (i) understand the structure and depth profile of representative showers, (ii) identify outliers and reconstruction failures, and (iii) iterate on the design of the calorimeter and of the downstream reconstruction algorithms [Beaudette et al. 2023]. The de-facto tool used inside CMS, Fireworks, is a desktop C++ application coupled tightly to the ROOT framework [Brun et al. 2020]; it is precise but heavy to install, hard to share with non-CMS collaborators (including the calorimeter designers themselves), and offers little support for the perceptual challenges introduced by HGCal-scale event sizes.

A large multi-institutional collaboration like CMS spans thousands of physicists, calorimeter designers, and engineers distributed across more than 50 countries; many of the people who need to look at HGCal simulations are not part of the CMS software cluster and have neither the privileges nor the appetite to install the full ROOT/Fireworks stack. A pure-browser client based on open web standards (WebGL 2.0 [Khronos Group 2022] for rendering, plain JavaScript, JSON for data exchange) requires no installation, runs on any reasonably modern desktop or laptop, can be shared by URL, and integrates naturally with the existing CMS data services. This kind of open-standards, zero-install access has, in our experience, been the single biggest unlock for collaboration with external partners. The cost is that WebGL is a constrained graphics target: it does not expose atomic operations, generic shader storage buffers, compute shaders, or programmable blending, and lacks robust support for per-pixel linked lists or A-buffer-style structures. Most modern dense-line techniques, such as per-pixel linked-list opacity optimization [Günther et al. 2013], voxel-based GPU ray casting [Kanzler et al. 2018], multi-layer alpha blending [Salvi and Vaidyanathan 2014], moment-based order-independent transparency [Münstermann et al. 2018], cannot be ported directly. Any solution for the browser must stay within a multi-pass rasterization model using only what WebGL 2.0 actually exposes (multiple render targets, texture sampling, blending, mipmaps).

Contributions. We present HGCalVis, a complete, web-based 3D visualization system designed in close collaboration with the CMS HGCal simulation, reconstruction, and event-display teams. Our contributions are: (i) a domain-driven task analysis from a year-long collaboration with CMS and ROOT-EVE (Section 3); (ii) a coordinated browser-only interface combining a 3D shower view, a 2D hierarchical schematic view, multi-attribute filtering, and two animation modes (Section 5); (iii) an *Importance-Ordered Rendering* (IOR) algorithm for dense 3D line sets under the WebGL 2.0 capability envelope, which partitions trajectories into importance clusters and composites them through Gaussian-filtered masks (Section 5.3); (iv) a performance evaluation on four production CMS datasets (~2k–~23k tracks) on both workstation and integrated-GPU laptop hardware (Section 6); (v) a preliminary within-subjects user study (N=19, Section 7) that isolates the contribution of each main visual feature on domain-motivated tasks; and (vi) qualitative feedback and use cases from senior CERN collaborators (Section 8). HGCalVis is built on top of the RenderCore [Bohak et al. 2023] WebGL engine, the same engine used by ROOT-EVE, which paves the way for future integration into the CMS data analysis stack. A live, interactive demo and the full source code will be released publicly upon acceptance; both are anonymized for review.

2 Related Work

2.1 Visualization of Particle Showers and Detector Events

Several systems target the visualization of particle tracks from collider or air-shower experiments. TrackVis [Wei et al. 2001] uses volume rendering of voxelized track densities for the RHIC experiment, with filtering, clustering and a fisheye view. Chiriţoi et al. [Chiriţoi et al. 2016] render extensive air showers simulated with CORSIKA via Blender, optimizing for visual aesthetics rather than interactivity. Belle2VR [Duer et al. 2018] is a Unity-based VR experience that immerses users in SuperKEKB collisions for educational purposes. More recently, Eschbach et al. [Eschbach et al. 2022] introduced a semi-automatic particle-tracking and visualization pipeline for particle detector data, focused on supporting track reconstruction in a desktop setting.

Inside CMS, the established event display Fireworks is integrated with the ROOT framework [Brun et al. 2020] and is the natural reference point for our work. None of these tools is web-based; none specifically addresses the depth and occlusion problems that arise when a single event contains tens of thousands of finely spaced trajectories such as those produced by HGCal. We position HGCalVis as a complement to Fireworks: it targets the same data but trades some precision for accessibility, perceptual clarity in dense events, and easier collaboration with external partners.

2.2 Rendering of Dense 3D Line Sets

Dense 3D line fields are a recurring topic in scientific visualization, with three broad families of techniques that we briefly review in turn.

Transparency-based methods. A natural way to relieve occlusion is to make less important lines transparent. Correct order-independent transparency (OIT) requires either back-to-front sorting or per-pixel structures, realized by the A-buffer [Carpenter 1984], the k-buffer [Bavoil et al. 2007], depth peeling [Everitt 2001], or per-pixel linked lists [Yang et al. 2010]. Günther et al. [Günther et al. 2013] use linked lists to optimize opacity as a bounded-variable least-squares problem on dense line fields. Approximate methods (adaptive transparency [Salvi et al. 2011], stochastic transparency [Enderton et al. 2010], weighted-blended OIT [McGuire and Bavoil 2013], multi-layer alpha blending [Salvi and Vaidyanathan 2014], moment-based OIT [Münstermann et al. 2018], Fourier opacity optimization [Rojo et al. 2019]) trade exact visibility for performance; Kern et al. [Kern et al. 2020] survey them on transparent line sets. All of these either require atomic counters / SSBOs / image load–store or large fragment-side state, none of which is available in WebGL 2.0.

Clustering and density control. A second family aggregates lines into clusters to reduce visual mass. Kanzler et al. define a line cluster hierarchy [Kanzler et al. 2016] and extend it to an interactive exploration tool [Kanzler and Westermann 2018]. Kern and Westermann [Kern and Westermann 2019] compare clustering strategies for ensembles of jet-stream core lines. These methods deliver significant clutter reduction, but require an offline geometric preprocessing step that conflicts with the interactive, per-event nature of HGCal exploration, where users routinely load new simulation outputs and adjust the importance metric on the fly. Kanzler et al.'s voxel-based representation [Kanzler et al. 2018] also relies on a GPU ray-caster.

Illustrative and importance-driven rendering. A third family is rooted in non-photorealistic and illustrative visualization. Everts et al. [Everts et al. 2009] use depth-dependent halos to enhance the legibility of dense lines. Eichelbaum et al. [Eichelbaum et al. 2012] extend ambient occlusion to dense line bundles in LineAO. Mallo et al. [Mallo et al. 2005] revisit the classical illuminated-lines model. Stoll et al. [Stoll et al. 2005], Svakhine et al. [Svakhine et al. 2005], and Grabli et al. [Grabli et al. 2010] explore stylized line drawings and procedural strokes. Importance-driven volume visualization, as formalized by Viola et al. [Viola et al. 2004, 2005] and extended by ClearView [Krüger et al. 2006] and others, articulates a view-dependent focus+context model. Rheingans and Ebert [Rheingans and Ebert 2001] and Bruckner and Gröller [Bruckner and Gröller 2007] combine NPR cues with volumetric data to highlight features. The state-of-the-art report by Brambilla et al. [Brambilla et al. 2012] surveys the broader area of illustrative flow visualization.

