StageAR: Markerless Mobile Phone Localization for AR in Live Events

Tao Jin*
Carnegie Mellon University

Shengxi Wu†
Carnegie Mellon University

Mallesham Dasari‡
Northeastern University

Kittipat Apicharttrisorn§
Nokia Bell Labs

Anthony Rowe¶
Carnegie Mellon University
Bosch Research

2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR) | 979-8-3503-7402-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/VR58804.2024.00119

Abstract

Localizing mobile phone users precisely enough to provide AR content in theaters and concert venues is extremely challenging due to dynamic staging and variable lighting. Visual markers are often disruptive in terms of aesthetics, and static pre-defined feature maps are not robust to visual changes. In this paper, we study several techniques that leverage sparse fixed infrastructure to monitor and adapt to changes in the environment at runtime to enable robust AR quality pose tracking for large audiences. Our most basic technique uses one or more fixed cameras in the environment to prune away poor feature points due to motion and lighting from a static model. For more challenging environments, we propose transmitting dynamic 3D feature maps that adapt to changes in the scene in real-time. Users with a mobile phone camera can use these maps to accurately localize across highly dynamic environments without explicit markers. We show the performance trade-offs resulting from StageAR's different reconstruction techniques, ranging from multiple stereo cameras to cameras paired with LiDAR. We evaluate each approach in our system across a wide variety of simulated and real environments at auditorium/theater scale and find that our most accurate technique can match the performance of large (1.5x1.5m) back-lit static markers without being visible to users.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Mixed / augmented reality;

1 Introduction

Mobile Augmented Reality (AR) has opened a realm of new possibilities for enhancing user experiences in entertainment venues. By leveraging mobile devices, artists have a new medium for presenting virtual content that can be choreographed with live performances and shared across the audience. Early examples of these types of performances can be seen in "Elements of Oz" [5], one of the first AR-enhanced live theater productions, and bands like Miro Shot, who recently won best XR experience at SXSW [6], with a live hybrid XR performance. We are also seeing major label bands like U2 and BTS adding in-venue live AR effects in their shows, along with a number of immersive theater startups like AR Show [7]. Beyond these entertainment use cases, one could also imagine natural extensions like AR-assisted services in concert or sporting venues for navigation, friend-finding, advertisements, concessions, etc.

The main difficulty in providing AR experiences in entertainment environments is accurately estimating the 6-Degree-of-Freedom (DOF) pose of each user's mobile device. Current mobile AR applications achieve this through visual feature-based optical registration or with specifically designed optical marker tags. Unfortunately, most vision-based techniques struggle in entertainment scenarios due to dynamic lighting and stage set changes that alter the visual event appearance. Some pioneer performances have used self-illuminated markers that are both large, intrusive, and must be carefully integrated into the aesthetics of the performance. To combat scene dynamics, recent work [17,39,55] has explored using learning-based approaches to improve the accuracy of localization results. However, these designs either rely on mobile device depth maps or need to train scene-specific neural networks. Depth maps from mobile devices can only be accurate at a short range (e.g., iPhone LiDAR), and training on specific scenes is not realistic for live events.

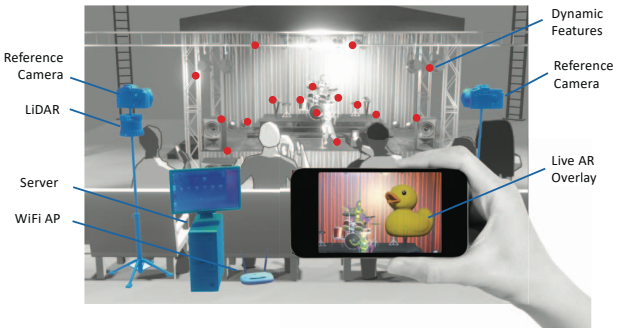


Figure 1: StageAR concept. Deploying fixed infrastructure sensors to provide accurate localization for mobile AR.

In order to truly facilitate reliable device pose estimation in entertainment settings, three key challenges need to be addressed. First, a localization technique should be robust to lighting changes and remain unaffected by significant scene shifts. Some existing systems attempt to localize based on naturally occurring visual features in the venue, which proves challenging in environments like theaters or concerts where the backdrop dramatically alters with new sets. Second, the system needs to operate from a considerable distance and over a broad viewing angle. This aspect is particularly crucial in entertainment applications to prevent them from being overly intrusive and hence, potentially distracting from the main event or disrupting a storyline. Finally, the system should provide an accurate camera pose estimate without requiring substantial device movement and operate across many users. This means supporting commodity phone hardware without requiring users to add external peripherals like IR sensors, which is a hassle, expensive, and logistically does not scale well to large crowds of people.

In this paper, we present StageAR, a system that uses several simple, but powerful approaches to enable scalable instant-on localization on standard mobile phones in highly dynamic live event environments. StageAR uses the key insight that a small amount of

*e-mail: taojin@andrew.cmu.edu
†e-mail: shengxiw@andrew.cmu.edu
‡e-mail: m.dasari@northeastern.edu, work done while at CMU
§e-mail: kittipat.apicharttrisorn@nokia-bell-labs.com
¶e-mail: agr@andrew.cmu.edu

979-8-3503-7402-5/24/$31.00 ©2024 IEEE
DOI 10.1109/VR58804.2024.00119

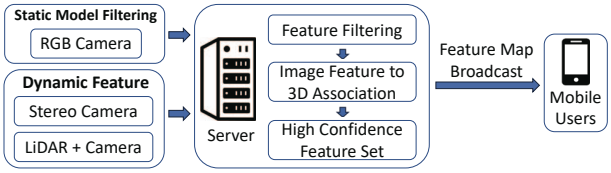


Figure 2: StageAR design includes three techniques to increase positioning accuracy. All three techniques use a centralized server to filter features and select the best ones to send to mobile users.

calibrated fixed infrastructure in the environment (e.g., cameras or depth sensors) can either filter poor visual features or even create a dynamic set of features that can be periodically broadcast to users for localization. In effect, by continuously updating features from fixed calibrated sources, the entire current state of the stage becomes the reference visual map for audience devices. However, creating these live feature maps is challenging because not only do they need to update rapidly when the scene or lighting changes, but the points need to be registered in 3D for devices to determine 6-DoF pose across a wide seating area. Figure 1 illustrates the high-level idea of StageAR with fixed infrastructure sensors in a live event setting. It is worth noting that once a device is able to localize, it can use onboard visual-inertial tracking for relative changes over time. This means that StageAR only needs to localize a device each time a session is started or resumed from a paused state.

While exploring the design space of dynamic infrastructure-aided visual localization, we found a number of solutions that trade off accuracy and reliability with cost and complexity. StageAR provides three approaches that each improve performance at the cost of additional fixed infrastructure hardware (shown in Figure 2). The first approach builds from a visual feature-based mobile phone localization technique that assumes an installer creates a 3D map of feature points while setting up the event. This is conceptually how AR Kit/Core [9, 22] performs localization on pre-recorded maps. Since dynamic components in the scene cause significant errors, StageAR uses a fixed camera at a known position to filter out poorly performing feature points. The fixed camera can then periodically broadcast the most reliable subset of features to the audience. This simple approach can work extremely well for environments that have a reasonable number of static surfaces that are not completely affected by lighting and occlusions. Our second approach goes one step further by dynamically creating new 3D feature points through the addition of multiple fixed cameras. The fixed cameras are at known positions, allowing them to perform stereo depth estimation of prominent features. The system also leverages the filtering of dynamic regions of the scene used in our first approach. This technique works better than a static pre-scanned model but suffers in terms of depth accuracy, especially if there are errors in the fixed camera calibration. Stereo correspondence also struggles in low light conditions with large baselines. To further improve system performance, we present our most advanced system that uses one or more fixed cameras in the audience along with a co-located 3D LiDAR that can directly and more accurately determine the depth of visual features detected by the cameras. In the case of LiDAR, we demonstrate a mesh reconstruction approach that improves feature point depth estimates given relatively sparse depth information. Meshes can also be used in real time to aid the occlusion of AR content based on the position of set components and actors.

