Toward In-Context Environment Sensing for Mobile Augmented Reality

Yiqin Zhao
Worcester Polytechnic Institute
Worcester, MA, USA
yzhao11@wpi.edu

Ashkan Ganj
Worcester Polytechnic Institute
Worcester, MA, USA
aganj@wpi.edu

Tian Guo
Worcester Polytechnic Institute
Worcester, MA, USA
tian@wpi.edu

Abstract

Environment sensing is a fundamental task in mobile augmented reality (AR). However, on-device sensing and computing resources often limit mobile AR sensing capability, making high-quality environment sensing challenging to achieve. In recent years, in-context sensing, a new sensing system design paradigm, has emerged with the promise of achieving accurate, efficient, and robust sensing results. In this work, we first formally define the in-context sensing design paradigm. We summarize its primary challenges as the uncertainty of environmental information availability. To quantify the impact of sensing context data, we present two in-depth case studies that show how it can impact different aspects of mobile AR sensing systems.

CCS Concepts

• **Computing methodologies** → **Mixed / augmented reality**; • **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**.

Keywords

Mobile AR; context awareness; environment understanding

ACM Reference Format:
Yiqin Zhao, Ashkan Ganj, and Tian Guo. 2024. Toward In-Context Environment Sensing for Mobile Augmented Reality. In *The 30th Annual International Conference on Mobile Computing and Networking (ACM MobiCom '24), November 18–22, 2024, Washington D.C., DC, USA*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3636534.3696211

ACM MobiCom '24, November 18–22, 2024, Washington D.C., DC, USA

1 Introduction

Understanding the physical environment is a fundamental task for mobile augmented reality (AR), which aims to integrate virtual and physical content seamlessly. This goal requires accurate and robust environment sensing across multiple properties, including, but not limited to, device motion, object distances, and environmental lighting [9]. The quality of environment sensing directly contributes to the mobile AR user experiences. For example, camera depth information is often needed to place virtual objects at the correct distance in virtual try-on applications [27].

Traditional AR systems typically rely on computer vision methods to extract the environment information from the current AR device camera image [4]. However, as a single input source, the current camera image often does not offer sufficient environmental information to support the desired immersive mobile AR user experiences. As the software and hardware ecosystem evolves, mobile AR environment sensing systems have started shifting to taking more comprehensive environment information from multi-sensor and multi-modal inputs. Unfortunately, sensing systems on mobile AR devices are often constrained by many physical limitations in computing and sensory devices, rendering the raw on-device sensing power insufficient to match the complexity of the physical environment.

More recently, new research works have started to leverage contextual information from *user, device, and environment elements* to improve mobile AR sensing systems' capability [7, 19, 31]. In later sections, we refer to this context-aware environment-sensing paradigm as *in-context sensing*. By leveraging information from the sensing context, the in-context sensing design promises a future of AR environment sensing with better accuracy, efficiency, and robustness. In §2, we present a formal definition of the in-context sensing design paradigm.

However, adopting the in-context design for mobile AR sensing systems also presents unique challenges in acquiring and managing context data. Through a comprehensive survey of environment sensing tasks and AR system design support, we summarize the primary challenge of in-context

Table 1: *A survey of in-context environment-sensing system design and their context data usage.*

Category	Paper	Sensing Context Data		Sensing Context Data Contribution		
		Modality	Multi-timestamp	Accuracy	Robustness	Efficiency
Device Tracking	ORB-SLAM3. [3]	Acceleration, Orientation	✗	✓	✓	N/A
	Kimera-VIO. [17]	Acceleration, Orientation	✗	✓	✓	N/A
Object Detection	Zhang, Zhishuai, et al. [28]	Semantics	✗	✓	N/A	N/A
	Chen, Chenglizhao, et al. [6]	Depth	✗	✓	N/A	✓
Object Tracking	BundleTrack [22]	Depth	✓	N/A	N/A	N/A
	BundleSDF [23]	Depth	✓	✓	✓	N/A
Lighting Estimation	Xihe [30]	Depth, Device Pose	✓	✓	✓	✓
	LitAR [31]	Device Pose, Ambient Light, Depth	✓	✓	✓	N/A
Depth Estimation	Sparse SPN [21]	Device Pose	✓	N/A	N/A	✓
	Sartipi et al. [18]	Device Pose	✓	✓	N/A	✓

sensing system design as *the uncertainty of environment information availability*. In other words, although critical environment information may be presented and extracted from sensing context data, their presence is not guaranteed. In §3 and §4, we present two case studies to investigate: *(i)* how could different environment context data presence impact the accuracy of environment sensing? And *(ii)* what systematic designs can be made to address the uncertainty of context data? Our investigation looks into two representative sensing tasks: *metric depth estimation*, a task that demands precise environment observations, and *lighting estimation*, a task that requires broad environment observations.

