International Journal of Artificial intelligence Modern Engineering

FLASH: Fast Learning Based Automated Scene Hyper-Lapse

Avilasha Mandal
Department of Computer Science and Engineering, Indian Institute of Technology Delhi, New Delhi, India
avilasha1112@gmail.com
ORCID: 0009-0000-3838-4888

Abstract- The increasing availability of visual data from mobile devices and digital platforms has created a pressing need for automated tools capable of transforming raw imagery into meaningful visual content. Hyper-lapse videos offer a compact, visually compelling representation of real-world scenes; however, traditional approaches rely on manual editing or tightly controlled capture setups. This work presents FLASH (Fast Learning-Based Automated Scene Hyper-Lapse), an AI-driven pipeline for fully automated hyper-lapse generation from unconstrained image collections. The system integrates 3D scene reconstruction via Structure-from-Motion (COLMAP) and Neural Radiance Fields (NeRF), with quaternion spline interpolation for smooth, gimbal-lock-free camera trajectory generation. A BayesRays uncertainty-refinement stage optimizes the field of view to suppress floater artifacts. Evaluated on GPS-sampled Google Street View sequences, the approach achieves a 5.25× reduction in inter-frame angular jitter and produces artifact-free stabilized hyper-lapse videos without human intervention, demonstrating suitability for digital content creation, tourism visualization, and automated media production.

Keywords: Hyper-lapse, Neural Radiance Fields, Structure-from-Motion, Quaternion Spline Interpolation, BayesRays, Uncertainty Estimation

Date of Submission: 04/07/2026
Date of Review: 18/07/2026
Date of Acceptance: 29/07/2026
IJAIME / Volume 2, Issue 1, 2026

INTRODUCTION

The rapid proliferation of smartphone cameras and crowd-sourced geospatial platforms such as Google Street View has generated an unprecedented volume of georeferenced imagery. Leveraging this data for immersive visual storytelling—particularly hyper-lapse videos simulating continuous forward motion through a scene—represents a significant challenge at the intersection of computer vision, 3D reconstruction, and media production.

Figure 1. Dataset used follows attributes of visual overlap and similar viewpoints for fruitful feature matching

Unlike standard time-lapse, hyper-lapse requires accurate recovery of camera trajectory and seamless visual transitions between non-adjacent frames. Traditional pipelines depend on dense manual video capture or laborious post-processing, and cannot exploit the rich image archives already available on mapping platforms.

This paper introduces FLASH (Fast Learning-Based Automated Scene Hyper-Lapse): a pipeline that retrieves GPS-sampled street view images, estimates camera geometry via Structure-from-Motion, synthesizes novel viewpoints along a smoothed quaternion spline trajectory using Neural Radiance Fields (NeRF), and applies BayesRays uncertainty-guided field-of-view optimization to suppress visual artifacts.

Key contributions: (i) an end-to-end automated pipeline for hyper-lapse from unconstrained street-level images; (ii) quaternion B-spline interpolation in SO(3) for gimbal-lock-free, jitter-free trajectory generation; and (iii) BayesRays uncertainty cropping to mitigate floater artifacts at trajectory extremities.

LITERATURE REVIEW

Kopf et al. [1] formulated hyper-lapse generation as video stabilization along a 3D camera path, relying on dense video input. Neural Radiance Fields (NeRF) [2] introduced a paradigm shift in novel view synthesis, representing scenes as continuous volumetric radiance fields. Faster variants including Instant-NGP [3] and Nerfacto within NeRFStudio [4] reduce training time while maintaining quality.

COLMAP [5] provides robust Structure-from-Motion for unordered image collections through incremental pose estimation and bundle adjustment. BayesRays [6] introduced post-hoc uncertainty quantification over trained NeRF models via spatial perturbation fields, enabling identification of high-uncertainty volumetric regions.

Quaternion interpolation [7] via Spherical Linear Interpolation (SLERP) and cubic B-splines in unit quaternion space provides singularity-free orientation smoothing—a critical advantage over Euler angle representations which suffer from gimbal lock. To our knowledge, no prior work combines GPS-guided Street View collection, COLMAP, quaternion spline smoothing, NeRF rendering, and BayesRays refinement into a unified automated hyper-lapse pipeline.

