An Optimized Pipeline for Image-Based Localization in Museums from Egocentric Images

Nicola Messina[1] [0000−0003−3011−2487], Fabrizio Falchi[1] [0000−0001−6258−5313], Antonino Furnari[2] [0000−0001−6911−0302], Claudio Gennaro[1] [0000−0002−3715−149X], and Giovanni Maria Farinella[2] [0000−0002−6034−0432]

[1] ISTI-CNR, via G. Moruzzi, 1 - 56017 Pisa, Italy
nicola.messina@isti.cnr.it
[2] University of Catania, Viale A. Doria 6 - 95125 Catania, Italy

Abstract. With the increasing interest in augmented and virtual reality, visual localization is acquiring a key role in many downstream applications requiring a real-time estimate of the user location only from visual streams. In this paper, we propose an optimized hierarchical localization pipeline by specifically tackling cultural heritage sites with specific applications in museums. Specifically, we propose to enhance the Structure from Motion (SfM) pipeline for constructing the sparse 3D point cloud by a-priori filtering blurred and near-duplicated images. We also study an improved inference pipeline that merges similarity-based localization with geometric pose estimation to effectively mitigate the effect of strong outliers. We show that the proposed optimized pipeline obtains the lowest localization error on the challenging Bellomo dataset [11]. Our proposed approach keeps both build and inference times bounded, in turn enabling the deployment of this pipeline in real-world scenarios.

Keywords: Localization · Camera Pose Estimation · Structure From Motion · Egocentric Vision

1 Introduction

Virtual Reality (VR) is becoming a game-changing technology in many scenarios – from gaming to medical applications – and is being increasingly applied to the exploration and preservation of cultural heritage sites [12, 2]. Visual localization is a critical task to enable stable and reliable VR applications on these sites, where it can be used to enhance the visitor experience and receive contextualized information about the visited rooms and observed artworks [14]. Furthermore, visual localization is employed in other real-world applications – like in robot navigation, mixed reality, and self-driving vehicles – becoming a critical computer vision task. In recent years, a significant amount of research has focused on developing deep learning-based methods to directly regress 3D camera coordinates from raw images. However, such methods require extensive network

This is a pre-copyedit version of this article. The final authenticated version is available online at https://doi.org/10.1007/978-3-031-43148-7_43

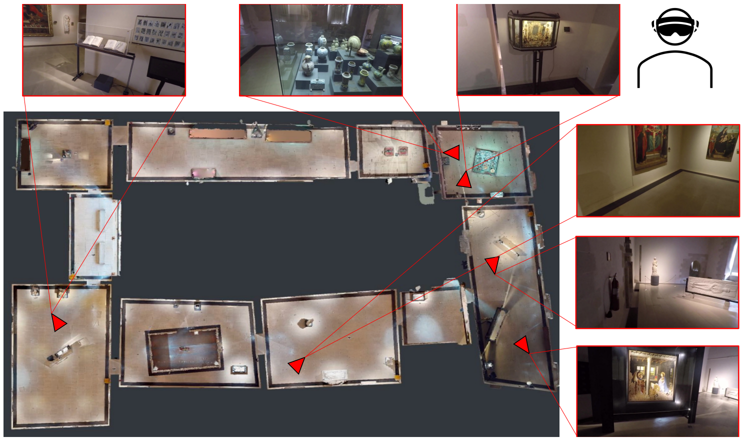


Fig. 1: The considered image-based localization problem consists in localizing a visitor of a museum from egocentric images collected through a wearable or mobile device. The figure shows some examples from the dataset proposed in [11], along with their positions in the map.

training time and resources. Furthermore, this expensive training should be performed once for every scenario, limiting the applicability of these approaches to real-world cases. Previous methods have proposed to use hierarchical localization (HLOC) approaches leveraging global feature matching and Structure from Motion (SfM) for fine-grained localization [15]. Despite their flexibility, these approaches have some known limitations in real environments, where image data can be noisy or blurred, and the presence of strong outliers invalidates their effectiveness.