In the first task, we investigate how metric depth estimation accuracy varies when metric depth estimation models are deployed to AR devices with different camera focal length configurations. We also show how the accuracy and efficiency of metric depth estimation models can be reliably improved by using camera parameter information and simple controls of camera focal length. In the second case study, we investigate how accumulated environment observation point clouds can contribute to the lighting estimation task. Compared to natural user mobility in an object placement task, lighting estimation accuracy can improve up to 40% with guided user movements. Similarly, point cloud sharing between nearby users can improve the estimation accuracy significantly by 33%. In both experiments, the interaction between the lighting estimation task and the information provides shows promising aspects of in-context sensing.

We summarize our main contribution as follows:

- We present a formulation of *in-context environment sensing*, an emerging environment sensing design paradigm for mobile AR that promises higher environment sensing quality with better accuracy, robustness, and efficiency.
- We present a survey on recent context-aware environment sensing system design and identify the primary design challenge as *the uncertainty of environment information availability*.
- We present two case studies on representative in-context sensing system designs of metric depth estimation and lighting estimation. Through the studies, we identify three opportunities to address the primary challenge: *(i)* sensory device manipulation, *(ii)* guided user mobility, and *(iii)* connected context sources.

2 Promises and Challenges

Pioneer research in context-aware AR systems demonstrated that important environment information can be extracted from camera frames for task planning and decision-making [20]. Taking inspiration from the prior research on how context can augment the sensing process, we define mobile AR in-context environment sensing as:

An environment sensing process that combines current AR camera image with information retrieved through interactions with an AR device, user, and environmental elements.

Broadly, sensing context data can be collected from on-device or externally connected sensors throughout the AR application session.

Promises of sensing context data. Table 1 summarizes our survey on recent mobile AR environment sensing systems with their respective usage of context data. We make several key observations from this survey. First, sensing context data improves sensing accuracy. Many systems utilize multi-modal and multi-timestamp context data to provide complementary information to camera image data. For example, device pose data generated by IMU sensors are often used in metric depth estimation systems to reduce the ambiguity of metric scale estimation [11, 12, 26]. These complementary data often provide information that is hard to extract from camera images. In particular, we have noticed device tracking and camera depth data are used across several sensing categories, potentially due to their critical role in reconstructing spatial and temporal environments.

Secondly, we noticed that sensing context data contributes to the overall robustness of the sensing system. In sensing

tasks with temporal optimizations, such as VI-SLAM systems [3, 17], the improvement of device tracking accuracy is associated with increased historical views. Additionally, awareness of environmental factors, such as localization data, can help reduce the ambiguity of the sensing goal. This information can be leveraged to help sensing systems adapt to different environments. For example, the estimation methods often differ for lighting estimation between indoor and output scenes [15, 16, 31].

Thirdly, sensing context data also contributes to the efficiency of environment sensing systems. As mobile AR sensing context data provides an understanding of the environment information, historical context can often be reused even when the device is moved to new positions. Some environment information, such as 3D geometry, can often be extracted from environment mapping results in sensing context data [29]. This information can help to reduce the dependencies of complicated 3D geometry processing in many DNN-based sensing systems and reduce the complexities of the DNN model. Additionally, reusing sensing context data can reduce sensory device activation [18], potentially saving mobile AR system power consumption.

Lastly, although sensing context contributes to overall accuracy improvements, common grounds of system-wide sensing context support has not been reached. We noticed that, although on-device sensor data can be directly leveraged for environment-sensing tasks, interacting between custom sensing systems and the sensors is difficult due to the limitations in hardware access. On the one hand, this limitation protects the information privacy of AR users [14, 19]. On the other hand, it prevents AR developers and researchers from building more sophisticated sensing systems. Allowing safe and flexible sensor hardware access to sensing systems is one of the primary open research questions. Additionally, cross-device connectivity is often limited to specific devices or proprietary solutions on current AR systems.

