

AnchorLoc: Large-scale, Real-Time Visual Localisation through Anchor Extraction and Detection

Chun Ho Park*, Ahmad Alhilal*, Tristan Braud*, and Pan Hui†*,

*Hong Kong University of Science and Technology, Hong Kong

†Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China

chpark@connect.ust.hk; aalhilal@connect.ust.hk; braudt@ust.hk; panhui@ust.hk

Abstract—Pervasive Augmented Reality (AR) requires accurate pose registration of the device in real-time at a neighbourhood-to-city scale. At such a scale, most pose registration techniques suffer from exponential computational and storage costs and a significant data collection burden. This paper introduces AnchorLoc, a framework that relies on visual anchors (stable and highly recognisable visual elements in a scene) to perform fast and accurate pose registration. AnchorLoc automatically identifies these anchors from large image sequences to optimise the search space in later image retrieval and pose registration. As such, it significantly improves the computational efficiency of existing hierarchical localisation pipelines without compromising accuracy. We collect a large-scale localisation dataset consisting of image sequences and 3D reconstruction of a university campus. AnchorLoc reduces localisation runtime by 83% on our campus dataset and 69% on the Cambridge Landmarks dataset without significantly increasing mean pose estimation errors. It is also more accurate and faster than SLD, a localisation algorithm that takes a comparable approach at the keypoint level. This work informs the development of more efficient pervasive AR applications that rely on both absolute and relative camera pose tracking on image sequences.

Index Terms—Camera relocalisation, Visual-Inertial Odometry, Augmented Reality

I. INTRODUCTION

Visual localisation is a core component of modern augmented reality (AR). It plays a vital role in overlaying virtual objects in the environment by estimating the location and the orientation of the device within a map. Recent developments in visual localisation methods have allowed camera pose estimation, which is highly accurate to centimetre levels on large-scale environments [1], paving the way towards pervasive AR applications at neighbourhood or city-scale.

Most localisation pipelines involve large amounts of storage and computational resources, which render them impractical. Persistent AR experiences require virtual objects to remain in the same physical location and be displayed consistently on different user devices over long periods of time. Implementing such applications at a large scale incurs heavy computation and storage for data capture, 3D map building and online camera localisation. Structure-based methods [2]–[5] rely on structure-from-motion (SfM) [6], [7] that constructs accurate 3D representations when given a set of images taken in the environment. Both 3D reconstruction and localisation require

significant computation and storage, and their accuracy is highly sensitive to visual changes in the environment. Adding posters on walls, moving furniture, or naturally growing vegetation requires frequent map updates to preserve accurate camera localisation. Pose regression models [8]–[10] implicitly encode the 3D map into a convolutional neural network (CNN) and directly regress the camera pose from an input image. While this reduces the load on both computation and storage, they tend to overfit their training set and may yield highly inaccurate pose estimations [11]. As such, most implementations rely on server-side pose estimation, which incurs large latencies, raises privacy concerns (and legal concerns in some countries [12]), and centralises pervasive AR experiences in the hands of the few actors who can combine large amounts of visual data with considerable server-side computation and storage.

This paper introduces AnchorLoc, a framework for fast on-device pose estimation without compromising accuracy. AnchorLoc quickly recognises visible objects, associating them with specific locations to infer the device’s pose. It abstracts a large visual positioning dataset as a finite collection of small-scale point clouds centred around visual anchors that are highly recognisable and stable objects in the dataset. AnchorLoc builds upon hierarchical localisation methods [1], [13] to optimise the search space using object detection for fast and accurate localisation. Figure 1 illustrates the overall localisation process of AnchorLoc and its runtime optimisations. The localiser uses a real-time object detection model such as YOLO [14] to detect anchors in the query image, which are then matched to the anchor information stored in the database to limit the search space during image retrieval and keypoint matching. While many AR applications rely on relative localisation algorithms such as visual-inertial odometry (VIO), they are also prone to errors resulting from drift. Thus, frequent relocalisation using fast and accurate absolute localisation algorithms is essential. Upon detecting an anchor, AnchorLoc (re)positions the AR experience, relying on relative localisation, such as VIO, to track the device’s pose between anchors.

AnchorLoc also presents a novel method for automated anchor extraction that identifies anchors from the image sequences and point clouds of a large-scale environment. We

define anchors as elements in the environment that are (1) highly recognisable, (2) visually distinctive, and (3) stable. These properties overcome several limitations of existing approaches, enabling reliable and persistent anchoring of virtual objects in AR. The automatic extraction of anchors reduces the memory footprint of the localisation system by replacing the large point cloud database in areas rich with anchors with a finite set of anchors extracted from the database. By being feature-rich and distinctive, anchors also facilitate the later steps of the localisation pipelines. As such, anchor detection also ensures accurate localisation, solving the issue of when to (re) localise the experience. Finally, its automated nature facilitates adapting to scene changes by running on newly collected data and updating the anchor database accordingly.

The automated anchor extraction module, in combination with the anchor-based localisation mechanism, allows the implementation of a real-time camera localisation system with improved memory efficiency and automatic adaptation to long-term scene changes. Focusing on a collection of small-scale areas also simplifies the data collection process, reduces the need for frequent map updates, and improves privacy. We believe AnchorLoc is the first system to enable on-device persistent AR experiences at large scales, towards pervasive AR. We demonstrate the effectiveness of AnchorLoc on two popular indoor and outdoor datasets [8], [15], and a new campus dataset, resulting in up to 70~80% runtime reduction over all datasets.

The contribution of our paper is fourfold:

- **Propose** a novel localisation pipeline based on visual anchor identification in the training dataset and subsequent detection to reduce the search space.
- **Collect** a large university-scale dataset with corner cases that mimic real-life localisation scenarios (repetitive textures, low-feature areas, poor coverage of visible areas).
- **Develop AnchorLoc**, a complete localisation system.
- **Evaluate** AnchorLoc against two leading localisation methods (HLoc and SLD) over three datasets. AnchorLoc improves runtime by over 69% without a significant reduction in accuracy compared to HLoc while improving the runtime by 66% and accuracy by 39% compared to SLD.

II. RELATED WORKS

Visual localisation is an active topic in the computer vision community. Prominent works include structure-based localisation, object-based localisation, and localisation methods focusing on scalability. Other works provide representative datasets to evaluate visual localisation algorithms.