Our IOR algorithm is closest in spirit to Viola et al.'s importance-driven framework, but lives in object space rather than volume space, operates on dense line sets, and is restricted to a multi-pass rasterization implementation that fits WebGL 2.0.

2.3 Web-Based Scientific Visualization

Mwalongo et al. [Mwalongo et al. 2016] survey the design space of web-based scientific visualization. The 3D-web community has produced web-deployed scientific visualization systems for domains as diverse as hardware-accelerated multivariate scatter plots [Wagner et al. 2022], immersive ultrastructural neuroscience [Cali and Agus 2024], multi-resolution exploration of billion-star catalogs [Alghamdi et al. 2025], and panoramic indoor exploration [Pintore et al. 2023, 2024]; these share with our work the design pressure of running on open standards and reaching users without an installation step. WebGL 2.0 remains a strict subset of OpenGL 4.x: no compute shaders, no atomic operations on storage buffers, limited binding counts, no programmable blending [Khronos Group 2022], so ports of advanced techniques are not straightforward. WebGPU promises to relax several of these constraints, and the RenderCore engine [Bohak et al. 2023] we build upon is already adopting it (Section 9).

3 Domain Background and Research Design

3.1 HGCal and the Simulation Pipeline

The High Granularity Calorimeter [CMS Collaboration 2017] replaces the existing endcap calorimeters of CMS for the HL-LHC era. It comprises roughly 50 sampling layers of silicon sensors and plastic scintillator tiles, divided into an electromagnetic and a hadronic section, providing precise three-dimensional reconstruction of energy depositions. When a primary particle (e.g., a photon, electron, or hadron) enters the calorimeter, it interacts with the detector material to produce a cascade of secondary particles, a *shower*. Electromagnetic showers (initiated by photons or electrons) are governed by bremsstrahlung and pair production; hadronic showers, initiated by quark-composed particles, are governed by both electromagnetic and nuclear processes and are therefore broader, deeper, and substantially more complex to model. The HGCal simulation pipeline propagates each primary particle through the detector geometry, recording the trajectory of every produced particle as a sequence of segments characterized by position, momentum, time, particle type (encoded by a PDG identifier [Garren et al. 2000]), and deposited energy at both endpoints. The resulting set of simulated trajectories and energy deposits is then processed by reconstruction algorithms that build sensor-level *layer clusters* and, in later stages, *tracksters* [Beaudette et al. 2023]: connected groups of layer clusters that approximate the trajectory of one primary particle through the calorimeter. The Trackster-based reconstruction is the centerpiece of the CMS TICL framework currently under development.

3.2 Analysis Questions

We worked with different stakeholders, from engineers in charge of the visualization framework of ROOT/CMS, to the scientist developing the HGCal simulation software, PhD students working on the trackster reconstruction algorithms, several CMS calorimeter physicists involved in design validation, and external collaborators from two partner institutions with no prior exposure to CMS internals. From semi-structured interviews and weekly meetings we compiled a list of recurring analysis questions:

- **Q1.** What does the high-energy "spine" of this shower look like? How many primary high-energy trajectories carry most of the deposited energy?
- **Q2.** How deep does the shower penetrate before depositing the bulk of its energy, and how does that depend on the incident particle type?
- **Q3.** For a given parent track, which descendants account for most of the secondary energy?
- **Q4.** Are there outliers (tracks that travel unusually far, or fragments that deposit energy in unexpected detector regions)?
- **Q5.** For trackster reconstruction debugging: do clusters reconstructed by TICL correspond to the same physical particle, or do they wrongly merge tracks from different parents?

3.3 Tasks and Design Requirements

Following design study methodology [Sedlmair et al. 2012], we translated these questions into a set of design requirements (R_1–R_6), and derived three evaluation tasks (T_1–T_3, Table 3), each isolating one of the system's main visual features, used in the study of Section 7. Each task is a controlled proxy for one or more analysis questions: T_2 (IOR) for Q1, T_3 (schematic view and filtering) for Q3 and the hierarchy navigation underlying Q5, and T_1 (depth cues) for the spatial judgments underlying Q2 and Q4; Q4 and Q5 are additionally examined through the expert use cases of Section 8. The design requirements are:

- **R1: Browser-only.** The tool must run unmodified in any modern desktop or laptop browser, with no installation. This is the primary accessibility constraint that motivates the WebGL target.
- **R2: Interactive frame rates (≥30 FPS) on consumer hardware.** For all features active on the largest production datasets (~23k tracks).
- **R3: Importance-driven emphasis.** The most physically meaningful tracks must remain visible regardless of camera angle and density, with low-energy context preserved.
- **R4: Multi-attribute exploratory filtering.** On energy, PDG type, time, and tree branch, all interactively combinable.
- **R5: Hierarchy navigation.** The parent–child structure of the shower must be navigable separately from the 3D view and linked to it.
- **R6: Algorithm flexibility.** Importance metrics must be replaceable (energy is the default, but the same machinery should support PDG-derived or geometry-derived metrics for future use cases).

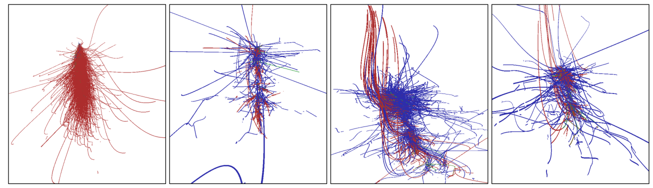

Figure 2: The four production HGCal datasets used in this paper. (a) Electromagnetic shower from a 200 GeV photon (12,507 tracks). (b) Hadronic shower from a 100 GeV kaon (3,751 tracks). (c) Hadronic shower from a 600 GeV kaon (22,604 tracks). (d) Hadronic shower from a 200 GeV proton (1,983 tracks).

Table 1: Production HGCal simulation datasets used in this paper.

