AR Journey Into Movies

Zhihao Cai
ETH Zürich
caizhi@ethz.ch

Junyi Huang
ETH Zürich
junyhuang@ethz.ch

Yiwei Pang
ETH Zürich
pangy@ethz.ch

Li Wa Tang
ETH Zürich
litang1@ethz.ch

Abstract

Film tourism is popular, but reproducing an iconic movie shot on site is not that straightforward: even at the correct location, matching the exact camera viewpoint is often a trial-and-error process. In addition, mobile AR tracking can drift, causing the overlay to gradually misalign. This work presents AR Journey into Movies, an iOS AR application that anchors a target movie frame in the live camera view so users can navigate to the correct viewpoint and capture cinematic photos. The system uses prebuilt Structure-from-Motion (SfM) models for filming locations and localizes incoming frames via hierarchical localization (HLOC) with DISK features. A multithreaded server selects informative keyframes via Farthest Point Sampling and estimates a global similarity (Sim(3)) transform: scale, rotation, and translation, by fitting correspondences between localized SfM camera poses and AR session poses. To mitigate ARKit/ARCore drift and occasional mislocalization, we estimate the SfM-AR alignment as a Sim(3) transform with RANSAC, rejecting geometrically inconsistent pose correspondences. On two test sites, the approach achieves average deviations of 0.748 m/ 5.01° outdoors and 0.567 m/ 0.84° indoors, enabling stable overlays. The application further supports AR character overlays and camera filters to enhance the on-site experience.

1. Introduction

Filming locations have become destinations in their own right, yet recreating an iconic movie shot in person remains surprisingly difficult: visitors may reach the correct place and still miss the precise camera viewpoint. The challenge is not only geometric (position and orientation) but also practical, guidance must stay stable while users walk, rotate, and rescan the scene.

Mobile augmented reality provides an intuitive interface by rendering a target frame directly in the live camera view. However, AR tracking is defined in a local session coordinate system and may drift over time, especially in large outdoor motions or challenging visual conditions. This drift can cause an overlay that initially looks correct to gradually slide away, undermining both navigation and the final photo.

This paper presents AR Journey into Movies, a client-server mobile AR system that grounds an AR session to a prebuilt 3D model of the location. The mobile client streams camera images and AR poses to a remote server, which performs image-based localization within an SfM reconstruction and estimates a robust similarity alignment between the SfM and AR coordinates. To remain reliable under tracking drift and real-world perturbations, the alignment is computed from a spatially well-distributed set of pose pairs and filtered by a RANSAC consistency check before updating the AR visualization. We evaluate the system across representative indoor and outdoor scenes and further analyze localization robustness under illumination and appearance changes, demonstrating stable on-site overlays for viewpoint reproduction and an enhanced user experience with cinematic capture features.

2. Related Work

3D reconstruction and localization Three-dimensional reconstruction and localization methods can be broadly divided into geometry-based and learning-based approaches.

- **Geometry-based methods** explicitly model scene structure using geometric constraints. Structure-from-Motion (SfM)[5] reconstructs a static 3D map from multiple images and enables accurate camera pose estimation via 2D-3D correspondences, while SLAM systems perform reconstruction and localization online from continuous image streams. Although SLAM is widely adopted in real-time AR, it relies on continuous tracking and may suffer from drift, which limits its ability to precisely align predefined visual content.
- **Learning-based methods** aim to infer 3D structure directly from visual input using deep networks. Approaches such as CUT3R[10] maintain implicit scene representations through learned persistent states, enabling continuous 3D perception without explicit geometric maps. However, these methods are less suited for precise single-image localization and verifiable geometric alignment.