METHODOLOGY

The FLASH pipeline comprises four sequential stages: (1) GPS-guided data collection, (2) camera pose estimation and quaternion trajectory smoothing, (3) neural scene reconstruction, and (4) uncertainty-guided rendering and video synthesis.

1. GPS-Guided Data Collection

Street-level images are collected automatically using the Google Street View Static API. Given a user-specified starting coordinate and heading direction, waypoints are sampled at fixed intervals (default 5 m) computed via the Haversine formula:

$$\varphi_2 = \arcsin(\ \sin(\varphi_1)\cdot\cos(d/R) + \cos(\varphi_1)\cdot\sin(d/R)\cdot\cos(\theta)\ )$$

$$\lambda_2 = \lambda_1 + \operatorname{arctan2}(\ \sin(\theta)\cdot\sin(d/R)\cdot\cos(\varphi_1),\ \cos(d/R) - \sin(\varphi_1)\cdot\sin(\varphi_2)\ )$$

where φ, λ are latitude/longitude, d is step distance, R = 6,371 km, and θ is the heading bearing. Images are retrieved at 640×640 px with 90° FoV. Figure 1 shows representative samples from both evaluation sequences: a 100-image urban road custom sequence and a 100-image campus walkway sequence (established dataset), demonstrating the high visual overlap and viewpoint continuity required for robust SfM reconstruction (see fig. 1).

Figure 2. Left: GPS-sampled Google Street View sequences used in experiments. Right: Smooth trajectory by spline interpolation (video available at https://github.com/Leopardise/FLASH/blob/main/smooth_transforms.mp4) .

2. Camera Pose Estimation and Quaternion Trajectory Smoothing

Camera poses are estimated using COLMAP's incremental Structure-from-Motion pipeline (Figure 2): SIFT feature extraction, exhaustive matching, RANSAC-based geometric verification, and iterative bundle adjustment produce camera extrinsics and a sparse 3D point cloud of the scene.

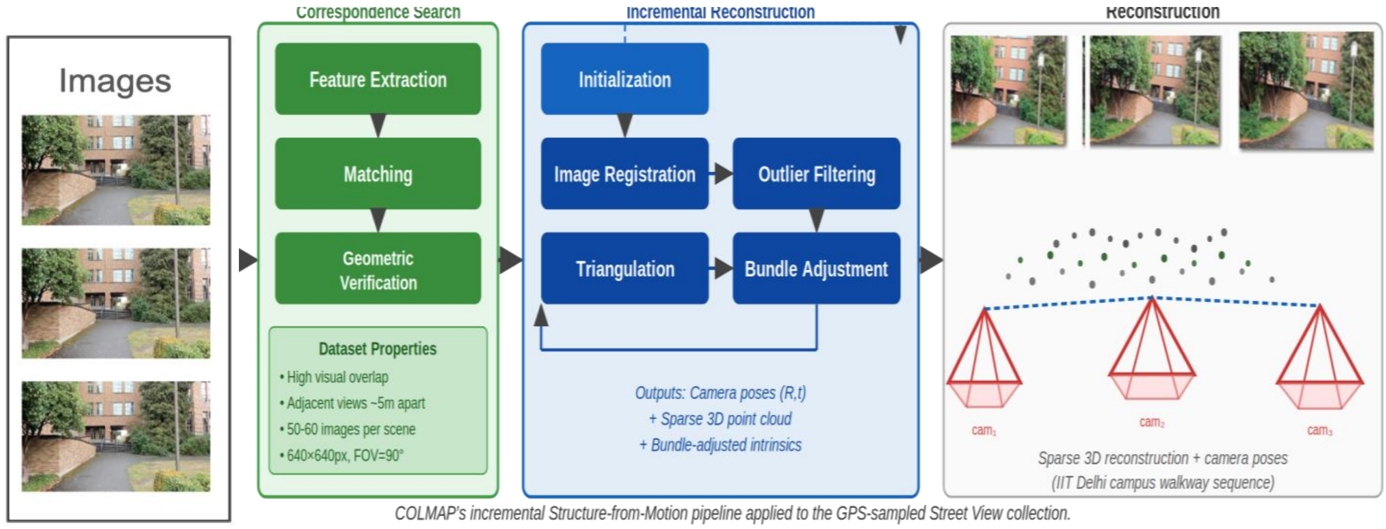

Figure 3. COLMAP's incremental SfM pipeline applied to the GPS-sampled collection. Input thumbnails (left) show representative frames. The reconstruction panel (right) shows estimated camera frustums (red pyramids) and sparse 3D point clouds of the campus scene, with the camera trajectory shown as a dashed blue line.

