Robust Absolute Headset Tracking for Extended Reality

Robert M. Tenny
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas
rmtenny@utexas.edu

Lisong C. Sun
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas
codey.sun@utexas.edu

Alperen Duru
Department of Electrical and Computer Engineering
The University of Texas at Austin
Austin, Texas
aduru@utexas.edu

Todd E. Humphreys
Department of Aerospace Engineering and Engineering Mechanics
The University of Texas at Austin
Austin, Texas
todd.humphreys@utexas.edu

***Abstract*—This paper presents a novel headset tracking framework designed for extended reality (XR) applications. The growth of XR demands accurate and robust tracking mechanisms that are suitable for both indoor and outdoor environments and offer anchoring to a global reference frame. By loosely coupling a visual simultaneous localization and mapping (SLAM) algorithm to a tightly-coupled carrier phase differential GNSS (CDGNSS) and inertial sensor subsystem, the proposed system aims to achieve centimeter-accurate, globally-referenced tracking that persists during extended periods of GNSS degradation. Collaborative and persistent XR experiences are enabled through accurate map creation utilizing a bundle adjustment approach for map generation and maintenance. Cloud or near-edge offloading of computationally demanding steps in the pipeline is explored to reduce the computational demand on the headset. Robust tracking performance is evaluated in terms of odometric drift under GNSS outages. This paper also explores the benefit of additional headset tracking constraints offered by direction-of-arrival measurements to nearby cellular base stations. Such measurements will become available as future wireless standards make increasing use of mmWave frequencies.**

***Index Terms*—CDGNSS, SLAM, extended reality**

I. Introduction

Extended reality (XR) is poised to dramatically alter the way people interact with the world. Fully-immersive XR experiences require determination of the position and orientation of the user's headset, allowing the user's motion to be reflected in a virtual environment. Outdoor headset tracking, this paper's focus, is possible under open-sky conditions and brief periods of GNSS signal degradation [1], [2], but an immersive XR experience will require a precise headset position and orientation (pose) during the longer-duration GNSS signal degradation that occurs with outdoor use in urban environments.

Outdoor XR will allow for unlimited users to share collaborative and interactive XR experiences with persistent virtual objects. A globally-referenced position is necessary to unlock the full potential of XR in outdoor environments, and GNSS can provide this [1]. Precise outdoor headset localization has been achieved using carrier-phase differential GNSS (CDGNSS), but it struggles in environments with poor GNSS coverage [1]. Nonetheless, some form of GNSS coupling is crucial for a globally-referenced tracking solution. The method of coupling CDGNSS with visual SLAM proposed in this paper has the advantage of building off of previous work in [1], [3]–[5] providing high-availability decimeter-accurate CDGNSS positioning even in environments with urban canyons and significant multipath.

Currently, commercial XR headsets are limited to indoor and digitally fenced-in areas. Lighthouse-based tracking systems, which rely on external lighthouse stations to track the user, provide impressive sub-millimeter precision, but are by nature limited in usable space [6]. Furthermore, any occlusion of the lighthouse signals interferes with tracking. Such an approach is not scalable for a global XR experience. Alternatively, camera-based inside-out tracking uses simultaneous localization and mapping (SLAM) algorithms to determine the user's relative pose without external hardware [7], [8]. Inside-out tracking utilizes a generated and stored map of the user's surrounding area, typically limiting the user to a local coordinate frame that is not globally-referenced. Current state-of-the-art SLAM algorithms such as OKVIS1/2 [9], [10], ORB-SLAM1/2/3 [11]–[13], ICE-BA [14], ROVIO [15], GEOSLAM [16], SLAM++ [17] each have different trade-offs based on computational complexity, precision, and persistence [18]–[20]. While real-time applications have adopted filtering methods, windowed and global bundle-adjustment SLAM algorithms have an advantage in accuracy and map generation. For XR applications, a generated map will be essential for shared experiences. Thus, a SLAM framework based on BA is most attractive provided its computational cost can be managed.

Given the complementary nature of GNSS and visual SLAM, researchers have worked to couple solutions to create a well-rounded tracking system. In particular, the outdoor, global-referencing capabilities of GNSS complement the limitations of urban, locally-referenced SLAM. Coupling SLAM with GNSS is often used for ground vehicles and unmanned aerial vehicles to create an accurate local map and locate this map in the global frame [16], [21]–[28]. The result of the SLAM computation is typically coupled with standard GNSS, providing meter-level precision as opposed to the centimeter-accurate CDGNSS. However some previous work has also coupled SLAM with CDGNSS using a filter based approach [29]–[34]. As compared to filtering, an approach based on bundle adjustment (BA), as proposed in the current paper, is more accurate [18] and lends itself to batch partitioning and cloud offloading of the most computationally expensive steps in the estimation pipeline.

With recent increases in wireless communication speeds, computationally expensive tasks are increasingly being offloaded to the cloud for small robotics applications [35]–[39]. A similar approach can be applied to XR headset tracking. The process of image-based positioning has components that are ideal for cloud offloading. After feature detection occurs, the feature points and their descriptors are sparse, whereas the BA calculation is computationally expensive. This means a small amount of data can be sent over the network to exploit fast cloud computing at large data centers or in near-edge resources (e.g., arrays of graphical processing units (GPUs) co-located with cellular base stations). Likewise, the result of the BA process is lightweight and easily transported back over the network, consisting primarily of the locally-relevant portion of the 3D map. In addition, cloud-based processing allows for user-contributed maps to be combined, forming larger, collaborative maps that multiple users can exploit for both pose estimation and XR applications [38]. This differs from current approaches, which transport entire images to be processed in the cloud, requiring higher data rates [36].

Future network architectures such as 6G may be the key to providing the high-reliability, low-latency communications needed for XR. In addition to increased bandwidth, 6G will utilize beam forming to steer the receiver's antenna towards a base station to increase communications reliability. The direction-of-arrival (DOA) of these signals from 6G base stations can be exploited to further constrain the pose of the XR headset [40].

Against the backdrop of the foregoing observations, this paper makes three primary contributions. First, it develops a framework for robust and precise globally-referenced XR headset tracking that couples BA-based visual SLAM with inertial and CDGNSS sensing. Second, it explores cloud offloading of the BA to lighten the headset's computational load. Third, it evaluates DOA measurements, the by-product of beamforming at the headset, as additional constraints for further reducing headset pose errors.

II. Open-World Virtual Reality

Open-world virtual reality (OWVR), first coined in [1], is a concept that envisions a seamless connection between virtual and physical worlds—indoors and out—through correspondence, customizability, and persistence. Correspondence refers to a 3D reconstruction of the real world being mapped to the virtual world. Customizability is the ability to alter this 3D reconstruction on demand. Persistence is the permanence of this customization, allowing virtual objects to remain in the same location forever. One can imagine a virtual XR experience where users drop items or leave notes in the virtual map for any user to find in the future. The objects would remain persistently accessible at their original precisely georeferenced location.

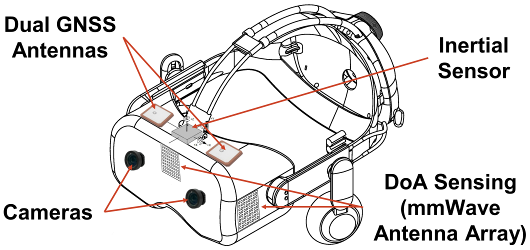


Fig. 1: Concept design for how sensors and communications antennas could be incorporated into a XR headset. The base sketch is the Valve Deckard VR headset.

Previous work [1] achieved a rough implementation of OWVR by separating the three requirements. A local map was reconstructed via photogrammetry, altered within a game engine, and traversed using a tightly coupled CDGNSS and inertial unit. This tracking solution does indeed provide the centimeter-and-degree accurate pose estimation needed for a convincing XR experience while also being globally referenced. However, its use cases depend greatly on an environment suited for CDGNSS integer fixing. In other words, while one can traverse the open-sky rooftop of a parking garage, one cannot walk through a busy city street lined with buildings and overhangs. This limitation prevents the experience from being truly open-world.

Achieving true OWVR requires a robust positioning solution. Many environments such as urban cities and other high multipath environments do not allow for continuous centimeter-accurate positioning. On the other hand, the same dense urban environments that a are detrimental to precise CDGNSS based positioning are rich in landmarks and features to support a robust visual SLAM based pose estimate. The complementary nature of vision-based positioning and GNSS allows their combination on a single headset—as envisioned in Fig. (1)—for precise and robust pose estimation.

The block diagram in Fig. (2) gives an overview of this paper's headset tracking framework. Block-level subsystems will be further described in following sections.

