Asia Pacific Academy of Science Pte. Ltd.

Metaverse 2026, 7(1), 8394.
https://doi.org/10.54517/m8394

Review

Augmented Reality Pose Estimation: A Systematic Review of Visual Localization and SLAM

Zhiqian Zhang[1], Hai Huang[1,2,*], Tong Wu[1], Wenpeng Huang[1], Zhenghan Zhong[1]

[1] Beijing University of Posts and Telecommunications, Beijing 100876, China
[2] Hainan Advanced Digital Technology and Systems Laboratory, Beijing University of Posts and Telecommunications, Hainan 572426, China
*** Corresponding author:** Hai Huang, huanghai@bupt.edu.cn

CITATION

Zhang Z, Huang H, Wu T, Zhong Z. Article title. Augmented Reality Pose Estimation: A Systematic Review of Visual Localization and SLAM. 2026; 7(1): 8394.
https://doi.org/10.54517/m8394

ARTICLE INFO

Received: 6 January 2026
Accepted: 17 March 2026
Available online: 25 March 2026

Abstract: Augmented Reality (AR) has attracted increasing attention as it enhances user perception and interaction by overlaying virtual content onto the physical world. Accurate and real-time six degrees of freedom (6-DoF) pose estimation is a core requirement for reliable spatial registration. To help researchers understand recent advances and select appropriate methods for AR applications, this survey provides a systematic overview of AR pose estimation algorithms from two complementary perspectives: visual localization with prior scene knowledge and simultaneous localization and mapping (SLAM), which performs online pose estimation and mapping in unknown environments. For visual localization, pose estimation methods can be grouped into three major lines, including feature-matching-based localization, scene coordinate regression, and pose regression; their evolution toward improved robustness and scalability is also discussed. For SLAM-based pose estimation, representative approaches are organized into traditional visual SLAM (VSLAM), deep learning–enhanced SLAM, rendering-based SLAM, highlighting key design choices as well as the strengths and limitations of each category under AR constraints. In addition, evaluation metrics and commonly used benchmarks are reviewed, and reported performance of representative algorithms on selected datasets is summarized. Finally, the review summarizes the current state of AR pose estimation and outlines future research directions.

Keywords: Augmented Reality; Pose Estimation; Visual Localization; Visual Simultaneous Localization and Mapping; Absolute Pose Error; Relative Pose Error

1. Introduction

Over the past decade, AR technology has developed rapidly and has been applied in fields such as healthcare, education, and entertainment [1,2]. Common devices include head-mounted displays and smartphones, which can overlay virtual content onto the real world, bringing stronger interactivity and a more lasting immersive experience. To further enhance interaction and perception, AR/VR systems have increasingly integrated auxiliary sensing modalities such as eye-tracking in recent years [3].

Meanwhile, AR ecosystem products compatible with the devices are also gradually being refined [4]. For local tracking and mapping, Apple ARKit and Google ARCore expose World Tracking as a core capability, and describes it as being based on visual inertial odometry for continuous device pose estimation. HUAWEI AREngine similarly provides motion tracking and explains that it estimates the device pose by combining camera observations with inertial sensing. HiScene's HiAR platform advertises SLAM for 3D positioning and mapping in AR applications. Beyond local tracking, some platforms explicitly support map based localization at larger scales. ARCore Geospatial API utilizes a Visual Positioning System (VPS).

By leveraging pre-built environmental maps, it rapidly determines the device's spatial orientation. Niantic Lightship can quickly perform visual localization based on Lightship maps and enable AR navigation. Immersal platform also focuses on pre-building large-scene maps and achieving real-time positioning based on these maps. Furthermore, the EasyAR SDK integrates both SLAM and VPS capabilities, enabling AR effects in small unfamiliar environments and large known environments respectively.

Based on the AR implementation process described above, pose estimation methods can be broadly categorized into two approaches: visual localization based on prior references and prior-free SLAM. Both types of methods have been extensively studied. However, previous reviews typically discuss them separately [5–7], making comparison within the same framework difficult. Therefore, this paper attempts to integrate and analyze these methods from different perspectives. At the same time, it focuses on discussing their accuracy, robustness, and deployability in AR scenarios.

This paper is structured as follows: Section 2 introduces the geometric foundations of pose estimation and discusses key challenges in practical AR applications. Section 3 reviews visual localization-based pose estimation methods, focusing on typical implementation processes and common network architectures. Section 4 summarizes SLAM-based pose estimation methods, including geometric constraints, deep learning augmentation, and novel rendering techniques. Section 5 introduces commonly used evaluation metrics and benchmark datasets for pose estimation, and presents a comparison of representative methods on these benchmarks. Finally, Section 6 summarizes the entire paper and looks forward to potential future research directions.

2. Theoretical Fundamentals

In AR, reliable spatial registration relies on the ability to estimate a stable 6-DoF pose estimate in real time with respect to the surrounding environment, characterized by a 3D translation and a 3D rotation. Although the frameworks of pose estimation methods vary, these geometric foundations—such as coordinate system definitions, pose representations, and projection relationships—remain largely consistent.

AR systems typically employ the world coordinate system, the camera coordinate system, and the pixel coordinate system to represent the physical world, the camera space, and the image plane, respectively. The world coordinate system provides a unified anchor point and reference for the entire system. The camera coordinate system, with the camera's optical center as its origin, describes the imaging geometry. The essence of pose estimation lies in solving the rigid body transformation between the camera coordinate system and the world coordinate system, thereby establishing the constraint relationship between 2D observations and 3D points.

Mathematically, a rigid-body motion is defined by a rotation $R \in SO(3)$ and a translation $t \in \mathbb{R}^3$. Given the camera pose (R,t), a 3D point X_w is mapped to the camera frame as in Equation (1):

$$X_c = RX_w + t \tag{1}$$

where X_w denotes the world coordinates of the point, and X_c denotes its coordinates in the corresponding camera frame. **Figure 1** illustrates the various coordinate systems and the transformation relationship between them.

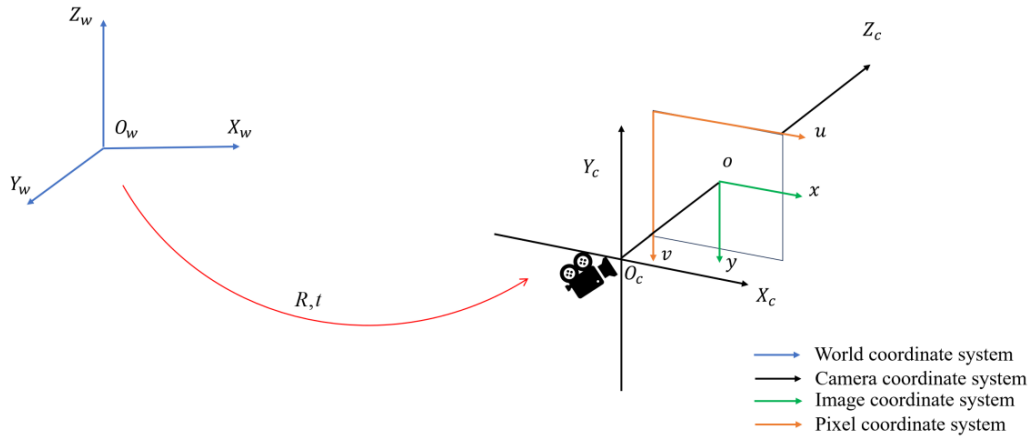

Figure 1. Coordinate frames and transformation relationship.

To facilitate the composition of transformations and subsequent derivations, rotation and translation are represented in a unified homogeneous form, as shown in Equation (2):

$$T = \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix}, \begin{bmatrix} X_c \\ 1 \end{bmatrix} = T \begin{bmatrix} X_w \\ 1 \end{bmatrix} \tag{2}$$

where $T \in SE(3)$, which denotes the pose represented as a homogeneous transformation matrix.

Pose estimation relates camera pose parameters to pixel observations, where the perspective projection model plays a central role. Given the camera intrinsic matrix K, a world point X_w and its corresponding homogeneous pixel coordinate $p = [u, v, 1]^T$ satisfy Equation (3):

$$\lambda p = KX_c = K(RX_w + t), \lambda > 0 \tag{3}$$

where λ denotes the scale factor in homogeneous coordinates, which corresponds to the depth of the 3D point in the camera coordinate system. This equation shows that the pose (R,t) determines how a 3D point is projected onto the image plane.

According to Equation (4), a pixel point p can be back-projected into 3D via the inverse intrinsic transformation, yielding a ray that passes through the camera's optical center:

$$X_c = \lambda K^{-1} p \tag{4}$$

In monocular imaging, the value of λ cannot be determined from a single image frame. Therefore, multi-view images, geometric prior data, or additional sensors are required to determine the scale and depth information of the monocular camera.