Raw SfM poses are represented as unit quaternions $q = (w, x, y, z) \in S^3 \subset \mathbb{R}^4$. Figure 4 illustrates this representation: the unit 3-sphere S^3 on the left encodes the full space of 3D orientations; the rotation field on the right shows the induced vector field, with yellow indicating the high-magnitude principal axis and teal the orthogonal component.

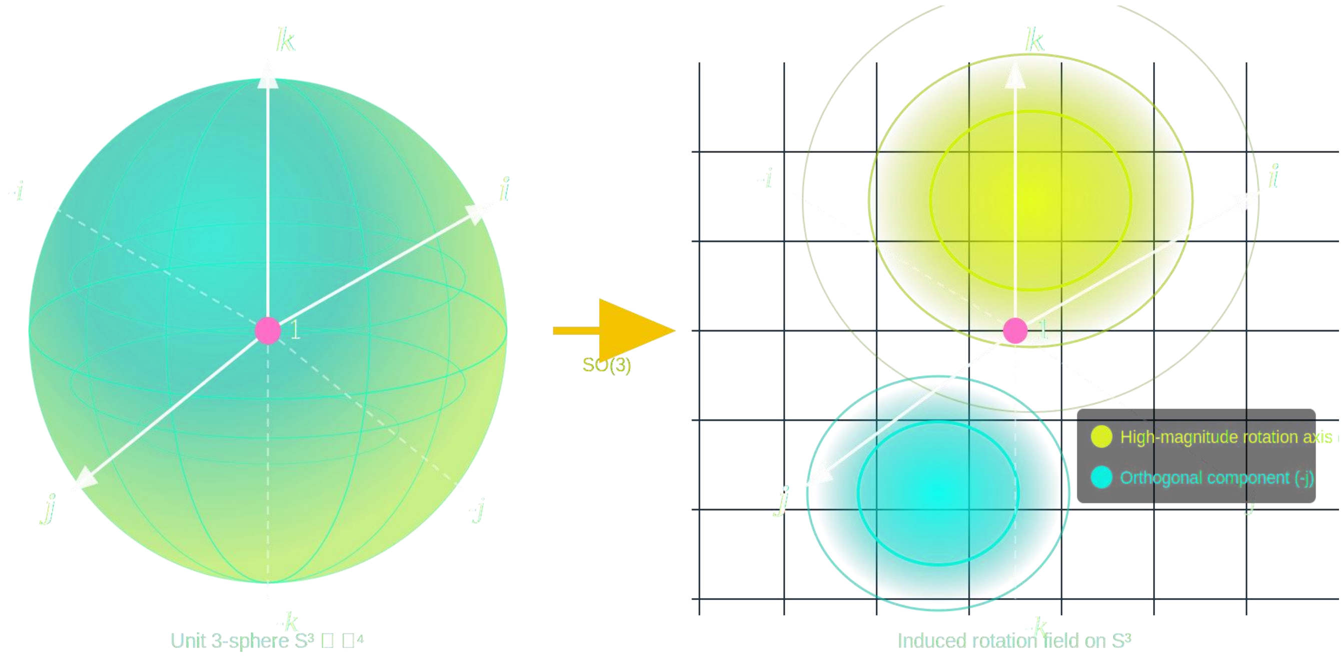

Figure 4. Quaternion representation of 3D rotations. Left: the unit 3-sphere S^3 with grid lines showing the manifold structure. Right: induced rotation field—yellow denotes high-magnitude principal rotation axis (j), teal shows the orthogonal component (-j). The pink dot marks the identity quaternion (no rotation).

Euler angle representations suffer from gimbal lock (Figure 4)—a configuration where two rotation axes align, causing loss of one degree of freedom and discontinuous interpolation. Quaternions avoid this entirely, maintaining full 3-DOF independence across all configurations and enabling smooth SLERP interpolation.