In this paper, we propose an optimized pipeline for image-based localization based on the hierarchical localization idea. We test it in a museum environment, where the main downstream use case would be capturing user location through wearable devices or smartphones to optimize the visitor experience (see Figure 1). Using a streamlined pipeline that discards blur images and duplicates, we propose a method that constructs a sparse 3D point cloud from raw reference images in under a couple of minutes, surpassing by a large margin long deep network training times and improving the HLOC framework by obtaining a 20x boost on build times with limited localization degradation. The 3D point model built using SfM allows us to perform both global image matching for coarse-grained localization and local matching for fine-grained localization using geometric camera pose estimation. One of the contributions proposed for mitigating the high variance of the geometric camera pose estimation is the fusion of fine- and coarse-grained localization pipelines. The choice of the localization method is simply driven by the estimated quality of the match between the query image and the ones registered in the 3D point cloud. The proposed approach efficiently queries the 3D point model and achieves high accuracy with less than one-second latency on a standalone desktop computer embedded with a mid-end graphic card, and it can be further engineered to run on portable localization devices such as smartphones or smart glasses – particularly relevant in the context of cultural sites. We evaluate the approach on four rooms from the Bellomo dataset [11], and the results demonstrate the effectiveness and efficiency of our proposed approach compared to current state-of-the-art methods.

To summarize, the main contributions are as follows:

- We propose a streamlined pipeline for efficiently and effectively constructing a sparse 3D point cloud using SfM from raw references, leveraging on the filtering of blurred and duplicated images.
- We employ both local and global localization outputs provided by the hierarchical localization pipeline to efficiently and effectively estimate the correct location.
- We perform extensive experimentation on four rooms from the Bellomo dataset, obtaining good accuracies with less than 1-second latencies.

2 Related Work

Localization methods based on classification rely on a discretization of the space in cells and train a classifier to infer the correct cell from an input RGB image. In this context, only the rough camera location is estimated, while its orientation is not predicted. Seminal works [21] considered the problem of inferring the room in which the user is located with classification approaches based on handcrafted features. More recent methods used Bag of Words representation [8, 4], while others are based on CNNs [23]. Others [7, 13] performed classification-based localization considering an *open-set* problem in which the camera may also acquire images of new locations not initially included at training time.

Among the approaches based on camera pose estimation, a line of works approximates the location of a test image assigning it the pose of the most similar image in the training set, as predicted by image retrieval methods [1, 23].

Some methods treated camera pose estimation as a regression problem in which camera coordinates are predicted directly from monocular images. Most of these methods are based on a backbone CNN to extract features, later used to regress the camera pose [9, 22]. Others predict relative camera pose [3, 10]. (i.e., the pose of a test image relative to one or more training images).

Localization methods based on local feature matchings are the most accurate ones, as they directly link 2D local features extracted from the image to 3D scene coordinates. Matchings can be obtained with a descriptor matching algorithm [17] or regressing 3D coordinates from image patches [16, 18].

Visual SLAM (Simultaneous Localization and Mapping) [6] is another widely studied set of methods for acquiring the location of a moving agent using camera sensors. However, SLAM makes assumptions quite different from our scenario and may present problems in our specific use case. In particular, i) SLAM assumes the environment is not known, while the model of our environment is always available and contains some labeled ground-truth positions; ii) SLAM has known issues when the camera is abruptly shaken – which often happens if the camera is from a smartphone or mounted on smartglasses iii) SLAM should rely on separate localization methods if the video stream is discontinuous, due to the so-called *kidnapped robot problem*. This problem may arise in our scenario, where the device may be activated only when the visitor needs it.

Particularly related to this paper are hierarchical localization (HLOC) works based on the combination of image retrieval approaches and geometric correspondences [15]. These approaches are based on a database of localized images for the first image retrieval step and the construction of a 3D model of the scene through Structure from Motion (SfM) to perform accurate camera pose estimation. Specifically, the work in [15] employs COLMAP [20] for performing SfM and constructing a 3D point cloud from a set of pictures taken in the environment. The method employs a monolithic approach, based on a shared CNN-based visual backbone, to produce both global descriptors through a NetVLAD head [1] and local descriptors using the efficient SuperPoint decoder [5].

3 Method

We rely on the hierarchical localization framework presented in [15], and we propose an optimized pipeline that obtains the best effectiveness and efficiency in cultural heritage sites like museums. In fact, although this approach obtains stable results, it adds some complexity that prevents its usability in real-time real-world scenarios where acquired data are noisy and redundant. Following, we describe in detail the improvements introduced in the pipeline to handle unclean data and enhance the framework for use in a specific downstream scenario.