Primary challenge. Supporting in-context environment sensing is a difficult task with challenges ranging from multiple aspects, including data quality, network connectivity, and computational efficiency. Here, we summarize the primary challenge as *the uncertainty of environment information availability*. In other words, the availability of sensing context data strongly depends on the interactivity between sensing tasks and information providers, causing uncertainty in the overall sensing quality. A main impacting factor is the intended usage behavior of AR applications. In particular, the trajectory of the user's movement is usually influenced by the AR applications' design rather than the user's interest in environment sensing. For example, maximizing environment observation coverage through a moving camera is a key goal in achieving high-quality environment tracking and lighting estimation. Mismatched mobility interests usually cause harm to the quality of environmental sensing as they limit the environmental observation coverage, blur camera motions, and cause sensor drifts. Solving this challenge requires new system designs that model the uncertainty of sensing context and provide new ways of ensuring its quality, such as multi-sensor collaboration or guided user mobility.

3 Case Study: Metric Depth Estimation

Metric depth estimation plays an important role in AR applications, enabling the seamless integration of virtual objects into the physical world. However, single image depth estimation models, a simple and popular method for depth sensing, often face challenges like scale ambiguity, overfitting to specific camera models, and high model complexity, as discussed in our previous work [10] and various studies [11, 12, 26]. In this case study, through analyzing recent works, we demonstrate that high-quality depth estimation results on mobile AR can be achieved using camera parameters and simple controls of camera focal length.

Experiments Setup. For our experiments, we used three state-of-the-art (SOTA) models: ZoeDepth-M12-N [2], DepthAnything [24], and HybridDepth [8]. The first two are heavy and large single-image depth estimation models, while HybridDepth is a depth model that utilizes additional camera data (focal stack) for depth estimation. In particular, the focal stack is a set of images created with intentional manipulation of camera focus distances. We use this hardware manipulation method to demonstrate how metric depth estimation tasks can interact with device hardware.

We selected the ARKitScenes [1], an AR-focused dataset captured with mobile cameras. We loaded the models with the provided pre-trained weights and evaluated them on the official evaluation set of ARKitScenes. All evaluations were performed on an NVIDIA RTX 4090 GPU. We processed and resized the input images to the desired size for each model before feeding them into the models. To measure inference time, we recorded the average inference time of each model on the ARKitScenes dataset's evaluation set.

Device hardware-awareness. *(1) Scale Ambiguity.* Single Image Depth models cannot reliably determine the absolute scale of objects within a scene. This scale ambiguity arises because the models rely solely on visual cues without additional context, leading to inaccuracies in depth prediction. Even human eyes can be misled into thinking that the last two photos are taken at the same distance, but the actual measurements reveal that these images were taken at two different distances. This discrepancy occurs due to the nature of camera intrinsics, which include parameters like focal length and sensor size. These intrinsic parameters influence how

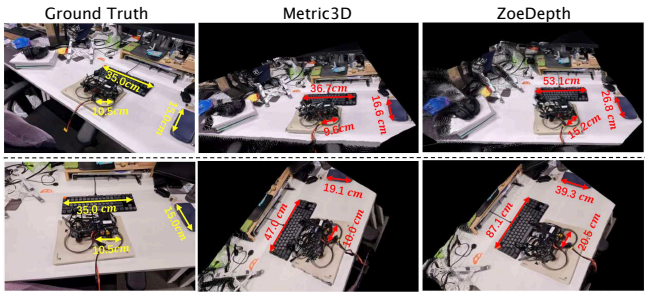


Figure 1: ***Visual examples of metric depth estimation results.*** Camera parameters provide important cues on metric depth estimation. With this information, Metric3D [13] significantly outperforms ZoeDepth [2], which uses only a single image as the input data.

the 3D world is projected onto the 2D image plane, causing similar objects at different distances to appear the same size. Without additional contextual information, such as focus distances or multiple viewpoints, the depth estimation model cannot resolve these ambiguities.