In real-time pose estimation, common solution paradigms can be broadly categorized into two types, which are often combined in practical systems. One is absolute pose estimation: when a system has access to certain 3D priors, the pose (R,t) can be solved by establishing 2D–3D correspondences, with the perspective-n-point (PnP) formulation being a representative example. The key constraint in PnP is projection consistency, meaning that the coordinates of the 3D projected points under the current pose should be as close as possible to the corresponding 2D observed coordinates. Accordingly, the reprojection error is formulated as a cost function in Equation (5), which is minimized to estimate the camera pose as shown in Equation (6).

$$L_{reproj}(R,t) = \sum_{i=1}^{N} \| p_i - \pi(K(RX_i + t)) \|^2 \tag{5}$$

$$(R^*, t^*) = \arg\min_{R,t} L_{reproj}(R,t) \tag{6}$$

where $\pi(\cdot)$ denotes the camera projection function that maps a 3D point in the

camera coordinate system to its corresponding 2D pixel coordinates, and (R^*, t^*) represents the updated camera pose. In practice, an initial estimate is usually made using an approximation method and then improved through a few rounds of updates.

Another paradigm is relative pose estimation. This method estimates the relative motion between consecutive frames and then combines it with the reference frame's pose to obtain the camera trajectory. This method is commonly referred to as visual odometry (VO). It offers high update rates and is suited to real-time operation.

Random sample consensus (RANSAC) is one of the most widely used frameworks for robust outlier rejection. This method generates candidate models through random sampling and then uses interior points to improve the estimation. It can prevent a few outliers from affecting the pose solution.

While key technologies for AR pose estimation are relatively mature, numerous unfavorable factors in real-world environments directly impact the accuracy and stability of pose estimation. For example, dynamic objects and non-rigid body changes can violate static scene assumptions, introducing more outliers and causing pose drift; changes in lighting and appearance also make relocalization more difficult. Furthermore, errors accumulate continuously in large-scale scenes and over long periods of operation. Therefore, constructing a robust and scalable pose estimation method is crucial. The following sections will further review various pose estimation methods.

3. Camera Pose Estimation via Visual Localization

Visual localization estimates the camera's 6-DOF pose by using prior scene references. Scene references can be either 3D models reconstructed using Structure from Motion (SfM) or a database of reference images containing pose and feature indices. During pose inference, the system extracts features from the query image and associates them with the scene references to form pose constraints. Based on the form of these constraints, visual localization for pose estimation can be broadly categorized into three paradigms: (i) feature-matching-based localization that builds explicit correspondences and solves pose with geometric solvers, (ii) scene coordinate regression for predicting 2D-3D constraints in scene alignment, and (iii) pose regression that directly maps images to poses. Three visual pose estimation methods and their general framework are shown in **Figure 2**.

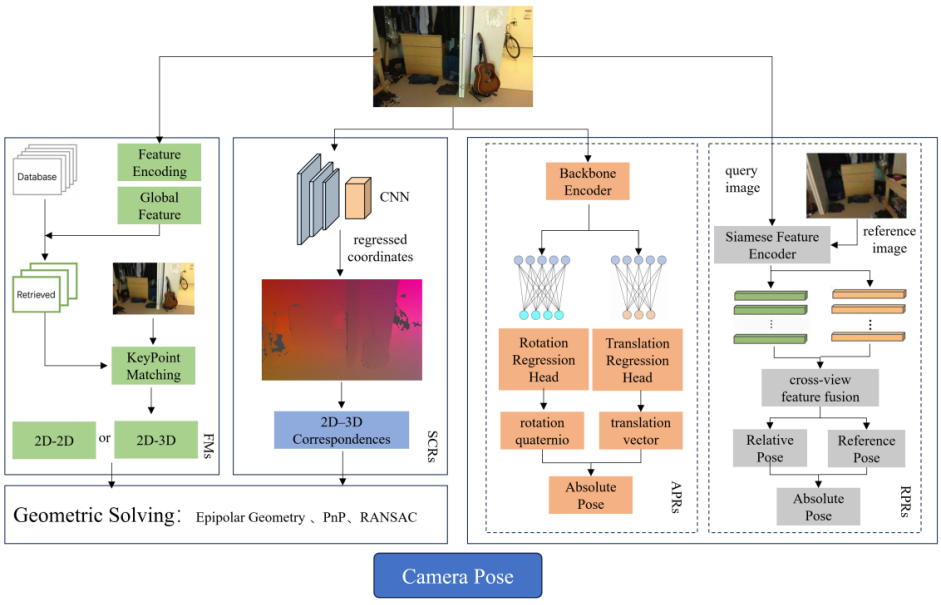

Figure 2. Framework of Visual Localization Pose Estimation Methods.

3.1. Feature-Matching-based Localization for Pose Estimation

Visual localization based on feature matching extracts local features from the query image and matches them against scene reference information to establish 2D-3D or 2D-2D correspondences, as shown in **Figure 3**. Subsequently, geometric algorithms such as the PnP solver are employed to determine camera pose. This method is simple to implement and has a high accuracy rate, but the explicit map consumes a lot of memory.

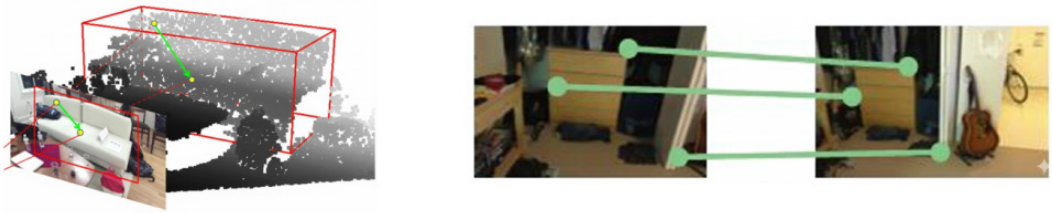

Figure 3. Feature Matching Relationships.

As research has progressed, feature-matching-based localization has evolved from early direct matching processes to more flexible and scalable systems. It introduces global retrieval to filter candidate frames from a large number of references, and then performs more reliable local matching within those candidate frames. A common approach in recent years is to combine global retrieval and local matching into a coarse-to-fine framework. This enables faster pose estimation and makes it more suitable for running on mobile devices with lower computational and storage costs.

Sattler et al. [8] propose converting large-scale 3D maps into a fine-grained visual vocabulary with superpoints. This approach enables efficient 2D–3D matching without storing descriptors for every individual 3D point. It reduces memory usage and accelerates matching. However, quantization may discard discriminative details and increase ambiguity in repetitive environments. Consequently, many pipelines adopt hierarchical strategies that combine image retrieval with feature matching. InLoc [9] employs a NetVLAD-style [10] retrieval mechanism and dense descriptors based on convolutional neural networks, then estimates candidate poses through view synthesis. This improves robustness in weakly textured indoor scenes.

HF-Net [11] predicts global descriptors for retrieval and local features for geometric verification within a single network. It enables efficient candidate selection before expensive 2D–3D matching. SuperGlue [12] improves correspondence estimation using a graph neural network. This method simultaneously considers descriptor similarity and spatial context, enabling the generation of globally consistent matching results under variations in viewpoint and lighting. HLoc [11,12] combines HF-Net with SuperGlue. It demonstrates outstanding performance across multiple benchmarks.

PixLoc [13] takes an image and a 3D model as input. By directly aligning multi-scale depth features, it transforms the camera localization problem into metric learning, enabling end-to-end estimation of camera pose. Recent studies have focused more on how to make systems lighter and easier to deploy. For example, PRAM [14] discards both global and local descriptors. It employs landmark verification instead of global searches and exhaustive matching, resulting in performance 2-3 times faster than previous hierarchical structures. LazyLoc [15] does not rely on explicit 3D maps. It estimates the camera pose of a query image solely from a database of images with known poses. The key to achieving this lies in performing motion averaging on

database-query pairs.

3.2. Scene Coordinate Regression for Pose Estimation

The core idea of Scene Coordinate Regression (SCR) is to directly predict 3D scene coordinates from input images. Given training images with known camera poses, the network learns a mapping from image pixels to 3D scene points. Specifically , the model predicts: $f:(I,p)\rightarrow(x,y,z)$, where I is the input image and p is a pixel location. During the inference phase, the network predicts scene coordinates and generates a 2D-3D correspondence map. The camera pose is then estimated with a PnP solver, typically combined with RANSAC to reject outliers and incorrect correspondences.

This method does not require pre-building an explicit map, thereby reducing memory usage. However, in practice, it faces several challenges, including high training costs, sensitivity to scene changes, and limited cross-scene generalization. To address these challenges, subsequent studies have extended the SCR framework along two directions: (i) introducing hierarchical architectures and richer contextual information to mitigate ambiguities arising from repetitive structures; (ii) redesigning scene representations or integrating renderable scene models to improve efficiency and scalability.

Early research focused on integrating geometric pose estimation methods into learnable SCRs. Brachmann et al. [16] propose a differentiable pose estimation method that incorporates RANSAC as a hypothesis sampling mechanism into SCR. The network first predicts scene coordinates for each pixel, and then uses a reprojection error-driven objective function to optimize pose estimation. Subsequently, DSAC++ [17] further improves the training method. It only allows the network to regress to the visible scene surface, making the training process more stable and reducing incorrect guesses in unseen areas. Based on this, Brachmann et al. [18] also combine a deep learning version of RANSAC with geometric and appearance priors to improve pose estimation accuracy in complex scenes. This makes the system more robust to noise, occlusion, and outliers.