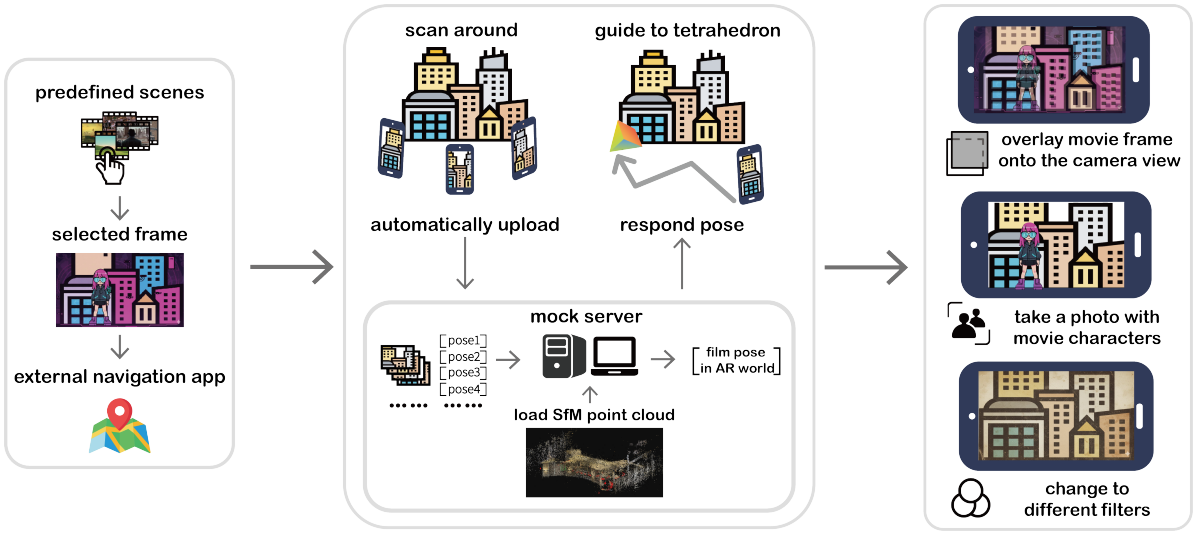
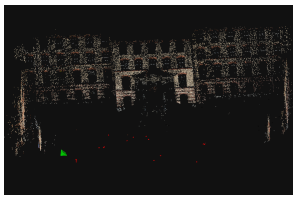
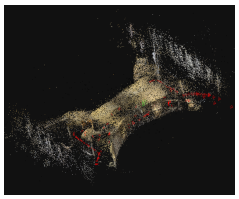


Figure 1. Workflow overview of *AR Journey into Movies*.The workflow starts with movie frame selection and approximate location navigation on the mobile client. During on-site scanning, camera view images and poses in AR session world are uploaded to a remote server, where HLOC-based localization and coordinate transformation between the SfM and AR worlds are estimated. The refined movie frame pose is returned for real-time AR visualization and interaction.

In contrast, our system adopts an SfM-based representation together with an HLOC-style localization pipeline[6, 7], which enables accurate pose estimation from individual images and avoids drift through explicit geometric verification. This design is well suited for aligning individual movie frames with real-world scenes.



((a)) ETHz CAB(outdoor).



((b)) ETHz HG EO-Nord(indoor).

Figure 2. SfM reconstruction results.

Local Features Learning and Matching HLOC offers a modular localization pipeline that supports different combinations of local feature extractors and matching strategies. In our system, we select DISK as the local feature extractor and LightGlue as the matcher.

- **DISK** is a learned local feature extractor that jointly performs keypoint detection and description and is optimized for matching robustness, enabling it to select keypoints that are more likely to form reliable correspondences under large viewpoint and appearance changes. This property is particularly beneficial for our project, where movie frames need to be matched against real-world images captured under significantly different imaging conditions.
- **LightGlue** is a lightweight transformer-based feature matching model designed to efficiently establish accurate correspondences between local features[4]. LightGlue achieves a favorable balance between robustness and computational efficiency.This property is important in our real-time localization software which needs relatively short runtime on server.

3. Method

The system follows a front-end and back-end separated architecture. The overall workflow is shown in Figure 1. The system is described in terms of data preparation and management, front-end, and back-end components.

3.1. Data preparation and management

Due to the lack of access to the true filming locations in Zurich, ETHz CAB and HG were selected as surrogate film scenes. We captured images of these scenes to construct individual SfM models using the HLOC pipeline with DISK as the local feature extractor.

DISK jointly learns keypoint detection and description and is optimized for matching performance, providing reliable 2D-2D and 2D-3D correspondences[8]. This leads to stable SfM reconstructions under large viewpoint and appearance variations. The results of SfM models are shown in Figure 2, where both outdoor and indoor scenes exhibit good reconstruction quality.

To support efficient real-time data retrieval, the SfM

models and the precomputed poses of film frames in SfM world are organized in a hierarchical directory structure on the local PC. It allows dynamic loading of the relevant model and pose data and remains scalable if we extend to more real films and construct a larger database.