(2) Overfitting to Specific Camera Models. Single image depth models are often tailored to the characteristics of the training dataset, which typically involves images captured by specific camera models with a unique focal length and sensor size. As we discussed in the first challenge, this will cause some scale problems if we use a camera different from the training dataset. For instance, Figure 1 compares two depth models: Metric3D [12, 26], which integrate camera parameters during training, and ZoeDepth [2], which relies only on visual information. The same scene and objects were captured using two different mobile cameras with distinct camera parameters. ZoeDepth [2], which relies solely on visual data, shows significant variation in its measurement of the object (keyboard), with the perceived size changing based on the camera used, resulting in high errors compared to the ground truth values (yellow). Conversely, Metric3D, which incorporates camera parameters as additional data, produces more stable and robust results. Another interesting issue with relying solely on visual data is the impact of different viewing angles, which can significantly affect depth estimation. This reliance on specific camera models and solely visual data leads to poor generalization when the models are applied to images from different cameras or environments, which questions their capability in AR scenarios.

(3) Model complexity and inference time. Based on our experiments and recent works [8, 12, 26], we realized that incorporating additional data can address the mentioned challenges of scale ambiguity and overfitting. Furthermore, it can lead to more accurate and robust depth estimation with smaller and faster models, which are more mobile-friendly

Table 2: ***Zero-shot evaluation comparison of current state-of-the-art (SoTA) models, trained on NYU Depth V2, on the ARKitScenes validation set. Bold indicates the best results.***

Model	RMSE ↓	AbsRel ↓	#Params ↓
ZoeDepth-M12-N [2]	0.61	0.33	344.82M
ZeroDepth [11]	0.62	0.37	233M
DepthAnything [24]	0.53	**0.32**	335.79M
HybridDepth [8]	**0.367**	0.40	**65.6M**

Table 3: ***Performance comparison of the three SOTA models on Nvidia RTX 4090. Bold values represent the best results.***

Model	Inference time	Size	#Params
ZoeDepth-M12-N [2]	86 ± 6 ms	1.28 GB	344.82M
DepthAnything [24]	57 ± 5 ms	1.25 GB	335.79M
HybridDepth [8]	**25 ± 2 ms**	**0.24 GB**	**65.6M**

and better suited for real-time AR scenarios. As shown in Tables 2 and 3, the HybridDepth model demonstrates excellent zero-shot performance on AR-specific datasets such as ARKitScenes [1]. Specifically, HybridDepth achieved an RMSE of 0.367 and an AbsRel of 0.40 with only 65.6 million parameters, significantly outperforming other SOTA models such as ZoeDepth-M12-N and DepthAnything, which have larger model sizes and more parameters.

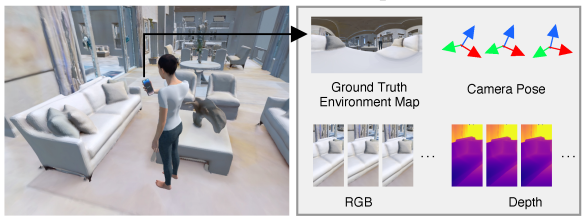
In terms of inference time and model size, HybridDepth also excels. It has an average inference time of 25 ± 2 ms, which is approximately 65% faster than DepthAnything (57 ± 5 ms) and 71% faster than ZoeDepth-M12-N (86 ± 6 ms). The model size of HybridDepth is 0.24 GB, which is around 81% smaller than DepthAnything (1.25 GB) and ZoeDepth-M12-N (1.28 GB). This demonstrates that integrating additional data sources, such as focus distances and focal lengths, can produce smaller, faster, and more efficient models without compromising accuracy, resulting in more robust and accurate depth estimation.

Key Takeaway: Hardware parameters, e.g., camera focal length, present important opportunities for building accurate, efficient, and robust in-context sensing systems. Acquiring context information from intentionally manipulated hardware, such as depth from defocus clues, has shown to be a promising way to create reliable context data.