III. Reference Frames

It will be convenient to define the following reference frames:

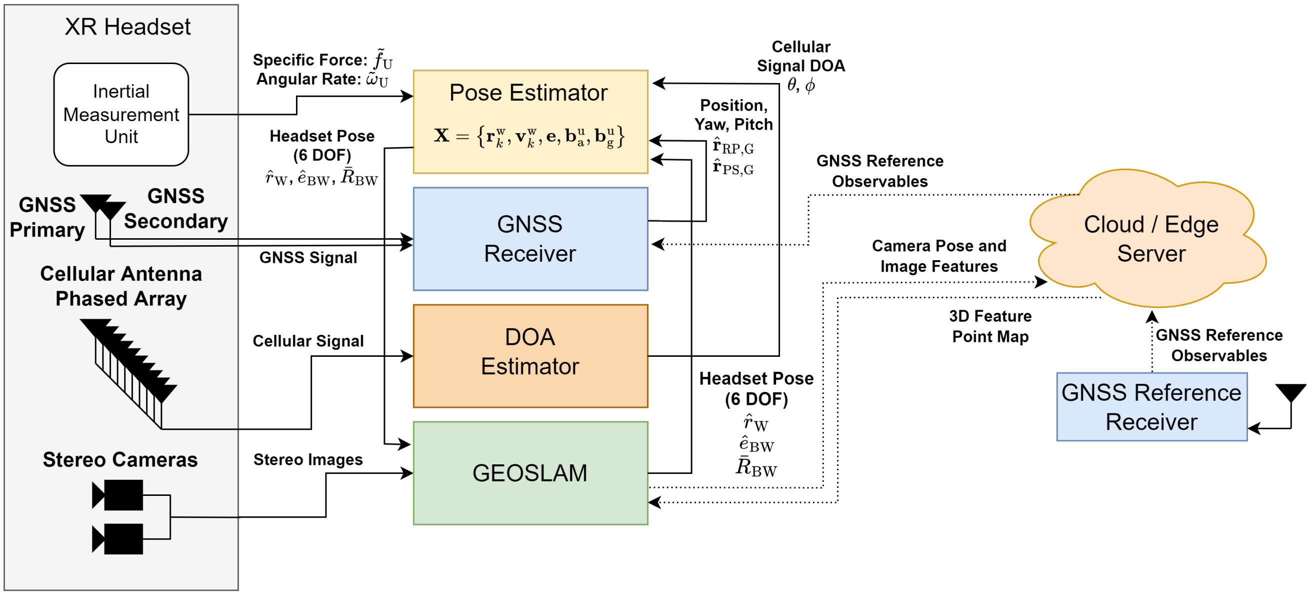

XR Headset
Inertial Measurement Unit
Specific Force: $\tilde{f}_{\mathrm{U}}$
Angular Rate: $\tilde{\omega}_{\mathrm{U}}$
Pose Estimator
$\mathbf{X} = \{\mathbf{r}_k^{\mathrm{w}}, \mathbf{v}_k^{\mathrm{w}}, \mathbf{e}, \mathbf{b}_{\mathrm{a}}^{\mathrm{u}}, \mathbf{b}_{\mathrm{g}}^{\mathrm{u}}\}$
Headset Pose (6 DOF)
$\hat{r}_{\mathrm{W}}, \hat{e}_{\mathrm{BW}}, \bar{R}_{\mathrm{BW}}$
Cellular Signal DOA
θ, ϕ
Position, Yaw, Pitch
$\hat{\mathbf{r}}_{\mathrm{RP,G}}$
$\hat{\mathbf{r}}_{\mathrm{PS,G}}$
GNSS Primary
GNSS Secondary
GNSS Signal
GNSS Receiver
GNSS Reference Observables
Cloud / Edge Server
Camera Pose and Image Features
Cellular Antenna Phased Array
Cellular Signal
DOA Estimator
3D Feature Point Map
GNSS Reference Observables
GNSS Reference Receiver
Headset Pose (6 DOF)
$\hat{r}_{\mathrm{W}}$
$\hat{e}_{\mathrm{BW}}$
$\bar{R}_{\mathrm{BW}}$
Stereo Cameras
Stereo Images
GEOSLAM

Fig. 2: Diagram of the full pose estimator.

C: The *camera frame* is centered at the camera center of the primary camera with its x axis aligned with the long axis of the focal plane pointing to the right. Its z axis is boresight to the focal plane and the y axis is aligned to complete the right-handed triad.

S: The *local SLAM frame* is equivalent to the camera frame corresponding to the first image.

U: The *IMU frame* is centered at and aligned with the IMU accelerometer triad.

B: The *body frame* has its origin at the phase center of the headset's primary GNSS antenna. Its x axis points towards the phase center of the secondary antenna, its z axis is aligned with the boresight vector of the primary antenna, and its y axis completes the right-handed triad.

G: The *global frame* is the Earth-centered, Earth-fixed (ECEF) reference frame

W: The *world frame* is a fixed geographic East-North-Up (ENU) frame with its origin at the phase center of the reference GNSS antenna, which is located at a fixed location with known ECEF coordinates.

IV. GEOSLAM

This paper's visual SLAM framework is called Globally-referenced Electro-Optical SLAM (GEOSLAM). It is the evolution of the framework of the same name developed in [16]. This section offers an overview of GEOSLAM and explains what makes it uniquely suitable for XR headset pose estimation. It then details how GEOSLAM performs feature extraction, determines structure from motion, and performs bundle adjustment. It also discusses GEOSLAM's ability to constrain the pose estimate with CDGNSS measurements, and the creation of collaborative maps.

A. GEOSLAM Overview

Unlike CDGNSS, SLAM algorithms are often used in current XR headsets due to their cost effectiveness and their tracking performance in most common indoor settings. Requiring only a camera at minimum, SLAM algorithms extract feature points from images and track those feature points to derive a map reconstruction and camera pose. This works especially well in areas with many distinct feature points, e.g., a living room with a variety of furniture or a dense urban sprawl lined with buildings. Conversely, environments without distinct features, such as an open field, are not suitable for SLAM: they lack the necessary feature points to robustly calculate optical flow. From this, one can make the connection that SLAM and CDGNSS complement each other as each excel in environments where the other fails. Furthermore, a tracking solution combining SLAM and CDGNSS will satisfy two of OWVR's three requirements: an appropriate SLAM algorithm can create a precise correspondence through 3D reconstruction, and CDGNSS provides the global reference frame needed for persistent objects.

Two common approaches to SLAM are filtering and keyframe-based bundle adjustment (BA). In terms of computational complexity, the authors of [18] discovered that with N feature points and M keyframes, filtering SLAM is $O(MN^3)$ whereas BA SLAM is $O(NM^2 + M^3)$; importantly, the authors also show that an increase in feature points increases accuracy while an increase in keyframes increases robustness. The current paper chooses a BA approach for two primary reasons: (1) this paper aims to construct dense 3D maps that provide centimeter-accurate visual tracking; the cubic feature-number scaling of filtering would severely limit the density of these maps, and (2) this paper aims to create a framework suitable for cloud offloading, and this is done more cleanly with batches of data from windowed BA than with filtering.

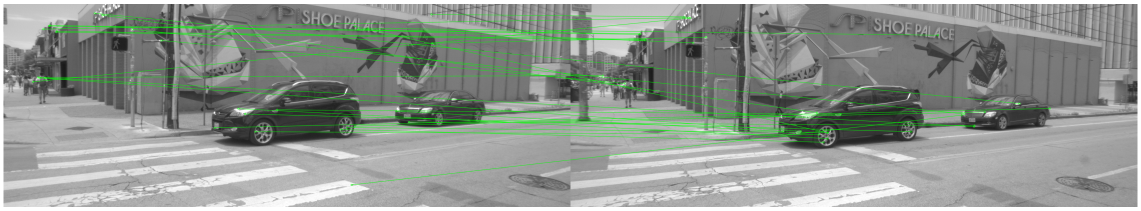


Fig. 3: Frame from the TEX-CUP dataset [43] with features stereo matched between the left and right cameras.

The core of GEOSLAM is a bundle adjustment algorithm that creates and store maps across multiple sessions. This method supports the open-world XR concept by allowing headset tracking to be a collaborative effort. For example, a user walking through an area under marginal CDGNSS conditions can create a map that may only be partially complete. Then, if this map is stored on the cloud, another user can walk through the same area and refine and extend the map by leveraging the prior map to improve their CDGNSS robustness. Additional iterations can further refine and extend the map until it is fully connected and accurately globally-referenced. The following sections will outline the details of the GEOSLAM algorithm and its cost functions.

B. Feature Extraction and Matching

As shown in Fig. (1) the envisioned XR headset is equipped with CDGNSS antennas, an inertial measurement unit, and multiple cameras. The headset cameras are synchronized with the CDGNSS sampling to capture globally shuttered images whose timing is traceable to the GNSS receiver's clock. A collection of images taken simultaneously is referred to as a frame. These images are undistorted according to each camera's calibration parameters, and stereo pairs are rectified. Then, feature points are extracted according to the Scale-Invariant Feature Transform (SIFT) to obtain each feature point's SIFT descriptor and 2D coordinates [41]. Fig. (3) shows extracted and stereo-matched feature points. These feature points are matched to a set of 3D map points via the Fast Library for Approximate Nearest Neighbors [42]. As described in the Perspective-n-Point(PnP) problem [16], a camera pose can be estimated from sufficient point matches, with random sample consensus (RANSAC) being used to filter out any outliers among the matched points. This initial pose estimation allows for the reprojection of 3D map points onto the estimated image plane. For each map point successfully reprojected onto the image plane, a k-nearest-neighbor brute-force matching is performed on all feature points within a threshold distance of the reprojection, following which PnP with RANSAC is performed once more.

As the headset moves through space, its cameras detect new points not present in the current map database. If more than one camera captures the same feature point, and if the baseline between the cameras is known, then the point's approximate 3D location can be derived and added to the map.

Let $\boldsymbol{u}_i^n$ denote the reprojection of map point i onto the camera's image plane on frame n. $\boldsymbol{u}_i^n = \emptyset$ if i is not within the camera's view. For the nth frame, GEOSLAM's tracking module assembles:

1) a set of estimated 3D map points
$$M_n \triangleq \{i \ : \ \boldsymbol{u}_i^n \neq \emptyset\}$$
In addition, let $\boldsymbol{m}_\mathrm{S}^i \in \mathbb{R}^3$ denote the estimated 3D coordinates of point i in the S frame.
2) a set of measured feature point matches
$$U_n \triangleq \{\tilde{\boldsymbol{u}}_i^n \ : \ i \in M_n\}$$
where $\tilde{\boldsymbol{u}}_i^n \in \mathbb{R}^2$ denotes the 2D coordinates of the measured SIFT feature corresponding to point i in the nth frame.
3) an estimated camera pose
$$(\boldsymbol{c}_\mathrm{S}^n, \boldsymbol{\theta}_\mathrm{CS}^n)$$
denoting the 6DoF (six-degrees-of-freedom) pose of the primary camera on the nth frame where $\boldsymbol{c}_\mathrm{S}^n$ is the location in the S frame and $\boldsymbol{\theta}_\mathrm{CS}^n$ is the angle-axis representation of the attitude of S with respect to the camera frame C. For purposes of matrix multiplication, $R(\cdot)$ is the direction cosine matrix of an input angle-axis orientation, e.g., $R(\boldsymbol{\theta}_\mathrm{CS}^n)$ is the rotation matrix corresponding to $\boldsymbol{\theta}_\mathrm{CS}^n$.