However, structural ambiguity remains a key challenge in SCR, particularly in scenes with repetitive texture cues. To address this, Li et al. [19] proposed a hierarchical strategy. First, regions are ranked using global constraints. Subsequently, finer-grained local regression is performed within candidate regions. By propagating contextual information across different levels, this method enhances the consistency of pose prediction results. HSCNet++ [20] extends upon HSCNet by decomposing the regression task of a single network into classification tasks across multiple networks, and predicts pixel scene coordinates through a layer-by-layer refinement approach. This enables the training of more compact models, thereby better adapting to large-scale environments.

The generalization capability and operational speed of SCR systems constitute a significant area of research. ACE [21] decouples the repositioning network into a scene-agnostic feature backbone and a scene-specific prediction head. Encoder pre-training requires training only a regression head. Experiments demonstrate that this approach significantly accelerates scene coordinate regression. The decoupling operation also enhances the algorithm's generalization across different scenes. GLACE [22] addresses the issue of visual localization errors in large-scale scenes by introducing co-visibility. This approach leverages feature diffusion techniques and a position decoder to utilize global encoding information more effectively. Simultaneously, by integrating pre-trained global and local encodings, SCR can be

scaled more efficiently to large-scale scenes. R-ScoRe [23] reduces computational complexity during training by lowering the dimensionality of local features and employing a global covisibility encoding. It also refines the reprojection loss to enhance the accuracy of scene coordinate estimation, thereby improving the reliability of pose estimation in large-scale scenes.

Recent studies have explored geometric structure constraints and novel scene representations. Xu et al. [24] integrated keypoint detection with SCR. By introducing sparse geometric features, the method guides the algorithm to focus on more critical regions, ensuring more efficient and accurate pose estimation. Tang et al. [25] encode geometric information into learnable latent variables and employ a Transformer decoder to predict scene coordinates. This approach achieves a compressed representation of the scene, balancing accuracy with memory consumption. NeRF-Loc [26] uses the neural radiance field (NeRF) [27] instead of traditional maps to create continuous neural rendering. It also uses Transformer-based implicit 3D descriptors to match 2D query images for localization.

3.3. Camera Pose Regression

Pose regression formulates camera localization as an end-to-end learning problem. The model directly maps images (or image pairs) to 6-DOF camera poses, eliminating the need for feature matching or explicit solving of complex geometric processes during inference. Typically, this method is primarily divided into two major categories: absolute pose regression (APR) and relative pose regression (RPR). APR directly predicts the camera pose using a single image in a fixed scene coordinate system. RPR first estimates the relative transformation between the query image and one or more reference images with known poses. It then recovers the absolute pose by incorporating the poses of the reference images.

These methods do not require map construction and enable end-to-end pose estimation, making them more appealing for mobile AR. However, compared to feature-matching-based localization, these methods have weaker geometric constraints. This results in lower accuracy in pose estimation. They are also sensitive to limited training coverage and scene changes. In recent years, researchers have attempted to address these issues in various ways, including introducing stronger geometric supervision and using synthetic data to improve generalization.

3.3.1. Absolute Pose Regression

APR takes a single image as input and directly estimates the camera pose. The typical architecture of this algorithm usually consists of two parts: an encoder for feature extraction and a regression head for outputting translation and rotation.

PoseNet [28] is a representative work of this type of method. This method directly regresses camera pose from a single RGB image using high-level convolutional features. It established an early benchmark for APR algorithms. However, due to the use of a relatively simple weighted regression loss, its accuracy is usually not high. In order to solve this problem, Kendall et al. [29] introduce geometric constraints and add scene reprojection error to the loss function. This enables the network to strike an optimal balance between positional and directional errors, thereby enhancing positioning accuracy across various indoor and outdoor datasets.

Attention mechanisms have also been shown to be effective for pose regression. AtLoc [30] introduces a self-attention module based on PoseNet. This enables the network to focus more on static structures and reliable features, thereby ignoring distractions caused by moving objects and lighting variations. Attention-based

methods can further enhance pose estimation performance.

3.3.2.Relative Pose Regression

RPR is used to estimate the transformation relationship between a query image and reference images with known poses. In a typical process, reference images are first retrieved from the database, and then the relative transformation relationship between the query image and each reference image is regressed. Finally, the absolute pose is recovered by combining the relative transformation with the known poses of the reference images.

RelocNet [31] is a representative approach that uses a CNN network to learn global image descriptors. The model is trained using continuous metric learning and computes the relative pose transformation by calculating the difference between the descriptors of the query image and the reference image. AnchorNet [32] employs the same retrieval method but additionally incorporates a set of explicit anchors. This approach detects the anchor most relevant to the query frame, converting the offset relative to the reference image into a relative offset relative to the anchor. By combining this offset with the known anchor pose, the absolute pose can be recovered. CamNet [33] builds upon this foundation to construct a coarse-to-fine retrieval and regression pipeline. This method first performs coarse retrieval based on images, followed by fine-grained retrieval guided by pose offsets and subsequent optimization. Additionally, it learns separate feature embedding spaces for retrieval and pose regression to minimize interference between the two tasks. The approach achieves significant improvements in pose accuracy across both indoor and outdoor benchmarks. Reloc3r [34] further demonstrates the potential of large-scale relative pose learning. Reloc3r achieves real-time localization while maintaining high accuracy by training a relative pose regressor on approximately 8 million image pairs and fusing information from multiple frames using a lightweight moving average module. It also shows excellent generalization ability to new scenes on multiple public datasets.

Regression-based localization has often been viewed as less accurate than feature-matching-based methods [35], but recent studies indicate that the accuracy bottleneck in pose regression stems more from limited training data and biased viewpoint distribution [36]. Therefore, the fusion of geometric priors and synthetic data has demonstrated significant effectiveness in enhancing pose regression performance. LENS [37] uses the NeRF algorithm to make realistic new images from a small number of calibrated images. It greatly alleviates data scarcity and significantly reduces localization errors. With a virtually unlimited supply of training images, the accuracy of deep regression models can approach that of classical feature-based localization methods. Similarly, DFNet [38] improves robustness by combining APR with direct feature matching. Additionally, this method proposes a novel viewpoint synthesis technique incorporating exposure adaptation to address the drastic variations in outdoor lighting conditions. Furthermore, Chen et al. [39] synthesize dense new viewpoint feature maps from learned 3D feature fields and injected implicit geometric constraints into APR. By aligning rendered features with query features and iteratively optimizing the initial regression pose, this method significantly narrowed the gap between APR and structure-based pose estimation.

4. Camera Pose Estimation via SLAM

In AR devices, pose estimation often needs to run continuously in unknown or partially unknown environments. Unlike the visual localization paradigm in the

previous section, which assumes a known scene model, SLAM jointly estimates camera trajectories and builds a scene map within one framework. SLAM algorithms iteratively estimate the camera pose from sequential observations while maintaining an incrementally updated map. They further suppress drift through loop closure detection and global optimization, thereby ensuring global consistency. **Figure 4** illustrates AR VSLAM and the corresponding pose estimation methods. AR pose estimation based on SLAM can be categorized into three types: traditional VSLAM, deep learning–enhanced SLAM, and rendering-based SLAM.

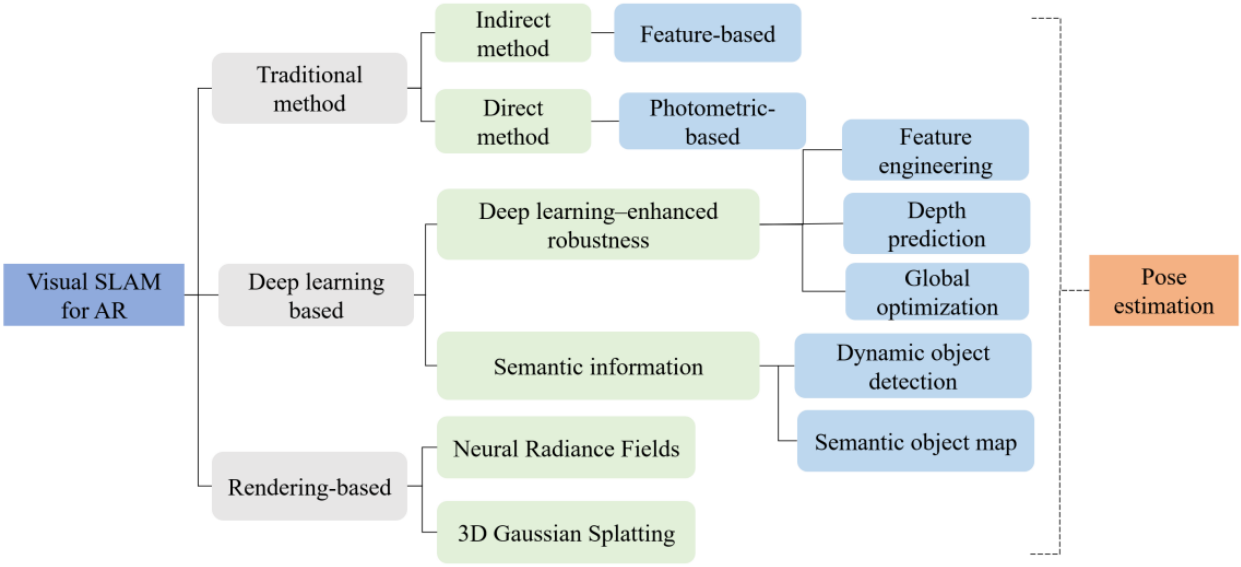

Figure 4. VSLAM pose estimation.