4 Case Study: Lighting Estimation

Lighting estimation is a fundamental environment-sensing task that estimates omnidirectional lighting from limited

(a) Simulation Setup



(b) Data Generation

Figure 2: ***Overview of our simulation environment semi-synthetic data generation process.***

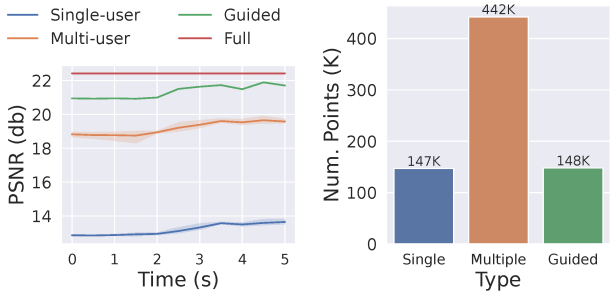
environment observations. It plays an important role in rendering visually coherent virtual objects for mobile AR applications. In this section, we investigate how in-context lighting estimation systems can improve sensing quality by interacting with AR users and external sources. Specifically, we focus on two application scenarios where context data are collected: *(i)* guided user movements and *(ii)* Multi-user sensing context data sharing.

Simulation environment setup. We use a simulation-based approach to control user mobility and evaluate its impact on lighting estimation. To address the difficulties in minimizing the synthetic-to-real gaps, we present a novel approach to facilitate the process by using *a semi-synthetic simulation-based experiment environment*. Specifically, our experimentation design differs from traditional simulation-based experiments in two key respects: *(i) photorealistic rendering* and *(ii) physically-accurate human modeling*. First, we set up the simulation environment with the RCareWorld [25] platform and high-fidelity assets from Matterport3D [5]. The scanned 3D rooms simulate the indoor environment where our experiments will occur. Next, to produce a realistic simulation of device camera movements, we use the human avatars RCareWorld [25]. In this human-centric simulation environment, the human avatar joints are derived from clinical data. Last but not least, we set the virtual AR device to use a real-world mobile phone (iPhone 14 Pro) form factors and camera parameters.

To generate our experiment data, we manually selected 12 different environment positions in the simulated room to execute the virtual object placement task. For each placement position, we apply an animation on the avatar's wrist joint to simulate the *look-around* moving pattern. We illustrate the data generation process in Figure 2. Throughout the entire interaction, we extracted the camera pose, RGB, and depth images. To allow quantitative evaluation of the lighting estimation environment understanding task, we have also extracted the ground truth environment map at the placement position using a virtual panoramic camera in Unity. To simulate the multi-user scenario, we manually set each engaged user to stand around the placement position in a circular formation. In total, we generated 36 sets of experiment data, each containing 150 frames of camera pose, RGB, and depth images. Our simulation-based experiment combines photorealistic rendering with physically accurate human modeling, representing a step forward in flexible and controllable AR experiment design.

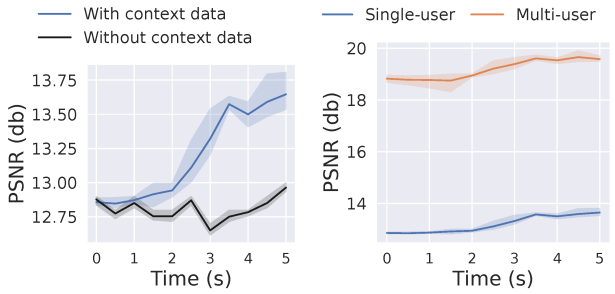
To test the AR system's performance in real-world scenarios, we perform *the AR virtual object placement task*, which represents one of the most common use cases of mobile AR, e.g., virtual furniture shopping. The workflow of this task involves a participant using an AR application to place a virtual object on a physical environment surface, e.g., a room floor, and move around their AR device to observe the object. During the process, we require the participants only to move their hands and arms and keep the AR device looking at the virtual object placement position from different angles. This movement simulates the "look-around" patterns in the AR application. This task represents a challenging environment understanding scenario where the AR systems are required to leverage the device mobility to discover the most environment information to deliver a plausible experience.

Guided user mobility. We first measure the impact of environment observation coverage in the single-user scenario with different user mobility patterns when executing the object placement task. For each selected placement position, we use the generated camera RGB, depth, and tracking data to simulate the sensory data stream available to the AR device. We removed LitAR's environment observation constraints, i.e., the number of environment observations to keep, to test its full capabilities of lighting estimation. Figure 4a shows the LitAR-generated environment map PSNR changes during the application time. On average, LitAR with environment point cloud collected by guided user mobility outperforms the baseline, where only the current camera frame is used (i.e., without context data), but by only 0.9db PSNR. This is largely due to the imperfect context data capture process, i.e., only relying on the user's hand movements. Upon manually inspecting the camera images and movements, we found that although the virtual object surrounding environment is clearly captured by the AR device camera, the user's hand



(a) Guided **(b)** Data Size

Figure 3: ***The impact of guided context collection.*** Guided context collection outperforms multi-user context sharing by 10% while using only 33% of the memory.