Name	Type	Primary	Energy	#Tracks
Electromagnetic	EM	γ	200 GeV	12,507
Hadronic-1	Hadronic	$K^{\pm}$	100 GeV	3,751
Hadronic-2	Hadronic	$K^{\pm}$	600 GeV	22,604
Hadronic-3	Hadronic	p	200 GeV	1,983

4 Particle Shower Data

Each simulation event is delivered as a hierarchical tree whose nodes are individual particle trajectories. A trajectory is described by:

- **Endpoint state vectors.** Position $\vec{X}$ and momentum $\vec{P}$ at both the start ($t = 0$) and end ($t = 1$) of the trajectory, together with the elapsed simulation time and the kinetic energy at each endpoint.
- **Particle identity.** Mass, charge, and PDG ID [Garren et al. 2000].
- **Hierarchy.** A pointer to the parent trajectory and a contiguous range of daughter indices, together with bookkeeping counts of daughters skipped due to the simulation's energy tracking cut.

Between endpoints, the actual particle path is reconstructed analytically. For charged particles in CMS's ~3.8 T solenoidal magnetic field, the Lorentz force produces a helical motion that we approximate with a cubic Hermite polynomial parameterized by the two endpoint positions and momenta. The resulting curve is then sampled into a small number of line segments (typically 100, see Section 6). Each segment is GPU-rendered as a tangent-aligned quadrilateral with cylinder-emulating per-fragment normals (Section 5.2). Table 1 summarizes the four datasets we use throughout the paper, all shown in Figure 2. They were selected by our CMS collaborators as representative of the HGCal design study: an electromagnetic baseline (clean, narrow shower) and three hadronic showers of increasing complexity. The largest, *Hadronic-2*, exceeds 22,000 individual trajectories and exemplifies the density problem that motivated this work.

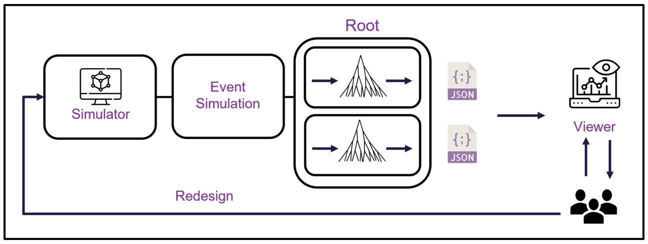

Figure 3: System architecture. HGCal simulation events are stored in ROOT files, exported to JSON, and consumed by the WebGL client running in the user's browser. The client renders the shower, the schematic tree view, and the interactive controls.

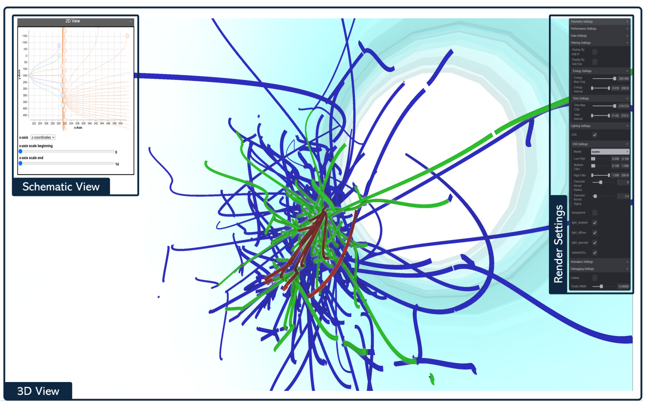

Figure 4: HGCalVis interface. The central 3D shower view (here, an electromagnetic shower) is flanked by the hierarchical schematic view on the left and the settings panel on the right. The two views are linked: interacting with one updates the other.

5 The HGCalVis Visualization System

HGCalVis is structured as a server–client web application (Figure 3). The server, written in C++ and embedded in ROOT, owns the simulation data and exposes events on demand; for the experiments reported in this paper, the server side is replaced by a thin C program that converts ROOT files to JSON for prototype-friendly iteration. The client, written in JavaScript and running entirely in the browser, performs all rendering and interaction. Rendering is built on top of the RenderCore engine [Bohak et al. 2023], the same WebGL 2.0-based engine used by ROOT-EVE; this commonality was an explicit design goal so that HGCalVis can later be folded into the official CMS event display pipeline.

The user interface (Figure 4) is organized in three coordinated regions: the main 3D scene at the center, the hierarchical schematic view on the left, and a collapsible settings panel on the right that exposes data loading, rendering toggles, filters, animation controls, and the IOR configuration. All views are linked: filtering or expanding a node in the schematic view immediately re-renders the 3D scene, and vice versa.

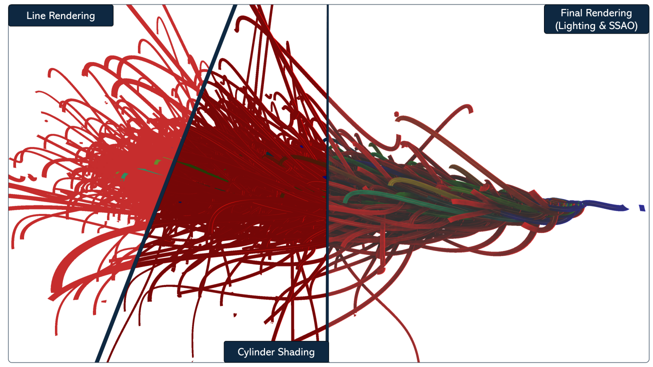

Figure 5: Effect of progressively activating the visual features. Left: plain colored lines. Central: cylinder-emulating per-fragment normals. Right: addition of screen-space ambient occlusion (SSAO). Depth and shape become substantially easier to read as features accumulate.

5.1 Hierarchical Schematic View

The shower is a tree, and the 2D schematic view (Figure 4, left) makes that structure first-class. It is specified declaratively in Vega-Lite [Satyanarayan et al. 2017] (which keeps the implementation small and reusable). Each node represents a trajectory; edges encode parent–child relationships. Nodes are positioned along the horizontal axis either by their creation time or by their longitudinal coordinate (z) along the beam axis, a toggle that addresses two distinct physics questions: shower *development* (time) versus shower *depth profile* (z). On hover, a tooltip exposes the trajectory's ID, the number of direct daughters, and the size of its sub-tree, and the path from the root to the hovered node is highlighted.

The schematic view directly satisfies design requirement R5: it provides a structural map for the otherwise overwhelming 3D view, and it doubles as a filtering control. Clicking on a node *restricts the 3D view to the corresponding sub-tree*, which is the single most-used interaction in our expert sessions.

5.2 Visual Features

Cylinder-shaded lines. Native WebGL line primitives are limited to one-pixel widths on most desktop GPUs and offer no per-fragment shading control. We therefore expand each curve into a tangent-aligned quadrilateral strip in the geometry stage. To make these flat quads look like three-dimensional tubes without the geometry cost of actual cylinders (an essential trick at our scales), each fragment rotates its surface normal around the segment tangent by an angle proportional to its signed distance from the quad's centerline:

$$\vec{N}_{\mathrm{rot}} = \vec{N} \cos\theta + \vec{B} \sin\theta,$$

where $\vec{N}$ and $\vec{B}$ are the segment's normal and binormal in view space. The rotated normal is then fed into a standard Phong illumination model [Phong 1975]. The technique is closely related to the illuminated-lines model of Mallo et al. [Mallo et al. 2005], but is fully expressed in a fragment shader and incurs no extra geometry.