Structure-based Localisation. Structure-based methods estimate the camera pose by extracting keypoint correspondence between the query image and a 3D reconstruction of the environment. They present a high accuracy at the cost of significant computation and storage. The 3D reconstruction is often created using a sequence of images through Structure-from-Motion (SfM) methods such as COLMAP [6], [7]. Numerous

works improve the localisation by using learned features [16]–[18] for keypoint extraction and matching. Specifically, R2D2 [18] detects discriminative keypoints for improved matching. Comparatively, AnchorLoc focuses on discriminative and stable *objects* to ensure adaptability to dynamic environments. SuperGlue [19] performs context-aware keypoint matching using graph neural networks (GNNs) to obtain 2D-3D correspondences. Li et al. [20] combine visual and depth data to optimise the accuracy of the 3D map. Recent works on structure-based localisation also focus on alternative scene representations such as lines [21] and dense 3D meshes [22] to improve accuracy, robustness, and flexibility. Although these methods are the most accurate, they lack the computational efficiency required in time-sensitive applications. AnchorLoc maintains the high accuracy required for large-scale AR applications with improved efficiency.

Scalable Localisation. Localisation in large-scale environments tends to be resource-intensive, making scalable localisation an open research area. Hierarchical localisation approaches use image retrieval methods to limit the search space when looking for 2D-3D correspondences [23], [24]. Sarlin et al. [13] perform coarse-grained matching to retrieve images, followed by covisibility clustering and fine-level feature matching to increase scalability. However, this approach depends on the accuracy of the image retrieval method. Sattler et al. [3] propose a prioritised keypoint matching scheme using visibility information from SfM. Yang et al. [25] utilise clustering, pruning, and quantisation to compress the 3D model of the scene while maintaining localisation accuracy. Do et al. [15] extract scene landmarks or discriminative keypoints from the constructed 3D model and perform localisation by predicting the scene landmark locations within the query image. Although significant progress has been made in the scalability of localisation algorithms, these works do not address realistic challenges posed in AR applications, such as lacking estimation accuracy in large-scale, dynamic environments. AnchorLoc’s unique anchor-based approach maintains a high accuracy and resilience to changes in the environment and offers better privacy guarantees through on-device operation.

Object-based Localisation. Several works use information regarding objects in the environment for visual localisation. Weinzaepfel et al. [26] uses a CNN to detect and segment Objects-of-Interest (OOIs), or highly descriptive objects in the environment to obtain dense keypoint matches. Benbihi et al. [27] propose object-guided localisation that detects objects of a single class to guide keypoint matching. Zins et al. [28] improve the Simultaneous Localisation And Mapping (SLAM) system by using high-level object landmarks by building an automated system that detects and tracks objects with 3D ellipsoids. However, the object detection in these approaches is either limited to a single class [26], [27], requires manual annotation [26], or is tested only in small-scale, controlled indoor [28], or synthetic environments [26]. AnchorLoc automatically extracts and detects multiple classes of distinct and stable objects and produces fast and accurate pose estimations even in large-scale (indoor and outdoor) settings.

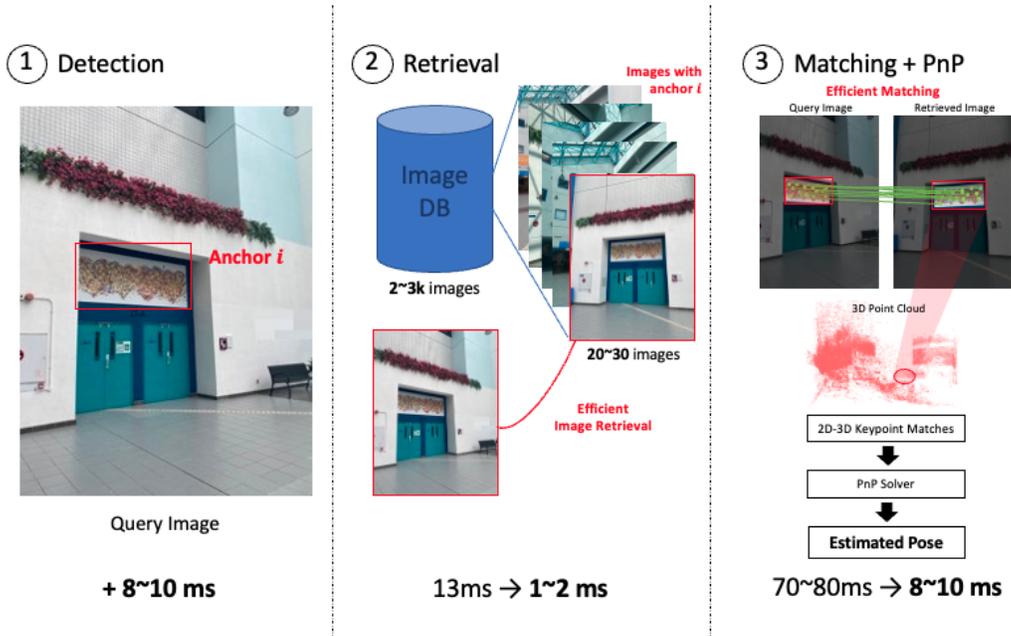


Fig. 1. Overview of AnchorLoc’s visual localisation process. AnchorLoc 1) rapidly detects anchor i (previously registered by our anchor extraction pipeline) from the query image, 2) retrieves the most similar image out of the set of images that contain anchor i and 3) performs keypoint matching to obtain 2D-3D keypoint correspondences to the 3D keypoints corresponding to the anchor, and runs PnP solver to obtain the pose estimation. Runtime reductions from existing image retrieval-based localisation method [13] on our Campus localisation dataset are shown at the bottom.

Localisation Datasets. Existing datasets are intended to provide challenges that could be encountered in real-life localisation scenarios, especially the dynamic nature of the environment. These include illumination changes (day-night) [15], [29], seasonal variations [30]–[32] and dynamic objects in the environment (vehicles, people) [8], [33]. The datasets vary in their scale from small indoor [4], [15], [34], [35] to large indoor and outdoor scenes [8], [29]–[33]. Our newly collected campus-scale dataset aims to test localisation performance in large indoor environments where data was collected from multiple users over a long period.

III. ANCHOR-BASED LOCALISATION

AnchorLoc extracts distinctive objects in sight, namely *anchors*, from sequences of frames taken within the environment. These anchors are leveraged to provide real-time structure-based visual localisation methods in large-scale environments.

A. Anchor Extraction

We extract the anchors from a database of image sequences used to build the 3D reconstruction of the environment. Each sequence is assumed to be captured at different times. Anchor extraction from the image database is performed through 1) anchor candidate identification and extraction and 2) candidate scoring and ranking.