Through a set of evaluations with real and simulated data, we demonstrate the effectiveness of StageAR on multiple fronts. We show it only requires a small number of fixed sensors deployed in the environment to provide accurate crowd user localization. Most importantly, it is able to operate in highly dynamic lighting and staging environments for an extended period of time where state-of-the-art methods fail.

In summary, the contributions of this paper are:

1. A set of techniques for "instant-on" phone localization in dynamic lighting and staging entertainment venues.
2. An intelligent feature filtering technique that extracts robust features based on geometry and image.
3. Sensitivity study of using various hardware in visually challenging environments.
4. Open-source implementation of StageAR.

2 Related Work

2.1 Passive Markers

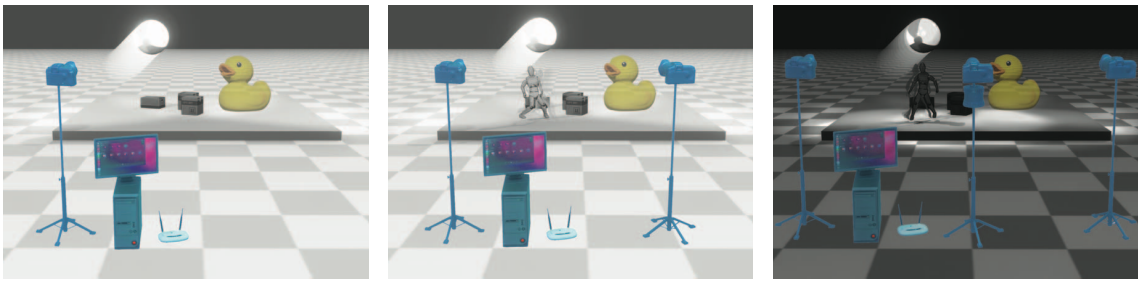
Visual markers (fiducial markers) are a commonly used solution for content registration in AR [26]. For the past few decades, there has been extensive literature in terms of different coded visual markers and optical markers, such as AprilTags [38], AR Tags [18], and ARToolKit [52]. These markers/tags can be very effective in terms of localization accuracy, low cost, and ease of use, but they have a limited range and can often be quite obtrusive. In addition, marker-based solutions are also shown to be vulnerable to security issues because they are easy to copy and spoof [47]. On the other hand, systems like Vuforia [57] can learn features from arbitrary images, thereby avoiding such issues. However, these image-based systems have restrictions on the number of tags they can support and often exhibit lower robustness compared to specialized positioning tags.

2.2 Active Markers

A key limitation of the above passive visual markers is their high reliance on lighting conditions and their ability to operate only within short ranges depending on their sizes. To address these issues, researchers have explored the use of active tags [8, 10, 12, 27, 42, 47, 58], which offer improved resilience to lighting conditions. Among these approaches, visible light communication (VLC)-based techniques [27, 42, 56] offer relatively coarse localization, work at short ranges, require very low camera exposure settings to avoid saturation (which is not ideal for AR), and require high blinking rates (1KHz+ rates) that are not compatible with commodity displays. Other active visual tags [8, 12, 47] are not practical in common mobile devices as they either require a special vision sensor [12], high camera frequencies (not yet compatible with state-of-the-art AR tools such as ARKit/ARCore) [8], support relatively short ranges [8, 58], or are computationally very expensive and susceptible to motion blur [47]. A more recent solution, FLASH [31], attempts the extreme opposite, where data is decoded from potentially a single pixel across multiple frames, avoiding the above issues. However, FLASH also suffers from obtrusiveness and requires high resolution on large displays to create a reasonably well-defined quad shape for tracking.

2.3 Model-based Approaches

Model-based methods localize users without the help of any markers or tags in the environment. For instance, researchers have demonstrated camera pose tracking methods for outdoor AR [43], which can be utilized for estimating the pose given a known 3D model of the environment and an initial camera position. Simultaneous localization and mapping (SLAM) solutions can estimate the pose relative to a known 3D model using visual or depth sensors [13, 16, 23, 36, 48]. Many modern headsets [28, 33, 51] and mobile AR platforms like ARKit [9] and ARCore [22] employ SLAM to determine the device's pose without relying on initial camera positions. However, SLAM necessitates acquiring a model of the space before determining a location, leading to increased acquisition latency, and it struggles in low-feature environments or when the scene undergoes changes. These approaches often formulate the problem of determining the camera pose through correspondences between 3D reference points and their 2D image projections as the Perspective-n-Point (PnP) problem [19]. A well-known and efficient formulation [30], which is adopted by the ORB-SLAM solver [36].



(a) One camera filter scene dynamics with a static model

(b) Stereo cameras generate dynamic feature maps and depth estimation

(c) LiDAR and cameras provide live accurate dense 3D mesh model

Figure 3: System configuration comparisons with StageAR's most basic approach of static model dynamics filtering (a), dynamic features reconstructed from stereo cameras (b), and then LiDAR + Camera based live model and feature generation (c). The full LiDAR-based system can accommodate the most dynamics in terms of obstructions and lighting.

Table 1: Taxonomy of different pose tracking techniques for AR.

Technique	Visible	Ability to Adapt to Scene Dynamics	Infrastructure Cost
Backlit Tags	Yes	Good	Low
Image Markers	Yes	Poor	None
Static Scan	No	Poor	None
Camera Filtering	No	Poor	Low
Stereo	No	Medium	Medium
LiDAR + Camera	No	Good	High

2.4 Trackers and Machine Learning Approaches

Specialized Trackers: Most of the current commercial solutions adopt beacons or trackers to localize the headsets. HTC Vive [24] employs a sweeping IR laser to accurately detect horizontal and vertical angles with high precision. However, this method necessitates powered beacons to be installed in the environment and is not optimized for long-range applications. Interestingly, the Oculus [37] utilizes blinking IR LEDs on the headset, which are detected and decoded by a stationary IR camera. However, it demands extremely tight synchronization between the LEDs and the camera, which is not feasible to achieve on mobile phones without specialized hardware.

Other approaches, such as motion capture systems [50] or RF-based systems like GPS [40], 3D RFID tracking [32], and UWB localization [14], have demonstrated increasing potential in supporting AR applications. These methods involve measuring the pose of the tag and the device directly from an external system, enabling the computation of a relative location in an image. However, they require costly infrastructure, additional hardware, and are not resilient in cluttered or high multi-path environments or over large distances.

Recent Learning-based Approach [17, 39, 55] shows success in dealing with scene dynamics in room-scale environments. However, their reliance on high-resolution depth maps or scene-specific training is not applicable to the live performance application. The user's mobile device is incapable of long-range depth sensing, and training scene-specific models does not adapt to live performance events.

Different from all of the above approaches, StageAR adopts a non-obtrusive, highly accurate, and robust instant-on phone localization by leveraging fixed sensor infrastructure support. StageAR is robust to dynamics (e.g., lighting or changing stage objects) in the environment because it uses highly confident features from external cameras. As a result, StageAR offers high accuracy pose tracking for scalable AR experiences in highly dynamic environments.

3 StageAR: System Design

Figure 3 shows StageAR's three configurations. The first (a) uses a single camera to periodically broadcast optimally selected feature points from a static model, ensuring a broad spread and avoiding blocked or dynamic erroneous features. Configurations (b) and (c) illustrate methods for capturing dynamic feature points, noting that dynamic point generation in a single-camera setup is not possible due to the need for 3D data. Our dynamic system employs either additional cameras for stereo depth or a more expensive, yet compact and less noisy, LiDAR. Table 1 displays a taxonomy of various techniques. StageAR's markerless approaches, including Camera Filtering, Stereo, and LiDAR + Camera, balance accuracy against hardware costs, offering suitable options for different scene dynamics. This section further details our feature filtering methods and dynamic 3D feature selection process.