4.1. Traditional VSLAM for Pose Estimation

Traditional VSLAM is the foundation and core of SLAM pose estimation. A typical system includes sensor input, visual odometry, back-end optimization, loop closure detection, and mapping, as shown in **Figure 5**. The VO estimates inter-frame motion through geometric constraints. The backend module optimizes pose and map variables by integrating local estimates with closed-loop constraints. Traditional VSLAM is commonly divided into indirect and direct paradigms.

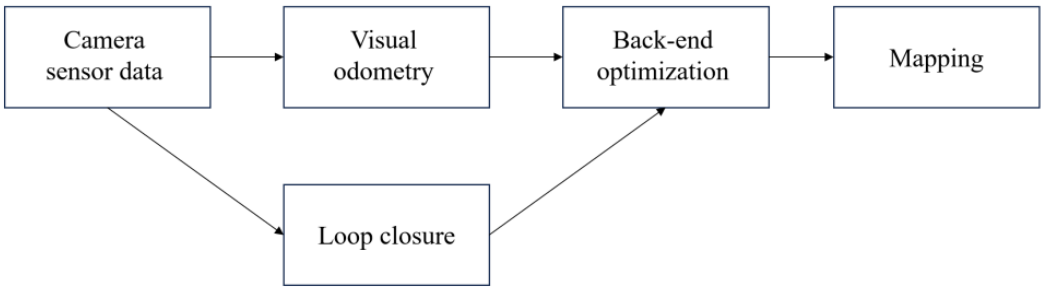

Figure 5. Classic SLAM architecture.

4.1.1. Indirect SLAM

Indirect methods recover camera pose by extracting and matching features. These methods are typically robust to illumination changes and can operate in real time, but their performance degrades significantly in texture-poor scenes or under

rapid motion.

MonoSLAM [40] is an early framework in the development of real-time monocular SLAM. It tracks a small number of feature points and then uses an Extended Kalman Filter (EKF) to simultaneously estimate both the camera pose and sparse landmarks.

PTAM [41] separates tracking and mapping into two parallel threads. Simultaneously, it employs a keyframe mechanism combined with bundle adjustment (BA) to optimize pose and map. This approach primarily addresses augmented reality requirements for small handheld devices.

ORB-SLAM [42] tracks based on ORB features and combines a bag-of-words (BoW) model [43] for relocalization. This method utilizes keyframe mechanisms to reduce computational load. Simultaneously, it employs loop closure detection to mitigate cumulative errors in large-scale scenes, thereby achieving more precise pose estimation. ORB-SLAM2 [44] further enhances the algorithm based on ORB-SLAM. Building upon the original monocular camera input, it additionally supports stereo and RGB-D inputs. Compared to the scale ambiguity inherent in monocular cameras, stereo and RGB-D cameras can directly compute true depth, thereby yielding more accurate pose estimates. ORB-SLAM3 [45] further supports inertial sensor inputs. It also introduces maximum a posteriori probability-based initialization and a multi-map mechanism. These make pose estimation more stable during rapid motion and extended continuous operation.

4.1.2. Direct SLAM

The direct method no longer explicitly extracts and matches feature points. It estimates camera pose by minimizing photometric errors on dense or semi-dense pixels. This method can utilize more pixel information, achieve tracking even in regions with less texture, and obtain a denser scene representation.

However, these methods also have significant drawbacks. They typically require more computational resources. Furthermore, because direct methods rely on the assumption of photometric consistency, changes in illumination will significantly reduce the stability of pose estimation.

Direct SLAM has evolved from dense methods to more efficient semi-dense and sparse methods, reducing computation while retaining effective photometric constraints for pose estimation. DTAM [46] is an early dense direct SLAM system that estimates camera pose by minimizing photometric errors of almost all pixels. It leverages GPU parallel computing to reconstruct dense scene maps through the synergistic combination of non-convex objective functions and regularization.

LSD-SLAM [47] simplifies the traditional dense alignment method into a semi-dense framework by introducing a keyframe mechanism. It mainly optimizes in the high gradient region of the image. This method significantly reduces the number of pixels that need to be optimized, thus significantly reducing the computational cost.

DSO [48] optimizes only a small number of pixels selected from regions of high image gradient. It further reduces the density of matching relationships, thereby enhancing the efficiency of pose estimation. This method utilizes BA to simultaneously optimize camera pose and depth information. DSO is closer to VO technology. Subsequent integration of loop closure detection and global optimization modules can suppress cumulative drift.

Building on these direct formulations, BAD SLAM [49] extends direct tracking to RGB-D and revisits bundle adjustment as the core backend. This method combines the advantages of the direct approach and bundle adjustment to enable real-time

processing of RGB-D data. By enhancing the backend module of the direct approach, it achieves high-precision collaborative optimization of camera trajectories and maps.

4.2. Deep Learning-Enhanced SLAM for Pose Estimation

Traditional VSLAM pose estimation achieves high accuracy under ideal conditions, but these algorithms often degrade in complex real-world scenarios, such as illumination changes, texture-poor environments, and motion blur. In addition, the motion of foreground objects disrupts geometric consistency. This makes the system more prone to pose drift and may even result in a complete loss of tracking capability.

4.2.1. Enhancing SLAM Robustness

In the initial stages, deep learning was primarily used to enhance the robustness of modules within SLAM systems.

D3VO [50] predicts depth maps through deep learning, thereby reducing scale uncertainty in monocular cameras. Simultaneously, it integrates depth information into DSO to achieve more accurate pose estimation.

With learned depth priors, CNN-SLAM [51] incorporates CNN-predicted dense depth into a direct monocular SLAM pipeline. Monocular depth prediction is used to initialize the scene map. Subsequently, the system updates camera pose and map variables through photometric optimization. This learned depth prior enhances the accuracy of monocular camera pose estimation.

Learning-based components can also improve long-term loop closure detection and global optimization. DeepTAM [52] draws inspiration from DTAM. However, it employs CNNs to transform pose estimation and mapping into an end-to-end learnable framework. This approach achieves pose estimation accuracy comparable to traditional RGB-D based methods using only monocular camera input. Combining learning-based prior information with geometric optimization makes localization more accurate and dense mapping better.

4.2.2. Semantic-Aware SLAM

Semantic-aware SLAM methods have evolved from early dynamic region removal to high-level semantic map construction. For example, DS-SLAM [53] extends ORB-SLAM by incorporating a semantic segmentation thread and motion consistency detection capabilities. It enables the system to more accurately locate dynamic interference regions and determines the camera pose based only on static background information.

CubeSLAM [54] simultaneously uses point features and detected objects for dual data association, jointly estimating camera motion and the trajectory of dynamic objects. The method uses static objects as anchor points and correlates SLAM map points with 2D detection results to infer the 3D motion of objects. Compared to baseline methods that only perform dynamic masking, CubeSLAM often tracks more stably in dynamic environments and constructs a more complete map. Based on this, OA-SLAM [55] further combines object-level semantics with traditional point features, improving the ability to relocalize and construct semantic maps. Recognizable objects serve as additional positioning landmarks. When tracking is lost, the system can rapidly recover its pose through object matching. Compared to traditional feature points, object-level landmarks are more robust. Therefore, this method typically remains more stable during drastic changes in viewpoint.

4.3. Rendering-based SLAM for Pose Estimation

Rendering-based SLAM methods leverage differentiable scene representations

to provide denser cues for pose estimation. Unlike traditional pipelines, they jointly optimize the camera pose by minimizing photometric residuals between rendered views and the observed frames. For AR, this can improve pose recovery under texture-poor conditions and significant appearance changes.

NeRF is a novel approach for rendering scenes. It employs an implicit neural representation method. This method relates camera viewpoints to observations through differentiable volume rendering. It can directly optimize camera poses based on photometric residuals on dense pixels. Another approach employs 3D Gaussian Splatting (3DGS) [56] as a renderable scene model. It enhances efficiency by rendering scenes using anisotropic Gaussian primitives through differentiable splatting. Beyond enabling faster novel view synthesis, 3DGS also provides explicit scene maps, making it more suitable for AR scenarios.

4.3.1. Neural implicit SLAM

Neural implicit SLAM technology has evolved from its initial global implicit field mapping to a hybrid mapping method that integrates geometric structure information. iMAP [57] is one of the first systems to use a single multilayer perceptron (MLP) to improve both the camera pose and an implicit scene model. It transforms pose tracking into a continuous optimization process for SE(3), while simultaneously optimizing the differentiable photometric residual between keyframes and the current frame. However, this algorithm encodes scene information into a MLP network, causing contamination data to easily compromise the overall scene accuracy during backpropagation.

NICE-SLAM [58] uses a hierarchical voxel grid to store local information instead of encoding the entire scene in a MLP. This method is more adaptable to large scenes and more conducive to long-term operation. However, it still requires high computational resources to ensure real-time performance. Co-SLAM [59] constructs a hybrid representation by combining coordinate encoding with sparse parameter encoding. It achieves both precise camera pose estimation and efficient memory management.