We selected the anchors from a set of candidate objects that were captured in the environment. To obtain the candidate set, we used an existing open-world object detection method [36] to detect objects of classes that are not necessarily predefined. The open-world property of this class of detectors allows the

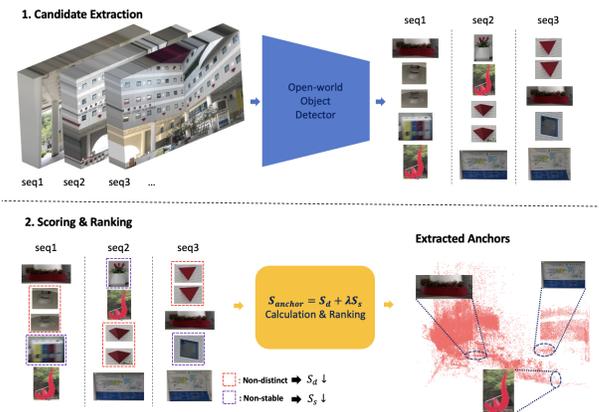


Fig. 2. Illustration of the anchor extraction pipeline. 1) Anchor candidates are extracted from image sequences through open-world object detection, 2) scored and ranked according to distinctiveness and stability, and 3) saved to the database along with detected frames D_i and keypoints K_i .

extraction algorithm to choose from a set of objects that are not confined to a small set of classes, thus allowing diverse objects with distinct visual elements to be included in the set. For an accurate localisation, we assume that anchors must fulfil two properties: *distinctiveness* and *stability*. Each candidate is scored and ranked by a metric that captures both aspects. *Distinctiveness* of each anchor is vital in ensuring that each anchor detection is associated with one unique set of keypoints corresponding to the anchor. *Stability* requires

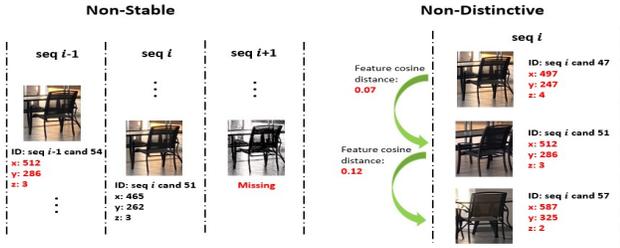


Fig. 3. Non-stable and non-distinctive objects, showing which factors contribute to the two properties. (Left) Candidate 51 at sequence i is mapped to a different position at sequence $i-1$ and not found in sequence $i+1$, leading to low stability. (Right) There exist other candidates with high feature similarity (low cosine distance) but at different locations, showing low distinctiveness.

detections of each anchor to be consistently found in different sequences taken at different times and mapped to the same set of point clouds in the constructed 3D model. A lack of either of these properties would likely lead to inaccurate pose estimations. For example, detecting a non-distinct, unstable (i.e. moving) object such as a chair could be mapped to a different instance of a similar-looking chair, or the detected chair might have moved to another location, leading to spurious keypoint matches. Figure 3 illustrates an example of a non-stable, non-distinct object in the environment to clarify the properties of stability and distinctiveness.

The anchor score S_{anchor} is a weighted sum of S_d and S_s , each measuring the *distinctiveness* and *stability* properties, respectively. The scoring algorithm is elaborated in Algorithm 1, which outputs anchor score S_{anchor} of candidate c given balancing factor λ , thresholds for image feature and keypoint distances ϵ_f and ϵ_k , and a total number of sequences N_{seq} . S_s is proportional to the number of sequences that include detections with high feature similarity (similar appearance) and keypoint maps to the same location. S_d is inversely proportional to the number of candidates within the same sequence with high feature similarity but maps to a different location. Therefore, the score prefers candidates that were detected more times in other sequences at the same location (*stability*) but do not appear at different locations within the same sequence (*distinctiveness*). We calculate the feature vector for each candidate c using a CNN backbone network [37] and define the keypoint location as the centre (x, y, z) coordinate of all the 3D keypoints corresponding to the candidate. The image feature distance δ_f and keypoint distance δ_k are calculated between c and other candidates in each iteration. δ_f is calculated as the cosine similarity between the image feature vectors of each candidate, and δ_k is calculated as the Euclidean distance between the keypoint locations of each candidate. $\delta_f \leq \epsilon_f$ implies that the two candidates have a similar appearance, and $\delta_k \leq \epsilon_k$ means the 3D keypoints of the two candidates map to the same location. We chose candidates with top S_{anchor} as anchors for fast localisation when detected. During the ranking process, candidates

with similar image features and the same keypoint coordinates in the 3D model are grouped as one candidate since multiple detections of the same object exist among different images. Additionally, these multiple detections of the same candidate were used as training data for the object detection algorithm used to detect anchors in query images. Also, we performed image retrieval on this set of images, labelled D_i , during the localisation process for each selected anchor i . Image set D_i , and the set of 3D keypoints K_i corresponding to each anchor i were stored in the database.

Algorithm 1 Anchor Score S_{anchor} Calculation

Require: $c, \lambda, \epsilon_f, \epsilon_k, N_{seq}$

f_c, K_c, seq_c = feature vector, keypoint set, and sequence number of candidate c respectively

$n_d, n_s, n_{all} = 0$

for $i = 1, 2, \dots, N_{seq}$ **do**

C_i = set of all candidates in sequence i

for $c' \in C_i, c' \neq c$ **do**

δ_f, δ_k = feature and keypoint distance between c and c' respectively

if $seq_c = i$ **then**

$n_d = n_d + 1$ if $\delta_f \leq \epsilon_f$ and $\delta_k > \epsilon_k$

$n_{all} = n_{all} + 1$

else

$n_s = n_s + 1$ if $\delta_f \leq \epsilon_f$ and $\delta_k \leq \epsilon_k$

break

end if

end for

end for

$S_d = \frac{1}{n_d}, S_s = \frac{n_s}{N_{seq}}$

return $S_{anchor} = S_d + \lambda S_s$

B. Anchor Detection and Localisation

To localise query images, we first detect anchors using a trained single-stage object detection algorithm [14]. We use the information on the anchor stored in the database to limit the search space for image retrieval and keypoint matching, thereby allowing real-time visual localisation.

Single-stage detectors [14], [38], unlike two-stage detectors [39]–[41], operate in real-time, which allows them to integrate other tasks which require real-time performance. We fine-tune a pre-trained YOLO [14] detector on the set of anchors collected during the anchor extraction process, with a train/validation split of 0.7:0.3. We only use anchors that show at least 0.9 mAP (mean average precision) on the validation set, which is achieved on anchors with at least 20 training samples ($|D_i| \geq 20$).