3.1 Static Model with External Camera Filtering

StageAR's most basic approach for markerless mobile user localization is to use RGB image feature matching against a pre-scanned 3D model of the environment. The pose can be obtained by solving the PnP problem. Typically, in the face of scene dynamics, the pose estimation robustness and accuracy are improved using outlier rejection techniques, such as Random Sample Consensus (RANSAC), that search across all point pairs in hopes of minimizing error. However, when there are significant changes to the scene, the assumption of RANSAC, that the majority of the feature points are accurate, no longer holds, leading to localization failure.

StageAR tackles this by employing fixed infrastructure cameras to provide a frame of reference, filtering the moving features and improving the accuracy. This idea and its associated filtering technique serve as a foundation for all of StageAR's three configurations. First, we can select the most prominent features that are not part of a dynamic region, which we refer to as *Geometry Based Feature Filtering*. Next, instead of randomly selecting points like RANSAC, we have a *Spatially Aware Feature Selector* that picks prominent points of interest.

3.1.1 Geometry Based Feature Filtering

The most intuitive approach to perform feature filtering given a static scene model and fixed 2D camera is identifying feature motion in 2D image space. One can keep track of corresponding features across frames and filter them based on relative 2D pixel location differences. However, a critical limitation of this approach is that, due to camera perspective projection and the lack of depth information, minor image space motion may not accurately reflect minimal 3D movements. Equation 1 shows that for a 3D point (X,Y,Z) in the world coordinates, if its position on the XY plane is fixed, its projection in image (u,v) space is proportional to its depth scaled

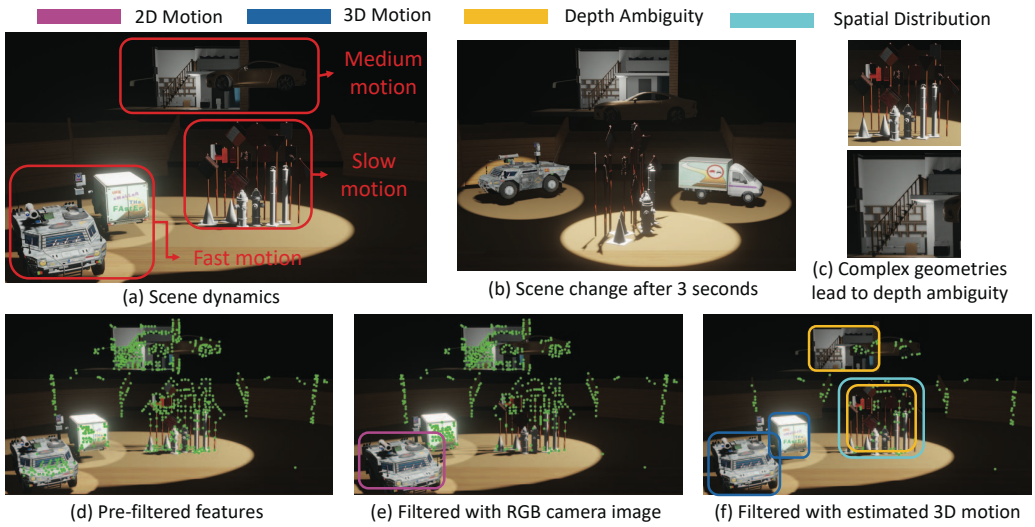


Figure 4: Impact and effectiveness of StageAR's various filtering techniques. (a) simulation scene is constructed with 3 different levels of motion. (b) scene changes after 3 seconds. (c) challenging scene geometries that can lead to inconsistent depth estimation across frames. (d) full feature set before filtering, where green points denote features extracted. (e) filtered feature from a fixed RGB camera. (f) significant improvement in feature filtering with the aid of 3D geometry information.

by w and the camera focal length f. This means that the pixel shift of the object in image (u,v) space is inversely proportional to the distance Z between the object and camera.

Figure 4(a) shows a simulation scene setup with dynamically moving objects placed at different distances with respect to the camera. After applying a fixed threshold for 2D feature filtering, the result can be seen in Figure 4(e). Compared to the full feature set in Figure 4(d), only the fast-moving features that are closest to the camera are filtered. To mitigate this, we can exploit a unique opportunity given by the combination of an RGB camera and a pre-scanned 3D model. By using the 3D model to derive the image feature's real-world coordinates, we can adaptively determine the feature filtering threshold. We show in Section 4 that this filtering alone significantly improves performance.

$$\begin{bmatrix} u & v \end{bmatrix}^T = w \begin{bmatrix} f(X/Z) & f(Y/Z) \end{bmatrix}^T \quad (1)$$

3.1.2 Spatially Aware Feature Selection

Once we pruned away the feature set using §3.1.1, we next identify the subset of features that exhibit good spatial distribution properties. We use two criteria to determine which features to broadcast.

On the Camera Side: we consider the spatial distribution of feature points across the scene to ensure the robustness of the PnP algorithm. Spatial distribution involves both the spatial spread of features in the image as well as their corresponding depth variations. An optimal feature arrangement involves a well-distributed spatial spread of feature points in 2D images at different depths, enhancing the robustness of PnP pose estimates. Conversely, having an excessive clustering of nearby features in a small region, coupled with relatively sparse features in the rest of the scene, leads to pose ambiguity [29, 34]. To achieve this, we divide the image into even block regions and calculate the percentage occupancy of blocks that have at least 1 feature point. Then, we downsample features in each block to a maximum of 5 features. Selecting the block size is highly scene-dependent; we use an 8x8 block size in our evaluation, and this is demonstrated in Figure 4f.

On the User Device Side: we prioritize frames with reduced motion and sharper images for feature extraction and matching. Unlike theaters that can afford expensive fixed infrastructure cameras with high frame rates, resolution, and large sensor sizes. Average user mobile cameras often have significantly worse image quality in the presence of high scene dynamics or user motion. Image features may appear less sharp due to the camera's shutter speed in low light conditions. For image feature extraction, sharp images offer more distinct edges, corners, and texture details, providing stronger gradients and unique features. To identify frames that favor image feature extraction, we use the Laplacian variance [49] as the criteria to determine user image sharpness and select frames that have the sharpest details. The variance threshold is calculated at the beginning of every session where users hold the mobile phone steadily. This threshold is highly dependent on lighting, camera, etc., in our experiment in §4 we used 33.56 as the threshold.

3.2 Dynamic Feature Map Generation

While employing fixed RGB cameras and static scene models (as discussed in Section 3.1) effectively identifies small-scale object motions and leverages stationary features, this approach has limitations. It struggles particularly in environments with high scene dynamics, such as performance events characterized by dramatic lighting and staging changes. These changes often lead to localization failures, as most scene features are altered under dynamic conditions. This challenge leads us to explore an essential question: How can infrastructure cameras be utilized to update 3D feature points in real time, thereby supporting more dynamic scenes?

Our first approach in dynamically creating feature maps uses a multi-camera based stereo depth estimation to obtain prominent 2D to 3D feature correspondence. The second and most advanced method combines a LiDAR sensor with the camera to achieve more accurate 3D feature coordinates, and the intermediate dense mesh created by this method can be used for effective occlusion handling when placing AR content.

3.2.1 Dynamic Features from Fixed Stereo Depth

Robust 3D Feature From Stereo Depth: With fixed stereo camera pairs, we can obtain depth maps by performing feature matching on calibrated camera pairs to produce a disparity map, which can later be processed into a depth map. Despite this seemingly promising approach to obtaining 3D features in real time, retrieving accurate

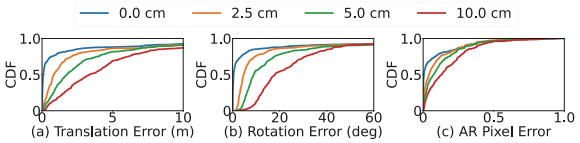


Figure 5: Impact of different calibration error on pose estimation accuracy shown as a Cumulative Distribution Function (CDF) plot. (a, b) shows the translation and rotation error, whereas (c) shows the AR Pixel Error (more in §4.2)

depth information from stereo cameras is challenging for two reasons. First, stereo depth estimation relies on using accurate and dense feature correspondence to infer and interpolate pixel depth. The nature of live events (lighting change, scene dynamics) causes noisy feature correspondences, leading to noisy stereo depth maps. Second, stereo triangulation produces noisy results due to imprecise camera calibration (both intrinsic and extrinsic). To mitigate these issues, we employ two criteria to better select reliable 3D features from stereo camera pairs.