Starting from these initial data and estimates, GEOSLAM refines the camera poses and map points via bundle adjustment.

C. Structure from Motion

Given a sequence of images with 2D feature points, Structure from motion (SfM) is the process of both estimating the 6DoF pose of the camera corresponding to each image and estimate the 3D points corresponding to the 2D features. Consider a pinhole camera model with focal length f, principal point (p_x, p_y), and pose $(\boldsymbol{c}_\mathrm{S}^n, \boldsymbol{\theta}_\mathrm{CS}^n)$. The projection of 3D map point i on the camera's image plane of frame n is [44]

$$\boldsymbol{u}_i^n = \begin{bmatrix} \frac{x_i^n}{z_i^n} \\ \frac{y_i^n}{z_i^n} \end{bmatrix}, \tag{1}$$
$$\begin{bmatrix} x_i^n \\ y_i^n \\ z_i^n \end{bmatrix} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \left[R(\boldsymbol{\theta}_\mathrm{CS}^n) \mid -R(\boldsymbol{\theta}_\mathrm{CS}^n)\boldsymbol{c}_\mathrm{S}^n \right] \begin{bmatrix} \boldsymbol{m}_\mathrm{S}^i \\ 1 \end{bmatrix}$$

A single camera can capture a map resolved to within a similarity transform. However, the envisioned XR headset is equipped with multiple cameras. If the baseline distances between cameras are known, the rendered map will have correct scaling. Suppose that, in addition to the primary camera, the headset contains an alternate camera with intrinsics $\bar{f}$ and $(\bar{p}_x, \bar{p}_y)$. These two cameras are synchronized, i.e., the nth image of the primary camera is taken at the same time as the nth image of the alternate camera. Let $(\bar{\boldsymbol{c}}_{\mathrm{C}}, \bar{\boldsymbol{\theta}}_{\bar{\mathrm{C}}\mathrm{C}})$ denote the known transformation from the primary camera to the alternate camera. Then the alternate camera's reprojection function is modified from (1) as

$$\bar{\boldsymbol{u}}_i^n = \begin{bmatrix} \frac{\bar{x}_i^n}{\bar{z}_i^n} \\ \frac{\bar{y}_i^n}{\bar{z}_i^n} \end{bmatrix},$$

$$\begin{bmatrix} \bar{x}_i^n \\ \bar{y}_i^n \\ \bar{z}_i^n \end{bmatrix} = \begin{bmatrix} \bar{f} & 0 & \bar{p}_x \\ 0 & \bar{f} & \bar{p}_y \\ 0 & 0 & 1 \end{bmatrix} \left[R_{\bar{\mathrm{C}}\mathrm{S}}^n \mid -R_{\bar{\mathrm{C}}\mathrm{S}}^n \bar{\boldsymbol{c}}_{\mathrm{S}}^n \right] \begin{bmatrix} \boldsymbol{m}_{\mathrm{S}}^i \\ 1 \end{bmatrix}$$

where

$$\bar{\boldsymbol{c}}_{\mathrm{S}}^n = \boldsymbol{c}_{\mathrm{S}}^n + R^{\mathsf{T}}(\boldsymbol{\theta}_{\mathrm{CS}}^n)\bar{\boldsymbol{c}}_{\mathrm{C}},$$
$$R_{\bar{\mathrm{C}}\mathrm{S}}^n = R(\boldsymbol{\theta}_{\bar{\mathrm{C}}\mathrm{C}})R(\boldsymbol{\theta}_{\mathrm{CS}}^n)$$

Note that all cameras will have known, rigid transformations relative to each other, so only one camera's pose needs to be estimated. However, reprojections are calculated with respect to all cameras in which a map point is visible.

Given a matched feature point measurement $\tilde{\boldsymbol{u}}_i^n$, the reprojection error is defined as

$$\boldsymbol{e}_i^n = \tilde{\boldsymbol{u}}_i^n - \boldsymbol{u}_i^n$$

Bundle adjustment finds the $(\boldsymbol{c}_{\mathrm{S}}^n, \boldsymbol{\theta}_{\mathrm{CS}}^n)$ and $\boldsymbol{m}_{\mathrm{S}}^i$, $i \in M_n$ that minimize a cost defined across all feature points. The cost function for frame n is

$$C_n = \sum_{i \in M_n} \rho\left((\boldsymbol{e}_i^n)^{\mathsf{T}} (\Omega_i^n)^{-1} \boldsymbol{e}_i^n \right) \tag{2}$$

where $\rho(\cdot)$ is a loss function of choice (e.g., least squares or Huber loss) and $\Omega_i^n \in \mathbb{R}^{2\times 2}$ is the covariance matrix associated with $\tilde{\boldsymbol{u}}_i^n$. GEOSLAM solves this minimization—and all subsequent BA minimization problems—using Google's Ceres solver [45].

1) Keyframe Bundle Adjustment: In a continuous video stream, map points are expected to be shared between frames. Thus, (2) must be modified to account for this temporal connection. In the SLAM literature, certain frames called keyframes, are designated as representative of nearby frames. Only keyframes participate in the full BA. Keyframes are stored to save past pose and reprojection information. GEOSLAM chooses keyframes based on distance traveled and number of new map points discovered, e.g., a frame is declared a keyframe if it is taken more than 1 meter from the nearest keyframe or if it contains fewer than 200 map matches. Each feature designated as a map point must be present in at least one keyframe, and the 3D coordinates of each map point must be consistent across all keyframe reprojections.

The covisibility window of frame n is the set of all keyframes that share at least T map points with n. This can be applied recursively X times to retrieve an X-level covisibility window, mathematically represented as

$$\mathrm{cov}(n, X) \triangleq \begin{cases} \{k \;:\; |M_k \cap M_n| \geq T\} & X = 1 \\ \{k \;:\; |M_k \cap \left(\cup_{y \in \mathrm{cov}(n, X-1)} M_y\right)| \geq T\} & X > 1 \end{cases}$$

where $|A|$ denotes the cardinality of the set A.

The cost function across the full X-level covisibility window of frame n is written as

$$C_n^{\mathrm{cov}} = \sum_{k \in \mathrm{cov}(n,X)} \sum_{i \in M_n} \rho\left((\boldsymbol{e}_i^k)^{\mathsf{T}} (\Omega_i^k)^{-1} \boldsymbol{e}_i^k \right)$$

2) CDGNSS Aiding: An unaided bundle adjustment is computed in the S frame such that its origin aligns with the pose of the first camera frame. The S frame is thus made relative to the first camera pose. To enable map permanence, the S frame must be anchored to a global reference frame G. GEOSLAM does this by incorporating CDGNSS measurements into the BA. These measurements (1) allow GEOSLAM to compute an affine transformation between the headset's local S frame and the global G frame, and (2) constrain visual odometric drift.

In order to align the local S frame and the global G frame, GEOSLAM performs an initialization procedure in which visual BA is performed while storing CDGNSS measurements for each keyframe. After N keyframes, two sets of 3D points are created: keyframe positions in the S frame by visual BA and keyframe positions in the G frame by CDGNSS. Let $\boldsymbol{z}_{\mathrm{S}}^n$ denote the antenna position according to visual BA and $\tilde{\boldsymbol{z}}_{\mathrm{G}}^n$ denote the antenna position measurement from CDGNSS, for frame n. Given the known baseline between the camera and antenna, $\boldsymbol{z}_{\mathrm{S}}^n$ is derived from the camera pose $(\boldsymbol{c}_{\mathrm{S}}^n, \boldsymbol{\theta}_{\mathrm{CS}}^n)$. Given CDGNSS measurements and camera images, these two rigid sets of camera poses will be similar to an affine transformation. For a known stereo baseline between the cameras, the two sets will also be of the same scale. Thus, a rotation and translation from the G frame to the S frame, R_{SG} and $\boldsymbol{t}_{\mathrm{SG}}$ respectively, can be calculated by aligning the two sets of points. As shown in [46], this can be modeled as the least squares optimization problem

$$(R_{\mathrm{SG}}, \boldsymbol{t}_{\mathrm{SG}}) = \underset{R \in SO(3), \boldsymbol{t} \in \mathbb{R}^3}{\operatorname{argmin}} \sum_{n=1}^{N} \|(R\tilde{\boldsymbol{z}}_{\mathrm{G}}^n + \boldsymbol{t}) - \boldsymbol{z}_{\mathrm{S}}^n\|^2 \tag{3}$$

and solved using singular value decomposition.

Once initialized, GEOSLAM requests a CDGNSS pose for each frame. Let

$$\boldsymbol{e}_z^n \triangleq (R_{\mathrm{SG}}\tilde{\boldsymbol{z}}_{\mathrm{G}}^n + \boldsymbol{t}_{\mathrm{SG}}) - \boldsymbol{z}_{\mathrm{S}}^n$$

denote the CDGNSS error for frame n. This CDGNSS error is added as a parameter block in the BA cost function to form the augmented cost function

$$C_n^{\mathrm{G}} = \sum_{k \in \mathrm{cov}(n,X)} \left[\sum_{i \in M_n} \rho\left((\boldsymbol{e}_i^k)^\mathsf{T} (\Omega_i^k)^{-1} \boldsymbol{e}_i^k \right) + (\boldsymbol{e}_z^k)^\mathsf{T} (\Gamma^k)^{-1} \boldsymbol{e}_z^k \right]$$

where Γ^k is the covariance matrix of $\tilde{\boldsymbol{z}}_{\mathrm{G}}^k$.