To further improve the stability of tracking, researchers have begun to introduce more geometric information into the tracking process. Vox-Fusion [60] combines voxels with neural SDF to introduce geometric constraint information. It optimizes camera pose by minimizing the geometric residual of depth observation. Compared with pure photometric targets, depth constraints are more stable, especially in areas with significant lighting changes or less texture. In addition, octree-based spatial management allows the system to expand the scene scale as needed, reducing the memory usage of scene information.

To better meet real-time demands, ESLAM [61] reduces representation and optimization overhead. This method employs an efficient encoder combined with a lightweight decoder to predict color and the Truncated Signed Distance Function (TSDF). By avoiding dense ray sampling, ESLAM has further increased the frequency of pose updates.

4.3.2. 3DGS SLAM

Early 3DGS SLAM methods primarily focused on integrating 3D Gaussian primitives into scene maps, enabling real-time pose tracking and incremental mapping through a differentiable tile rasterizer. GS-SLAM [62] builds the Gaussian map via an adaptive expansion strategy. New Gaussians are introduced when previously unseen regions are observed, while redundant Gaussians are pruned to control map

growth. This adaptive maintenance improves map quality and supports more accurate pose refinement. MonoGS [63] achieved the first monocular SLAM method based on 3DGS. This approach employs a standard dual-threaded framework for tracking and mapping. It is also compatible with stereo and RGB-D inputs. Photo-SLAM [64] combines the front-end of ORB-SLAM3 with 3DGS scene representation. This approach achieves real-time view composition and high-quality rendering while maintaining robust tracking.

SplaTAM [65] utilizes dense RGB-D constraints and differentiable rendering to improve pose estimation. As new observations are added, the system iteratively updates the camera pose and map by minimizing photometric and depth rendering errors. Based on dense photometric alignment, this method provides stable pose estimation in scenes with less texture. RD-SLAM [66] also estimates camera pose based on RGB-D input. It fuses the generalized ICP and 3DGS algorithms. This method employs a semantic keyframe strategy to progressively encode Gaussian representations in newly observed regions. Simultaneously, it introduces a regularization term to suppress excessive Gaussian growth and geometric distortion, thereby achieving a favorable balance between pose accuracy and computational speed.

In AR scenarios, a challenging problem is that the environment is constantly changing, which significantly increases the difficulty of mapping and tracking. DyGS-SLAM [67] uses semantic cues and multi-view geometric consistency to minimize the impact of moving objects on the system. This method first constructs and optimizes a baseline Gaussian map. Subsequently, it refines this map using motion compensation data to reduce interference caused by moving objects. Dy3DGS-SLAM [68] employs a monocular camera to capture the scene. It utilizes optical flow to obtain dynamic object masks. Simultaneously, it incorporates monocular depth estimation to provide supplemental spatial information. During pose estimation, both the dynamic mask and depth information are integrated into the objective function, enabling more precise pose solutions through optimization.

However, 3DGS SLAM is also prone to structural degradation because it mainly relies on photometric information for optimization. SEGS-SLAM [69] addresses this issue by introducing more explicit geometric priors. This method uses point clouds generated by ORB-SLAM3 as anchor points to initialize Gaussian primitives. Simultaneously, it defines a threshold for primitive optimization to maintain structural consistency during subsequent optimization. Experiments demonstrate that these designs can enhance the pose accuracy of monocular, stereo, and RGB-D cameras.

RTG-SLAM[70] employs a compact Gaussian parameterization scheme and an efficient real-time optimization mechanism to construct a large-scale environment real-time 3D reconstruction system. By configuring each Gaussian model as either opaque or nearly transparent, it effectively reduces redundant information, ensuring the optimization process remains sufficiently rapid. This approach achieves real-time pose estimation in large-scale scenes.

5. Evaluation Metrics and Datasets

In AR scenarios, precise alignment of virtual content with the real environment and the ability to prevent cumulative drift during long-term operation are critical importance. Based on these requirements, this section first introduces pose estimation metrics. Subsequently, it summarizes commonly used public datasets and their applicable scenarios, providing a reference basis for experimental analysis and

method optimization.

5.1. Evaluation Metrics

In AR systems, the accuracy of camera pose estimation directly affects the quality of spatial registration. Absolute Pose Error (APE) is commonly used to quantify algorithmic performance. It directly measures the error between the estimated trajectory and the actual trajectory. Since estimated trajectories and actual trajectories are typically represented in different coordinate systems, coordinate system transformation must be performed in advance. For stereo or RGB-D SLAM, where the metric scale is known, the least-squares method can be used to estimate an alignment transformation $S \in SE(3)$ to align the estimated trajectory to the ground truth. For monocular systems, where the scale is ambiguous, an alignment by a similarity transformation $S \in Sim(3)$ is required. The APE at frame i is defined as in Equation (7):

$$F_i = Q_i^{-1} S P_i \tag{7}$$

Here, P_i and Q_i denote the estimated and ground-truth poses at frame i, respectively, and S denotes the alignment transformation. APE measures the accuracy of the algorithm's pose estimation by computing the root mean square error (RMSE) of the translation residuals after alignment.

Relative Pose Error (RPE) serves as a metric for measuring the real-time drift in camera pose estimation. It calculates the difference between the true relative pose and the predicted relative pose over a fixed time interval Δ. The RPE at frame i is defined as in Equation (8):

$$E_i = \left(Q_i^{-1} Q_{i+\Delta}\right)^{-1} \left(P_i^{-1} P_{i+\Delta}\right) \tag{8}$$

Where $P_1,.....,P_n \in SE(3)$ denotes the estimated poses, $Q_1,.....,Q_n \in SE(3)$ denotes the ground-truth poses, and Δ denotes the time interval. The relationship between APE and RPE is shown in **Figure 6**.

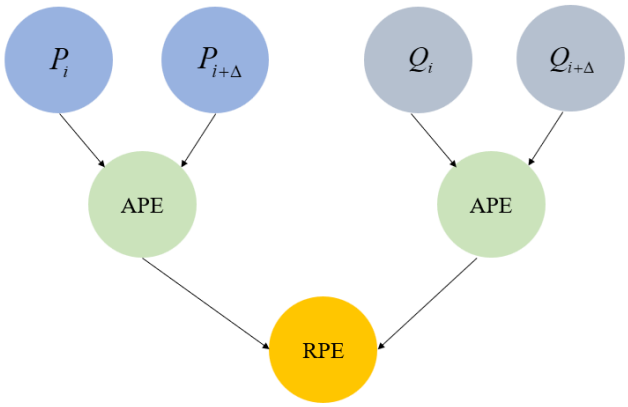

Figure 6. The relationship between APE and RPE.

5.2. Visual Localization Datasets

Commonly used public datasets for visual localization include 7Scenes, 12Scenes, InLoc, Cambridge Landmarks, Aachen Day-Night, and CMU.

For indoor localization, 7Scenes is a classic benchmark for indoor camera pose estimation, as shown in **Figure 7**. It covers multiple room-level scenes and provides RGB images, depth, and ground-truth poses from sequential captures. This dataset more closely resembles real-world, complex indoor conditions and is currently the most widely used indoor dataset. 12Scenes is also an indoor dataset and provides

multi-scene RGB-D data with ground-truth poses, which can be used to evaluate of generalization across different indoor layouts and appearance conditions. InLoc emphasizes large-scale indoor localization. It constructs a large RGB-D image database and provides separate query images. The dataset includes significant viewpoint changes, occlusions, and layout variations, making it suitable for evaluating absolute pose estimation under complex scene changes.

The Cambridge Landmarks dataset is used to evaluate outdoor localization. It captures images of five different scenes around Cambridge University using mobile devices and provides ground truth camera pose values. The data acquisition process included real-world interference factors such as pedestrians, vehicles, lighting changes, and motion blur. This dataset is currently the most widely used outdoor dataset in the field of camera pose estimation. The Aachen Day-Night dataset captures images of the same city streets during the day and night, and can be used to evaluate the robustness of algorithms to drastic changes in scene lighting. The CMU dataset considers weather and seasonal variations, captures data from both urban and suburban scenes. It provides 6-DoF ground truth pose values for each query image and 3D scene models. **Table 1** compares the median position error (cm) and orientation error (°) for several pose estimation methods tested on the 7Scenes dataset.