The first criterion is to use only the depth value produced by triangulating matched features instead of generating depth for the full camera view. This provides better accuracy because it avoids using noisy values generated from interpolation between matched features. The second criterion leverages the depth estimation result to track features' 3D motion and preserves only slow-moving features. We track each feature's 3D location across consecutive frames to derive 3D velocity and only keep features that exhibit slow-speed motion. On top of this, we observed stereo depth estimation generally does not work well in situations where there is complex geometry, causing depth ambiguity. This is demonstrated in Figure 4(c); it is well known that complex scene geometry contributes to depth estimation ambiguity. Such ambiguity will cause depth readings to fluctuate significantly and present themselves as sudden 3D motion that is filtered. We can obtain a set of highly confident 2D to 3D feature correspondences through these two techniques, shown in Figure 4(f).

Impact of Stereo Camera Configuration: The key to accurately triangulating 3D feature points is the precise calibration of stereo pairs and the baseline separation between cameras. The accuracy of stereo depth estimation increases with a larger baseline between two cameras. However, as the stereo baseline increases, it becomes more difficult to match points due to the diverging camera field-of-view (FOV). This effect is observed and evaluated in Figure 11. As the stereo baseline increases, the system becomes more susceptible to camera calibration error in practice.

To understand the impact of calibration error on tracking performance and whether stereo depth estimation is feasible when considering fixed installations in theater settings, we study pose tracking accuracy under different calibration error with stereo pairs. Figure 5 shows the pose accuracy in a simulated scene using our feature selection method with varying calibration errors. We simulated a perfect pinhole camera and introduced translation and rotation errors to the camera's extrinsic pose calibration. We observed a significant drop in accuracy with increasing calibration errors. As shown in Figure 5a and b, we find that the median translation and rotation errors increase by 630% and 681%, respectively, when there is a 10 cm calibration error compared to the ideal case with no calibration error. We also quantify an AR pixel error metric (defined in §4.2) that more faithfully reflects AR experiences. Figure 5c shows that AR pixel error also increases as the calibration error increases. The result shows that a realistically well-calibrated stereo pair (2.5cm average error) can perform well for localization tasks. Ground truth surveying equipment such as Total Station [4] can be used to calibrate the infrastructure camera at millimeter accuracy.

In addition to the pose accuracy and AR pixel error, we also found a significant number of failures in localization with stereo depth-

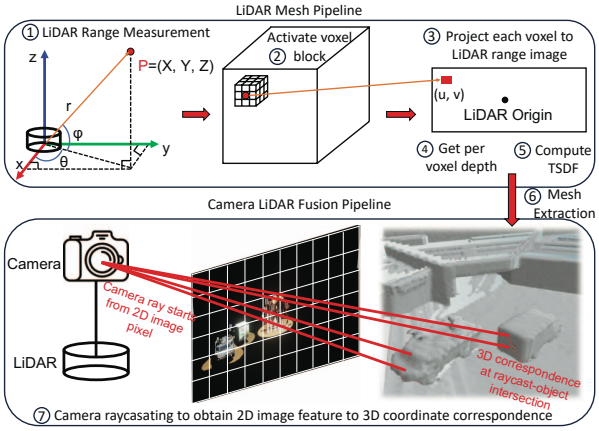


Figure 6: LiDAR and camera fusion pipeline to (1) reconstruct mesh of environment, (2) retrieve 2D feature to 3D correspondence

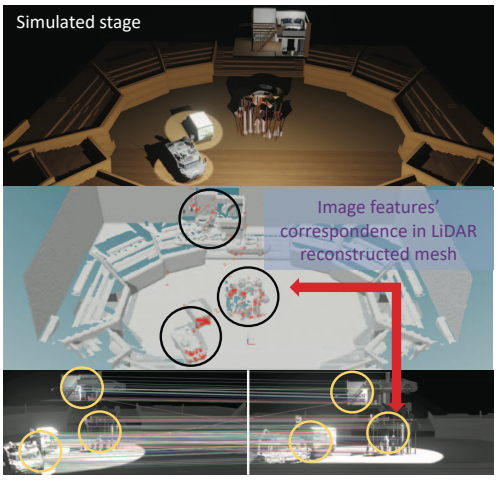


Figure 7: Visual representation `StageAR`'s LiDAR dynamic feature map generation. The top and middle figures show the stage setup and its mesh reconstructed from 3 LiDARs. The middle and bottom figure shows 2D image features to 3D correspondence.

assisted features. Even with perfect camera calibration, we found around 40% failures in the results under dynamic lighting conditions. This is mainly due to the limited number of matched features when using only two stereo cameras. Because of the limited number of cameras, only a subset of matched stereo features is selected for the final pose estimation, resulting in a failure to find a solution to localize the users. This effect is also reflected in Figure 11. A natural solution to this issue is deploying more camera sensors around the venue to obtain better coverage and provide more candidate features. Although having more cameras can decrease the number of failures, it still suffers from high pose error because of inaccurate stereo depth estimation. To address these issues, we introduce LiDAR-assisted pose tracking, which we will explain in the next section.

3.2.2 LiDAR + Camera based Pose Estimation

To improve depth sensing performance, our final system measures depth with a LiDAR paired with one of the cameras. This brings up one final challenge: LiDAR points tend to be significantly sparser in resolution compared to high-resolution cameras. While commercially available LiDARs generally provide a spatial resolution that ranges from 2048 x 128 to as low as 512 x 16 points, it is far from adequate to associate precisely feature points extracted from high-resolution camera images with 3D LiDAR point clouds. Sev-

eral methods can be used to improve the resolution of point clouds (we show the trade-off between each method in Figure 13). First, we can upsample the sparse LiDAR point cloud to camera resolution, ensuring pixel-to-pixel correspondence for each feature point. However, upsampling a low-resolution image to higher resolution does not recover the already lost high-frequency information, and interpolation at depth discontinuity causes artifacts. Second, instead of upsampling depth images, we can use a nearest-neighbor search to associate the 2D feature point to the nearest 3D point location. This results in even higher error due to incorrect depth information.

Instead of interpolating sparse point clouds or performing nearest-neighbor searches to gain data association, we employ mesh reconstruction to use dense surfaces to retrieve accurate 3D correspondence. Dense mesh reconstructed from a sparse point cloud acts as surface interpolation and avoids the issues presented in point cloud interpolation, such as artifacts at depth discontinuity.

We take a two-step solution to fuse sparse LiDAR point cloud with the camera image feature; the process is described in Figure 6. First, starting from a LiDAR range measurement, we obtain all point cloud coordinates and use them to obtain (activate) a set of neighboring voxel blocks. Once all the voxel blocks neighboring the points are retrieved, we can compute and update the implicit surface with Truncated Signed Distance Function (TSDF) representation [25], a well-known technique from computer graphics. The reconstruction of mesh from TSDF provides robustness to model accuracy due to: (1) TSDF is tolerant to noisy sensor data, as it estimates the underlying surface by taking a weighted average of multiple noisy sensor readings. (2) every triangle in the mesh is extracted considering neighboring surface information, leading to more accurate and smoother surfaces. After obtaining the mesh, we fuse camera images to find the 2D image feature to 3D surface correspondence. This fusion is achieved by marching a ray from a 2D image plane coordinate until the camera ray intersects with a surface. The resulting 3D coordinate of the ray-mesh intersection is the position of the image point in the 3D world. Figure 7 visually shows the intermediate results of this process. In the middle of the figure, we see a dense mesh reconstructed from 3 simulated LiDAR sensors (each with a resolution of 2048 x 128); the red dots show the 3D correspondence of matched image features.