3) Pose-graph Optimization: After BA is performed, if $\boldsymbol{e}_z^n$ is large and $\tilde{\boldsymbol{z}}_{\mathrm{G}}^n$ has low variance, a trajectory pose-graph optimization is performed. This may occur e.g., after a CDGNSS outage. Once a high confidence measurement $\tilde{\boldsymbol{z}}_{\mathrm{G}}^n$ is available, pose-graph optimization will be performed. For neighboring keyframes k and j, the delta-pose is defined as

$$\begin{aligned} \delta \boldsymbol{c}_{\mathrm{S}}^{kj} &\triangleq \hat{\boldsymbol{c}}_{\mathrm{S}}^k - \hat{\boldsymbol{c}}_{\mathrm{S}}^j, \\ \delta \boldsymbol{\theta}_{\mathrm{CS}}^{kj} &\triangleq \boldsymbol{\theta}\left(R\left(\hat{\boldsymbol{\theta}}_{\mathrm{CS}}^k\right) R^\mathsf{T}\left(\hat{\boldsymbol{\theta}}_{\mathrm{CS}}^j\right)\right) \end{aligned}$$

where $(\hat{\cdot})$ denotes a state prior to pose-graph optimization, and $\boldsymbol{\theta}(\cdot)$ outputs the angle-axis representation of an input direction cosine matrix. GEOSLAM aims to estimate new keyframe poses that preserve the delta-poses while ensuring consistency with CDGNSS measurements and smoothness along the full path. As the first step, frame n's pose is adjusted assuming that the CDGNSS measurement is correct. The keyframe poses

$$(\boldsymbol{c}_{\mathrm{S}}^k, \boldsymbol{\theta}_{\mathrm{CS}}^k) \forall k \in \mathrm{cov}(n, X_p)$$

over an X_p-level covisibility window are used to define the pose-graph delta-pose error:

$$\begin{aligned} \boldsymbol{e}_c^{kj} &\triangleq \left(\boldsymbol{c}_{\mathrm{S}}^k - \boldsymbol{c}_{\mathrm{S}}^j\right) - \delta \boldsymbol{c}_{\mathrm{S}}^{kj}, \\ \boldsymbol{e}_\theta^{kj} &\triangleq \boldsymbol{\theta}\left(R\left(\boldsymbol{\theta}_{\mathrm{CS}}^k\right) R^\mathsf{T}\left(\boldsymbol{\theta}_{\mathrm{CS}}^j\right)\right) - \delta \boldsymbol{\theta}_{\mathrm{CS}}^{kj} \end{aligned}$$

The pose-graph optimization can be performed by minimizing the cost due to the delta-pose errors

$$C^{\mathrm{pgo}} = \sum_{k \in \mathrm{cov}(n, X_p)} \sum_{j \in \mathrm{cov}(k,1)} \left((\boldsymbol{e}_c^{kj})^\mathsf{T} \boldsymbol{e}_c^{kj} + (\boldsymbol{e}_\theta^{kj})^\mathsf{T} \boldsymbol{e}_\theta^{kj} \right)$$

subject to the constraint that all keyframes with valid CDGNSS measurements—including n—as well as keyframes at the edge of an X_p-level covisibility window have their poses held fixed; denote this fixed set K_f.

$$\left(\boldsymbol{c}_{\mathrm{S}}^k, \boldsymbol{\theta}_{\mathrm{CS}}^k\right) = \left(\hat{\boldsymbol{c}}_{\mathrm{S}}^k, \hat{\boldsymbol{\theta}}_{\mathrm{CS}}^k\right) \forall k \in K_f$$

This pose-graph optimization is followed by another BA with an expanded covisibility window. By increasing the level of the covisibility window, GEOSLAM extends its optimization window beyond any CDGNSS outages.

4) Collaborative Mapping: After the BA operations, a map is created which contains the collection of keyframes with their 6DoF poses, feature points with descriptors, and CDGNSS measurements. The included map points contain the 3D G frame coordinates and SIFT feature descriptors. Future GEOSLAM runs, e.g., future users, can draw upon this stored map to aid pose estimation with or without CDGNSS measurements. If a user ventures into unmapped regions he or she will extend the map. For a user with an *a priori* map, after BA is performed for frame n, GEOSLAM checks for a merge event by matching to the set of map points not present in recent keyframes $\{i \ : \ i \notin \cup_{k \in \mathrm{cov}(n,X)} M_k\}$. If enough matches are found, a map merge is executed. In this event, GEOSLAM initially assumes the *a priori* map to be true and adjusts the current frame's pose to be consistent with it. Afterward, pose-graph optimization is performed in the same manner as noted in Sec. IV-C2. Following this, duplicate map points are removed and a local BA is performed on the merged map.

V. Cloud Offloading

XR headsets designed for outdoor use will have tight constraints on power consumption and processing power to maximize usable battery life. Offloading a portion of the headset's pose estimation to a cloud or near-edge resource will reduce the computational demand on the XR headset. The most computationally expensive portion of the pose estimator is GEOSLAM. Table I shows the relative processing time for each step of the GEOSLAM pipeline found using Python's native deterministic profiler. Note that this instance of GEOSLAM was executed using only a central processing unit; the addition of a GPU may speed up certain sections of the pipeline. Nonetheless, it is clear that bundle adjustment is the most computationally expensive part of GEOSLAM. This section will explore multiple possible offloading paradigms and the data rate required to support them.

TABLE I: Relative Processing Time

Process	Relative Processing Time
Feature Extraction & Stereo Matching	00.825%
Map Matching	00.860%
Bundle Adjustment	97.925%
Misc.	00.390%

TABLE II: Data Type and Needed Transmit Data Rate.

Data Type	Data Size	Data Rate Required
Images	1.5 MB / frame	240 Mbps
Video (MPEG)	141.19 kB / frame	22.59 Mbps
Images Features (SIFT)	134 B × 1000 features	10.72 Mbps
Images Features (ORB)	38 B × 1000 features	3.04 Mbps
Map Matches	28 B × 300 matches	672 kbps
Pose	0.448 kB	35.84 kbps

Suppose that the full feature-extraction-to-BA pipeline is done on the cloud. In this case, the headset must send the full sequence of frames. Offloading the entire GEOSLAM stack for cloud processing would reduce the computational requirements of an XR headset; however, sending unprocessed images requires a relatively high data rate. The dataset used in this paper, described in detail in Sec. VII-A1, contains stereo pairs of 2048×732 8-bit grayscale images at 10 Hz. Sending the uncompressed stereo images results in a data rate of 240 Mbps, as shown in Table II. A higher resolution camera, additional cameras, or a higher camera frame rate would further increase the required data rate to offload the raw images. One option to reduce the data rate is to compress the video via MPEG or H.264 [47] or computer-vision-specific compression such as Compact Descriptors for Visual Search (CDVS) [49]. However, lossy compression leads to loss of localization precision [48].

Feature extraction can be easily done locally on the headset due to lower computation time. In this case, the headset must send for each feature point (1) 2D pixel coordinates (two 16-bit integers) and (2) a SIFT feature descriptor (a 128-element 8-bit vector). In a properly rectified image, stereo matched points will share a single axis coordinate. This is multiplied by the number of feature points, which varies between images. On this paper's dataset and implementation, GEOSLAM images contained on average 1500 feature points, reduced to $\sim$1000 stereo matches. The approximate data rate to send the feature points to the cloud is shown in Table II assuming an average of 1000 stereo-matched feature points are extracted each frame. This is limited by the number of pixels since there cannot be more feature points than pixels. A similar analysis is performed with the ORB descriptor [50]. Which is more compact than the SIFT feature descriptor, but comes with the cost of reduced precision [51]. If an image is particularly dense with features, the data size could exceed sending the image itself. It has been shown in the literature that image features can be compressed [52], [53] to further reduce the required data-rate.

Most feature points will be eliminated in the map matching process, allowing for a reduced set of features to be sent to the cloud. However, because the map database is stored on the cloud, if matching is to be done within the headset then the headset must download the map beforehand, e.g., during initialization. This paradigm enables the headset to send map keys in place of heavy feature descriptors; a map key may be a map index or hash as long as each map point can be uniquely identified. Table II assumes 300 map matches—each with 2D feature point coordinates and 3D map point coordinates—for each frame. The headset will still need to occasionally send descriptors for new map points (in the case of exploration), but the required data rate will will be low compared to sending every feature point's descriptor. The headset must also perform feature matching, a process whose computational expense is comparable to feature extraction, as shown in Table I.

All above paradigms must also send timestamped CDGNSS based pose estimates to allow for globally-referenced BA; this overhead is captured in Table II. For the return path, the cloud provides the geo-referenced 3D map points local to the XR headset. Treating these post-BA points as trustworthy markers, the headset can perform parallel tracking and mapping [7], [8].

Offloading visual processing to the cloud has the added benefit of allowing the map points and keyframes to be stored on the cloud. This allows large precise maps to be built collaboratively as described in Sec. IV-C4.

VI. DOA Based Positioning

Immersive outdoor XR experiences will require high data rates to deliver XR content to the user. In order to achieve high data rates mmWave or subTHz spectrum is expected to be used in future standards requiring high gain phased array antennas. The narrow beams of the phased array antennas require the receiver to perform precise alignment with the base station. The precise DOA estimates produced by the cellular radio can be utilized to aid in pose estimation of the XR headset. This section outlines how a signal's DOA estimates can be utilized for pose estimation.

A. Time Constants for DOA Estimates

Beam coherence time is the average time interval a beam formed by a phased array antenna is deemed to be valid. The beam coherence time reveals insight into the rate DOA estimation can be performed. The beam coherence time of a channel is often longer than the channel coherence time, causing DOA methods performed at each channel coherence time to be unnecessary [54], [55]. When DOA estimates are used for pose estimation, the beam coherence time represents the time interval that a DOA estimate is reliable.

For slow rotations of the XR headset the beam coherence time will be large and allow reliable DOA estimates. During periods of high rate XR headset motion the beam coherence may be quite small and highly dynamic motion may cause the beam coherence time to be shorter than the time to compute a DOA estimate. In a very low beam coherence time scenario, the XR headset must omit the DOA estimates since they will not accurately represent the pose. The beam coherence time for translational motion was introduced in [55] but the rotation included version is an open research field.