(a) Single User **(b)** Multi User

Figure 4: ***The impact of contextual environment observation.*** We show that accumulating environment observations over time pose positive but limited feedback on the lighting estimation performance. In the multi-user scenario, inter-user observation sharing allows significant lighting estimation performance improvement.

movements only induce overlapped views, resulting in only a small amount of observation coverage increase.

Finally, we test the impact of context data collected by guided user movement. Before executing the virtual object placement task, we collect environment observation data using the bootstrapping movement pattern introduced in LitAR. During the guided movement, LitAR collects large coverage environment observations and converts them into sparse point clouds. Figure 3a compares the lighting estimation performance difference. We see that guided context collection leads to better performance when compared to opportunistic context collection. For example, it outperforms multi-user context sharing by 10%. Additionally, when comparing the performance of using full environment observations, guided context collection has only 0.5db lower PSNR. Furthermore, guided context collection only incurs 33% of the memory usage when compared to the multi-user scenario. This suggests the need for using a well-designed context-capturing process rather than just naively relying on multi-user sharing.

Near-by multi-user context sharing. Next, we test the multi-user scenario where three users are engaged in the virtual object placement task and share environment observations with each other. In this scenario, environment point clouds are collected from user movement and multi-user sharing. Figure 4b compares the lighting estimation performance. We observe that by sharing environment point clouds from multiple users, the lighting estimation performance significantly improves by an average of 40%, compared to the single-user scenario. This is because the multi-user scenario can capture point clouds from different angles, which leads to higher environmental coverage. In practice, we believe the multi-user application scenario can be extrapolated to more generalized multi-source context data collection cases, such as edge IoT-assisted AR [19], multi-application context sharing, and multi-time context sharing.

Key Takeaway: Constructing multi-view environment observations, e.g. point cloud, can be beneficial to some environment sensing tasks. The natural user mobility in AR applications, however, may misaligned with the optimal movement for environment scanning. Actively instructing users with low-disruptive movement and retrieving shared environment observations from nearby devices can create more helpful context for in-context sensing.

5 Conclusion and Future Work

In this paper, we investigate the promises and challenges of in-context environment sensing for mobile AR. We first define the in-context sensing task and survey how recent sensing systems use sensing context data regarding the data characteristics and their contributions. Then, through two case studies, we quantified the benefits of the impacts of sensing context awareness on metric depth estimation and lighting estimation. We summarize three promising opportunities to address context uncertainty challenges: *(i)* sensory device manipulation, *(ii)* guided user mobility, and *(iii)* connected context sources. Due to the limitations of the human models in the current simulation environment, our current evaluation is limited to simple modeling of human joints. In the future, we plan to enhance the simulator by adding human muscle models to better simulate user movement and thus quantify different context data benefits.

Acknowledgments

We thank Ruolin Ye from Cornell University for her assistance in setting up the experiment simulation environment. We thank the anonymous reviewers for their constructive reviews. This work was supported in part by NSF Grants #2105564, #2236987, #2346133, and a VMware grant.

References

[1] G. Baruch, Z. Chen, A. Dehghan, T. Dimry, Y. Feigin, P. Fu, T. Gebauer, B. Joffe, D. Kurz, A. Schwartz, and E. Shulman. ARKitscenes - a diverse real-world dataset for 3d indoor scene understanding using mobile RGB-d data. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[2] S. F. Bhat, R. Birkl, D. Wofk, P. Wonka, and M. Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv:2302.12288*.

[3] 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*, 37(6):1874–1890, 2021.

[4] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic. Augmented reality technologies, systems and applications. *Multimedia tools and applications*, 51:341–377, 2011.

[5] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017.

[6] C. Chen, J. Wei, C. Peng, and H. Qin. Depth-quality-aware salient object detection. *IEEE Transactions on Image Processing*, 30:2350–2363, 2021.

[7] L. Duan, Y. Chen, and M. Gorlatova. Demo abstract: Biguide: A bi-level data acquisition guidance for object detection on mobile devices. In *Proceedings of the 22nd International Conference on Information Processing in Sensor Networks*, pages 368–369, 2023.