3.2. Front-end Mobile AR Application

The mobile front-end is an iOS-based AR application responsible for user interaction, data acquisition, and real-time visualization. It guides users to the target filming location and assists viewpoint alignment through AR cues.

User Interaction Flow. The interaction starts with movie frame selection and map navigation to the filming location. Once on site, users enter AR scanning mode, where directional arrows and a tetrahedron frustum guide the device toward the target pose. As users move and rotate the device, these cues update continuously based on server feedback, helping users move towards the target point.

AR Data Uploading and Synchronization. During scanning, the client continuously captures RGB images and corresponding AR session poses. Each frame, together with camera intrinsics and timestamps, is uploaded asynchronously to the server. Uploading and localization run in parallel, allowing uninterrupted user interaction during server-side processing.

Interactive AR Visualization and Capture. After receiving a valid film pose, the client updates the movie frame overlay and tetrahedron accordingly. Users can interactively adjust transparency, apply visual filters, and insert extracted movie characters. The application supports capturing composited AR photos that combine the real scene with aligned cinematic elements.

3.3. Server Architecture

The server is deployed on a PC and implemented as a Python-based application. To avoid upload queue blocking during visual localization, a multi-threaded architecture is adopted to process uploads receiving and frame localization simultaneously. The overall pipeline in server is illustrated in Fig. 3.

HLOC-based Camera Localization. We chose HLOC as the baseline for the per-frame localization because it provides accurate and robust camera pose estimation within a pre-build SfM model and it is highly modularized, which is convenient for modification and adaption. The overall localization pipeline includes disk feature extraction, features matching with Lightglue, and camera pose estimation.

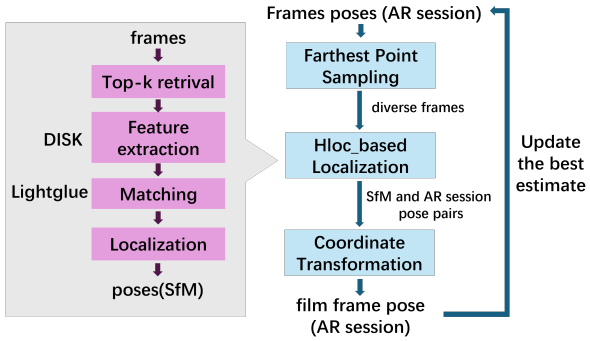


Figure 3. Server work pipeline. After farthest point sampling, selected frames are localized using modified HLOC to obtain SfM poses, which are then used to estimate the transformation matrix and map the film pose into the AR session world.

However, the original HLOC is too inefficient for frame-by-frame positioning. We made the following adjustments to improve computational speed, making it more suitable for our software:

1. **Top-K Retrieval for Matching:** As matching a query image against all images in the SfM database is computationally expensive, we perform a top-10 image retrieval step to select candidate database images after DISK feature extraction. However, the built-in retrieval methods in HLOC rely on additional global descriptor models, such as NetVLAD, which introduce extra memory usage and computational overhead due to redundant feature extraction. To address this, we construct a lightweight 128×1 global image descriptor by pooling DISK features, eliminating the need for separate global descriptor models while maintaining efficient retrieval performance.
2. **Single-image Localization Design:** PyTorch profiling showed that the original Dataloader consumed about 74% of the CUDA time. To enable per-frame processing and remove unnecessary overhead, the DataLoader and dataset traversal logic were replaced with a single forward pass per image, reducing the processing time from roughly 40 s to 25 s per frame.

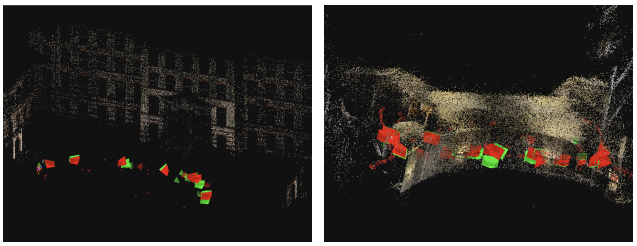
Coordinate Transformation Estimation. To align camera poses estimated in the SfM coordinate frame with users' real-time mobile AR session space, we estimate a similarity transformation $\mathrm{Sim}(3)$, consisting of scale, rotation, and translation, following the standard least-squares formulation for point set alignment [9]. This step is necessary because SfM reconstructions are defined up to an arbitrary scale, whereas ARKit/ARCore operate in a metric coordinate system.