B. DOA Estimation Methods

The concept design XR headset structure in Fig. (1) contains mmWave uniform planar antenna arrays that contribute to the headset pose estimation by preforming DOA estimation of the incoming cellular (5G/6G) signals. DOA estimation leverages a multi-element antenna array to and the phases of an incoming signal at each antenna element to determine the signal DOA [56], similar to how a phased array antenna is steered.

One such DOA estimator is the Multiple Signal Classification (MUSIC) algorithm [57]. The algorithm decomposes the covariance matrix of the received signal to generate a signal subspace. Using the orthogonality of the signal and the noise subspace, the MUSIC algorithm estimates the DOA [56], [58], [59]. The MUSIC algorithm measures different received signals simultaneously with high precision, making it a good candidate for DOA estimation for XR headset with multiple base stations.

There are other versions of DOA estimators and MUSIC algorithms with various various trade offs in implementation and computation time [60], [61]. The MUSIC algorithm is suitable for arbitrary system geometry but computationally inefficient [61]. Despite the computationally inefficiency of

MUSIC, it was selected as a basis for the analysis of DOA estimates due to the tractability of its analysis.

C. DOA Estimate Error Covariance Bounds

For a linear array with n elements the MUSIC signal model for the received signal from each transmitter $\boldsymbol{y}(t)$ when m signals are present is a function of the complex signal $\boldsymbol{x}(t) = [\gamma_1(t)e^{j\omega_1^{\text{rx}}t}, \ldots, \gamma_n(t)e^{j\omega_n^{\text{rx}}t}]^{\text{T}}$ with amplitudes γ_n and phase ω_n^{rx} at each antenna element.

$$\boldsymbol{y}(t) = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{x}(t) + \boldsymbol{e}(t) \tag{4}$$

The matrix $\boldsymbol{A}(\boldsymbol{\theta})$ represents a matrix of transfer vectors $\boldsymbol{a}(\omega) = [1, e^{j\omega}, \ldots, e^{j(m-1)\omega}]^{\text{T}}$ where $\boldsymbol{A}(\boldsymbol{\theta}) = [\boldsymbol{a}(\omega_1), \ldots, \boldsymbol{a}(\omega_n)]$ with the vector $\boldsymbol{\theta} = [\omega_1, \ldots, \omega_n]$ of angles of arrival that produce m unique signals in $\boldsymbol{y}(t)$. The noise $\boldsymbol{e}(t)$ is an n dimension zero mean and Gaussian distributed with covariance $\sigma^2\boldsymbol{I}$. In order to analyze the benefit of utilizing DOA estimates to aid headset positioning the estimator's error covariance is needed. The covariance matrix $\boldsymbol{R}$ of the received signal is defined by the covariance matrix $\boldsymbol{P}$ of the signal $\boldsymbol{x}(t)$ and the transfer vectors $\boldsymbol{A}(\boldsymbol{\theta})$ as shown below.

$$\boldsymbol{R} = \mathbb{E}[\boldsymbol{y}(t)\boldsymbol{y}(t)^*] = \boldsymbol{A}(\boldsymbol{\theta})\boldsymbol{P}\boldsymbol{A}(\boldsymbol{\theta}) + \sigma^2\boldsymbol{I} \tag{5}$$

The error covariance of an angle of arrival estimate $\hat{\boldsymbol{\theta}}$ is lower bounded by the Cramér–Rao bound (CRB). The CRB for the MUSIC angle of arrival estimator is well studied in the literature [62] and the explicit CRB for the MUSIC estimator is known. The CRB of a signal with angle of arrival θ, is a function of θ is dependent on the matrix $\boldsymbol{A}$, a derivative matrix $\boldsymbol{D} = [\boldsymbol{d}(\omega_1), \ldots, \boldsymbol{d}(\omega_n)]$ where $\boldsymbol{d}(\omega_n) = \frac{d\boldsymbol{a}(\omega)}{d\omega}$ and $\boldsymbol{X} = \boldsymbol{x}(t)\boldsymbol{I}^{n\times n}$.

$$\begin{aligned}\text{CRB}(\hat{\boldsymbol{\theta}}) = \frac{\sigma}{2}\Bigg\{\sum_{t=1}^{N}\text{Re}[\text{X}^*(\text{t})\text{D}^*\\ [\boldsymbol{I} - \boldsymbol{A}(\boldsymbol{A}^*\boldsymbol{A})^{-1}\boldsymbol{A}^*]\boldsymbol{D}\boldsymbol{X}(t)]]\Bigg\}^{-1}\end{aligned} \tag{6}$$

This however becomes intractable to as it is dependent on the amplitude of the noise free signal at each array element $\boldsymbol{X}(t)$. As the number of array elements m and the number of measurements N of the received signal increases the CRB approaches a lower bound that can be used to approximate the variance of the angle of arrival estimate $\hat{\boldsymbol{\theta}}$ [62] if the signals are uncorrelated.

$$\text{CRB} \geq \frac{6}{\text{m}^3\text{N}}\text{I}^{\text{m}\times\text{m}}\text{S} \tag{7}$$

Where $\boldsymbol{S}$ is a matrix of the SNR values of each of the m signals.

$$\boldsymbol{S} = [\text{SNR}_1, \text{SNR}_2, \ldots, \text{SNR}_\text{m}]^{\text{T}} \tag{8}$$

For the high frequencies that will be used in the communications link for an XR headsets a large number of elements will be required in the phased array satisfying the first assumption. The waveforms will likely be OFDM which can be modeled as Gaussian in the time domain. If neighboring base stations utilize different pilot locations in the OFDM waveform, and the number of sub carriers is large, signals from neighboring base stations can be considered uncorrelated allowing the approximate CRB shown in (7) to be used.

D. Simulating DOA Estimates

A simulated set of DOA estimates can be generated using the know pose of a simulated cellular receiver. The poses of the receiver $\boldsymbol{r}_{\text{W}}$ shown in Fig. (5) defined in the world frame are used to generate simulated DOA estimates. The cellular antenna is modeled as two colocated linear arrays aligned with the B frame x and z axis. The location of simulated base stations Fig. (4) are each defined as $\boldsymbol{b}_{\text{W}}$ in the world frame. A unit vector $\hat{\boldsymbol{d}}_{\text{B}}^k$ at time k in the body frame defined according to the model (9) pointing from the receiver to the base station where $\bar{\boldsymbol{R}}_{\text{BW}}$ is the initial rotation matrix between the world frame and the body frame and $\boldsymbol{R}(\hat{e})$ is the estimated error rotation matrix as a function of $\hat{\boldsymbol{e}}$ the estimated error Euler angle.

$$\hat{\boldsymbol{d}}_{\text{B}}^k = \frac{\boldsymbol{r}_{\text{W}}^k - \boldsymbol{b}_{\text{W}}}{||\boldsymbol{r}_{\text{W}}^k - \boldsymbol{b}_{\text{W}}||}(\boldsymbol{R}(\hat{e})\bar{\boldsymbol{R}}_{\text{BW}}) \tag{9}$$

The DOA estimate at time k is simulated as azimuth and elevation angles. The azimuth $\hat{\theta}^k$ and the elevation angle $\hat{\phi}^k$ at time index k are defined as

$$\hat{\theta}^k = \arctan\left(\frac{\hat{\boldsymbol{d}}_{\text{B}}^k \cdot \boldsymbol{u}_{\text{By}}}{\hat{\boldsymbol{d}}_{\text{B}}^k \cdot \boldsymbol{u}_{\text{Bx}}}\right) + v_\theta \tag{10}$$

$$\hat{\phi}^k = \arctan\left(\frac{\hat{\boldsymbol{d}}_{\text{B}}^k \cdot \boldsymbol{u}_{\text{Bz}}}{\sqrt{(\hat{\boldsymbol{d}}_{\text{B}}^k \cdot \boldsymbol{u}_{\text{By}})^2 + (\hat{\boldsymbol{d}}_{\text{B}}^k \cdot \boldsymbol{u}_{\text{Bx}})^2}}\right) + v_\phi \tag{11}$$

in terms of the unit vector $\hat{\boldsymbol{d}}_{\text{B}}^k$ pointing to the base station and the unit vectors $\boldsymbol{u}_{\text{Bx}}$, $\boldsymbol{u}_{\text{By}}$, $\boldsymbol{u}_{\text{Bz}}$ pointing the x,y,z directions in the body frame.

The noise component of the DOA estimates v_θ and v_ϕ are modeled as zero mean and variance equal to the CRB shown in (7). The MUSIC estimator is asymptotically efficient as m and N are increased [62] so the CRB will represent a lower bound on the estimate error covariance. This assumes the MUSIC DOA estimator is an efficient estimator and achieves the CRB in order to lower bound the variance of the estimator to represent the best possible performance achievable from MUSIC based DOA estimates.

E. DOA Aided Pose Estimation

The XR headset pose and twist are estimated by incorporating the measurement models of the DOA measurements into PpEngine, the Radionavigation Laboratory's (RNL) precise positioning engine [1], [3]–[5]. PpEngine was modified to incorporate DOA estimates produced by the DOA simulator into PpEngine as DOA measurement. The DOA measurement models utilized by PpEngine are the same models (10) and (11) used to generate the simulated estimates of the azimuth

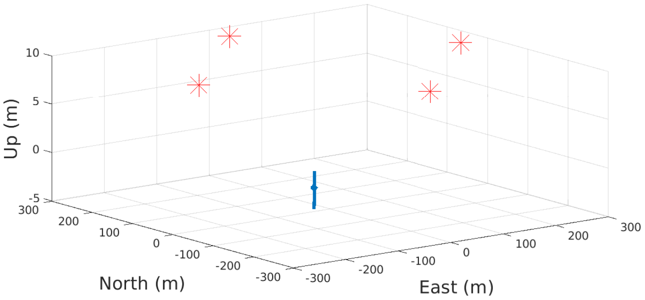


Fig. 4: Plot of simulated base station locations. The red asterisk represent the location of the simulated cellular base stations in the W frame. The blue line shows the trajectory of the cellular receiver relative to the base stations.