3.3 Implementation

`StageAR` proposes using fixed infrastructure sensors deployed around the stage to snapshot the live environment and localize mobile users. Its implementation consists of three main components: (1) image feature extraction and matching, (2) image to depth data association, and (3) feature broadcasting.

Feature Extraction and Depth Association: To perform 2D feature extraction and matching, we use SuperPoint [15], a popular and robust neural feature extractor for 2D image feature extraction, followed by SuperGlue [45], a relibale graph neural network based feature matcher. We use OpenCV's [11] stereo triangulation for stereo dynamic feature creation to get the 2D feature's corresponding depth. For the static model and LiDAR mesh construction, we use Open3D [60] TSDF implementation and its raycasting based on the Intel Embree library [53]. After obtaining image features and their 3D locations, we solve for user pose with the EPnP algorithm [30] in OpenCV.

Feature Broadcasting: In order to deliver AR quality localization to a large crowd of users, we design `StageAR` to be scalable. After obtaining a set of feature points and their 3D locations, we broadcast them to all users, and the PnP algorithm is executed on individual user devices. Solving the PnP problem requires a calibrated camera, and we obtain this information from user devices. Camera calibration information such as intrinsic and distortion can be obtained easily in both Andriod and iOS devices through API calls [44].

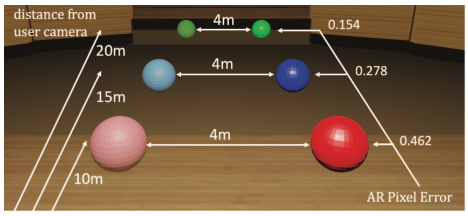


Figure 8: Virtual object with same translation error exhibit dramatically different screen-space error at different distances

4 Evaluation

4.1 Experimental Setup

We evaluate `StageAR` in both simulation and real-world scenarios. For simulation, we use Blender [20] to create live events with programmable lighting and moving objects, with a combination of photogrammetry models and synthetic models. Figure 10 shows two different complexity scenes.

We simulate two types of audience configurations for each simulated scene: (1) 9 users spread across the audience seats and (2) a large-scale setup with 600 users. 3 virtual RGB cameras and a downscaled Ouster [2] LiDAR-like depth camera are placed at the rear of the stage. To mimic real-world sensor calibration error, an average of 2.5 cm translation and 2° rotation error were introduced to all fixed cameras. For the LiDAR data, an average 2.5 cm depth noise was added according to the LiDAR hardware datasheet.

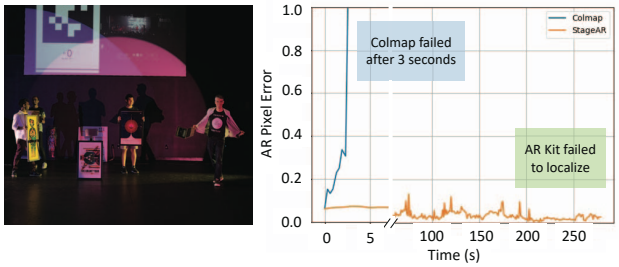
Besides extensive simulation, we conducted a real-world evaluation in a Studio theater. We set up the lighting and stage dynamics of a live performance with changing spotlights directed at the stage and projected concert videos on the stage screen. A dozen people stood and moved around the stage to replicate the movement dynamics typical of live performers. This setup is shown in Figure 9a and Figure 12.

Hardware setup: We use Sony α7RV [3] for all of our fixed RGB cameras, capturing videos at 4K 60 FPS. Ouster OS-0-128 [2] LiDARs were used to collect point clouds at 10 Hz, at a resolution of 2048 x 128. All camera intrinsics are calibrated with a checkerboard, and extrinsics are calibrated with 1.5 meter Apriltag [54]. Calibration between LiDAR and cameras is obtained using line and plane correspondences [59]. Geometric calibration accuracy is verified by reprojecting the Near-IR image returned by LiDAR to the camera frame. They are synchronized with a starting soundtrack via post-processing. The LiDAR point cloud data is processed with a Linux laptop with i9-12900H, 64GB RAM, and RTX 3080 Ti GPU. User devices are iPhone 12 Pro, shooting at 4K 60 FPS with main camera.

4.2 Evaluation Methodology

To precisely measure how localization error affects AR content overlay, we adopt AR Pixel Error (APE) metric, based on the display-proportional error metric by Miller `et al.` [35]. We also calculate translation and rotation error, where the translation is calculated as the Euclidean distance between two positions in meters, and rotation error is computed using axis-angle representation [21] in degrees. The APE metric quantifies the pixel shift on screen space to reflect the impact of camera perspective projection on AR object placement. Illustrated in Figure 8, this metric demonstrates how objects at varying depths relative to the user show different screen space pixel shifts. Specifically, objects farther from the user exhibit smaller pixel shifts compared to nearer objects, clearly linking the metric to the actual user experience, beyond just geometric errors.

APE is computed as per Equation 2, where we define ε_{xy} to be the xy component of the geometric translation error, and ε_z as the z component error. $dist$ is the Euclidean distance between the user camera and the target object, f_x represents the camera focal length (pixels), and H_x refers to the horizontal screen resolution.



(a) Theater stage setup. (b) Localization error over 4.5 mins.

Figure 9: StageAR's real-world accuracy when compared to AR Kit and Colmap. Evaluation is conducted during a performance in a theater environment. Apriltag for ground truth only.

$$APE = \frac{\varepsilon_{xy}}{|dist + \varepsilon_z|} * \frac{f_x}{H_x} \quad (2)$$

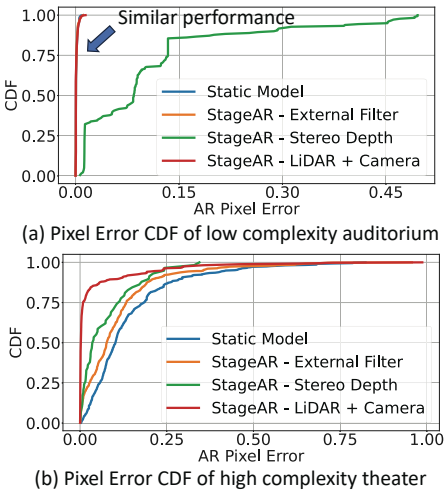
Beyond the APE metric, we also qualitatively compare StageAR by rendering AR content linked to physical features in the environment for reference.

Comparing Approaches. We compare StageAR's three configurations against Colmap, AR Kit, and a Static Model approach.

- **Colmap [46]** is a Structure-From-Motion library that serves as our baseline; we first gather around 40 camera images of the theater stage before the performance. With these images, we construct a dense point cloud using Colmap. This map is then utilized to localize user cameras during the performance, and we resize the point cloud to a real-world scale based on LiDAR measurements.
- **Apple AR Kit** Multi-User demo application [1] was custom adapted for baseline evaluation. This process involves two iPhones. The first pre-scans the stage using LiDAR and a camera to create a 3D feature set, then broadcasts this to the second iPhone. The second device uses its own visual features and LiDAR (when range allows) to perform localization against the received map or switches to local tracking if localization fails.
- **Static Model** of the environment obtained from photogrammetry, and the model is adjusted to the actual scale based on LiDAR point cloud. Based on this pre-scanned model, we perform image feature matching and solve for user pose while the actual stage is introduced with dynamic lighting and object motion.
- **StageAR - External Filter (§3.1)** is the first method introduced in StageAR using a fixed camera deployed within the stage to perform geometry and spatial feature filtering.
- **StageAR - Stereo Depth (§3.2.1)** is our second approach that creates dynamic feature maps from fixed stereo pairs. It takes snapshots of the evolving scene and extracts the most confident dynamic features to perform localization.
- **StageAR - LiDAR + Camera (§3.2.2)** leverages a LiDAR sensor deployed within the scene to reconstruct mesh in real-time. It then uses a co-located RGB camera to provide image feature matching and extracts 3D features from the mesh geometry.