The coordinate transformation module is integrated into the server pipeline shown in Figure 3 and runs concurrently

with frame uploading and localization. Since per-frame SfM pose queries are computationally more expensive than frame acquisition on the mobile device, the system does not attempt to use all incoming frames for transformation estimation in a very early stage. Instead, we make the following design choices to ensure robustness and efficiency:

1. **Parallel Pose Acquisition and Transformation Update:** As illustrated in Figure 3, frame uploading and SfM pose querying are handled in parallel threads. While frames are continuously uploaded from the mobile AR session, another worker localizes selected frames in the SfM model and updates the global alignment. This design prevents blocking the upload queue and allows transformation estimation to proceed asynchronously.
2. **Farthest Point Sampling for Frame Selection:** To avoid biased Sim(3) estimation caused by spatially clustered viewpoints [3], we apply Farthest Point Sampling (FPS) on AR session poses to select a geometrically diverse subset of frames. FPS ensures that pose pairs used for transformation estimation are well distributed in space, reducing centroid drift and improving the numerical stability of scale and translation estimation.
3. **Robust Similarity Transformation Estimation and Update Strategy:** We estimate the similarity transform between the AR session and SfM coordinate systems from matched pose pairs using a RANSAC-based scheme [2]. For temporal stability, avoid updating the transform on every new match and instead use a conservative policy based on geometric consistency and incremental gains. A new similarity is accepted only when (i) the number of inlier pose pairs increases over the previous solution and (ii) the re-projection or alignment RMSE remains within a specified tolerance of its prior value. These rules reject unstable updates caused by temporary localization noise or brief ARKit/ARCore drift. The transform is computed only when at least four geometrically consistent inlier poses are available, the minimum needed for a similarity, and is then progressively refined as additional frames are processed, leading to a more stable alignment between the SfM and AR frames.

Before computing the similarity transform, we first align the coordinate conventions of the AR session and the SfM reconstruction. Since ARKit/ARCore and SfM use different handedness and axis directions, especially for the camera forward axis, we apply a fixed conversion via predefined rotation matrices so that both pose sets share a common convention before Sim(3) estimation. Once the transform is obtained, we apply the inverse conversion to return the aligned film frame poses to the AR session's native coordinates, preserving compatibility with the AR rendering pipeline while maintaining a consistent global alignment.



((a)) Outdoor trajectory. ((b)) Indoor trajectory.

Figure 4. ARKit/ARCore camera poses aligned in the SfM coordinate frame. Larger opaque frustum pairs denote inlier pose pairs selected for Sim(3) estimation, while smaller translucent frustum pairs correspond to filtered-out pose pairs.

4. Experiments

We evaluate the proposed AR system under different practical conditions to assess its robustness and accuracy in real-world usage.

Specifically, experiments are conducted across different application scenes (indoor and outdoor) to analyze scene-dependent performance, and under illumination and minor appearance variations to examine the robustness of the localization pipeline. Both qualitative visualizations and quantitative metrics are reported.

4.1. Evaluation across different scenes

We evaluate the proposed system in two representative application scenarios: an outdoor scene (ETH CAB) and an indoor scene (ETH HG EO-Nord). For each scenario, ARKit/ARCore camera poses are aligned to the corresponding SfM coordinate frame using a robust Sim(3) estimation with RANSAC-based inlier selection.

Figure 4 illustrates the camera pose trajectories obtained from the proposed alignment pipeline in indoor and outdoor scenes. Red frustums correspond to ARKit/ARCore session poses transformed into the SfM coordinate frame via the estimated Sim(3) transformation, while green frustums denote reference camera poses directly retrieved from the SfM model through HLOC-based localization. Only geometrically consistent pose pairs are retained for alignment, whereas inconsistent poses are discarded by the robust filtering strategy. Compared to the indoor case, the outdoor scene shows a larger number of rejected poses, indicating increased sensitivity to long-range motion and accumulated AR tracking drift.

To further quantify the alignment accuracy, we analyze the distribution of translation and rotation deviations using box plots, as shown in Figure 5. The indoor scenario exhibits a more compact error distribution with fewer outliers, while the outdoor case shows increased variance, particularly in rotation, reflecting higher sensitivity to accumulated ARKit/ARCore drift.