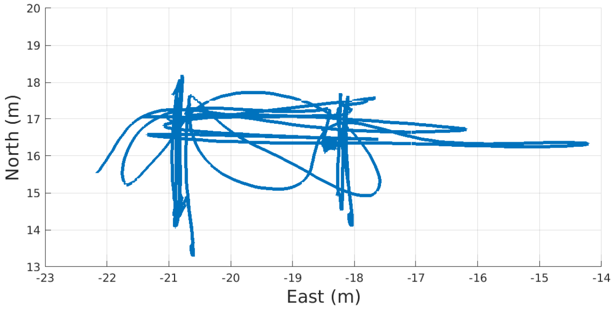


Fig. 5: Plot of XR headset trajectory in the W frame used to generate simulated DOA estimates.

and elevation angles respectively. The DOA measurements are brought into the estimator as shown in Fig. (2) along with the measurement error covariance. Measurements from an inertial measurement unit (IMU), GNSS observables from the GNSS receiver are also incorporated into PpEngine. The state of the PpEngine estimator $\boldsymbol{X}$ contains the pose and twist in addition to IMU bias parameters as shown below:

$$\boldsymbol{X} = \left\{\boldsymbol{r}_{\mathrm{W}}^{k}, \boldsymbol{v}_{\mathrm{W}}^{k}, \boldsymbol{e}, \boldsymbol{b}_{\mathrm{U}}^{\mathrm{a}}, \boldsymbol{b}_{\mathrm{U}}^{\mathrm{g}}\right\} \tag{12}$$

The position $\boldsymbol{r}_{\mathrm{W}}^{k}$, $\boldsymbol{e}$, velocity $\boldsymbol{v}_{\mathrm{W}}^{k}$, IMU accelerometer bias $\boldsymbol{b}_{\mathrm{U}}^{\mathrm{a}}$, and IMU gyroscope bias $\boldsymbol{b}_{\mathrm{U}}^{\mathrm{g}}$ are estimated by PpEngine. The pose and twist estimates produced by PpEngine are then passed to GEOSLAM Fig. (2) to constrain the SLAM BA as describe in previous sections.

VII. Results

This section contains the results from performing visual SLAM using the GEOSLAM algorithm against a test dataset and evaluating the results. A simulation incorporating DOA measurements into the pose estimator was performed to determine the benefit to the pose estimator during simulated GNSS outages.

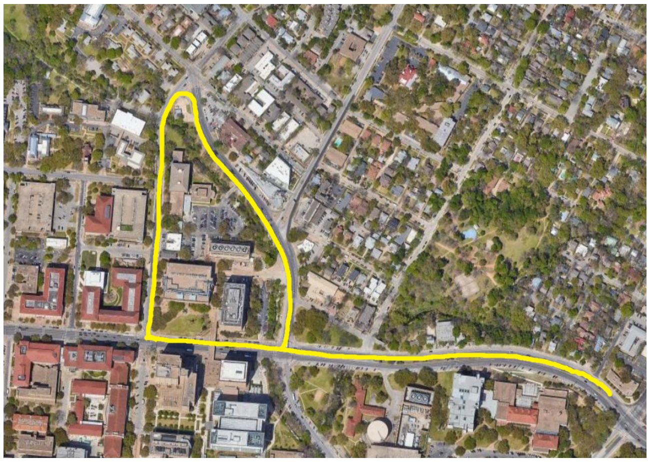


Fig. 6: Map showing the portion of the TEX-CUP dataset tested. Starting from the bottom right, the car travels west and then performs two loops as shown. Each loop is approximately 1 km.

A. GEOSLAM Results

1) Dataset: The GEOSLAM pipeline was tested on the Texas Challenge for Urban Positioning (TEX-CUP) dataset [43]. This dataset contains monochromatic 2048x732 resolution images captured at 10Hz via stereo cameras, shown in Fig. (3). The camera intrinsics were calibrated using Kalibr [63]. In addition, the TEX-CUP dataset contains dual frequency (L1/L2) GNSS observables recorded using the RNL's *RadioLynx* GNSS front-end and processed with the PpRx software-defined GNSS receiver [4], [64]–[67] and inertial measurements from a LORD MicroStrain 3DM-GX5-25 AHRS industrial-grade IMU [43]. Other sensing sources are included in the dataset but were not utilized in this experiment.

Ground truth for TEX-CUP is provided by the iXblue ATLANS-C's post-processing software and is sub-centimeter-accurate.

2) GEOSLAM Performance: The performance of GEOSLAM was tested using two different sections of the TEX-CUP dataset consisting of two passes over the same area, shown in Fig. (6). During the first pass through the loop, GEOSLAM utilized stereo camera images and pose estimates provided by PpEngine to produce camera poses and a globally-referenced feature map. The errors in pose estimate produced by GEOSLAM shown in Fig. (7) exhibit a position RMSE of 1.38 cm and an orientation RMSE of 1.30°.

The second test was performed on the segment of the TEX-CUP dataset corresponding to the second loop around the block shown in Fig. (6). For this test, no CDGNSS based pose estimates were utilized by GEOSLAM. Instead, the feature map generated from the previous test was used to test the accuracy of the map. During this test Fig. (7) shows that a larger error in the camera poses was observed with a position RMSE of 5.47 cm. The map produced by GEOSLAM was able to constrain the orientation errors with an orientation RMSE

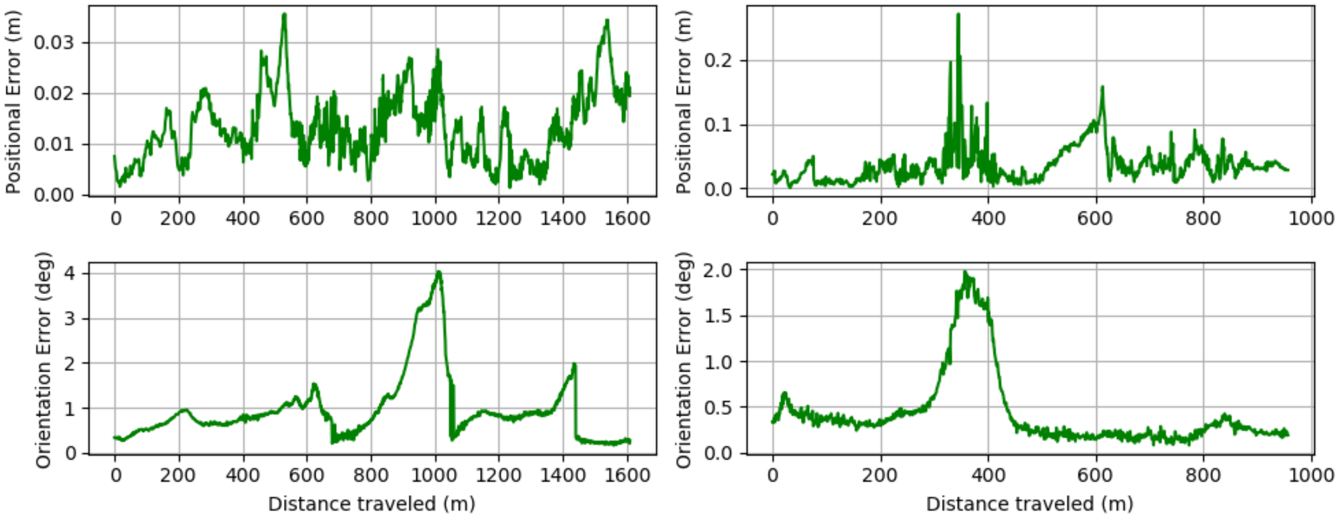


Fig. 7: Positional and orientation drift with respect to ground truth for the two loops. Left panels: Loop 1, performed with CDGNSS aiding on unmapped terrain. Right panels: Loop 2, performed without CDGNSS aiding on Loop 1's created map.

of 0.58°.

This visual-only test showed that the map created is nearly centimeter-accurate and that GEOSLAM can collaboratively add to existing feature maps. Areas of larger error correspond with poorer map quality. Further loops can prune out these poor quality keyframes and further reduce error. Importantly, the second loop exhibits higher amounts of jittering than the first test. When implemented into an XR headset, this jittering behavior may be disorienting for the user. This could be reduced by increasing the length of the BA window or by feeding back the camera poses into PpEngine, the GNSS based pose estimator, to jointly estimate the drift rate of the GEOSLAM camera poses. The latter would complete the diagram introduced in Fig.(2).

B. DOA Simulation Results

The effect of incorporating DOA measurements on headset pose estimation was performed by simulating DOA measurements, as discussed in Section VI using a dataset of a GNSS observables collected using the RNL's *RadioLynx* GNSS front-end and processed with the PpRx software-defined GNSS receiver [4], [64]–[67]. The GNSS observables were then coupled with IMU measurements and processed to create a cm-accurate trajectory shown in Fig. (5). The simulated DOA measurements were simulated using an array size of 16, an SNR of 10 dB, and assuming one measurement of the received signal is used for DOA estimation. The trajectory was then used to simulate noisy DOA measurements that could be incorporated into the pose estimator. It is important to note that the DOA measurements did not take into account multipath and assume the DOA of the signal is within line of sight.