Screen-space ambient occlusion (SSAO). To strengthen the perception of nearby occluders (a critical depth cue in dense bundles), we add an SSAO pass [Bavoil and Sainz 2008; Mittring 2007] applied to the linearized depth and normal G-buffer textures produced by the line shader. Unlike LineAO [Eichelbaum et al. 2012], our SSAO operates on the standard screen-space geometry buffers and therefore keeps the pipeline compatible with WebGL. Figure 5 shows the progressive effect of cylinder shading plus SSAO on a representative shower.

Multi-attribute filtering. HGCalVis supports four filtering dimensions that can be combined freely: sub-tree (set from the schematic view), PDG ID (multi-select), time interval (slider), and energy interval (slider). All filters are evaluated per-trajectory in the vertex shader through a uniform buffer of attribute ranges and a per-vertex attribute mask, which keeps the cost negligible compared to the rasterization pass. This satisfies R4. Filtering compounds with IOR: it is common in our user sessions to first restrict the shower to a sub-tree, then enable IOR to inspect the energy partition within that subset. At current event sizes this in-shader evaluation is sufficient; for substantially larger collections, dedicated multivariate range-query structures [Ageeli et al. 2023] would be the natural way to keep filtering interactive.

Animation modes. Two animation modes complement the static rendering. *Dashed-line animation* renders each track as a moving dashed pattern (with user-controlled speed, dash length, and gap length) to convey local direction of motion, helpful for densely entangled crossings. *Time-based animation* sweeps a temporal window across the shower's simulated time, revealing how the shower develops layer by layer. Both modes are retained because expert sessions showed they answer complementary questions.

5.3 Importance-Ordered Rendering

IOR addresses design requirement R3 within the WebGL 2.0 capability envelope (no atomics, no shader storage buffers, no compute, no programmable blending). The full pipeline is summarized in Figure 6.

Inputs. A line set $\mathcal{L} = \{l_i\}$, a per-track scalar importance $w_i \in \mathbb{R}_{\geq 0}$ (the default is deposited energy, but any user-supplied attribute works), and $K-1$ user-adjustable energy boundaries $E_1 < E_2 < \ldots$ partitioning the tracks into K ordered clusters $C_1, \ldots, C_K$ of increasing importance. We use $K = 3$ throughout this paper (low, medium, high).

Pass 1: Per-cluster G-buffers. Each cluster C_k is rasterized to its own multi-target framebuffer containing five textures: world-space position P_k, view-space normal N_k, linear depth D_k, shaded color L_k (the output of the cylinder-shading + Phong + SSAO pipeline of Section 5.2, computed from D_k and N_k), and a binary presence mask M_k (1 where the cluster wrote a fragment, 0 elsewhere). Each pass uses its own depth buffer, so within a cluster all standard hidden-surface elimination is preserved. *Across* clusters, however, the depth tests are independent.

Pass 2: Mask Gaussian blur. A two-pass separable Gaussian filter with user-adjustable kernel radius r and standard deviation σ is applied to each M_k to produce a soft, anti-aliased blurred mask $\widetilde{M}_k \in [0, 1]$. The blur is what produces the halo effect that smoothly fades context tracks under more important ones; without it, the boundaries between clusters appear as hard cutouts.

Pass 3: Cluster compositing. For each cluster from lowest to highest importance, we attenuate its color L_k by the union of all higher-importance blurred masks:

$$L'_k = L_k \cdot \prod_{j>k} (1 - \widetilde{M}_j).$$

The resulting L'_k are accumulated front-to-back according to the depth D_k, and finally the unattenuated highest-importance cluster L_K is overlaid. The result (Figure 1, right) is an image in which the highest-energy tracks dominate, but where the lower-energy context (compressed into a soft, depth-aware halo) is still legible.

Color encoding. By default we apply a red–green–blue colormap for high/medium/low clusters (Figure 1), but the system optionally supports a warm-to-cool depth-blended encoding inspired by Rheingans and Ebert [Rheingans and Ebert 2001] for users who prefer not to overload the importance channel with hue.

5.4 WebGL 2.0 Constraints

Several constraints of WebGL 2.0 directly shape IOR. Without atomics or SSBOs, per-pixel linked lists and A-buffer-like structures [Günther et al. 2013; Yang et al. 2010] are unavailable, so opacity optimization is out of reach; IOR uses only multiple render targets and texture sampling. Without compute shaders, the Gaussian blur is realized as two fragment-shader passes over a fullscreen quad, more expensive than a compute implementation but adequate at our resolutions. Without programmable blending, cluster compositing is performed in a dedicated fragment shader. To stay within the WebGL minimum of four color attachments, we pack mask and depth into a shared multi-channel texture per cluster. Finally, native `gl.LINES` primitives are 1 px wide on most desktop drivers, which mandates expanding lines into quads independently of IOR. Because all data is kept in regular 2D textures and only standard rasterization, blending, and texture sampling are used, IOR ports trivially to any WebGL 2.0 implementation. The trade-off is that IOR is a heuristic in the spirit of multi-layer alpha blending [Salvi and Vaidyanathan 2014]: depth ordering between clusters is approximated rather than exact, and the Gaussian blur compresses fine geometric detail of the context tracks. We discuss these limitations in Section 9. The same constraints recur whenever any of the techniques in Section 2 is ported to WebGL.

6 Performance Evaluation

We evaluated the performance of HGCalVis on two configurations that bracket the realistic accessibility envelope: a development workstation with a discrete NVIDIA RTX 3050 and a thin-and-light laptop with an integrated Intel Iris Xe GPU. Both run Chrome 129 on Windows 11 at 1920×1080 with default WebGL settings. The line-sampling parameter, which controls how many quadrilateral segments are emitted per curve, is fixed at 100 (the sweet spot between visual smoothness and frame rate, identified in preliminary testing; lower values produce visible kinking, higher values cost performance without visible benefit). Frame rates are reported as medians over five seconds of continuous orbiting after a five-second