For the localisation process, we build upon the hierarchical localisation scheme as in Sarlin et al. [13] by first running global image retrieval to retrieve the most similar database image, followed by feature matching on two images and performing pose estimation by solving the PnP problem [42] in RANSAC [43] loop. Figure 4 outlines the overall localisation process. Suppose a query image I_{query} is given, and anchor i

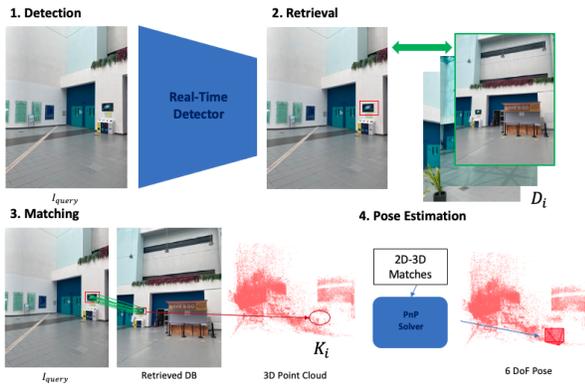


Fig. 4. Illustration of the localisation process. 1) Anchor i is detected at real-time speed, then 2) image retrieval among D_i is performed. Then, 3) keypoint matching with the retrieved image on anchor i returns 2D-3D keypoint matches with K_i , which is 4) used to perform pose estimation.

is detected. We run an image retrieval algorithm for I_{query} on D_i to retrieve the most similar database image that contains anchor i . Within the two images, we run keypoint matching on 2D keypoints, which exist inside the bounding box for the detected anchor. Then, we infer the 2D-3D match from the query image to the 3D point cloud K_i by transitivity using the 2D-2D keypoint matches and the 2D-3D keypoint correspondence information from the database image to 3D keypoints. Finally, we obtain the estimated six-degree-of-freedom (6 DoF) pose by solving the Perspective-N-Point (PnP) problem [42] using the 2D-3D keypoint matches within a RANSAC [43] loop.

The performance gain comes from the significant reduction in search space during the image retrieval and keypoint matching steps using the information from the detected anchor. Additionally, since the runtime complexity of PnP solver [42] is linearly proportional to the number of 2D-3D keypoint correspondences, it also results in a significant gain in the absolute pose estimation phase. The additional computational overhead for object detection during inference time is minimal, resulting in a significant overall inference time reduction.

IV. DATASETS

There are many datasets for evaluating visual localisation methods that pose challenges, such as the dynamic nature of the environment, especially seasonal or illumination changes [29]. Some datasets are created from crowdsourced images taken in a popular location [44], which could serve as a realistic scenario on a large-scale platform where crowdsourced user data is collected and used to improve the system.

The anchor-based localisation method outlined in the previous section is expected to be particularly useful in environments with plenty of objects and structures that can be used as anchors. Additionally, mining anchors from data collected over an extended period from multiple users would allow our method to collect stable anchors, add new anchors from newly collected data, and eliminate old ones. We evaluate our method on existing indoor and outdoor localisation datasets [8], [15]

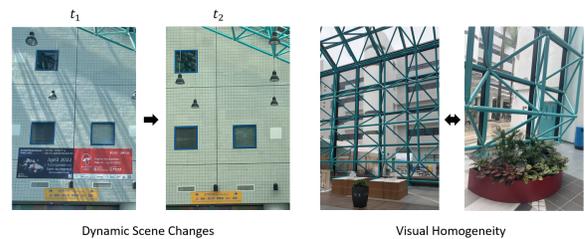


Fig. 5. Challenges introduced by the Campus Dataset. *Left* shows an example of a scene change over time t_1 and t_2 , one month apart, and *right* shows an example of visual homogeneity, with similar-looking visual structures 500 meters apart.

and a new large-scale, dynamic, and anchor-rich dataset captured on different devices by multiple users.

Large environments such as university campuses often contain objects and structures that exhibit distinctiveness and stability and could be used as anchors. Examples would include a large poster, a statue, or a distinctly recognisable part of a building facade. Our new dataset includes sequences of images and a 3D Structure-from-Motion (SfM) model taken in a large indoor area within a university campus.

On the other hand, a campus dataset introduces interesting challenges for our method, as depicted in Figure 5. As a living space, a campus changes significantly over time. Banners are added to promote events, furniture is moved around, and many feature-dense elements have low temporal stability. Another issue stems from the campus’s large scale and strong architectural identity. The university campus features many repetitive patterns and textures at different places, leading to low distinctivity of high-feature areas. As such, this dataset allows evaluating our method on a large-scale, complex scenario that could be commonly encountered in pervasive AR applications such as urban AR.

Images of a large indoor campus environment were collected from 3 participants using mobile phone cameras. Participants were instructed to take photos of the surrounding environment while walking through a large indoor area, approximately 800 m long. They were collected on different days spanning over two months, during which there were changes in the scene (e.g., new posters were hung on the wall). A total of 15 sequences were collected from 3 participants, each consisting of 70 to 80 images. All people captured in the images were blurred for privacy reasons. The 3D model of the large indoor campus area was built from COLMAP [6], [7], which includes the registered 6 DoF poses of each frame, the 3D point cloud and the 2D-3D keypoint correspondences.

V. EXPERIMENTAL RESULTS

We evaluate AnchorLoc’s localisation performance on three datasets with different scales: Cambridge Landmarks [8] dataset - St. Mary’s (large outdoor), Indoor 6 [15] (small indoor) and our CampusDataset (large indoor). These three datasets are the only datasets that present sequences of images captured at different times, allowing us to extract *stable* anchors and contain a sufficient number of images to run

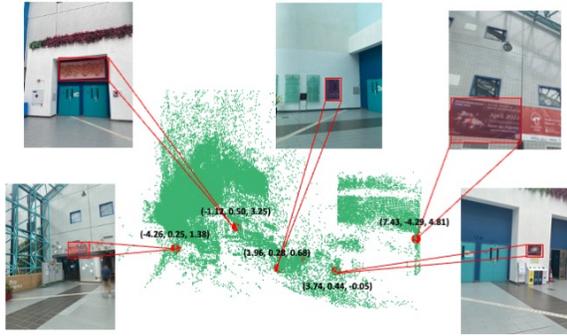


Fig. 6. Visualisation of a portion of the 3D point cloud built with COLMAP [6], [7] and positions of the selected anchors. Anchors are marked with a red point and their (x, y, z) coordinates. We also show one of the images and bounding boxes for each anchor.

Structure-from-Motion and our anchor extraction pipeline. Experiments on Cambridge Landmarks (approximately $4800m^2$ wide, 2000 frames) and our CampusDataset (approximately $7300m^2$ wide, 1200 frames) show localisation performance under large-scale areas, while the Indoor 6 dataset (small indoor area with 7000 frames per scene) focuses on small, home-scale indoor spaces.