4.3 Robustness Results of StageAR

Beginning with a real-world evaluation with a scene setup as shown in Figure 9a, we find that Colmap started to perform reasonably for only the first one or two seconds. After the scene setup changes, it failed to localize almost instantly (at 3 seconds). On the other hand, we used a pre-scanned map generated from the AR Kit to localize a new audience. The user never got localized at any point during the performance. This shows AR Kit does not perform well under significant scene changes. Our method performs consistently well (average 0.06 APE) throughout the 4.5-minute performance session.



(a) Pixel Error CDF of low complexity auditorium

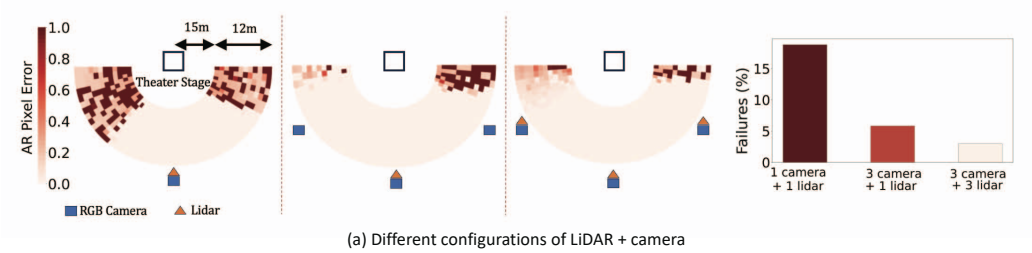
(b) Pixel Error CDF of high complexity theater

Figure 10: Simulation scene accuracy of various techniques StageAR provides. (a, b) show AR Pixel Error in low and high complexity environments. Worth noting in (a), all techniques except Stereo Depth performed similarly due to the low scene complexity.

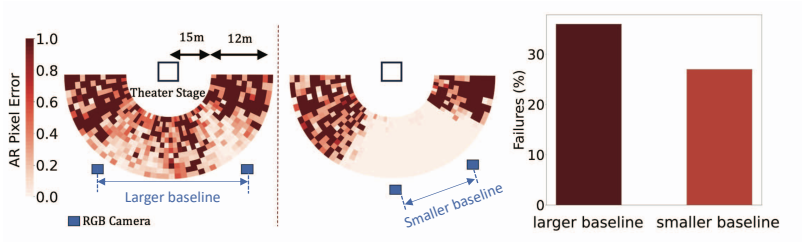
Regarding simulation environments, Figure 10 shows the cumulative distribution of APE with StageAR comparing with different strategies. We highlight the robustness of StageAR under a high complex (theater) environment in Figure 10b and compare it with a low complex (auditorium) environment in Figure 10a. As shown, most of the techniques in the low-complex environment perform similarly but differ significantly under highly complex scenes. The key difference is that most of the models have sufficient features to localize the users in a low-complex scene while they struggle to find highly confident features in more complex dynamic scenes. As we move from a static feature map (External Filter) to leverage dynamic features (Stereo) and more accurate depth estimation techniques (LiDAR mesh), the localization accuracy improves, with LiDAR being the most accurate one (300% more accurate than the static model on average). The static model approach clearly suffers due to scene dynamics. Note that the stereo depth based tacking doesn't perform well primarily due to calibration error and limited correspondence.

We also find a significant increase in failed attempts to localize users with high dynamic lighting and stage changes. We observe almost no change (both around 18%) in terms of failures for LiDAR + Camera setup and 20% increase (from 40% to 60%) in stereo depth based tracking. This is due to the fact that the stereo based solution requires taking an intersection of features that exist in at least 3 cameras (2 stereo cameras, 1 audience device), significantly limiting the number of available features. LiDAR + Camera does not suffer from the same due to 3D depth information and image features being independent of each other, eliminating the need to find common feature points across more than two cameras.

Impact of Fixed Camera Placement on Accuracy: To understand the impact of cameras and sensor location within the scene on overall accuracy, we evaluate the coverage of both stereo cameras and LiDAR + Camera. For this, we simulate a large scene with 600 users sitting in front of a theater stage (audience seats 15 - 27 m away) in a hemisphere. Figure 11 shows the spread of pose accuracy with three configurations of camera and LiDAR sensors. Figure 11 (a) demonstrates the coverage of the LiDAR + Camera system that ranges from 1 camera and 1 LiDAR up to 3 cameras and 3 LiDAR. From this study, we find that with just 1 camera and LiDAR, we can localize users seated within roughly 90° of the camera perspective. With the addition of cameras, we can cover a wider range of users with significantly lower localization failure. However, there is a



(a) Different configurations of LiDAR + camera



(b) Different configurations of stereo camera pair

Figure 11: Impact of different sensor choices and placement configurations on user experience, simulated with 600 users sitting around the theater stage. Heatmap (a) shows the distribution of AR Pixel Error for LiDAR + Camera configuration. It also shows that having more cameras will lead to fewer failures in localization. Heatmap (b) shows AR Pixel Error using two stereo cameras with different stereo baselines.

diminishing return by adding more LiDARs without changing the number of fixed cameras, albeit a significant decrease in the number of failure cases in localizing audience cameras.

We also compare the stereo depth based system with two different configurations as shown in Figure 11b: a smaller baseline (on the center and one side of the stage), and a large baseline with cameras on two sides of the stage. With stereo pairs spread across a larger baseline, the APE significantly increases compared to a smaller one. This shows that if we want to achieve similar localization accuracy for the same number of audience users across the scene, we need to deploy more cameras compared to LiDAR-based solutions. However, that only increases the localization success rate, not the accuracy of stereo-based tracking. `StageAR`'s LiDAR + Camera solution is the most accurate of all variants (despite the highest cost).

Table 2: Accuracy impact of time dilation (using features collected at an earlier timestamp to match against a later timestamp) on `StageAR`'s LiDAR + Camera approach in simulation.

Time Dilation (s)	Mean (AR Pixel Error)	Standard Deviation (AR Pixel Error)
0	0.000645	0.0000594
1	0.035	0.0105
2	0.0388	0.0228
3	0.0522	0.0281
4	0.0979	0.035

4.4 Impact of Time Dilation

In a realistic setting, audience members are seated at different locations within the scene, and their cameras take time to adjust to different lighting conditions, focus, and capture a frame with sharp detail. This means the amount of time varies between the server broadcasting the feature set and the actual time the audience device selects the best frame to perform pose estimation. We refer to this as time dilation and evaluate its impact on content overlay accuracy. Table 2 shows the APE with different time dilation values. We see that under 2 seconds of time dilation, the content drift in 2D screen space is averaged around 3.5%. As the time differences increase to 4 seconds, the screen space placement error increases to nearly 10%. This showcases the need for a dynamic feature generation and broadcasting frequency of around 2Hz (for acquisition) if the object is to achieve the best user experience. It is worth noting that onboard tracking (AR Kit, AR Core) will keep the AR content registered once the device is localized without needing to relocalize globally.