3.1 Model Building

Usually, SfM is exceedingly expensive, and the construction time increases with the number of matching image pairs. To decrease the number of image matches, the hierarchical localization framework allows employing only the most similar images to a given one to check for local feature matches. This is done by performing k nearest neighbor search using the NetVLAD [1] global feature. In the experiments, we refer to the k used to build the model as k_{build}. Even if beneficial, this smart pair filtering procedure is often not sufficient for reaching competitive localization performance and build times. With datasets like the Bellomo dataset [11], where frames are sampled from a video acquired by a wearable device, the quality of the acquired images is often limited. Specifically, many frames from the video are blurred or near-duplicated. While near-duplication mostly causes problems in efficiency due to the increasing number of less informative images that have to be considered by SfM, frame blurring also has disadvantages in reconstruction accuracy, given that blurred images cause many false matches. Hence, we apply near-duplicate removal and blur image filtering to optimize the model creation process. We show in the experiments how these pre-processing steps help achieve higher performance and better efficiency.

Near-duplicate removal. Near duplicate removal relies on the similarity between low-level descriptors for finding almost identical keyframes. We can reuse the NetVLAD global descriptor used during the first image search stage for filtering out duplicated images at model construction time. This can be obtained by scanning the list of vectors starting from the last element $i = N - 1$, and finding if there is at least of the previous elements $j < i$ such that $S(i, j) >= \delta_{\text{duplicate}}$, where $S(\cdot, \cdot)$ is the dot product in our case. If the above condition is met, element i is marked as duplicate. Notice that with this formulation, the first element $i = 0$ is never considered a duplicate, which makes sense in our scenario where images are sequentially obtained from a real-time acquisition device.

Blur image filtering. We rely on a simple approach based on the variance of the Laplacian of the image, which is computationally efficient and already suffices for our goal. In particular, to find blurred images, given the Laplacian of the pixel intensities I computed as $L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$, we can compute its variance across all the image pixels $\text{Var}[L]$ and check if this value is below a certain threshold δ_b. We then keep all the images such that $\text{Var}[L] > \delta_b$.

Model Geo-registration. SfM allows us to reconstruct a sparse point cloud of the environment, but it cannot infer the scale of the model until some of the points in the cloud are annotated with real-world coordinates. In order to estimate the location error in meters, we need to infer the correct scale of the point cloud. This can be achieved through the model geo-registration function of COLMAP, which takes the 3D coordinates of a subset of registered images as input and infers the model scale through a RANSAC estimator to be robust to possible outliers. Theoretically, only three images are sufficient for geo-registering a model. Nevertheless, usually, more images are used to diminish the effect of possible imprecise ground-truth annotations. Specifically, in our scenario, we used the 3D coordinates associated with the images in the training set to register the models of each room in the dataset.

3.2 Localization

Differently from the model building pipeline, the localization phase should happen in real-time. In this phase, we do not a-priori filter images based on their blurriness, as we could potentially ground every image in the 3D point cloud for deriving the user location. Drawing inspiration from the hierarchical localization method in [15], this method is based on a coarse-grained localization which uses an image-similarity-based approach, and a fine-grained localization, which instead relies on geometric pose estimation.

For coarse-grained localization, we employ k-nearest-neighbor search using global features to search the images registered in the 3D point cloud more similar to the query image. The images registered in the point cloud have been assigned 3D coordinates, so we can easily infer an approximate query location as follows:

$$\mathbf{x}_{\text{coarse}} = \frac{1}{k_{\text{infer}}} \sum_{i=1}^{k_{\text{infer}}} \mathbf{X}_i, \qquad \text{where} \quad \mathbf{X} = \{\mathbf{x}_i | i \in \text{search}(V_{\text{train}}, v_q, k_{\text{infer}})\}, \quad (1)$$

where search$(\cdot,\cdot,\cdot)$ finds the indexes of the k_{infer} most similar images to v_q in the training image set V_{train}, and $\mathbf{X}$ is the set of locations of the nearest neighbors registered images. Note that not all the training images are registered in the point cloud. Therefore, in some cases, it may happen that the resulting set $\mathbf{X}$ is empty. As it can be noticed, using $k_{\text{infer}} = 1$ we can simply localize the camera using the coordinates of the nearest neighbor only. If $k_{\text{infer}} > 1$, we are instead computing the centroid among the coordinates associated with the k-nearest-neighbors. We experimentally show that $k_{\text{infer}} = 1$ works the best in our scenario.