We compare AnchorLoc’s localisation accuracy and inference time to those of HLoc [13], a conventional hierarchical localisation method, and SceneLandmarkDetector(SLD) [15], a method that extracts stable keypoints named ‘scene landmarks’, comparable to anchors at keypoint level. We select these methods as baselines since (1) our method focuses on increasing the scalability of hierarchical localisation methods (HLoc), and (2) our method also aims to automatically identify stable visual elements in the environment and leverage them for efficient localisation (SLD).

A. Implementation Details

Anchor Extractor. We use Object Localization Network (OLN) [36] to get the bounding box for each candidate anchor. The detection confidence threshold is fixed at 0.8. To extract the image feature used for the scoring phase, we use EfficientNetB0 [37], which is a widely used CNN backbone architecture. The balancing factor λ during S_{anchor} calculation is set to 2.0 to ensure that both *stability* and *distinctiveness* contributed to anchor selection. The thresholds for image feature similarity ϵ_f and keypoint location ϵ_k between candidates are set to 0.3 and 0.1, respectively. We find that these values are adequate for checking if two instances have a similar appearance (with feature cosine similarity) or are at the same location (with Euclidean distance between centre 3D coordinates) within the 3D model.

Detector and Localiser. We use YOLOv5 [14] object detector for running the anchor detection. We use the YOLOv5m (medium) version out of several versions that differ in model size provided by Ultralytics [45]. The detector is trained by fine-tuning the last three layers of a detector pre-trained on the

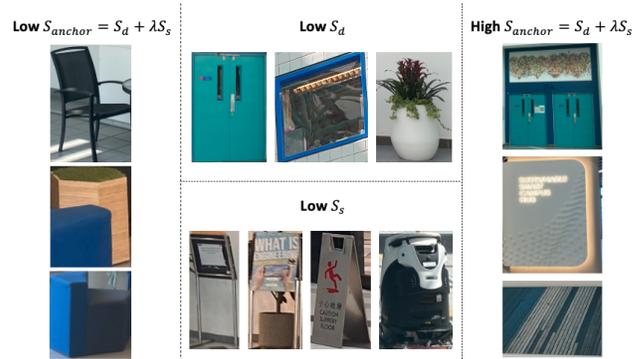


Fig. 7. Examples of anchor candidates with varying values for each component S_d , S_s and combined score S_{anchor} . It can be inferred that each score component S_d and S_s can measure distinctiveness and stability, respectively.

MS COCO [46] object detection dataset. We divide the detector training set into 0.7:0.3 train-validation split, where the entire training set consists of at least 20 instances per anchor. We use SuperPoint [16] feature descriptor, NetVLAD [23] global image descriptor for the image retrieval step and nearest neighbour keypoint matching method.

Hardware. All of the procedures, including 3D model construction using COLMAP [7] [6], anchor extraction, anchor detection, and query image localisation are performed on a PC equipped with NVIDIA GeForce RTX 3060 GPU.

B. Anchor Extraction

We conduct a qualitative evaluation of the anchor extraction scoring method to verify that the algorithm can quantify the required properties of anchors. Examining the candidates that scored low on each component S_d and S_s allows us to validate the scoring metric and extract the candidates that can be relied upon during the localisation process.

Figure 7 shows example candidates at different score levels for different components of S_{anchor} . As shown in the figure, candidates with overall low scores, both on distinctiveness and stability, are non-static objects of which several similar-looking instances exist in the environment (e.g. chairs). Candidates scoring low on either components S_d or S_s fall short on either of the properties. For example, stable but non-distinct objects such as doors and windows score low on S_d . Meanwhile, distinct but moving objects (e.g. moving signs and cleaning robots) score low on S_s . Candidates scoring high on both components exhibit both distinctiveness and stability and are thus suitable for localisation. Since a large portion of the candidates who score high on S_{anchor} are filtered out because of the lack of training samples for the detector, the amount of stable anchors extracted is expected to grow with the amount of collected data.

Moreover, a new sequence of 61 images was collected for the Campus dataset after approximately one year (original data collected during the summer of 2021 and new data collected during the summer of 2022) to test if the extracted anchors remained available after a long period. As a result, 43.75%



Fig. 8. Examples of modified/eliminated anchors and their anchor scores from the Campus dataset, after one year period

TABLE I

HLOC [13], SLD [15], AND ANCHORLOC'S LOCALISATION ACCURACY. MEAN TRANSLATION ERROR (CM), ROTATION ERROR($^{\circ}$) AND RECALL RATE AT $5\text{cm}\&5^{\circ}$ WERE RECORDED. THE LOWER THE VALUES WITH (\downarrow), THE BETTER, AND THE HIGHER THE VALUES WITH (\uparrow), THE BETTER. ALTHOUGH HLOC PRESENTS A SLIGHT IMPROVEMENT OVER ANCHORLOC, ANCHORLOC SIGNIFICANTLY OUTPERFORMS SLD BY CONSIDERING GROUPS OF FEATURES INSTEAD OF SINGLE KEYPOINTS.

Method	Cambridge	Indoor 6	Campus
	cm.(\downarrow)/deg.(\downarrow)/recall@5cm5 $^{\circ}$ (\uparrow)		
HLoc [13]	11.2/5.2/0.46	5.1/7.6/0.8	16.4/3.8/0.91
SLD [15]	-	9.8/20.2/0.42	-
AnchorLoc	14.1/5.5/0.39	6.0/9.4/0.71	19.2/4.5/0.78

of all the anchors extracted from the original sequences were detected in the new sequence. This shows that a large percentage of anchors cannot be used for localisation after a long period of time. Additionally, there was no apparent relationship between their anchor score components S_d , S_s and their availability in the new sequence. The mean S_d and S_s for the unavailable anchors were 1.0 and 0.778, respectively, and the mean S_d and S_s for available anchors were 0.972 and 0.633, respectively. Higher scores thus do not reflect a higher likelihood of remaining in the environment for a long time. Figure 8 shows example images and the scores of the anchors that were modified or eliminated in the environment. This analysis highlights the need for continuous periodic updates of the anchor database, possibly by running the anchor extraction algorithm periodically from newly collected data.