4.5 Qualitative Results from Studio Theater Live Event

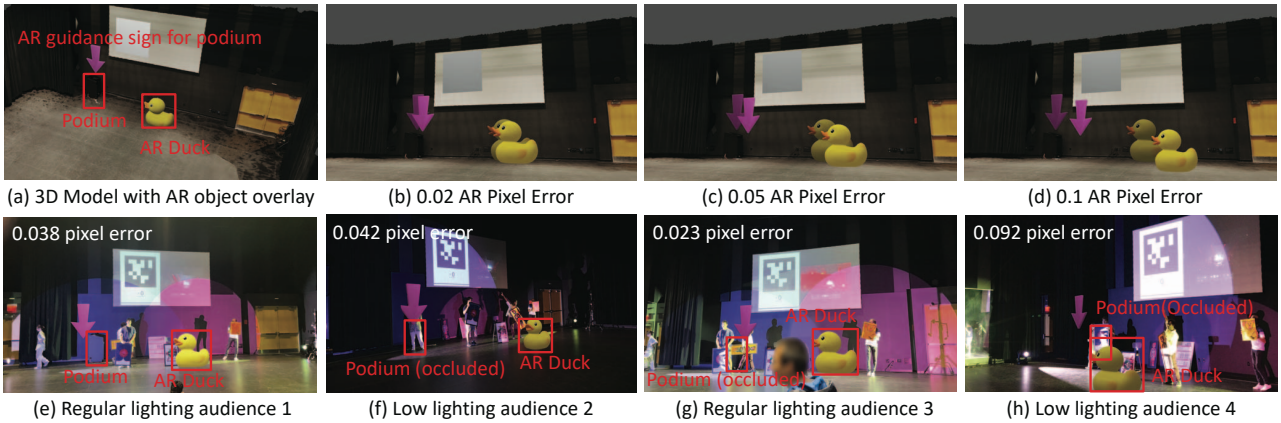
Next, we evaluate `StageAR`'s performance in real live theater environment. Figure 12 shows the screenshots of a photogrammetry model of our theater scene with an AR overlay showing demonstration content. This content would likely be authored in a virtual environment that is then overlaid into the real theater scene during a performance. The purple arrow in the scene is a visual guiding sign of where the podium is located, and a yellow duck is placed on the ground, acting as virtual content. Figure 12 (b-d) visually demonstrates the impact of different APE on what the users would see on their screen. In Figure 12 (e-h), we show `StageAR`'s performance from four audience camera perspectives, with AR content in two different lighting conditions and the corresponding APE. As shown, an audience camera with low lighting has a higher error rate compared to an audience with regular good lighting. The respective error can be gauged by referring to the pixel error impact on displaced overlay in Figure 12(b-d).

4.6 LiDAR + Camera Data Association Benchmarks

To evaluate our LiDAR + Camera system, we benchmark various point interpolation alternatives described in §3.2.2: (1) Bi-cubic interpolation, (2) KD-tree based nearest-neighbor search, and compare it with the mesh-based system. Figure 13 shows the impact of different LiDAR resolutions on APE with each of these variants. As shown on the left of Figure 13, `StageAR`'s mesh-based system outperforms Bi-cubic and KD-tree methods at the highest LiDAR depth resolution (2048x128) and performs even better with lower resolutions. More importantly, the right side of Figure 13 shows that `StageAR`'s mesh-based system has almost no failures in localization despite the stage dynamics because of its dense surface reconstruction. On the other hand, the upsampling and nearest-neighbor search methods have up to 40% failure rates in localizing the users. This demonstrates the effectiveness of our choice of LiDAR mesh reconstruction over interpolating point clouds for 3D correspondences in pose estimation.

4.7 Latency and Scalability Analysis

Computational Latency of `StageAR` is shown in Table 3 versus that of Colmap. We categorize the computation stages based on



(a) 3D Model with AR object overlay (b) 0.02 AR Pixel Error (c) 0.05 AR Pixel Error (d) 0.1 AR Pixel Error

(e) Regular lighting audience 1 (f) Low lighting audience 2 (g) Regular lighting audience 3 (h) Low lighting audience 4

Figure 12: Real world evaluation of StageAR in Studio Theater. (a) shows correct AR content overlay in a photogrammetry 3D model of the theater. (b-d) shows the perceptual difference of different levels of APE. (e-h) shows the real-world benchmark performance of our LiDAR approach from different audience angles and lighting conditions. Apriltag on the projector screen is used as the user pose ground truth.

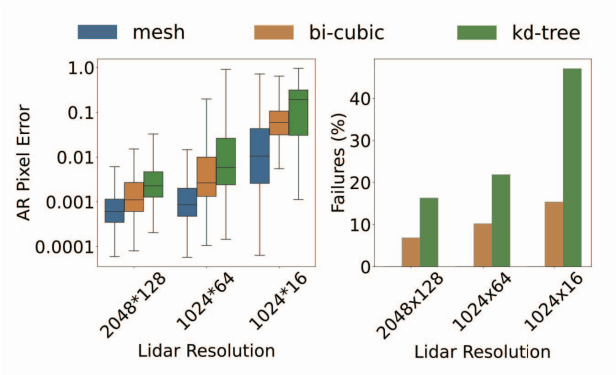


Figure 13: Different LiDAR data association techniques

where they are executed. 3D feature map generation is executed on a Linux laptop; Colmap takes nearly 1 minute to generate a sparse feature map from 40 images. This is mostly due to global bundle adjustment that refines camera registrations. In comparison, both of our methods perform extremely efficiently as they operate on a smaller and highly confident feature set that does not need global optimization. We compute the latency on the mobile side with a Jetson platform. Given the 3D feature maps are transmitted to the user device, we record the time it takes to localize a new frame. Similarly, both our methods finished with half of Colmap's time.

Scalability of StageAR is characterized as the bandwidth required for the server to broadcast features to mobile devices for feature matching. When the server transmits 1024 feature points and descriptors to one device under LZ4 or ZIP compression, the bandwidth is measured at around 1MB per frame. The number of features broadcast after StageAR feature filtering is around 200 feature points, measured at around 230 KB per frame.

Table 3: Computational latency comparison between StageAR's Stereo and LiDAR approaches and Colmap.

Method	Feature Map Generation	User Localization	
		Jetson	iPhone
Colmap	57.12 s	2.76 s	N/A
StageAR Stereo	0.32 s	1.09 s	0.98 s
StageAR LiDAR	0.14 s	1.09 s	0.98 s

5 Discussion

StageAR has a number of drawbacks in practical deployments. First, older mobile cameras struggle in low light conditions. Many of them also do not have accessible camera intrinsics. Thankfully, most modern phones with AR Kit/Core now have APIs that expose camera parameters. Another pain point is infrastructure calibration. One advantage of the LiDAR approach is that it can be pre-calibrated into a single sensor unit. It then just needs to be registered with the virtual content through software. It is also worth re-iterating that after a device is localized, it can internally track for tens of seconds to minutes. This means the system does not need to relocalize constantly but instead localize the device at the start of a new session.

In this paper, the mobile client was a custom application that could be run natively on a mobile phone or processed externally on a computer. We are working on a WebXR version of the system that doesn't require installing any software. Unfortunately, it is more challenging to implement efficient large-scale broadcast of data in web apps. This may require installers to provision more capable wireless in instrumented venues. We also envision a future where the audiences wear headsets with accommodation-supporting interactive 3D displays [41], where StageAR provides robust localization for accurate overlay of 3D interactive virtual content.

Finally, one might argue that the overall cost of our system is quite high when including equipment like LiDAR. In practice, this is well within the standard cost parameters of an industrial-grade production. For smaller-scale events, our single-camera solution, along with additional AR cue lighting, might be more than adequate.

6 Conclusion

In summary, localizing mobile users in dynamic settings like theaters can be efficiently managed using sparse infrastructure and real-time adaptive techniques. Our research demonstrates that using fixed cameras to filter out unreliable feature points due to environmental changes yields reliable results. Additionally, transmitting dynamic 3D feature point maps that adjust to real-time scene alterations offers a solid approach for more complex environments. We evaluated numerous reconstruction methods, establishing that mobile phone cameras can precisely localize in dynamic scenes without explicit markers. Our approach is shown to equal or exceed large static marker performance while being less intrusive, paving the way for scalable, high-quality AR experiences.

Acknowledgments

This work was supported by the National Science Foundation under the award CNS-1956095 and Bosch Corporate Research.

REFERENCES

[1] Creating a multiuser ar experience. https://developer.apple.com/documentation/arkit/arkit_in_ios/creating_a_multiuser_ar_experience. [Accessed Oct-2023].

[2] OS0 Ultra-wide field-of-view lidar sensor for autonomous vehicles and robotics — ouster.com. https://ouster.com/products/scanning-lidar/os0-sensor/. [Accessed 13-Jun-2023].

[3] Sony a7RV (a7R5) Mirrorless Camera — ILCE7RM5 — electronics.sony.com. https://electronics.sony.com/imaging/interchangeable-lens-cameras/full-frame/p/ilce7rm5-b. [Accessed 13-Jun-2023].

[4] Total stations for surveying, construction, and mapping. https://www.engineersupply.com/total-stations.aspx. [Accessed Oct-2023].

[5] The elements of oz. https://www.elementsofoz.com, 2015.