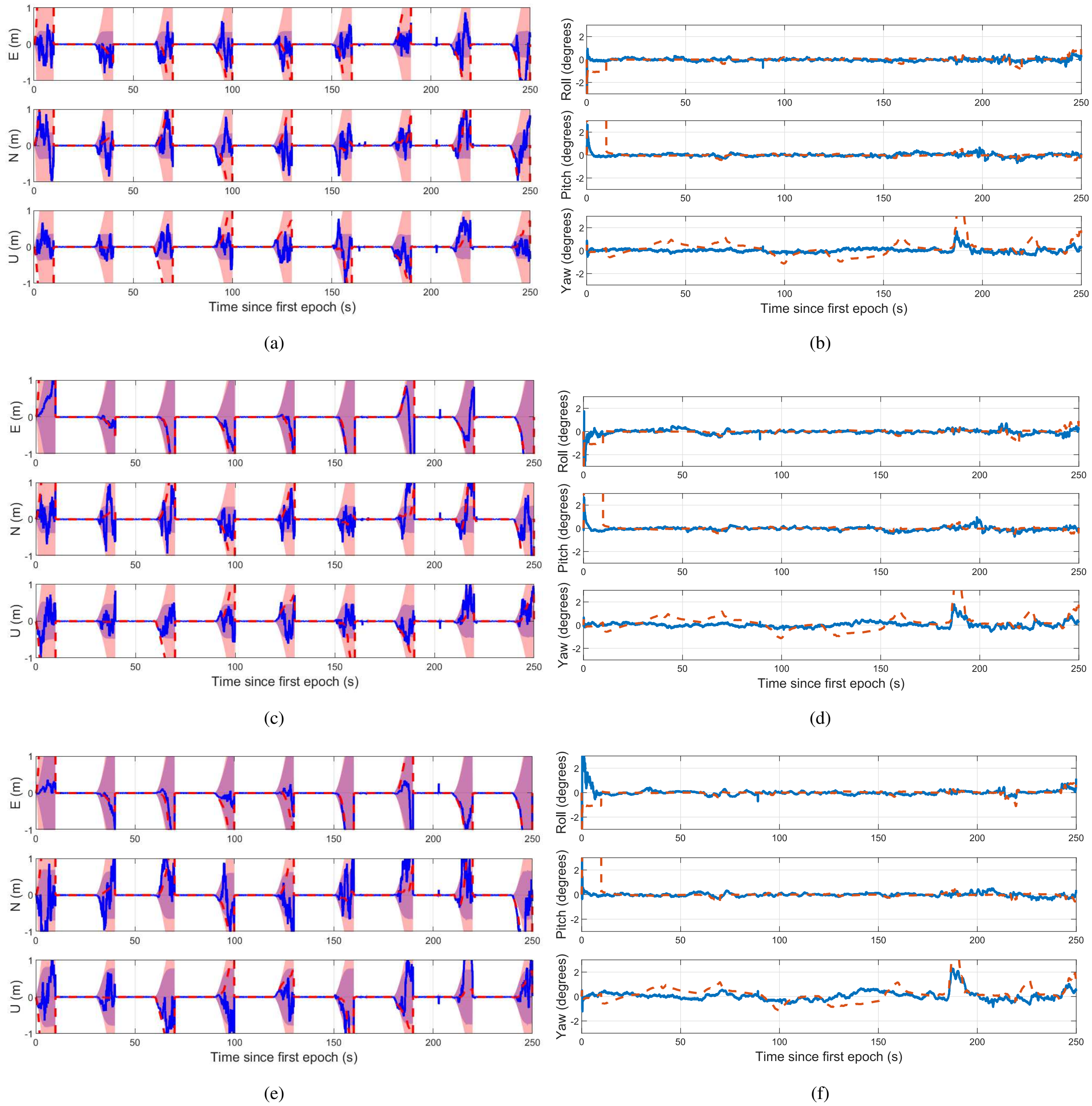
The estimator was then tested by selectively removing the GNSS observables to simulate a 10 second GNSS outage. The outages repeated every 30 seconds and the position and orientation errors were evaluated. The first experiment was performed by simulating DOA measurements to 4 separate base stations Fig (4) at a rate of 20 Hz. This resulted in a pose estimate and orientation estimate that constrained the error growth of the IMU during simulated GNSS outages as shown in Fig. (8), with a position RMSE below 15 cm and orientation RMSE of 0.17 degrees shown in Table III. This is a large improvement compared to the case when no DOA measurements were available during a simulated outage as shown in Table III.

TABLE III: Pose RMSE During GNSS Outage

Base Stations	DOA Rate (Hz)	RMSE (meters)	RMSE (degrees)
4	20	0.1438	0.1674
2	20	0.2266	0.2825
2	5	0.2446	0.5990
0	0	1.2174	0.9104

This experiment was repeated with more realistic simulated DOA measurements. It is unlikely that a XR headset will receive a signal from multiple base stations. To account for this scenario the number of base stations was reduced to 2. As a result the RMSE error increased as shown in Table III, and the error time histories are shown in Fig. (8).

A third experiment was performed to determine the effects of the DOA measurement rate. There is a trade-off between the amount of time a communications receiver will be processing a received signal for DOA estimation; for this reason, the rate of simulated DOA measurements was reduced to 5 Hz. In addition, one of the purposes of precise pose estimation is for beam forming and the cellular radio will likely rely on the estimated poses to inform beam forming during highly dynamic motion, and it is likely that the DOA of a signal will not be estimated. For this reason, for any simulated DOA



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 8: This figure shows the effect of simulated GNSS outages on the pose estimate in 6Dof. Panel (a) shows the plot of position errors in the W frame in meters during simulated 10 second GNSS outages with 4 visible base stations. The red line represents the position error without DOA measurements while the blue line is the position error with DOA measurements. The red and blue shaded regions are the one standard deviation error of the position estimate for each case. Panel (b) shows the attitude errors expressed as Euler angles during simulated 10 second GNSS outages with 4 visible base stations. The dotted orange line represents the attitude error without DOA measurements while the blue line is the position error with DOA measurements. Panels (c) and (d) are the same plots as (a) and (b) except with only 2 base stations visible to the receiver. Panels (e) and (f) are the same as (c) and (d) except the DOA measurements are only given to the pose estimator at 5Hz, and any measurement taken when the angular rate of the headset is above 180 degrees per second are thrown out.

measurement that occurred when the headset was rotating faster than 180 degrees per second (dps), the simulated measurement was discarded in the simulation. This resulted in less precise pose estimates as shown in the increased pose RMSE in Table III and in the time histories in Fig. (8). This simulation highlights the limitations of DOA measurements for pose estimation. Although the addition of DOA measurements under ideal conditions when multiple base stations are visible, under more realistic conditions provide a sub-meter accuracy on the position, and sub-degree accuracy of the orientation. The results demonstrate that DOA measurements constrain the error growth during a GNSS outage, but additional sources of measurements—such as visual SLAM—will need to be incorporated into the estimator to increase the precision of the pose estimate during GNSS outages.

VIII. Conclusion

This paper outlined a method of pose estimation providing a robust, outdoor, globally-referenced pose for XR headsets. GEOSLAM produced cm-accurate globally-referenced feature maps that were then used to produce cm-accurate pose estimates. An efficient method of cloud offloading GEOSLAM was developed which sends image feature points to allow bundle adjustment, the most computationally expensive portion of GEOSLAM, to be processed in the cloud. The proposed cloud processing regime also allowed for the creation of precise feature maps when CDGNSS is available that can be utilized for periods of GNSS outages with a position RMSE of 5.47 cm and orientation RMSE of 0.58 degrees. The pose estimator's resilience to GNSS outages was also demonstrated by incorporating simulated DOA measurements. The addition of DOA measurements was able to constrain the RMSE of the position estimate during simulated GNSS outages below 25 centimeters.

Acknowledgment

Research was sponsored by affiliates of the 6G@UT center within the Wireless Networking and Communications Group at The University of Texas at Austin.

References

[1] T. E. Humphreys, R. X. T. Kor, and P. A. Iannucci, "Open-world virtual reality headset tracking," in *Proceedings of the ION GNSS+ Meeting*, Online, 2020.

[2] M. Stranner, C. Arth, D. Schmalstieg, and P. Fleck, "A high-precision localization device for outdoor augmented reality," in *2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. IEEE, 2019, pp. 37–41.

[3] T. E. Humphreys, M. J. Murrian, and L. Narula, "Deep-urban unaided precise global navigation satellite system vehicle positioning," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 3, pp. 109–122, 2020.

[4] J. E. Yoder, P. A. Iannucci, L. Narula, and T. E. Humphreys, "Multi-antenna vision-and-inertial-aided CDGNSS for micro aerial vehicle pose estimation," in *Proceedings of the ION GNSS+ Meeting*, Online, 2020, pp. 2281–2298.

[5] J. E. Yoder and T. E. Humphreys, "Low-cost inertial aiding for deep-urban tightly-coupled multi-antenna precise GNSS," *Navigation, Journal of the Institute of Navigation*, vol. 70, no. 1, 2023.

[6] D. C. Niehorster, L. Li, and M. Lappe, "The accuracy and precision of position and orientation tracking in the HTC Vive virtual reality system for scientific research," *i-Perception*, vol. 8, no. 3, 2017.

[7] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE, 2007, pp. 225–234.

[8] D. P. Shepard and T. E. Humphreys, "High-precision globally-referenced position and attitude via a fusion of visual SLAM, carrier-phase-based GPS, and inertial measurements," in *Proceedings of the IEEE/ION PLANS Meeting*, May 2014.

[9] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, "Keyframe-based visual-inertial slam using nonlinear optimization," *Proceedings of Robotis Science and Systems (RSS) 2013*, 2013.

[10] S. Leutenegger, "Okvis2: Realtime scalable visual-inertial slam with loop closure," *arXiv preprint arXiv:2202.09199*, 2022.

[11] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[12] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[13] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[14] H. Liu, M. Chen, G. Zhang, H. Bao, and Y. Bao, "Ice-ba: Incremental, consistent and efficient bundle adjustment for visual-inertial slam," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1974–1982.

[15] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015, pp. 298–304.

[16] L. Narula, J. M. Wooten, M. J. Murrian, D. M. LaChapelle, and T. E. Humphreys, "Accurate collaborative globally-referenced digital mapping with standard GNSS," *Sensors*, vol. 18, no. 8, 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/8/2452

[17] V. Ila, L. Polok, M. Solony, and P. Svoboda, "SLAM++-a highly efficient and temporally scalable incremental SLAM framework," *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 210–230, 2017.