C. Localisation

We compare the localisation performance of HLoc [13], SLD [15], and AnchorLoc by examining their accuracy and the computational efficiency on the three datasets mentioned above. The comparison is conducted on the subset of query images in which anchors are detected, which amounted to 69.8% for Cambridge, 58.4% for Indoor 6, and 39.5% for our Campus Dataset. To provide comparable runtime measures, we set HLoc's image retrieval step to retrieve a single image. In practice, a higher number of images is recommended to ensure accuracy. Similarly, we only consider the anchor detected with the highest confidence score in AnchorLoc. The anchor detector, which was trained on the set of database images

TABLE II
RECALL RATES OF HLOC [13], SLD [15] AND ANCHORLOC AT DIFFERENT THRESHOLDS: $5\text{cm}\&5^{\circ}$, $25\text{cm}\&10^{\circ}$, AND $50\text{cm}\&20^{\circ}$. ANCHORLOC SIGNIFICANTLY IMPROVES RECALL RATES COMPARED TO SLD AND COMES CLOSE TO HLOC.

Method	Cambridge	Indoor 6	Campus
	recall@5cm5 $^{\circ}$ /25cm10 $^{\circ}$ /50cm20 $^{\circ}$ (\uparrow)		
HLoc [13]	.46/.76/.87	.80/.92/.95	.91/.97/.97
SLD [15]	-	.42/.82/.87	-
AnchorLoc	.39/.68/.82	.71/.84/.88	.78/.95/.97

TABLE III

LOCALISATION RUNTIME STATISTICS OF HLOC [13], SLD [15], AND ANCHORLOC IN MILLISECONDS(MS). MEAN, STANDARD DEVIATION(STD.), TOP 10%, AND 90% FOR EACH METHOD ON EACH DATASET ARE RECORDED. ANCHORLOC IS FIVE TIMES AS FAST AS HLOC AND HALVES THE LOCALISATION TIME COMPARED TO SLD, WITH SIGNIFICANTLY LOWER VARIANCE.

Method	Cambridge	Indoor 6	Campus
	mean/std./10%/90%(\downarrow)		
HLoc [13]	58/18/41/73	91/7.8/82/101	94/42/35/157
SLD [15]	-	45/9.5/32/59	-
AnchorLoc	18/2.5/14/20	20/1.8/18/23	16/1.6/14/19

which included extracted anchors, was able to achieve an accuracy of over 0.92 mAP (mean average precision). Even though the size of the training and validation set was small (around 20 images per class in total), it was able to achieve stable performance due to the fine-tuning approach. We report the localisation accuracy in Table I and the inference time statistics in Table III.

AnchorLoc shows localisation accuracy comparable to the accurate method HLoc [13] while significantly outperforming it in terms of computational efficiency. HLoc [13] shows the highest level of accuracy as it performs image retrieval over all database images, performs keypoint matching on all 2D keypoints on retrieved images, and uses them to perform the pose estimation. Although HLoc is the most accurate method, AnchorLoc's mean estimation error deviates from HLoc's error by only 1cm~3cm and $1^{\circ}\sim 3^{\circ}$ while reducing runtime significantly by approximately 70~80%. SLD [15] does not directly output the pose estimation from a neural network like other pose regression-based methods [8]. However, it outputs the 2D keypoint location through a neural network, which decreases runtime at the cost of lower accuracy. Meanwhile, AnchorLoc drastically reduces the runtime for all datasets without considerably degrading the accuracy, especially on the Campus dataset, which covers the largest area. As shown in Table III, AnchorLoc induces a runtime reduction of approximately 70 to 80% compared to HLoc and around 60% reduction compared to SLD [15]. We found that detecting and using stable and distinct object-level anchors instead of keypoint-level scene landmarks (as in SLD) for localisation yields higher accuracy at shorter runtime. Additionally, An-

TABLE IV

STEP-WISE MEAN RUNTIME PER QUERY ON HLOC [13] AND ANCHORLOC (IN MS), MEASURED ON OUR CAMPUS DATASET. ANCHORLOC ADDS, ON AVERAGE 8 MS FOR ANCHOR DETECTION, BALANCED BY A SIGNIFICANT (MIN 10X) DECREASE IN RETRIEVAL, MATCHING, AND PnP TIMES.

Method	Detection	Retrieval	Matching	PnP
HLoc [13]	-	13	37	43
AnchorLoc	8	1	1.3	5.9

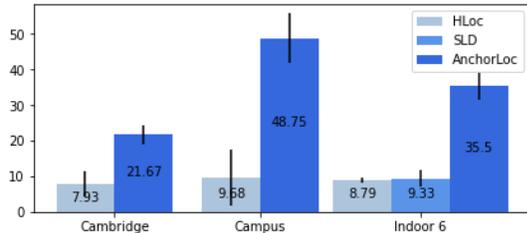


Fig. 9. Recall@5cm5°/Mean runtime (seconds) of HLoc [13], SLD [15] and AnchorLoc over all datasets. Error bars show the standard deviation. Larger values indicate higher accuracy relative to localisation runtime. AnchorLoc is more efficient, exhibiting higher accuracy over computational cost.

chorLoc is the only method out of the three tested methods to show fast performance at around 50 frames-per-second (FPS), which could be advantageous for augmented reality applications in overlaying virtual objects at a smooth frame rate while tracking the device within the environment.

To further illustrate the efficiency of our method, we show the recall rate at 5cm5° over the mean runtime (in seconds) of each method in Figure 9, which can be interpreted as the ratio between the level of accuracy and computation cost. AnchorLoc shows the highest level of accuracy relative to the computational cost over all datasets at different scales. Additionally, AnchorLoc shows a large reduction in the runtime variance in all cases. According to Table III, the standard deviations of runtime are significantly reduced by approximately 80% to 95% over all datasets. This shows that our method is not only able to reduce the computational load but also allows stable and consistent performance compared to previous methods, which is another essential aspect for deploying real-time localisation systems.

We compare the runtime taken by each component in the hierarchical localisation pipeline with HLoc [13] to show that AnchorLoc effectively reduces the search space and increases efficiency. Table IV shows the mean runtime taken by the detection, image retrieval, keypoint matching, and PnP solver [42] of the two methods on our large-scale indoor Campus Dataset. Although there is additional overhead for anchor detection in AnchorLoc, the performance gain in the rest of the pipeline from the reduced keypoint search space possible by the anchor detection drastically outweighs this factor. When an anchor is detected, image retrieval is only performed on the small subset of database images containing the anchor and the keypoint matching. The PnP solver only



Fig. 10. Example visualisations of keypoint matching between a query image (left) and database image(right) in AnchorLoc. Using the information from the detected anchor, AnchorLoc can boost the localisation efficiency by limiting the search space during image retrieval and reducing the number of processed keypoints during pose estimation.

has to process the keypoints within the detection bounding box, significantly boosting computational efficiency. Comparatively, we configured HLoc to retrieve a single image during the image retrieval phase. This phase should return several images in larger-scale or low-feature environments to minimise error. The rest of the pipeline would then process each image, increasing the runtime by a multiplicative factor.