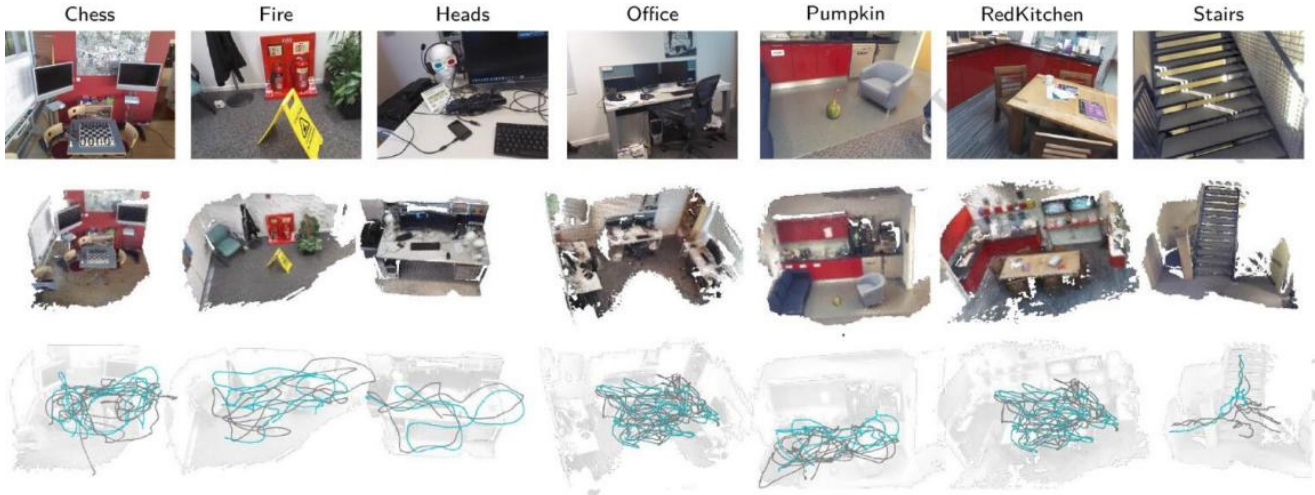

Figure 7. 7Scenes dataset.

Table 1. Pose accuracy of visual localization on the 7Scenes dataset.

Group	Method	7Scenes t(cm)/R(°)	Running speed/FPS
FMs	AS (RGB)[8]	5.14/2.46	N/A
	HLoc (RGB)[11]	3.42/1.07	0.46
	PRAM (RGB)[14]	1.0/0.3	1.1
	LazyLoc (RGB)[15]	2.83/0.95	2.5
	InLoc (RGB-D)[9]	4.14/1.38	0.7
	PixLoc(RGB)[13]	2.9/1.0	1.23
SCRs	HSCNet (RGB-D)[19]	3/0.9	N/A
	HSCNet++ (RGB)[20]	2.28/0.806	11.8
	DSAC* (RGB)[18]	2.7/1.4	55
	DSAC++ (RGB)[17]	3.57/1.1	5
	ACE (RGB)[21]	2.5/1.3	56
	NeRF-loc (RGB-D)[26]	2.3/1.3	N/A

Continuation Table:

Group	Method	7Scenes t(cm)/R(°)	Running speed/FPS
APRs	AtLoc (RGB)[30]	20/7.6	159
	LENS (RGB)[37]	8/3.0	N/A
	DFNet (RGB)[38]	12/3.7	12.2
	Posenet (RGB)[28]	24/7.9	109
	NeFeS[39]	2/0.8	6.9
RPRs	Reloc3r (RGB)[34]	38.1/0.52	24
	AnchorNet (RGB)[32]	9.0/6.74	N/A
	CamNet (RGB)[33]	4/1.7	N/A

[1] Running speed values are collected from the original publications and are provided for reference only.

5.3. Visual SLAM Datasets

Commonly used public visual SLAM datasets include KITTI, TUM RGB-D, EuRoC, Replica, and Bonn RGB-D Dynamic.

For outdoor scenarios, the KITTI dataset focuses on road environments and covers dynamic traffic participants and large-scale motion. It is suitable for evaluating trajectory consistency and scale stability over long distances, and is widely used to assess robust tracking under challenging outdoor conditions. For indoor and near-range scenarios, the TUM RGB-D dataset provides synchronized RGB-D sequences with high-precision pose, as shown in **Figure 8**. The dataset includes typical indoor environments such as offices and corridors, making it suitable for evaluating trajectory accuracy and stability under occlusion and weak texture environments. The EuRoC dataset provides challenging indoor sequences along with ground-truth trajectories and inertial measurements. It is widely used to evaluate both pure visual SLAM and visual–inertial SLAM.

The Replica is a 3D reconstruction dataset containing 18 high-quality indoor scenes. Each scene includes dense geometry, high-resolution textures, and information on glass and mirror surfaces. Furthermore, the dataset includes plane segmentation and instance segmentation data, making it suitable for pose estimation and scene generation tasks. The Bonn RGB-D Dynamic is an indoor scene dataset containing a large number of dynamic elements. It provides 24 dynamic sequences, each containing highly dynamic information and non-rigid changes, as well as 2 static sequences and ground truth camera pose values. Table 2 compares the ATE (cm) RMSE of different VSLAM pose estimation methods on the TUM RGB-D and EuRoC datasets. A short dash (-) indicates that the method is not suitable for the corresponding sensor input type.

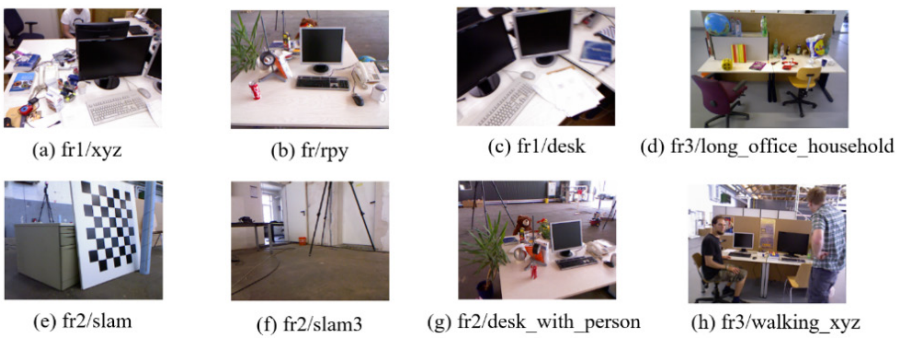

Figure 8. Example RGB-D images from the TUM RGB-D dataset, licensed under the Creative Commons Attribution 4.0 (CC BY 4.0) license .

Table 2. Pose accuracy of visual localization on the TUM RGB-D and EuRoC datasets.

Group	Method	RGB-D TUM R	Monocular TUM R	Stereo EuRoC	Running speed/FPS
Traditional SLAM	LSD-SLAM[47]	-	77.213	-	50(CPU)
	ORB-SLAM3[45]	2.196	46.004	10.907	35(CPU)
	BAD SLAM[49]	1.5	-	-	27
DL-Based	CNN-SLAM[51]	-	24.607	-	Real-time
	OA-SLAM[55]	-	80.8	-	22
Implicit Methods	Vox-Fusion[60]	10.3	-	-	2.11
	NICE-SLAM[58]	13.3	-	-	13.7
	iMAP[57]	6.1	-	-	0.64
	CoSLAM[59]	2.4	-	-	13.3
	Photo-SLAM[64]	1.870	1.539	11.203	40
	ESLAM[61]	2.0	-	-	18.11
3DGS Methods	SplaTAM[65]	4.215	-	-	1.7
	MonoGS[63]	1.502	4.009	49.241	3
	GS-SLAM[62]	3.7	-	-	8.34
	RTG-SLAM[70]	0.985	-	-	16.28
	SEGS-SLAM[69]	1.528	1.505	7.462	17

[1] Running speed values are collected from the original publications and are provided for reference only.

6. Conclusion

This paper systematically reviews existing AR pose estimation methods and classifies them into two main categories: (1) visual localization-based approaches that estimate pose using prior scene references; and (2) SLAM-based approaches that achieve continuous pose tracking without requiring pre-constructed reference maps. This paper further analyzes the implementation pipelines of different methods. Additionally, it compares the pose estimation performance of these two representative methods across different datasets.

Through research and analysis of existing methods, the challenge of AR pose estimation is not only to reduce accuracy errors, but also to ensure its stability, real-time performance, and generalization ability in practical applications. Especially under the limited computing power of mobile devices, how to achieve accurate and low-latency pose estimation in complex environments remains a problem that requires continuous exploration.

Beyond computational capability, power consumption and energy efficiency impose additional constraints on practical AR deployment. Since most AR systems operate on mobile and wearable devices such as AR glasses, the computing power-to-power consumption ratio becomes a critical design consideration. Although recent deep learning-based and rendering-based approaches have demonstrated remarkable accuracy improvements, their high computational and memory demands often lead to increased power consumption and thermal pressure, limiting real-time operation on resource-constrained hardware. This challenge highlights a key gap between algorithmic performance and deployability, emphasizing the need for hardware-aware algorithm design, lightweight models, and efficiency-oriented optimization strategies in future AR pose estimation systems.

Recently, pose estimation methods have shown a clear trend toward hybridization, with both visual localization and SLAM approaches evolving beyond

single-technique pipelines. Within each paradigm, multiple strategies are increasingly integrated to exploit complementary strengths, leading to improved robustness and stability in complex or degraded environments.

In visual localization pose estimation, the boundary between feature matching-based and regression-based methods is gradually blurring. Researchers are increasingly focusing on combining structured geometry with regression learning. By using geometric information as prior constraints and combining it with deep learning for robust feature matching and retrieval, pose estimation performance can be significantly improved. In SLAM pose estimation, state-of-the-art algorithms combine geometry-based and rendering-based hybrid representations to construct scenes. Geometry-based representations ensure stable spatial structures, while rendering-based representations fully utilize scene information. Furthermore, due to the differentiability of novel rendering methods, SLAM systems are evolving towards end-to-end differentiable frameworks.

In the future, with the continuous improvement of algorithmic frameworks and device computing power, AR pose estimation methods will evolve towards hybrid pipelines that couple geometric constraints with differentiable optimization while emphasizing computational efficiency and energy-aware design. By exploiting multiple complementary cues, AR systems can achieve high-precision and long-term consistent spatial registration under resource-constrained settings, enabling immersive interaction in large-scale and complex environments.