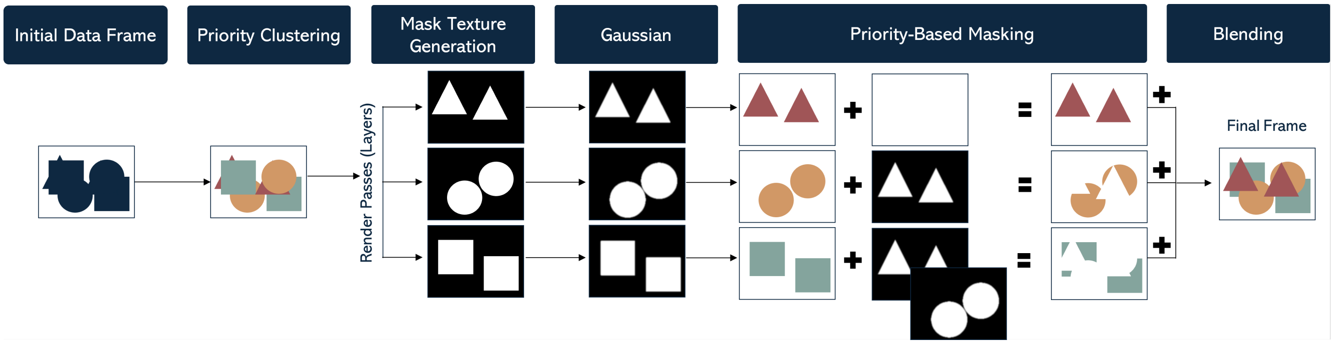

Figure 6: The Importance-Ordered Rendering algorithm. (1) Tracks are partitioned into K energy-based clusters (we use $K = 3$ throughout). (2) Each cluster is rasterized into its own G-buffer (position, normal, depth, color, and a presence mask) in an independent render pass. (3) The masks are blurred by a separable two-pass Gaussian filter. (4) From low importance to high, each cluster's G-buffer is attenuated by the union of higher-importance blurred masks before being composited. The final image preserves a soft halo of low-energy context while keeping high-energy tracks visually dominant.

Table 2: Rendering performance of HGCalVis on two hardware configurations. "FPS off" is the median frame rate with cylinder shading, SSAO and filtering active but with IOR disabled; "FPS on" is the same configuration with IOR enabled (three clusters, two-pass Gaussian, r=2, σ=1). *Workstation:* AMD Ryzen 7 5800H @ 3.20 GHz, NVIDIA RTX 3050, 16 GB RAM, Chrome 129, Windows 11. *Laptop (integrated GPU):* Intel Core i7-1260P, Intel Iris Xe (integrated), 16 GB RAM, Chrome 129, Windows 11. Resolution: 1920×1080.

	Workstation (RTX 3050)				Laptop (Iris Xe)			
Dataset	#Tracks	GPU mem	FPS off	FPS on	#Tracks	GPU mem	FPS off	FPS on
Electromagnetic	12,507	1.24 MB	91	44	12,507	1.24 MB	58	31
Hadronic-1	3,751	1.62 MB	81	51	3,751	1.62 MB	60	38
Hadronic-2	22,604	3.73 MB	64	37	22,604	3.73 MB	39	22
Hadronic-3	1,983	1.49 MB	112	60	1,983	1.49 MB	75	45

warm-up. Table 2 reports GPU memory and median FPS with and without IOR. On the workstation, HGCalVis stays above 30 FPS for all four datasets, including the largest (Hadronic-2, 22.6k tracks) with IOR enabled. On the integrated-GPU laptop, the same configuration drops to 22 FPS on Hadronic-2 (below the interactive threshold, but still usable for inspection); with IOR disabled the laptop comfortably exceeds 30 FPS on all four datasets, confirming usability of the browser-only deployment on hardware without a discrete GPU, one of the key accessibility requirements raised by past reviewers. IOR roughly halves the frame rate in both configurations. Its cost is dominated by the per-cluster G-buffer fills and the Gaussian blur, both of which scale with viewport size rather than with the number of tracks; as a consequence the relative cost of IOR is largest on the lightest datasets and smallest on the densest, which matches the property we want. Increasing the Gaussian kernel radius monotonically reduces the frame rate (Figure 7); we recommend r=2, σ=1 as defaults, yielding clearly visible halos with no measurable artifacts and stable performance across all datasets.

7 User Study

The three main visual contributions of HGCalVis are depth cues (cylinder shading + SSAO), IOR, and the linked schematic view with multi-attribute filtering. To gather quantitative evidence that each

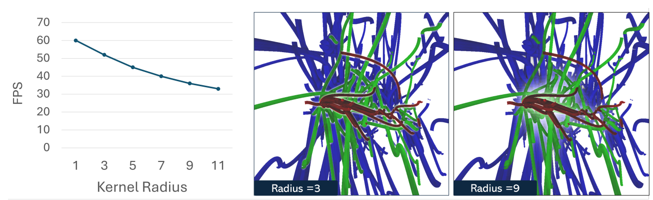

Figure 7: Effect of the two-pass Gaussian filter kernel radius on the IOR halo (Hadronic-3 dataset, σ=10). Larger radii produce softer, more diffuse halos at the cost of frame rate.

delivers a measurable benefit, we ran a web-based, preliminary user study with a paired within-subjects design. The study is deliberately compact: it covers exactly the claims the paper makes and was administered remotely, in the browser, in the same deployment the end users will eventually see.

Tasks and Conditions. We used three tasks (Table 3), each exercising one of the three feature groups. T_1 and T_3 contrast a control trial (relevant feature disabled) with a treatment trial (feature enabled). T_2 adds a third, intermediate condition: trajectories are color-coded by energy but rendered without importance ordering, which separates the effect of the color mapping from that of the ordered

Table 3: The three user-study tasks and the feature each isolates. T_1 and T_3 were run under two conditions (feature off/on); T_2 under three (no cue, color only, full IOR). Condition order was counterbalanced across participants.

ID	Task prompt	Feature under test
T_1	"Of the two highlighted trajectories, which one is closer to the camera?"	cylinder shading + SSAO
T_2	"Click on the highest-energy trajectory in this shower."	IOR
T_3	"Find all the direct daughter trajectories of the highlighted parent track."	schematic view + filtering

compositing. Each trial uses a different shower from the four production datasets, and all other features are held constant across conditions.

Participants. Nineteen participants completed the study, recruited via the authors' institutional and collaborator networks: 11 engineers or computer scientists, 6 physicists or physics students, and 2 from other technical backgrounds. Mean self-rated familiarity was 4.2/7 for particle physics and 4.9/7 for 3D scientific visualization. Participation was voluntary and uncompensated.

Procedure. The study was administered as a self-contained web page accessed through a personalised URL. Each session took about 15 minutes: (1) a brief consent and demographics form (1 min); (2) a short introduction of the background (2 min); (3) seven measured trials (T_1 and T_3 with two conditions each, T_2 with three), with condition order counterbalanced across participants and each trial preceded by a fresh dataset load and camera reset (~10 min); (4) a short subjective questionnaire (2 min). Task-completion time (in seconds, from prompt display to answer submission) and accuracy (binary for T_1; click within the bounding cylinder of the ground-truth track for T_2; exact recovery of the reference daughter set for T_3, with mean F1 as a secondary measure) were logged automatically.