While AnchorLoc’s main advantage is the increased efficiency in the localisation pipeline, its ability to retain a similar level of accuracy as the original hierarchical localisation methods is vital. This ability is assumed to be primarily due to the anchor scoring mechanism, as the stability and distinctiveness criteria aim to ensure the accuracy of the retrieved image and the keypoint matches. During retrieval and matching, the localisation algorithm is designed to focus on a small portion of the visual information. We also emphasise that anchor-based localisation is highly scalable, as the runtime is independent of the global dataset size once we reduce the search space using the detected anchors. This portion contains the most useful information for localisation thanks to its stable and distinct properties. The resource-intensive anchor extraction from the dataset is performed in advance on the server side, allowing accurate and efficient localisation.

VI. RESOURCE-CONSTRAINED SETTING

Mobile Augmented Reality often relies on devices with limited capabilities. Therefore, we perform additional experiments in a resource-constrained setting to confirm AnchorLoc’s improvements in computational efficiency for realistic deployment scenarios. Experiments were carried out on a mobile single-board computer with ARM architecture equipped with an 8-core CPU with 2.4 gigahertz processing speed.

Table V shows the runtime statistics of each method on all three datasets under the resource-constrained setting. The accuracy of each method remained the same as in the original experiment conditions since the change in hardware does not induce any change in algorithm output. As shown in the table, the mean runtime is reduced by around 60% on all

TABLE V
LOCALISATION RUNTIME STATISTICS OF HLOC [13], SLD [15], AND ANCHORLOC IN MILLISECONDS(MS) IN RESOURCE-CONSTRAINED SETTING. MEAN, STANDARD DEVIATION(STD.), TOP 10% AND 90% FOR EACH METHOD ON EACH DATASET ARE RECORDED.

Method	Cambridge	Indoor 6	Campus
	mean/std./10%/90%(↓)		
HLoc [13]	197/79/93/271	187/68/101/258	163/71/85/241
SLD [15]	-	119/46/72/165	-
AnchorLoc	75/19/56/106	71/21/47/96	67/23/31/102

three datasets compared to HLoc [13] and 40% compared to SLD [15]. Moreover, the standard deviations of the runtime distribution for each dataset were reduced by around 70 to 80% compared to previous methods, which shows that AnchorLoc shows similar improvements in computational efficiency on more realistic hardware settings.

VII. DISCUSSION

Summary of results: Our anchor-based localisation method shows superior computational efficiency (up to 70-80% faster) with only marginal increases in pose estimation errors (1-3 cm). AnchorLoc is a more practical method for pervasive AR applications, showing real-time performance at large-scale even on resource-constrained mobile devices.

Dynamic Environments: Realistic localisation scenarios present unstable visual elements that change over time, such as moving pedestrians, vehicles, or furniture. AnchorLoc’s automatically identifies and focuses on stable visual elements in the scene. Its automated anchor extraction pipeline does not depend on manual labelling, although it also allows manually extracted anchors. It may adapt to scene changes over time, as shown in Figure 8, by running periodically on newly collected data. These properties make AnchorLoc an efficient localisation method for large-scale AR platforms.

Long-term Anchor Persistence: One year after the initial data collection, we collected another sequence of images covering all the previously detected anchors in the dataset. 43.75% of the anchors were detected. The anchors that were not detected were either (1) posters and signs that got replaced over time; (2) large pieces of equipment (e.g., vending machines) that moved around campus; (3) existing anchors that got covered, either temporarily or permanently. This finding highlights the need for semantic classification of anchors to remove objects likely to move over a longer period and the need for periodic updates of the anchor database. In a real-life scenario, users would use anchors to relocalise the experience periodically. It would thus be easy to detect missing anchors and leverage users’ experiences to collect more data.

Limitations: A significant portion of the query image set did not contain anchors, which could potentially lead to inconsistencies in localisation speed. On the other hand, environments containing many objects (e.g. busy city centres) could make the anchor extraction costly since the algorithm’s

computational complexity is quadratic to the number of candidates. Semantic segmentation may be used to provide more contextual information to be used during the anchor extraction phase. Additionally, this work only considers a single anchor on the frame. Considering multiple anchors on a single frame could further improve the localisation accuracy while introducing computational overhead. Finally, evaluating the method under a more realistic localisation scenario, such as in a large-scale AR platform, would be necessary to bridge the gap between experimental settings and real-world applications.

Towards pervasive AR: This work aims to enable AR indoors and outdoors at a vast (city) scale. Current approaches are computationally intensive and require large amounts of storage. As such, only major industry players can partake in pervasive AR. Currently, only Google Geospatial API¹ and Niantic Lightship² offer such capabilities. However, Niantic only focuses on select environments, while Google’s visual positioning system requires immense computation and storage capabilities to handle the scale of its data. Academically, solutions such as HLoc offer good accuracy, but performing image retrieval over larger datasets requires selecting multiple candidates to minimise error, significantly increasing runtime. Anchorloc considers that most visual data collected in continuous sequences is not adapted to visual positioning (e.g., featureless environments, repetitive textures) and focuses on the stable, feature-rich areas. Besides accelerating localisation, it also ensures that localisation is performed over images that would yield high accuracy thanks to the uniqueness and density of the features. Anchors are detected with lightweight ML models such as YOLO, and the search is performed on a minimal-size point cloud. This allows mapping very large-scale environments with minimal data and enables on-device positioning, paving the way for city-scale pervasive AR scenarios such as augmented tourism, vehicular AR head-up displays, and AR navigation applications.

VIII. CONCLUSION

This work introduces AnchorLoc, a framework for extracting and integrating anchors into visual localisation for efficiency and scalability. We provide an automated pipeline for extracting anchors, which are stable and distinct visual elements that can be used for localisation. AnchorLoc leverages anchor detection to improve the computational efficiency of hierarchical localisation methods. We also introduce a new large-scale, indoor campus visual localisation dataset. Experiments on representative datasets prove our method’s efficacy for low-latency localisation on environments of different scales.

IX. ACKNOWLEDGEMENTS

This research was supported in part by a grant from the Guangzhou Municipal Nansha District Science and Technology Bureau under Contract No.2022ZD01 and the MetaHKUST project from the Hong Kong University of Science and Technology (Guangzhou).