Beyond camera-centric pose estimation, recent AR systems have increasingly incorporated eye-tracking technologies to capture users' gaze behavior and visual attention, which is particularly relevant for perceptually optimized rendering and interaction design. Although eye-tracking does not directly estimate the 6-DoF pose of the AR device or camera, it provides complementary cues about user intention and visual focus. Recent advances in eye-tracking hardware and gaze estimation algorithms indicate a growing trend toward attention-aware AR systems. Exploring the integration of gaze information with geometric pose estimation pipelines may represent a promising future research direction.

Author contributions: Conceptualization, Z. Zhang; methodology, Z. Zhang and H.H.; software, Z. Zhang; validation, Z. Zhang and H.H.; formal analysis, Z. Zhang; investigation, Z. Zhong and W.H.; resources, Z. Zhang; data curation, Z. Zhong and W.H.; writing—original draft preparation, Z. Zhang and H.H.; writing—review and editing, T.W. and W.H.; visualization, Z. Zhang; supervision, H.H.; project administration, T.W.; funding acquisition, H.H. and T.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflict of interest: The authors declare no conflict of interest.

References

1. Li G, Luo H, Chen D, Wang P, Yin X, Zhang J. Augmented Reality in Higher Education: A Systematic Review and Meta-Analysis of the Literature from 2000 to 2023. Education Sciences. 2025; 15(6):678. https://doi.org/10.3390/educsci15060678
2. Wei, W., Gao, J., Luo, X. et al. A scoping review of the applications of augmented reality in nursing. BMC Nurs 24, 1105 (2025). https://doi.org/10.1186/s12912-025-03750-1
3. Lin, L., Wu, Z., Lu, Y., Chen, Z., & Guo, W. Lightweight deep learning with multi-scale feature fusion for high-precision and low-latency eye tracking[J]. Displays, 2026, 91: 103260. https://doi.org/10.1016/j.displa.2025.103260
4. Syed TA, Siddiqui MS, Abdullah HB, Jan S, Namoun A, Alzahrani A, Nadeem A, Alkhodre AB. In-Depth Review of

Augmented Reality: Tracking Technologies, Development Tools, AR Displays, Collaborative AR, and Security Concerns. *Sensors*. 2023; 23(1):146. https://doi.org/10.3390/s23010146

5. Sheng X , Mao S , Yan Y ,et al.Review on SLAM algorithms for Augmented Reality[J].Displays, 2024, 84:102806. https://doi.org/10.1016/j.displa.2024.102806
6. Xu M , Wang Y ,Bin XuJun ZhangJian RenZhao HuangStefan PosladPengfei Xu.A critical analysis of image-based camera pose estimation techniques[J].Neurocomputing, 2024, 570(Feb.14):127125.1-127125.27. https://doi.org/10.1016/j.neucom.2023.127125
7. Chandio Y , Selialia K , Degol J ,et al.Lost in Tracking Translation: A Comprehensive Analysis of Visual SLAM in Human-Centered XR and IoT Ecosystems[J]. 2024. https://doi.org/10.48550/arXiv.2411.07146
8. Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient &effective prioritized matching for large-scale image-based localization. IEEE transactions on pattern analysis and ma chine intelligence, 39(9):1744–1756, 2016. https://doi.org/10.1109/TPAMI.2016.2611662
9. Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Ak ihiko Torii. Inloc: Indoor visual localization with dense matching and view synthesis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7199–7209, 2018. https://doi.org/10.1109/CVPR.2018.00752
10. R. Arandjelović, P. Gronat, A. Torii, T. Pajdla and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437-1451, 1 June 2018, doi: 10.1109/TPAMI.2017.2711011. https://doi.org/10.1109/TPAMI.2017.2711011
11. Sarlin, Paul-Edouard et al. “From Coarse to Fine: Robust Hierarchical Localization at Large Scale.” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2018): 12708-12717. https://doi.org/10.1109/CVPR.2019.01300
12. Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4938–4947, 2020. https://doi.org/10.1109/CVPR42600.2020.00499
13. Paul-Edouard Sarlin, Ajaykumar Unagar, M˚ans Larsson, Hugo Germain, Carl Toft, Victor Larsson, Marc Pollefeys, Vincent Lepetit, Lars Hammarstrand, Fredrik Kahl, and Torsten Sattler. Back to the Feature: Learning Robust Cam era Localization from Pixels to Pose. In CVPR, 2021. https://doi.org/10.48550/arXiv.2103.09213
14. Fei Xue and Ignas Budvytis and Roberto Cipolla. PRAM: Place Recognition Anywhere Model for Efficient Visual Localization. arXiv preprint arXiv:2404.0778, 2024. https://doi.org/10.48550/arXiv.2404.07785
15. Siyan Dong, Shaohui Liu, Hengkai Guo, Baoquan Chen, and Marc Pollefeys. Lazy visual localization via motion averaging. arXiv preprint arXiv:2307.09981, 2023. https://doi.org/10.48550/arXiv.2307.09981
16. E. Brachmann *et al*., "DSAC — Differentiable RANSAC for Camera Localization," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 2492-2500. https://doi.org/10.1109/CVPR.2017.267
17. E. Brachmann and C. Rother, "Learning Less is More - 6D Camera Localization via 3D Surface Regression," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 4654-4662. https://doi.org/10.1109/CVPR.2018.00489
18. E. Brachmann and C. Rother, “Visual camera re-localization from rgb and rgb-d images using dsac,” TPAMI, vol. 44, no. 9, pp. 5847–5865, 2022. https://doi.org/10.1109/TPAMI.2021.3070754
19. X. Li, S. Wang, Y. Zhao, J. Verbeek, and J. Kannala, “Hierarchical scene coordinate classification and regression for visual localization,” in CVPR, 2020. https://doi.org/10.1109/CVPR42600.2020.01200
20. Shuzhe Wang, Zakaria Laskar, Iaroslav Melekhov, Xiaotian Li, Yi Zhao, Giorgos Tolias, and Juho Kannala. Hscnet++: Hierarchical scene coordinate classification and regression for visual localization with transformer. International Jour nal of Computer Vision, pages 1–21, 2024. https://doi.org/10.48550/arXiv.2305.03595
21. Eric Brachmann, Tommaso Cavallari, and Victor Adrian Prisacariu. Accelerated coordinate encoding: Learning to relocalize in minutes using rgb and poses. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pat tern Recognition, pages 5044–5053, 2023. https://doi.org/10.48550/arXiv.2305.14059
22. F. Wang, X. Jiang, S. Galliani, C. Vogel and M. Pollefeys, "GLACE: Global Local Accelerated Coordinate Encoding," *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2024, pp. 5819-5828, doi: 10.1109/CVPR52733.2024.02037. https://doi.org/10.1109/CVPR52733.2024.02037