Subjective Questionnaire. After the trials, each participant answered three Likert items on a seven-point scale (1 = strongly disagree, 7 = strongly agree), plus one optional open-ended comment:

- **S1.** The 3D shading and ambient occlusion made the spatial layout easier to read.
- **S2.** The importance-ordered rendering helped me identify the high-energy trajectories.
- **S3.** The hierarchical schematic view and filtering helped me navigate the data.
- **S4 (open).** In one sentence: what did you find most useful?

Results. All 19 participants completed the seven trials. Two T_3 trials exceeded the pre-registered five-minute limit, both on low-end hardware; they are retained, as excluding them does not change the direction of any result. Table 4 reports accuracy and median completion time per condition. For T_2, accuracy increased from 0.11 under unordered rendering to 0.42 with IOR (exact McNemar test, p=0.03). Energy color-coding without importance ordering left accuracy unchanged (0.11), which locates the effect in the ordered compositing rather than the color mapping. The increase held for both physicists (0.33 to 0.67) and non-physicists (0.00 to 0.31). For T_3, exact recovery of the daughter set increased from 0.37 to 0.79 (p<0.01; mean F1 0.62 to 0.83), while median completion time increased from 27 to 50 s, consistent with participants tracing the hierarchy rather than answering more quickly. T_1 accuracy was near ceiling under both conditions (0.84 vs. 0.95); depth cues primarily reduced response time (median 7.7 to 6.9 s, 14 of 19 participants faster, Wilcoxon signed-rank p=0.01). Mean Likert agreement (1–7) was 5.8 for the IOR item (S_2), 5.1 for depth cues (S_1), and 4.9 for the schematic view (S_3). Full descriptive statistics, paired significance tests, per-participant data, and verbatim responses are provided in the supplementary material.

Table 4: User-study results (N=19): accuracy and median completion time per condition. Accuracy is the fraction of correct trials; for T_3 it is exact recovery of the daughter set, with mean F1 in parentheses. T_2 separates energy color-coding from full IOR.

Task	Condition	Accuracy	Median t (s)
T_1 (depth cues)	off	0.84	7.7
	on	0.95	6.9
T_2 (IOR)	no cue	0.11	37.9
	color only	0.11	20.1
	IOR	0.42	33.7
T_3 (schematic)	off	0.37 (0.62)	26.9
	on	0.79 (0.83)	50.3

Discussion. The results are consistent with the three claims the study was designed to test, most clearly for IOR: identifying the single highest-energy trajectory was difficult under unordered rendering (0.11) and was not improved by color-coding alone, but rose to 0.42 once importance ordering kept that trajectory unoccluded. The schematic view increased the completeness of daughter-set recovery while also increasing interaction time, suggesting that participants used it to trace the hierarchy deliberately rather than to answer faster. Depth cues acted mainly on response time, as the spatial-ordering task was already near ceiling. Open-ended responses were consistent with these measurements: one physicist participant noted that *"the visual aids helped… better understand how trajectories move through 3D space"*, and a participant with little domain background described the prototype as *"pretty user friendly"*. Returning to the analysis questions of Section 3, the study provides quantitative support that IOR helps locate the high-energy spine (Q1) and that the linked schematic view helps trace the parent–child structure (Q3, and the navigation underlying Q5), while depth cues support the spatial judgments behind Q2 and Q4. Q4 and Q5 are examined here only indirectly and remain primarily qualitative (Section 8). Several factors bound these conclusions. The sample is small (N=19) and unevenly split between physicists and non-physicists, and absolute accuracy on T_2 remains low even with IOR. Participants also reported limitations of the prototype interface

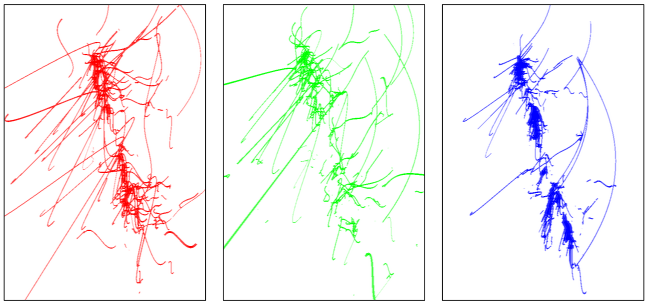

Figure 8: Use case: importance clustering of a hadronic shower. The three clusters (high in red, medium in green, low in blue) make the high-energy spine readable through the dense low-energy halo.

rather than of the techniques themselves, including the absence of an in-app tutorial, an inconsistent selection color between the 3D and 2D views, and reduced performance on low-end hardware; interface refinement was not an objective of this work, and the improvements observed despite an unpolished interface are consistent with the effect originating in the visual encodings.

8 Expert Feedback and Use Cases

Throughout the year-long collaboration with CERN, we exposed HGCalVis to five categories of stakeholders (Section 3). Three use cases recurred across these sessions, quoted (paraphrased and anonymized) where useful.

HGCal design validation. The HGCal simulation lead used HGCalVis to validate the shape of simulated electromagnetic and hadronic showers against expectations. The schematic view ordered by z-coordinate, combined with IOR (Figure 8), was the most-used pair of features: "*For the first time I can see the spine of the shower without running a separate analysis script. That this works in a browser means I can show it to the calorimeter mechanical team in their office.*"

TICL reconstruction debugging. PhD students working on TICL trackster reconstruction [Beaudette et al. 2023] used HGCalVis to debug merging failures in dense regions. The sub-tree filter driven from the schematic view was the dominant interaction: pick a parent, restrict the 3D view to its descendants, inspect.

External outreach. The ROOT-EVE lead used HGCalVis to demonstrate HGCal showers to non-CMS visitors. Because the tool is browser-only, the demonstration required no setup beyond sharing a URL, the operational reason for R1.

Recurring requests. The most-requested follow-ups across stakeholders were: (i) visualization of reconstructed tracksters [Beaudette et al. 2023] on top of the simulated shower, (ii) side-by-side comparison of two events, and (iii) multi-region brushing on the schematic view.

9 Discussion, Limitations and Future Work

Although described for particle showers, IOR applies to any dense line set with a scalar importance and a target number of clusters; informal experiments on streamline integration and DTI fiber bundles produced plausible results, but full cross-domain evaluation is future work. IOR is closest in spirit to Viola et al.'s importance-driven volume rendering [Viola et al. 2004, 2005]; it can be read as its line-rendering analogue under the WebGL 2.0 envelope. IOR does not produce a globally exact depth-ordered composition: depth tests are run per cluster and the compositing step assumes that the importance ordering dominates the depth ordering. In practice that is exactly the behaviour the domain requires (high-energy tracks must be visible regardless of depth), but small artifacts can appear when a high-importance track passes behind a low-importance bundle close to the camera. No expert session in our collaboration mistook these for real physical effects. Finally, IOR defaults to hue for the importance channel, which collides with any other use of hue (e.g., PDG); we therefore ship a warm-to-cool alternative inspired by Rheingans and Ebert [Rheingans and Ebert 2001].