¹<https://developers.google.com/ar/develop/geospatial>

²<https://lightship.dev/>

REFERENCES

- [1] P.-E. Sarlin, F. Debraine, M. Dymczyk, R. Siegwart, and C. Cadena, "Leveraging deep visual descriptors for hierarchical efficient localization," 2018.
- [2] T. Sattler, B. Leibe, and L. Kobbelt, "Fast image-based localization using direct 2D-to-3D matching," in *International Conference on Computer Vision (ICCV)*, 2011, pp. 667–674.
- [3] S. Gao, J. Wan, Y. Ping, X. Zhang, S. Dong, Y. Yang, H. Ning, J. Li, and Y. Guo, "Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2016.
- [4] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "InLoc: Indoor Visual Localization with Dense Matching and View Synthesis," in *Conference on Computer Vision and Pattern Recognition*, 2018.
- [5] L. Liu, H. Li, and Y. Dai, "Efficient Global 2D-3D Matching for Camera Localization in a Large-Scale 3D Map," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2391–2400.
- [6] J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition*, 2016.
- [7] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm, "Pixel-wise View Selection for Unstructured Multi-View Stereo," in *European Conference on Computer Vision*, 2016.
- [8] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization," in *International Conference on Computer Vision (ICCV)*, December 2015.
- [9] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Conference on Computer Vision and Pattern Recognition*, 2017.
- [10] F. Walch, C. Hazirbas, L. Leal-Taixe, T. Sattler, S. Hilsenbeck, and D. Cremers, "Image-based localization using LSTMs for structured feature correlation," in *International Conference on Computer Vision (ICCV)*, 2017.
- [11] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixé, "Understanding the Limitations of CNN-based Absolute Camera Pose Regression," in *Conference on Computer Vision and Pattern Recognition*, 2019.
- [12] C. B. Fernandez, T. Braud, and P. Hui, "Implementing gdpr for mobile and ubiquitous computing," in *Proceedings of the 23rd Annual International Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 88–94. [Online]. Available: <https://doi.org/10.1145/3508396.3512880>
- [13] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From Coarse to Fine: Robust Hierarchical Localization at Large Scale," in *Conference on Computer Vision and Pattern Recognition*, 2019.
- [14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [15] T. Do, O. Miksik, J. DeGol, H. S. Park, and S. N. Sinha, "Learning to Detect Scene Landmarks for Camera Localization," in *Conference on Computer Vision and Pattern Recognition*, June 2022.
- [16] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," in *CVPR Workshops*, 2018.
- [17] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-Net: a Trainable CNN for Joint Description and Detection of Local Features," in *Conference on Computer Vision and Pattern Recognition*, 2019.
- [18] R. Jerome, P. Weinzaepfel, C. De Souza, and M. Humenberger, "R2D2: Reliable and Repeatable Detectors and Descriptors," in *NeurIPS*, 2019.
- [19] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning Feature Matching with Graph Neural Networks," in *Conference on Computer Vision and Pattern Recognition*, 2020.
- [20] J. Li, C. Wang, X. Kang, and Q. Zhao, "Camera localization for augmented reality and indoor positioning: a vision-based 3d feature database approach," *International Journal of Digital Earth*, 2020.
- [21] S. Gao, J. Wan, Y. Ping, X. Zhang, S. Dong, Y. Yang, H. Ning, J. Li, and Y. Guo, "Pose Refinement with Joint Optimization of Visual Points and Lines," in *ICCV Workshop*, 2022.
- [22] V. Panek, Z. Kukulova, and T. Sattler, "MeshLoc: Mesh-Based Visual Localization," in *European Conference on Computer Vision*, 2022.
- [23] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," in *Conference on Computer Vision and Pattern Recognition*, 2016.
- [24] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 Place Recognition by View Synthesis," in *Conference on Computer Vision and Pattern Recognition*, 2015.
- [25] L. Yang, R. Shrestha, W. Li, S. Liu, G. Zhang, Z. Cui, and P. Tan, "SceneSqueezer: Learning To Compress Scene for Camera Relocalization," in *Conference on Computer Vision and Pattern Recognition*, 2022.
- [26] P. Weinzaepfel, G. Csürka, Y. Cabon, and M. Humenberger, "Visual Localization by Learning Objects-of-Interest Dense Match Regression," in *Conference on Computer Vision and Pattern Recognition*, 2019.
- [27] A. Benbihi, C. Pradalier, and O. Chum, "Object-Guided Day-Night Visual Localization in Urban Scenes," in *International Conference on Pattern Recognition (ICPR)*, 2022.
- [28] M. Zins, G. Simon, and M.-O. Berger, "Oa-slam: Leveraging objects for camera relocalization in visual slam," pp. 720–728, 2022.
- [29] T. Sattler *et al.*, "Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions," in *Conference on Computer Vision and Pattern Recognition*, 2018.
- [30] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.
- [31] T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, and T. Pajdla, "Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions," in *Conference on Computer Vision and Pattern Recognition*, 2018.
- [32] H. Badino, D. Huber, and T. Kanade, "The CMU Visual Localization Data Set," <http://3dvis.ri.cmu.edu/data-sets/localization>, 2011.
- [33] M. Humenberger, Y. Cabon, N. Guerin, J. Morat, J. Revaud, P. Rerole, N. Pion, C. de Souza, V. Leroy, and G. Csürka, "Robust Image Retrieval-based Visual Localization using Kapture," 2020.
- [34] E. Wijmans and Y. Furukawa, "Exploiting 2D Floorplan for Building-scale Panorama RGBD Alignment," in *Computer Vision and Pattern Recognition, CVPR*, 2017.
- [35] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, "Real-time rgb-d camera relocalization," in *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013, pp. 173–179.
- [36] D. Kim, T.-Y. Lin, A. Angelova, I. S. Kweon, and W. Kuo, "Learning Open-World Object Proposals without Learning to Classify," *IEEE Robotics and Automation Letters (RA-L)*, 2022.
- [37] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in *Proceedings of the 36th International Conference on Machine Learning*, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6105–6114.
- [38] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *European Conference on Computer Vision*, 2017.
- [39] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *NeurIPS*, 2015.
- [40] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Conference on Computer Vision and Pattern Recognition*, 2017.
- [41] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Conference on Computer Vision and Pattern Recognition*, 2017.
- [42] L. Kneip, D. Scaramuzza, and R. Siegwart, "A Novel Parametrization of the Perspective-three-point Problem for a Direct Computation of Absolute Camera Position and Orientation," in *Conference on Computer Vision and Pattern Recognition*, 2011.
- [43] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Commun. ACM*, vol. 24, no. 6, 1981.
- [44] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski, "Building Rome in a Day," in *International Conference on Computer Vision (ICCV)*, 2009.
- [45] G. Jocher *et al.*, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation," Nov. 2022.
- [46] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common Objects in Context," 2014.