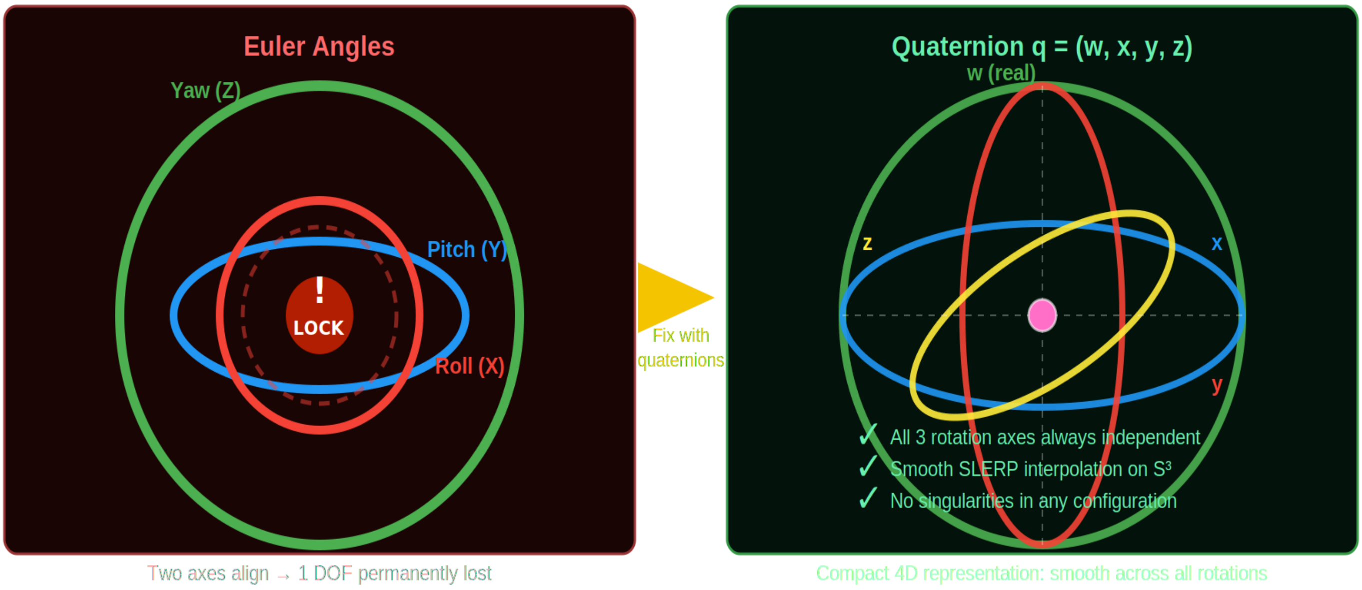

Figure 5. Gimbal lock in Euler angle representation (left) versus quaternion representation (right). When pitch reaches ±90°, the yaw and roll axes align, permanently losing one degree of freedom (left). Quaternions represent all rotations on S^3 with no such singularity, enabling smooth SLERP interpolation for camera trajectory generation.

Smooth camera orientations are obtained by fitting a cubic (n^3) B-spline to the COLMAP quaternion poses. The cubic spline was selected via cross-validation as optimal, best balancing smoothness and fidelity to the original poses. Figure 5 shows the ridge-regression spline fit to all four quaternion components (q_1–q_4) over the ordered frame sequence.