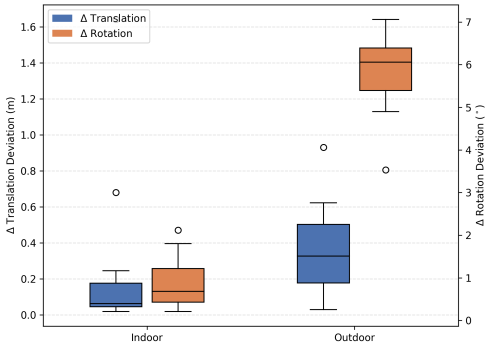


Figure 5. Distribution of translation and rotation deviations in indoor and outdoor scenes. Box plots show the median, interquartile range, and outliers (whisker length = 1.0 IQR) for both position and rotation errors.

Table 1. Mean deviation of position and rotation in both scenes.

Application Scenes	Outdoor	Indoor
Δ position (m)	0.748	0.567
Δ rotation (°)	5.01	0.84

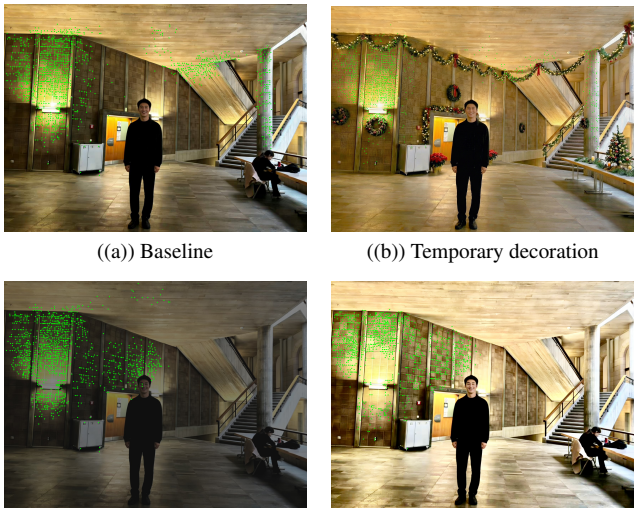
Mean translation and rotation deviations for both scenes are reported in Table 1. Consistent with the distributional analysis, the indoor scene achieves lower average errors in both position and rotation compared to the outdoor scene.

4.2. Effect of illumination and appearance changes

To assess robustness to common real-world illumination and appearance changes, such as varying exposure, lighting, or scene alterations caused by temporary objects or decorations, we focus on the SfM-based localization stage of the pipeline.

Since ARKit/ARCore session poses are obtained independently of image appearance, illumination changes mainly affect whether a query image can be reliably localized in the SfM model. Robustness is evaluated using the number of RANSAC inliers reported by the HLOC pipeline, together with the relative translation and rotation differences between images captured under special conditions and baseline images that closely match the SfM database.

Figure 6 visualizes the inlier keypoints used during PnP pose estimation for four different queries with our top-k pre-query strategy that narrows down candidate reference images and improves user-side response time. For the temporary decoration condition, the query image is synthetically modified using a generative model (Gemini) to introduce visual perturbations while preserving the overall scene geometry. The number of RANSAC inliers for the baseline, bright, dark, and appearance queries are 2891, 2221, 2779, and 1853, respectively. Despite significant appearance changes, including lighting conditions and synthetic scene modifications, a substantial number of geometrically consistent correspondences are retained. This indicates that the SfM pose query remains stable under moderate visual variations.



((a)) Baseline ((b)) Temporary decoration

((c)) Low illumination ((d)) High illumination

Figure 6. Visualization of PnP inlier keypoints for different illumination and appearance conditions. Only inlier correspondences used for pose estimation are shown.

Table 2. Pose differences under illumination and appearance variations, measured relative to the baseline query.

Condition	Δ position (m)	Δ rotation (°)
Bright illumination	1.06	1.95
Low illumination	0.69	1.68
Temporary decoration	1.03	0.78

Table 2 reports the relative pose differences with respect to the baseline query. Across all conditions, the relative pose deviations remain limited, showing that illumination and appearance changes have only a minor impact on the geometric consistency of the estimated poses.

Overall, these results demonstrate that the proposed system maintains robust image-based localization performance under common illumination variations and moderate appearance changes, which is critical for reliable on-site AR usage.

5. User Study

To assess the usability and overall experience of *AR Journey into Movies* in a realistic setting, we conducted a small user study with $N = 10$ participants (6 male, 4 female; 6 aged 18–24, 4 aged 25–34).