23. Jiang, Xudong et al. “R-SCoRe: Revisiting Scene Coordinate Regression for Robust Large-Scale Visual Localization.” *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2025): 11536-11546. https://doi.org/10.1109/CVPR52734.2025.01077
24. K. Xu, Z. Jiang, H. Cao, S. Yuan, C. Wang and L. Xie, "Enhancing Scene Coordinate Regression With Efficient Keypoint Detection and Sequential Information," in *IEEE Robotics and Automation Letters*, vol. 10, no. 10, pp. 9932-9939, Oct. 2025, doi: 10.1109/LRA.2025.3598670. https://doi.org/10.1109/LRA.2025.3598670
25. S. Tang, S. Tang, A. Tagliasacchi, P. Tan and Y. Furukawa, "NeuMap: Neural Coordinate Mapping by Auto-Transdecoder for Camera Localization," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Vancouver, BC, Canada, 2023, pp. 929-939, doi: 10.1109/CVPR52729.2023.00096. https://doi.org/10.1109/cvpr52729.2023.00096
26. J. Liu, Q. Nie, Y. Liu, and C. Wang, “Nerf-loc: Visual localization with conditional neural radiance field,” in ICRA, 2023. https://doi.org/10.48550/arXiv.2304.07979
27. B. Mildenhall, P.P. Srinivasan, M. Tancik, J.T. Barron, R. Ramamoorthi, R. Ng, Nerf: Representing scenes as neural radiance fields for view synthesis, Commun. ACM 65 (1) (2021) 99–106. https://doi.org/10.48550/arXiv.2003.08934
28. A. Kendall, M. Grimes, R. Cipoll. “PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization.” *2015 IEEE International Conference on Computer Vision (ICCV)* (2015): 2938-2946. https://doi.org/10.1109/ICCV.2015.336
29. Kendall, Alex and Roberto Cipolla. “Geometric Loss Functions for Camera Pose Regression with Deep Learning.” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017): 6555-6564. https://doi.org/10.1109/CVPR.2017.694
30. Wang, Bing et al. “AtLoc: Attention Guided Camera Localization.” *ArXiv* abs/1909.03557 (2019): n. pag. https://doi.org/10.1609/aaai.v34i06.6608
31. Balntas, Vassileios et al. “RelocNet: Continuous Metric Learning Relocalisation Using Neural Nets.” *European Conference on Computer Vision* (2018). https://doi.org/10.1007/978-3-030-01264-9_46
32. Soham Saha, Girish Varma, and CV Jawahar. Improved visual relocalization by discovering anchor points. arXiv preprint arXiv:1811.04370, 2018. https://doi.org/10.48550/arXiv.1811.04370
33. Ding, Mingyu et al. “CamNet: Coarse-to-Fine Retrieval for Camera Re-Localization.” *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019): 2871-2880. https://doi.org/10.1109/ICCV.2019.00296
34. Dong, Siyan et al. “Reloc3r: Large-Scale Training of Relative Camera Pose Regression for Generalizable, Fast, and Accurate Visual Localization.” *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024): 16739-16752. https://doi.org/10.1109/CVPR52734.2025.01560
35. Sattler, Torsten et al. “Understanding the Limitations of CNN-Based Absolute Camera Pose Regression.” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019): 3297-3307. https://doi.org/10.1109/CVPR.2019.00342
36. Chen, Changhao et al. “Deep Learning for Visual Localization and Mapping: A Survey.” *IEEE Transactions on Neural Networks and Learning Systems* 35 (2023): 17000-17020. https://doi.org/10.1109/tnnls.2023.3309809/mm1
37. Moreau, Arthur et al. “LENS: Localization enhanced by NeRF synthesis.” *Conference on Robot Learning* (2021). https://doi.org/10.48550/arXiv.2110.06558
38. Chen, Shuai et al. “DFNet: Enhance Absolute Pose Regression with Direct Feature Matching.” *European Conference on Computer Vision* (2022). https://doi.org/10.48550/arXiv.2204.00559
39. S. Chen, Y. Bhalgat, X. Li, J.-W. Bian, K. Li, Z. Wang, and V. A. Prisacariu, “Neural refinement for absolute pose regression with feature synthesis,” in CVPR, 2024, pp. 20987–20996. https://doi.org/10.1109/CVPR52733.2024.01983
40. Davison, Andrew J. et al. “MonoSLAM: Real-Time Single Camera SLAM.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (2007): 1052-1067. https://doi.org/10.1109/TPAMI.2007.1049
41. Klein, Georg S. W. and David William Murray. “Parallel Tracking and Mapping for Small AR Workspaces.” *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality* (2007): 225-234. https://doi.org/10.1109/ISMAR.2007.4538852
42. R. Mur-Artal, J.M.M. Montiel, J.D. Tardos. “ORB-SLAM: A Versatile and Accurate Monocular SLAM System.” *IEEE Transactions on Robotics* 31 (2015): 1147-1163. https://doi.org/10.1109/TRO.2015.2463671
43. Kesorn K, Poslad S. An enhanced bag of visual word vector space model to represent visual content in athletics images[J]. IEEE Trans on Multimedia, 2011,14(1): 211-222. https://doi.org/10.1109/TMM.2011.2170665

44. Mur-Artal, Raul and Juan D. Tardós. “ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras.” *IEEE Transactions on Robotics* 33 (2016): 1255-1262. https://doi.org/10.1109/TRO.2017.2705103
45. C. Campos, R. Elvira, J.J.G. Rodríguez, J.M. Montiel, J.D. Tardós. “ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM.” *IEEE Transactions on Robotics* 37 (2020): 1874-1890. https://doi.org/10.1109/TRO.2021.3075644
46. Newcombe, Richard A. et al. “DTAM: Dense tracking and mapping in real-time.” *2011 International Conference on Computer Vision* (2011): 2320-2327. https://doi.org/10.1109/ICCV.2011.6126513
47. J. Engel, T. Schöps, D. Cremers. “LSD-SLAM: Large-Scale Direct Monocular SLAM.” *European Conference on Computer Vision* (2014). https://doi.org/10.1007/978-3-319-10605-2_54
48. J. Engel, V. Koltun, D. Cremers. “Direct Sparse Odometry.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40 (2016): 611-625. https://doi.org/10.1109/TPAMI.2017.2658577
49. Schöps, Thomas et al. “BAD SLAM: Bundle Adjusted Direct RGB-D SLAM.” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019): 134-144. https://doi.org/10.1109/CVPR.2019.00022
50. Yang, Nan et al. “D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry.” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020): 1278-1289. https://doi.org/10.1109/CVPR42600.2020.00136
51. K. Tateno, F. Tombari, I. Laina, N. Navab. “CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction.” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017): 6565-6574. https://doi.org/10.1109/CVPR.2017.695
52. Zhou, Huizhong et al. “DeepTAM: Deep Tracking and Mapping.” *European Conference on Computer Vision* (2018). https://doi.org/10.48550/arXiv.1808.01900
53. Yu, Chao et al. “DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments.” *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2018): 1168-1174. https://doi.org/10.1109/IROS.2018.8593691
54. Yang, Shichao and Sebastian A. Scherer. “CubeSLAM: Monocular 3-D Object SLAM.” *IEEE Transactions on Robotics* 35 (2018): 925-938. https://doi.org/10.1109/TRO.2019.2909168
55. Zins, Matthieu et al. “OA-SLAM: Leveraging Objects for Camera Relocalization in Visual SLAM.” *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2022): 720-728. https://doi.org/10.48550/arXiv.2209.08338
56. Kerbl, Bernhard et al. “3D Gaussian Splatting for Real-Time Radiance Field Rendering.” *ACM Transactions on Graphics (TOG)* 42 (2023): 1 - 14. https://doi.org/10.1145/3592433
57. E. Sucar, S. Liu, J. Ortiz, A.J. Davison. “iMAP: Implicit Mapping and Positioning in Real-Time.” *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021): 6209-6218. https://doi.org/10.48550/arXiv.2103.12352
58. Z. Zhu, S. Peng, V. Larsson, W. Xu, H. Bao, Z. Cui, M.R. Oswald, M. Pollefeys. “NICE-SLAM: Neural Implicit Scalable Encoding for SLAM.” *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021): 12776-12786. https://doi.org/10.48550/arXiv.2112.12130
59. Zou, Danping and Ping Tan. “CoSLAM: Collaborative Visual SLAM in Dynamic Environments.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (2013): 354-366. https://doi.org/10.1109/TPAMI.2012.104
60. Yang, Xingrui et al. “Vox-Fusion: Dense Tracking and Mapping with Voxel-based Neural Implicit Representation.” *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (2022): 499-507. https://doi.org/10.48550/arXiv.2210.15858
61. Johari, M. M., Carta, C., & Fleuret, F. “ESLAM: Efficient Dense SLAM System Based on Hybrid Representation of Signed Distance Fields.” *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022): 17408-17419. https://doi.org/10.1109/cvpr52729.2023.01670
62. C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, X. Li. “GS-SLAM: Dense Visual SLAM with 3D Gaussian Splatting.” *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023): 19595-19604. https://doi.org/10.1109/CVPR52733.2024.01853
63. Hidenobu Matsuki, Riku Murai, Paul H.J. Kelly, and An drew J. Davison. Gaussian splatting slam. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 18039–18048, 2024. https://doi.org/10.1109/CVPR52733.2024.01708
64. Huajian Huang, Longwei Li, Hui Cheng, and Sai-Kit Yeung. Photo-slam: Real-time simultaneous localization and photo realistic mapping for monocular stereo and rgb-d cameras. In IEEE/CVF Conference on Computer Vision and Pattern

Recognition, pages 21584–21593, 2024. https://doi.org/ 10.1109/CVPR52733.2024.02039

65. N. Keetha, J. Karhade, K.M. Jatavallabhula, G. Yang, S. Scherer, D. Ramanan, J. Luiten. "SplaTAM: Splat, Track & Map 3D Gaussians for Dense RGB-D SLAM." *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023): 21357-21366. https://doi.org/10.48550/arXiv.2312.02126
66. Chaoyang Guo, Chunyan Gao,Yiyang Bai. "RD-SLAM: Real-Time Dense SLAM Using Gaussian Splatting." *Applied Sciences* (2024): n. pag. https://doi.org/10.3390/app14177767
67. Xinggang Hu, Chenyangguang Zhang, Mingyuan Zhao, et al. "DyGS-SLAM: Realistic Map Reconstruction in Dynamic Scenes Based on Double-Constrained Visual SLAM." *Remote Sensing* (2025): n. pag. https://doi.org/10.3390/rs17040625
68. Mingrui Li,Yiming Zhou,Hongxing Zhou, et al. "Dy3DGS-SLAM: Monocular 3D Gaussian Splatting SLAM for Dynamic Environments." *2025 IEEE International Conference on Robotics and Automation (ICRA)* (2025): 14572-14578. https://doi.org/10.1109/ICRA55743.2025.11127324
69. Tianci Wen, Zhiang Liu, Yongchun Fang. "SEGS-SLAM: Structure-enhanced 3D Gaussian Splatting SLAM with Appearance Embedding." (2025). https://doi.org/10.48550/arXiv.2501.05242
70. Zhexi Peng, Tianjia Shao, Yong Liu, Jingke Zhou, Yin Yang, Jingdong Wang, and Kun Zhou. Rtg-slam: Real-time 3d re construction at scale using gaussian splatting. In ACM SIG GRAPH, 2024. https://doi.org/10.48550/arXiv.2404.19706