[18] H. Strasdat, J. Montiel, and A. J. Davison, "Visual SLAM: Why filter?" *Image and Vision Computing*, 2012.

[19] B. Garigipati, N. Strokina, and R. Ghabcheloo, "Evaluation and comparison of eight popular lidar and visual SLAM algorithms," in *2022 25th International Conference on Information Fusion (FUSION)*. IEEE, 2022, pp. 1–8.

[20] A. George **et al.**, "Analysis of visual-inertial odometry algorithms for outdoor drone applications," 2021.

[21] S. Cao, X. Lu, and S. Shen, "GVINS: Tightly coupled GNSS–visual–inertial fusion for smooth and consistent state estimation," *IEEE Transactions on Robotics*, 2022.

[22] Z. Wang, M. Li, D. Zhou, and Z. Zheng, "Direct sparse stereo visual-inertial global odometry," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 403–14 409.

[23] J. Liu, W. Gao, and Z. Hu, "Optimization-based visual-inertial SLAM tightly coupled with raw GNSS measurements," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11 612–11 618.

[24] B. Congram and T. D. Barfoot, "Relatively lazy: Indoor-outdoor navigation using vision and GNSS," in *2021 18th Conference on Robots and Vision (CRV)*. IEEE, 2021, pp. 25–32.

[25] R. Zhai and Y. Yuan, "A method of vision aided GNSS positioning using semantic information in complex urban environment," *Remote Sensing*, vol. 14, no. 4, p. 869, 2022.

[26] R. Sun, Y. Yang, K.-W. Chiang, T.-T. Duong, K.-Y. Lin, and G.-J. Tsai, "Robust IMU/GPS/VO integration for vehicle navigation in GNSS degraded urban areas," *IEEE Sensors Journal*, vol. 20, no. 17, pp. 10 110–10 122, 2020.

[27] X. Chen, W. Hu, L. Zhang, Z. Shi, and M. Li, "Integration of low-cost GNSS and monocular cameras for simultaneous localization and mapping," *Sensors*, vol. 18, no. 7, p. 2193, 2018.

[28] W. Lee, P. Geneva, Y. Yang, and G. Huang, "Tightly-coupled GNSS-aided visual-inertial localization," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9484–9491.

[29] P. Henkel, A. Blum, and C. Günther, "Precise RTK positioning with GNSS, INS, barometer and vision," in *Proceedings of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017)*, 2017, pp. 2290–2303.

[30] X. Li, S. Li, Y. Zhou, Z. Shen, X. Wang, X. Li, and W. Wen, "Continuous and precise positioning in urban environments by tightly coupled integration of GNSS, INS and vision," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 458–11 465, 2022.

[31] T. Li, H. Zhang, Z. Gao, X. Niu, and N. El-Sheimy, "Tight fusion of a monocular camera, MEMS-IMU, and single-frequency multi-GNSS RTK for precise navigation in GNSS-challenged environments," *Remote Sensing*, vol. 11, no. 6, p. 610, 2019.

[32] S. Gu, C. Dai, F. Mao, and W. Fang, "Integration of multi-GNSS PPP-RTK/INS/vision with a cascading kalman filter for vehicle navigation in urban areas," *Remote Sensing*, vol. 14, no. 17, p. 4337, 2022.

[33] X. Li, X. Li, S. Li, Y. Zhou, M. Sun, Q. Xu, and Z. Xu, "Centimeter-accurate vehicle navigation in urban environments with a tightly integrated PPP-RTK/MEMS/vision system," *GPS Solutions*, vol. 26, no. 4, pp. 1–17, 2022.

[34] T. Liu, B. Li, L. Yang, L. He, J. He *et al.*, "Tightly coupled GNSS, INS and visual odometry for accurate and robust vehicle positioning," in *2022 5th International Symposium on Autonomous Systems (ISAS)*. IEEE, 2022, pp. 1–6.

[35] B. A. Erol, S. Vaishnav, J. D. Labrado, P. Benavidez, and M. Jamshidi, "Cloud-based control and vSLAM through cooperative mapping and localization," in *2016 World Automation Congress (WAC)*. IEEE, 2016, pp. 1–6.

[36] P. Sossalla, J. Rischke, J. Hofer, and F. H. Fitzek, "Evaluating the advantages of remote SLAM on an edge cloud," in *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, 2021, pp. 01–04.

[37] S. Dey and A. Mukherjee, "Robotic slam: a review from fog computing and mobile edge computing perspective," in *Adjunct Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services*, 2016, pp. 153–158.

[38] J. Salmerón-Garcı, P. Inigo-Blasco, F. Dı, D. Cagigas-Muniz *et al.*, "A tradeoff analysis of a cloud-based robot navigation assistant using stereo image processing," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 444–454, 2015.

[39] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile networks and Applications*, vol. 18, no. 1, pp. 129–140, 2013.

[40] A. Pages-Zamora, J. Vidal, and D. H. Brooks, "Closed-form solution for positioning based on angle of arrival measurements," in *The 13th IEEE international symposium on personal, indoor and mobile radio communications*, vol. 4. IEEE, 2002, pp. 1522–1526.

[41] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[42] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration." *VISAPP (1)*, vol. 2, no. 331-340, p. 2, 2009.

[43] L. Narula, D. M. LaChapelle, M. J. Murrian, J. M. Wooten, T. E. Humphreys, J.-B. Lacambre, E. de Toldi, and G. Morvant, "TEX-CUP: The University of Texas Challenge for Urban Positioning," in *Proceedings of the IEEE/ION PLANSx Meeting*, 2020.

[44] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, 2nd ed. Cambridge University Press, 2003.

[45] S. Agarwal, K. Mierle, and T. C. S. Team, "Ceres Solver," 3 2022. [Online]. Available: https://github.com/ceres-solver/ceres-solver

[46] O. Sorkine-Hornung and M. Rabinovich, "Least-squares rigid motion using SVD," *Computing*, vol. 1, p. 1, 2017.

[47] J. Martínez-Carranza, N. Loewen, F. Márquez, E. O. García, and W. Mayol-Cuevas, "Towards autonomous flight of micro aerial vehicles using orb-slam," in *2015 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*. IEEE, 2015, pp. 241–248.

[48] L. Qingqing, J. P. Queralta, T. N. Gia, H. Tenhunen, Z. Zou, and T. Westerlund, "Visual odometry offloading in internet of vehicles with compression at the edge of the network," in *2019 Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU)*. IEEE, 2019, pp. 1–2.

[49] L.-Y. Duan, J. Lin, J. Chen, T. Huang, and W. Gao, "Compact descriptors for visual search," *IEEE MultiMedia*, vol. 21, no. 3, pp. 30–40, 2014.

[50] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[51] H.-J. Chien, C.-C. Chuang, C.-Y. Chen, and R. Klette, "When to use what feature? SIFT, SURF, ORB, or A-KAZE features for monocular visual odometry," in *2016 International Conference on Image and Vision Computing New Zealand (IVCNZ)*. IEEE, 2016, pp. 1–6.

[52] D. Van Opdenbosch and E. Steinbach, "Collaborative visual slam using compressed feature exchange," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 57–64, 2018.

[53] L. Baroffio, M. Cesana, A. Redondi, M. Tagliasacchi, and S. Tubaro, "Coding visual features extracted from video sequences," *IEEE transactions on Image Processing*, vol. 23, no. 5, pp. 2262–2276, 2014.

[54] D. Astely and B. Ottersten, "The effects of local scattering on direction of arrival estimation with MUSIC," *IEEE transactions on Signal Processing*, vol. 47, no. 12, pp. 3220–3234, 1999.

[55] V. Va, J. Choi, and R. W. Heath, "The impact of beamwidth on temporal channel variation in vehicular channels and its implications," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5014–5029, 2016.

[56] H. Tang, "DOA estimation based on MUSIC algorithm," 2014.

[57] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276 – 280, Mar. 1986.

[58] P.-J. Chung, M. Viberg, and J. Yu, "DOA estimation methods and algorithms," in *Academic Press Library in Signal Processing*. Elsevier, 2014, vol. 3, pp. 599–650.

[59] P. Gupta and S. Kar, "MUSIC and improved MUSIC algorithm to estimate direction of arrival," in *2015 International Conference on Communications and Signal Processing (ICCSP)*. IEEE, 2015, pp. 0757–0761.

[60] F. Gross, "Smart antennas for wireless communications with MATLAB," *McGraw Hills*, 2005.

[61] N. Ruan, H. Wang, F. Wen, and J. Shi, "Doa estimation in b5g/6g: Trends and challenges," *Sensors*, vol. 22, no. 14, p. 5125, 2022.

[62] P. Stoica and A. Nehorai, "Music, maximum likelihood, and cramer-rao bound," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 5, pp. 720–741, 1989.

[63] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1280–1286.

[64] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, and P. M. Kintner, Jr., "GNSS receiver implementation on a DSP: Status, challenges, and prospects," in *Proceedings of the ION GNSS Meeting*. Fort Worth, TX: Institute of Navigation, 2006, pp. 2370–2382.

[65] E. G. Lightsey, T. E. Humphreys, J. A. Bhatti, A. J. Joplin, B. W. O'Hanlon, and S. P. Powell, "Demonstration of a space capable miniature dual frequency GNSS receiver," *Navigation*, vol. 61, no. 1, pp. 53–64, Mar. 2014.

[66] T. E. Humphreys, J. Bhatti, T. Pany, B. Ledvina, and B. O'Hanlon, "Exploiting multicore technology in software-defined GNSS receivers," in *Proceedings of the ION GNSS Meeting*. Savannah, GA: Institute of Navigation, 2009, pp. 326–338.

[67] Z. Clements, P. A. Iannucci, T. E. Humphreys, and T. Pany, "Optimized bit-packing for bit-wise software-defined GNSS radio," in *Proceedings of the ION GNSS+ Meeting*, St. Louis, MO, 2021, pp. 3749–3771.