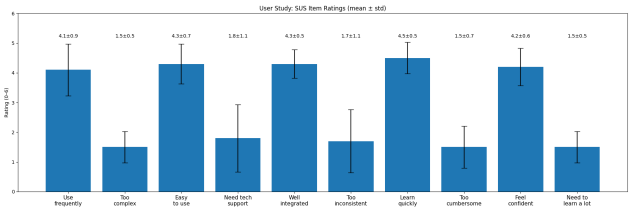


Figure 7. User study results on the 10 SUS items (mean ± std) using a 5-point Likert scale. Ratings indicate high perceived usability and learnability, with most positive items above 4 and negative items near 1–2.

Procedure. The evaluation consisted of three stages. First, participants received a brief introduction to the system goal (reproducing a target movie shot on site) and the interaction workflow. Second, they performed a hands-on task with the iOS app: scanning the environment, following the AR guidance to refine the viewpoint, and capturing a photo (optionally with AR character overlays and visual filters). Finally, participants completed a post-study questionnaire and provided free-form feedback.

Questionnaire. We used the System Usability Scale (SUS)[1] with 10 statements rated on a 5-point Likert scale (1: strongly disagree, 5: strongly agree), and added two open-ended questions to capture what users liked most and what was confusing or frustrating during alignment.

Quantitative results. Overall, the system received strong usability scores. The average SUS score was 83.5 ± 11.4 (min 65, max 100), indicating that participants generally found the app easy to learn and comfortable to use. As shown in Figure 7, the positive usability and learnability items were consistently rated high, while the negatively phrased items stayed low, suggesting low perceived complexity and low friction in the interaction flow. We observed the largest spread on the item related to needing technical support, implying that a minority of users would benefit from clearer in-app guidance during initial use.

Qualitative feedback. Participants most frequently praised the interaction and overall experience, describing the app as intuitive, with a simple GUI and engaging capture features (filters and character overlays). Several users also noted that localization felt responsive in the live workflow. The most common pain points were related to alignment clarity and expectations: some participants were unsure how to align the camera frustum or which on-screen cue to follow (e.g., confusion about the arrow guidance), and a few mentioned that scanning could take longer than expected or that the final viewpoint felt slightly off. Additional suggestions included providing clearer in-app guidance for the alignment step (e.g., step-by-step hints or a short tutorial), exposing camera/intrinsics control where appropriate, and clarifying the limitations of the approach (e.g., assumptions about planar/2D movie frames or applicability to heavily CGI scenes). These insights motivate future improvements in onboarding, alignment visualization, and robustness-oriented UI feedback (e.g., scan progress and confidence indicators).

6. Conclusion

This work presented AR Journey into Movies, a client–server mobile AR system that enables users to accurately reproduce iconic movie shots on site. By grounding a local AR session to a prebuilt SfM reconstruction through HLOC-based image localization and robust Sim(3) alignment, the system overcomes limitations of mobile AR drift and viewpoint ambiguity.

A multi-threaded server architecture allows continuous frame uploading, spatially diverse keyframe selection, and incremental pose refinement without blocking user interaction. On the front end, an interactive AR application integrates guidance cues, real-time overlay updates, and cinematic capture features, providing an intuitive and engaging user experience.

Experimental results in both indoor and outdoor scenes demonstrate that the proposed approach achieves stable and accurate alignment under varying illumination and appearance conditions. A user study further confirms the usability and practical value of the system for film tourism scenarios.

Overall, AR Journey into Movies shows that combining geometry-based localization with carefully designed AR interaction can transform static film frames into immersive, location-aware experiences. Future work includes improving onboarding guidance, extending the approach to more complex dynamic scenes, and integrating richer semantic understanding of movie content to further enhance realism and interactivity.

References

[1] John Brooke. *SUS: A 'Quick' and 'Dirty' Usability Scale*, pages 189–194. Taylor Francis, 1996. 6

[2] Martin A Fischler and Robert C Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981. 4

[3] Berthold K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(4):629–642, 1987. 4

[4] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *ICCV*, 2023. 2

[5] Linfei Pan, Dániel Baráth, Marc Pollefeys, and Johannes L. Schönberger. Global structure-from-motion revisited, 2024. 1

[6] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 2

[7] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 2

[8] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[9] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991. 3

[10] Qianqian Wang*, Yifei Zhang*, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3d perception model with persistent state. In *CVPR*, 2025. 1