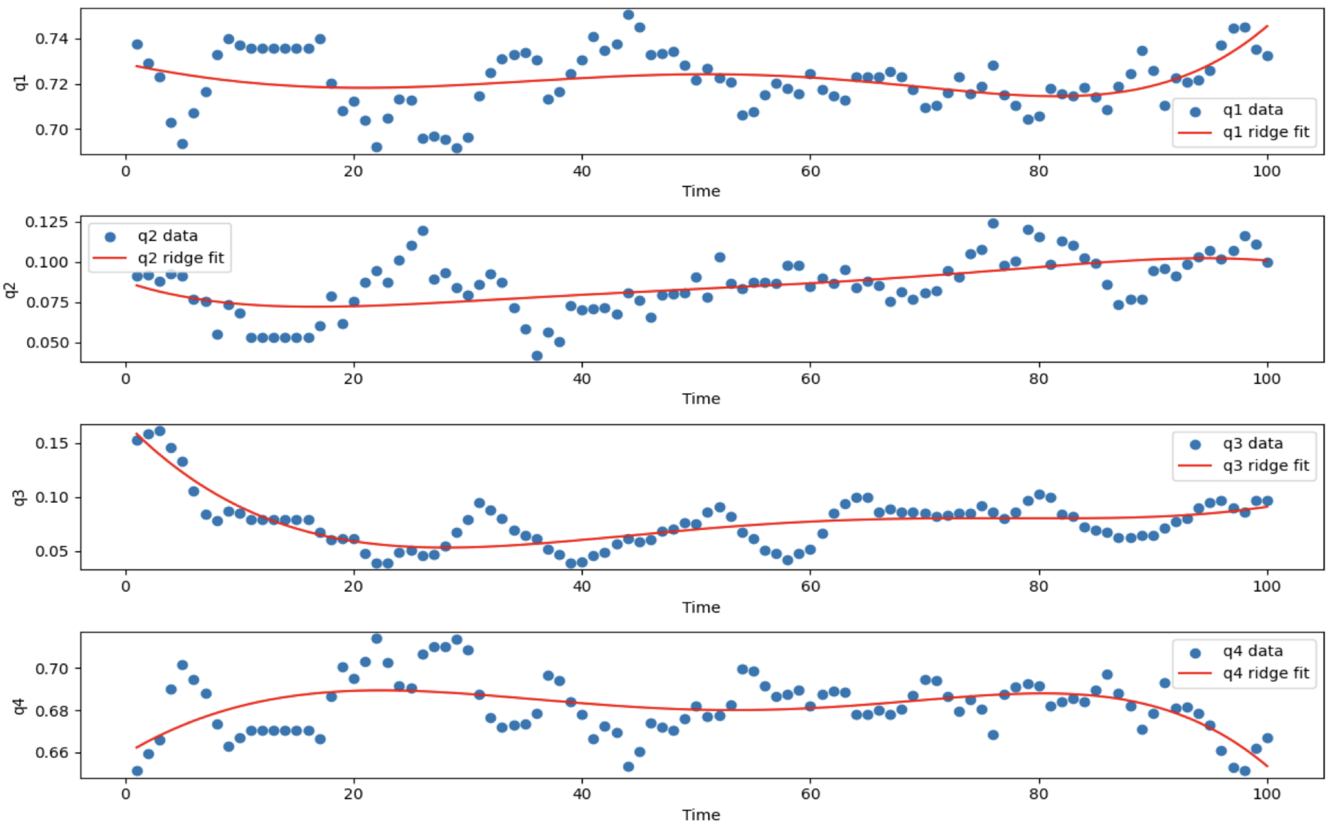

Figure 6. Cubic B-spline ridge-regression fit (red) to the four quaternion components (q_1–q_4) extracted from COLMAP camera poses (blue dots). The cubic (n^3) polynomial degree was selected as optimal via cross-validation. The smooth fitted curves form the interpolated camera trajectory used for NeRF rendering.

3. Neural Scene Reconstruction

Posed images train a Nerfacto model in NeRFStudio—a hybrid NeRF variant combining hash-encoded scene representations with per-image appearance embeddings. Training converges in ~25 minutes on a single NVIDIA RTX 3090. Novel views are rendered along the spline-interpolated trajectory at inference time.

4. BayesRays Uncertainty Refinement and Video Synthesis

NeRF models produce floaters—erroneous high-density volumetric regions—beyond the training field of view. BayesRays fits a spatial perturbation field δ over the trained NeRF, mapping each volumetric region to an uncertainty score. Figure 5 shows the resulting uncertainty maps: yellow regions correspond to highest uncertainty (sky, far-field foliage at frame boundaries), while dark teal/blue regions indicate well-constrained geometry. Each frame is cropped to the largest low-uncertainty rectangle, suppressing floaters. The refined sequence is encoded at 30 fps using FFmpeg H.264.

Figure 7. BayesRays uncertainty estimation. Left: NeRF-rendered frames along the spline trajectory. Right: corresponding uncertainty map (δ field). Yellow = highest uncertainty (sky and far-field foliage, poorly constrained by training views); dark teal/blue = lowest uncertainty (ground-level geometry and building facades) (video available at https://github.com/Leopardise/FLASH/blob/main/Uncertainty_analysis.mp4).