[6] Miro shot performance with ristband. https://vrscout.com/news/ar-technology-meets-music-at-sxsw-2022, 2022.

[7] Ar show. https://www.arshowpro.com/, 2023.

[8] K. Ahuja, S. Pareddy, R. Xiao, M. Goel, and C. Harrison. Lightanchors: Appropriating point lights for spatially-anchored augmented reality interfaces. In *UIST*. ACM, 2019.

[9] Apple. Arkit. https://developer.apple.com/arkit/, 2019.

[10] A. Ashok, M. Gruteser, N. Mandayam, J. Silva, M. Varga, and K. Dana. Challenge: Mobile optical networks through visual mimo. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom '10, p. 105–112. Association for Computing Machinery, New York, NY, USA, 2010. doi: 10.1145/1859995.1860008

[11] G. Bradski. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000.

[12] A. Censi, J. Strubel, C. Brandli, T. Delbruck, and D. Scaramuzza. Low-latency localization by active led markers tracking using a dynamic vision sensor. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 891–898, 2013. doi: 10.1109/IROS.2013.6696456

[13] K. Chen, T. Li, H.-S. Kim, D. E. Culler, and R. H. Katz. Marvel: Enabling mobile augmented reality with low energy and low latency. ACM SenSys '18, 2018.

[14] DecaWave. Decawave. https://www.decawave.com, 2019.

[15] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 224–236, 2018.

[16] A. Dhakal, X. Ran, Y. Wang, J. Chen, and K. K. Ramakrishnan. Slamshare: Visual simultaneous localization and mapping for real-time multi-user augmented reality. ACM CoNEXT '22, 2022.

[17] S. Dong, Q. Fan, H. Wang, J. Shi, L. Yi, T. Funkhouser, B. Chen, and L. J. Guibas. Robust neural routing through space partitions for camera relocalization in dynamic indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8544–8554, June 2021.

[18] M. Fiala. Artag: Fiducial marker system using digital techniques. In *CVPR*, 2005.

[19] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[20] B. Foundation. blender.org - Home of the Blender project - Free and Open 3D Creation Software — blender.org. https://www.blender.org/. [Accessed 13-Jun-2023].

[21] G. Gallego and A. Yezzi. A compact formula for the derivative of a 3-d rotation in exponential coordinates. *Journal of Mathematical Imaging and Vision*, 51:378–384, 2015.

[22] Google. Arcore. https://developers.google.com/ar/, 2019.

[23] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. *RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments*, pp. 477–491. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. doi: 10.1007/978-3-642-28572-1_33

[24] HTC. Htc vive. https://www.vive.com/us/, 2019.

[25] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pp. 559–568, 2011.

[26] H. Kato and M. Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Proceedings 2nd IEEE and ACM International Workshop on Augmented Reality (IWAR'99)*, pp. 85–94, 1999. doi: 10.1109/IWAR.1999.803809

[27] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta. Luxapose: Indoor positioning with mobile phones and visible light. In *MOBICOM*. ACM, 2014.

[28] M. Leap. Magic leap. https://www.magicleap.com/, 2019.

[29] V. Lepetit, F. Moreno-Noguer, and P. Fua. Ep n p: An accurate o (n) solution to the p n p problem. *International journal of computer vision*, 81:155–166, 2009.

[30] V. Lepetit, F. Moreno-Noguer, and P. Fua. Epnp: An accurate o(n) solution to the pnp problem. *Int. J. Comput. Vision*, 81(2):155–166, Feb. 2009. doi: 10.1007/s11263-008-0152-6

[31] E. Lu, J. Miller, N. Pereira, and A. Rowe. Flash: Video-embeddable ar anchors for live events. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 489–497. IEEE, 2021.

[32] Y. Ma, N. Selby, and F. Adib. Minding the billions: Ultra-wideband localization for deployed rfid tags. In *MobiCom*. ACM, New York, NY, USA, 2017. doi: 10.1145/3117811.3117833

[33] Microsoft. Hololens. https://www.microsoft.com/en-us/hololens/, 2019.

[34] E. M. Mikhail, J. S. Bethel, and J. C. McGlone. *Introduction to modern photogrammetry*. John Wiley & Sons, 2001.

[35] J. Miller, E. Soltanaghai, R. Duvall, J. Chen, V. Bhat, N. Pereira, and A. Rowe. Cappella: Establishing multi-user augmented reality sessions using inertial estimates and peer-to-peer ranging. In *2022 21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 428–440. IEEE, 2022.

[36] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5):1147–1163, Oct 2015. doi: 10.1109/tro.2015.2463671

[37] Oculus. Rift website, Oct 2020. Online. Accessed: 2020-10-20.

[38] E. Olson. Apriltag: A robust and flexible visual fiducial system. In *ICRA*. IEEE, 2011.

[39] M. A. Ouali, M. Bouguessa, and R. Ksantini. Graph attention network for camera relocalization on dynamic scenes. In *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10. IEEE, 2022.

[40] B. W. Parkinson, P. Enge, P. Axelrad, and J. J. Spilker Jr. *Global positioning system: Theory and applications, Volume II*. AIAA, 1996.

[41] Y. Qin, W.-Y. Chen, M. O'Toole, and A. C. Sankaranarayanan. Split-lohmann multifocal displays. *ACM Transactions on Graphics (TOG)*, 42(4):1–18, 2023.

[42] N. Rajagopal, P. Lazik, and A. Rowe. Visual light landmarks for mobile devices. In *IPSN*. IEEE, 2014.

[43] G. Reitmayr and T. W. Drummond. Going out: robust model-based tracking for outdoor augmented reality. In *2006 IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 109–118, 2006. doi: 10.1109/ISMAR.2006.297801

[44] R. P. . N. G. (RPNG). Android camera calibration. https://github.com/rpng/android-camera-calibration.

[45] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020.

[46] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[47] R. A. Sharma, A. Dongare, J. Miller, N. Wilkerson, D. Cohen, V. Sekar, P. Dutta, and A. Rowe. All that glitters: Low-power spoof-resilient optical markers for augmented reality. In *2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pp. 289–300, 2020. doi: 10.1109/IPSN48710.2020.00-27

[48] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.

[49] L. J. Van Vliet, I. T. Young, and G. L. Beckers. A nonlinear laplace

operator as edge detector in noisy images. *Computer vision, graphics, and image processing*, 45(2):167–195, 1989.
[50] Vicon. Vicon. https://www.vicon.com/, 2019.
[51] M. Vision. Meta2. https://www.metavision.com/, 2019.
[52] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Pose tracking from natural features on mobile phones. In *ISMAR*. IEEE Computer Society, 2008.
[53] I. Wald, S. Woop, C. Benthin, G. S. Johnson, and M. Ernst. Embree: a kernel framework for efficient cpu ray tracing. *ACM Transactions on Graphics (TOG)*, 33(4):1–8, 2014.
[54] J. Wang and E. Olson. Apriltag 2: Efficient and robust fiducial detection. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4193–4198. IEEE, 2016.
[55] J. Wang and Y. Qi. Deep 6-dof camera relocalization in variable and dynamic scenes by multitask learning. *Mach. Vision Appl.*, 34(3), mar 2023. doi: 10.1007/s00138-023-01388-0
[56] G. Woo, A. Lippman, and R. Raskar. Vrcodes: Unobtrusive and active visual codes for interaction by exploiting rolling shutter. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 59–64, 2012. doi: 10.1109/ISMAR.2012.6402539
[57] C. Xiao and Z. Lifeng. Implementation of mobile augmented reality based on vuforia and rawajali. In *ICSESS*. IEEE, 2014.
[58] J. J. Yang and J. A. Landay. Infoled: Augmenting led indicator lights for device positioning and communication. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, p. 175–187. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3332165.3347954
[59] L. Zhou, Z. Li, and M. Kaess. Automatic extrinsic calibration of a camera and a 3d lidar using line and plane correspondences. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5562–5569. IEEE, 2018.
[60] Q.-Y. Zhou, J. Park, and V. Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.