Limitations and Roadmap. HGCalVis stays interactive on integrated-GPU laptops, but other low-end mobile devices, such as tablets, remain untested. The most likely bottleneck is the Gaussian blur over fullscreen quads; a separable mipmap-based approximation, or a port to WebGPU compute shaders once browser support stabilizes, would address it, and the same port would unlock per-pixel linked-list opacity optimization [Günther et al. 2013]. Our roadmap is otherwise shaped by the recurring stakeholder requests (Section 8): (i) integrate reconstructed tracksters [Beaudette et al. 2023] alongside the simulated shower, with explicit support for visualising reconstruction failures, (ii) add side-by-side event comparison, (iii) generalise the schematic view to multi-region brushing, and (iv) export through a glTF extension carrying per-track importance, PDG, and energy so that HGCalVis data is consumable by other open-web tools. We also plan to integrate HGCalVis directly into ROOT-EVE and replicate the user study on a larger scale.

10 Conclusion

We have presented HGCalVis, a web-based 3D visualization system for simulated particle showers in CERN's HGCal calorimeter. Its main rendering contribution is IOR, an Importance-Ordered Rendering algorithm tailored to the WebGL 2.0 capability envelope. A preliminary user study (N=19) indicates that importance-ordered rendering improves identification of high-energy trajectories over unordered rendering, and that the linked schematic view improves recovery of a track's daughter set, while depth cues mainly reduce response time on an already-simple spatial task; qualitative feedback from CERN collaborators supports the practical utility of the tool. As HGCal moves towards construction, we hope HGCalVis will help as a low-friction, open-web bridge among physicists, simulation experts, reconstruction algorithm developers, and calorimeter designers who share the need to examine and better understand these showers.

Acknowledgments

This research was supported by the KAUST–CERN Collaboration Program and partially supported by the U.S. National Science Foundation under Cooperative Agreement PHY-2121686.

References

Amani Ageeli, Alberto Jaspe-Villanueva, Ronell Sicat, Florian Mannuss, Peter Rautek, and Markus Hadwiger. 2023. Multivariate Probabilistic Range Queries for Scalable Interactive 3D Visualization. *IEEE Transactions on Visualization and Computer Graphics* 29, 1 (2023), 646–656. doi:10.1109/TVCG.2022.3209439

Reem Alghamdi, Markus Hadwiger, Guido Reina, and Alberto Jaspe-Villanueva. 2025. Lactea: Web-Based Spectrum-Preserving Multi-Resolution Visualization of the GAIA Star Catalog. *Computer Graphics Forum* 44, 3 (2025). doi:10.1111/cgf.70117

G. Apollinari, I. Béjar Alonso, O. Brüning, P. Fessia, M. Lamont, L. Rossi, and L. Tavian. 2017. High-Luminosity Large Hadron Collider (HL-LHC): Technical design report. *CERN Yellow Reports: Monographs* 4 (2017).

Louis Bavoil, Steven P. Callahan, Aaron Lefohn, Joao L.D. Comba, and Cláudio T. Silva. 2007. Multi-fragment effects on the GPU using the k-buffer. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*. 97–104.

Louis Bavoil and Miguel Sainz. 2008. Screen space ambient occlusion. *NVIDIA Developer Information* (2008).

Florian Beaudette, Marco Rovere, Benedikt Maier, Felice Pantaleo, Shamik Ghosh, Alexander Schmidt, Abhirikshma Nandi, Alessandro Tarabini, Alice Savona, and Wahid Redjeb. 2023. *The TICL reconstruction at the CMS Phase-2 High Granularity Calorimeter Endcap.* Technical Report. CERN.

Ciril Bohak et al. 2023. *RenderCore: A WebGL-based rendering engine.* https://github.com/UL-FRI-LGM/RenderCore

Andrea Brambilla, Robert Carnecky, Robert Peikert, Ivan Viola, and Helwig Hauser. 2012. Illustrative flow visualization: State of the art, trends and challenges. (2012).

Stefan Bruckner and Eduard Gröller. 2007. Enhancing depth-perception with flexible volumetric halos. *IEEE Transactions on Visualization and Computer Graphics* 13, 6 (2007), 1344–1351.

Rene Brun, Fons Rademakers, et al. 2020. *ROOT – An Object Oriented Data Analysis Framework.* doi:10.5281/zenodo.3895860

Corrado Cali and Marco Agus. 2024. NeuroVerse: Immersive 3D ultrastructural brain reconstructions for education and collaborative analysis. In *Proceedings of the 29th International ACM Conference on 3D Web Technology (Web3D)*. ACM.

Loren Carpenter. 1984. The A-buffer, an antialiased hidden surface method. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*. 103–108.

G. Chiriţoi, A. Jipa, and E.M. Popescu. 2016. 3D visualization for simulated Extensive Air Showers (EAS) using CORSIKA and Blender 3D. *Romanian Reports in Physics* 68, 2 (2016), 563–570.

CMS Collaboration. 2008. The CMS experiment at the CERN LHC. *Journal of Instrumentation* 3, 08 (2008), S08004.

CMS Collaboration. 2017. The Phase-2 Upgrade of the CMS Endcap Calorimeter. *CERN-LHCC-2017-023* (2017).

Zach Duer, Leo Piilonen, and George Glasson. 2018. Belle2VR: A virtual-reality visualization of subatomic particle physics in the Belle II experiment. *IEEE Computer Graphics and Applications* 38, 3 (2018), 33–43.

Sebastian Eichelbaum, Mario Hlawitschka, and Gerik Scheuermann. 2012. LineAO – Improved three-dimensional line rendering. *IEEE Transactions on Visualization and Computer Graphics* 19, 3 (2012), 433–445.

Eric Enderton, Erik Sintorn, Peter Shirley, and David Luebke. 2010. Stochastic transparency. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. 157–164.

Robin Eschbach, Kai Messerschmidt, Ralf Keidel, and Alexander Wiebel. 2022. Semi-automatic particle tracking for and visualization of particle detector data. In *Vision, Modeling, and Visualization*, Jan Bender, Mario Botsch, and Daniel A. Keim (Eds.). The Eurographics Association. doi:10.2312/vmv.20221210

Lyndon Evans and Philip Bryant. 2008. LHC machine. *Journal of Instrumentation* 3, 08 (2008), S08001.