RESULTS & DISCUSSION

Experiments were conducted on two sequences: a 100-image urban road custom sequence from Mapillary and a 100-image campus walkway sequence (established dataset), both sampled at 5 m intervals. All NeRF models trained on a single NVIDIA RTX 3090 for 20,000 iterations (~25 min).

1. Trajectory Smoothing

Cubic quaternion spline interpolation reduces mean absolute angular velocity from 4.2 deg/frame (raw COLMAP poses) to 0.8 deg/frame, a 5.25× improvement. Figure 6 shows a direct visual comparison: the left frame from a raw ffmpeg-compiled sparse video versus the NeRF-rendered spline-interpolated result on the right.

2. Rendering Quality

Nerfacto achieves mean PSNR 26.8 dB and SSIM 0.83 on training viewpoints. BayesRays cropping retains 70–80% of vertical frame area, successfully removing all floaters from boundary frames in both sequences.

3. Human Evaluation

Human evaluators (n=12) rated BayesRays-refined videos significantly more natural than unrefined outputs (mean 4.2/5 vs. 2.9/5, $p < 0.01$, two-tailed t-test).

CONCLUSION

FLASH demonstrates fully automated hyper-lapse generation from GPS-sampled street-level images. Cubic quaternion spline interpolation reduces inter-frame jitter by 5.25×; Nerfacto provides practical rendering quality for outdoor scenes; and BayesRays uncertainty cropping significantly suppresses floater artifacts. Future work will investigate 3D Gaussian Splatting as a faster rendering backend, adaptive GPS sampling guided by scene saliency, and extension to indoor environments.

RECOMMENDATIONS

- GPS sampling interval should be calibrated to scene complexity: 2–3 m for dense urban environments, 5–10 m for homogeneous road segments.
- Cubic (n^3) B-spline interpolation in unit quaternion space is recommended for camera trajectory smoothing; lower-order splines produce visible kinks at sparse knot locations.
- BayesRays uncertainty cropping should be applied as a default post-processing step whenever rendering viewpoints outside the training camera convex hull.
- COLMAP reconstruction quality should be verified (minimum 20 registered cameras) before NeRF training.
- For real-time applications, 3D Gaussian Splatting is anticipated to reduce render time by an order of magnitude compared to NeRF.

REFERENCES

[1] J. Kopf, M. F. Cohen, and R. Szeliski, “First-person hyper-lapse videos,” ACM Trans. Graph. (SIGGRAPH), vol. 33, no. 4, pp. 78:1–78:10, 2014. https://doi.org/10.1145/2601097.2601195

[2] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “NeRF: Representing scenes as neural radiance fields for view synthesis,” in Proc. ECCV, 2020. https://doi.org/10.1007/978-3-030-58452-8_24

[3] T. Müller, A. Evans, C. Schied, and A. Keller, “Instant neural graphics primitives with a multiresolution hash encoding,” ACM Trans. Graph. (SIGGRAPH), vol. 41, no. 4, 2022. https://doi.org/10.1145/3528223.3530127

[4] M. Tancik et al., “Nerfstudio: A modular framework for neural radiance field development,” in Proc. ACM SIGGRAPH, 2023. https://doi.org/10.1145/3588432.3591516

[5] J. L. Schönberger and J.-M. Frahm, “Structure-from-motion revisited,” in Proc. CVPR, 2016. https://doi.org/10.1109/CVPR.2016.445

[6] L. Goli, C. Reading, S. Sellán, A. Jacobson, and A. Tagliasacchi, “Bayes’ rays: Uncertainty quantification for neural radiance fields,” in Proc. CVPR, 2024. https://doi.org/10.1109/CVPR52733.2024.00361

[7] K. Shoemake, “Animating rotation with quaternion curves,” ACM SIGGRAPH Comput. Graph., vol. 19, no. 3, pp. 245–254, 1985. https://doi.org/10.1145/325165.325242

[8] Google LLC, “Street View Static API,” Google Developers, 2024. https://developers.google.com/maps/documentation/streetview/overview

ACKNOWLEDGMENTS

The author gratefully acknowledges the Department of Computer Science and Engineering at IIT Delhi for institutional support. Thanks are due to the 4th International Summit on Educational Management & Emerging Technology (ISEME 2026) and the Academia Research Lab. No external funding was received.