[8] A. Ganj, H. Su, and T. Guo. Hybriddepth: Robust depth fusion for mobile ar by leveraging depth from focus and single-image priors, 2024.

[9] A. Ganj, Y. Zhao, F. Galbiati, and T. Guo. Toward scalable and controllable ar experimentation. In *Proceedings of the 1st ACM Workshop on Mobile Immersive Computing, Networking, and Systems*, ImmerCom '23, page 237–246, New York, NY, USA, 2023. Association for Computing Machinery.

[10] A. Ganj, Y. Zhao, H. Su, and T. Guo. Mobile ar depth estimation: Challenges & prospects. In *Proceedings of the 25th International Workshop on Mobile Computing Systems and Applications*, HOTMOBILE '24, page 21–26, New York, NY, USA, 2024. Association for Computing Machinery.

[11] V. Guizilini, I. Vasiljevic, D. Chen, R. Ambrus, and A. Gaidon. Towards zero-shot scale-aware monocular depth estimation. In *ICCV*, 2023.

[12] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024.

[13] M. Hu, W. Yin, C. Zhang, Z. Cai, X. Long, H. Chen, K. Wang, G. Yu, C. Shen, and S. Shen. A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. 2024.

[14] T. Jin, S. Wu, M. Dasari, K. Apicharttrisorn, and A. Rowe. Stagear: Markerless mobile phone localization for ar in live events. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 1000–1010. IEEE, 2024.

[15] C. LeGendre, W.-C. Ma, R. Pandey, S. Fanello, C. Rhemann, J. Dourgarian, J. Busch, and P. Debevec. Learning illumination from diverse portraits. In *SIGGRAPH Asia 2020 Technical Communications*, SA '20, New York, NY, USA, 2020. Association for Computing Machinery.

[16] J. Li, H. Li, and Y. Matsushita. Lighting, reflectance and geometry estimation from 360 panoramic stereo. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10586–10595. IEEE, 2021.

[17] A. Rosinol, M. Abate, Y. Chang, and L. Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020.

[18] K. Sartipi, T. Do, T. Ke, K. Vuong, and S. I. Roumeliotis. Deep depth estimation from visual-inertial slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

[19] T. Scargill, S. Eom, Y. Chen, and M. Gorlatova. Ambient intelligence for next-generation ar. *arXiv preprint arXiv:2303.12968*, 2023.

[20] T. Starner, B. Schiele, and A. Pentland. Visual contextual awareness in wearable computing. In *Digest of Papers. Second International Symposium on Wearable Computers (Cat. No. 98EX215)*.

[21] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool. Sparse and noisy lidar completion with rgb guidance and uncertainty. In *2019 16th International Conference on Machine Vision Applications (MVA)*.

[22] B. Wen and K. E. Bekris. Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*.

[23] B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Muller, A. Evans, D. Fox, J. Kautz, and S. Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. *CVPR*, 2023.

[24] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[25] R. Ye, W. Xu, H. Fu, R. K. Jenamani, V. Nguyen, C. Lu, K. Dimitropoulou, and T. Bhattacharjee. Rcareworld: A human-centric simulation world for caregiving robots. *IROS*, 2022.

[26] W. Yin, C. Zhang, H. Chen, Z. Cai, G. Yu, K. Wang, X. Chen, and C. Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9043–9053, 2023.

[27] Y. Zhang, T. Scargill, A. Vaishnav, G. Premsankar, M. Di Francesco, and M. Gorlatova. Indepth: Real-time depth inpainting for mobile augmented reality. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 6(1), mar 2022.

[28] Z. Zhang, S. Qiao, C. Xie, W. Shen, B. Wang, and A. L. Yuille. Single-shot object detection with enriched semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5813–5821, 2018.

[29] Y. Zhao and T. Guo. Pointar: Efficient lighting estimation for mobile augmented reality. In *European Conference on Computer Vision*, pages 678–693. Springer, 2020.

[30] Y. Zhao and T. Guo. Xihe: A 3d vision-based lighting estimation framework for mobile augmented reality. In *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '21, pages 28–40, 2021.

[31] Y. Zhao, C. Ma, H. Huang, and T. Guo. Litar: Visually coherent lighting for mobile augmented reality. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(3):1–29, 2022.