Cass Everitt. 2001. Interactive order-independent transparency. In *White paper, nVIDIA*.

Maarten H. Everts, Henk Bekker, Jos B.T.M. Roerdink, and Tobias Isenberg. 2009. Depth-dependent halos: Illustrative rendering of dense line data. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (2009), 1299–1306.

L. Garren, I.G. Knowles, T. Sjöstrand, and T. Trippe. 2000. Monte Carlo particle numbering scheme. *The European Physical Journal C* 15, 1–4 (2000), 205–207.

Stéphane Grabli, Emmanuel Turquin, Frédo Durand, and François X. Sillion. 2010. Programmable rendering of line drawing from 3D scenes. *ACM Transactions on Graphics* 29, 2, Article 18 (2010), 20 pages.

Tobias Günther, Christian Rössl, and Holger Theisel. 2013. Opacity optimization for 3D line fields. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 1–8.

Mathias Kanzler, Florian Ferstl, and Rüdiger Westermann. 2016. Line density control in screen-space via balanced line hierarchies. *Computers & Graphics* 61 (2016), 29–39.

Mathias Kanzler, Marc Rautenhaus, and Rüdiger Westermann. 2018. A voxel-based rendering pipeline for large 3D line sets. *IEEE Transactions on Visualization and Computer Graphics* 25, 7 (2018), 2378–2391.

Mathias Kanzler and Rüdiger Westermann. 2018. Interactive visual exploration of line clusters. In *Proceedings of the Conference on Vision, Modeling, and Visualization*. 155–163.

Michael Kern, Christoph Neuhauser, Torben Maack, Mengjiao Han, Will Usher, and Rüdiger Westermann. 2020. A comparison of rendering techniques for 3D line sets with transparency. *IEEE Transactions on Visualization and Computer Graphics* 27, 8 (2020), 3361–3376.

Michael Kern and Rüdiger Westermann. 2019. Clustering ensembles of 3D jet-stream core lines. In *Vision, Modeling and Visualization (VMV)*. The Eurographics Association.

Khronos Group. 2022. WebGL 2.0 Specification. https://www.khronos.org/registry/webgl/specs/latest/2.0/

Jens Krüger, Jens Schneider, and Rüdiger Westermann. 2006. ClearView: An interactive context preserving hotspot visualization technique. *IEEE Transactions on Visualization and Computer Graphics* 12, 5 (2006), 941–948.

Ovidio Mallo, Ronald Peikert, Christian Sigg, and Filip Sadlo. 2005. Illuminated lines revisited. In *VIS 05. IEEE Visualization, 2005*. IEEE, 19–26.

Morgan McGuire and Louis Bavoil. 2013. Weighted blended order-independent transparency. *Journal of Computer Graphics Techniques* 2, 4 (2013).

Martin Mittring. 2007. Finding next gen: CryEngine 2. In *ACM SIGGRAPH 2007 Courses*. 97–121.

Cedrick Münstermann, Stefan Krumpen, Reinhard Klein, and Christoph Peters. 2018. Moment-based order-independent transparency. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1, 1 (2018), 1–20.

Finian Mwalongo, Michael Krone, Guido Reina, and Thomas Ertl. 2016. State-of-the-art report in web-based visualization. *Computer Graphics Forum* 35, 3 (2016), 553–575.

Bui Tuong Phong. 1975. Illumination for computer generated pictures. *Commun. ACM* 18, 6 (1975), 311–317.

Giovanni Pintore, Alberto Jaspe-Villanueva, Markus Hadwiger, Enrico Gobbetti, Jens Schneider, and Marco Agus. 2023. PanoVerse: Automatic generation of stereoscopic environments from single indoor panoramic images for metaverse applications. In *Proceedings of the 28th International ACM Conference on 3D Web Technology (Web3D)*. ACM.

Giovanni Pintore, Alberto Jaspe-Villanueva, Markus Hadwiger, Jens Schneider, Marco Agus, Fabio Marton, Fabio Bettio, and Enrico Gobbetti. 2024. Deep synthesis and exploration of omnidirectional stereoscopic environments from a single surround-view panoramic image. *Computers & Graphics* 119 (2024), 103907. doi:10.1016/j.cag.2024.103907

Penny Rheingans and David Ebert. 2001. Volume illustration: Nonphotorealistic rendering of volume models. *IEEE Transactions on Visualization and Computer Graphics* 7, 3 (2001), 253–264.

Irene Baeza Rojo, Markus Gross, and Tobias Günther. 2019. Fourier opacity optimization for scalable exploration. *IEEE Transactions on Visualization and Computer Graphics* 26, 11 (2019), 3204–3216.

Marco Salvi, Jefferson Montgomery, and Aaron Lefohn. 2011. Adaptive transparency. In *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*. 119–126.

Marco Salvi and Karthik Vaidyanathan. 2014. Multi-layer alpha blending. In *Proceedings of the 18th meeting of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. 151–158.

Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. 2017. Vega-Lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics* 23, 1 (2017), 341–350.

Michael Sedlmair, Miriah Meyer, and Tamara Munzner. 2012. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics* 18, 12 (2012), 2431–2440.

Carsten Stoll, Stefan Gumhold, and Hans-Peter Seidel. 2005. Visualization with stylized line primitives. *IEEE Visualization 2005* (2005), 695–702.

N.A. Svakhine, Y. Jang, D. Ebert, and K. Gaither. 2005. Illustration and photography inspired visualization of flows and volumes. In *VIS 05. IEEE Visualization, 2005*. 687–694. doi:10.1109/VISUAL.2005.1532858

Ivan Viola, Armin Kanitsar, and Meister Eduard Groller. 2004. Importance-driven volume rendering. In *IEEE Visualization 2004*. IEEE, 139–145.

Ivan Viola, Armin Kanitsar, and M. Eduard Groller. 2005. Importance-driven feature enhancement in volume visualization. *IEEE Transactions on Visualization and Computer Graphics* 11, 4 (2005), 408–418.

Tobias Wagner, Daniel Limberger, Willy Scheibel, and Jürgen Döllner. 2022. Hardware-accelerated web-based 3D scatter plots with projected density fields and embedded controls. In *Proceedings of the 27th International ACM Conference on 3D Web Technology (Web3D)*. ACM.

Xiaoming Wei, Arie E. Kaufman, and Timothy J. Hallman. 2001. Case study: visualization of particle track data. In *Proceedings Visualization, 2001. VIS'01*. IEEE, 465–590.

Jason C. Yang, Justin Hensley, Holger Grün, and Nicolas Thibieroz. 2010. Real-time concurrent linked list construction on the GPU. In *Computer Graphics Forum*, Vol. 29. Wiley Online Library, 1297–1304.