Fine-grained localization in the HLOC framework depends on the results from the coarse-grained method. Specifically, we employ k-nearest-neighbor search using global features to search among the training images the ones registered in the 3D point cloud more similar to the query image. Then, we can perform local feature matching between the query image and the local features already registered on the point cloud for the k images:

$$\mathbf{x}_{\text{fine}} = \text{pose\_estimation}(\mathcal{M}^{\text{3D}}, \mathbf{M}_q^{\text{2D}}, \mathcal{I}), \tag{2}$$

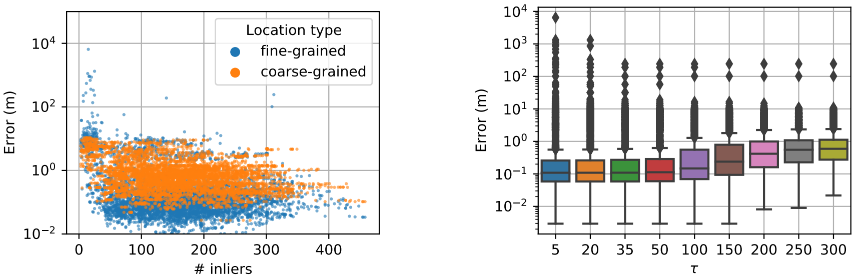
$$\text{where} \quad \begin{cases} \mathcal{M}^{\text{2D}}, \mathcal{M}^{\text{3D}} = \{(\mathbf{M}_i^{\text{2D}}, \mathbf{M}_i^{\text{3D}}) | i \in \text{search}(V_{\text{train}}, v_q, k_{\text{infer}})\} \\ \mathcal{I} = \text{match}(\mathcal{M}^{\text{2D}}, \mathbf{M}_q^{\text{2D}}) \end{cases} \tag{3}$$

$\mathbf{M}_i^{\text{2D}}$ and $\mathbf{M}_i^{\text{3D}}$ are the 2D and associated 3D coordinates of the found joints in the k_{infer} neighboring images, $\mathbf{M}_q^{\text{2D}}$ is the set of local features found in the query image, and $\mathcal{I}$ is the set of local feature inliers. The pose_estimation$(\cdot,\cdot)$ function is a COLMAP function[3] which performs geometric pose estimation using the matching 2D inliers to derive the actual pose, indicated as $\mathbf{x}_{\text{fine}}$. Note that $\mathcal{I}$ could be empty either if there are no retrieved images registered in the point cloud, or if there are no matching local features. In that case, the fine-grained position cannot be estimated, but we show in the experiments that this happens with an acceptable probability for a real-case scenario.

3.3 Mixing the Localization Outcomes

Although fine-grained localization has potentially higher accuracy, there may be some strong outliers due to failures in the geometric pose estimation. For this reason, we decided to prioritize the fine-grained over coarse-grained localization only if the number of inliers (indicated as $|\mathcal{I}|$) found from the local features matching phase is above a certain threshold τ. We argue that the number of matches is a good indicator of the quality of the fine-grained localization, and we prove it empirically in the experimental evaluation. Therefore, the final estimated position $\mathbf{x}$ is $\mathbf{x}_{\text{coarse}}$ if $|\mathcal{I}| < \tau$ and $\mathbf{x}_{\text{fine}}$ otherwise. The localization error is then computed using the standard Euclidean distance with the ground-truth position values $\mathbf{x}_{GT}$ provided within the dataset: $e = ||\mathbf{x} - \mathbf{x}_{GT}||_2$.

[3] https://github.com/colmap/pycolmap/blob/master/estimators/absolute_pose.cc



(a) Distribution of localization errors depending on the number of inliers for both coarse- and fine-grained estimated poses.

(b) Average localization error varying the threshold on the number of considered inliers.

Fig. 2: Analysis of the effectiveness of the mixing between fine- and coarse-grained pose estimations using the number of the inliers as the threshold.

4 Experiments

4.1 Dataset

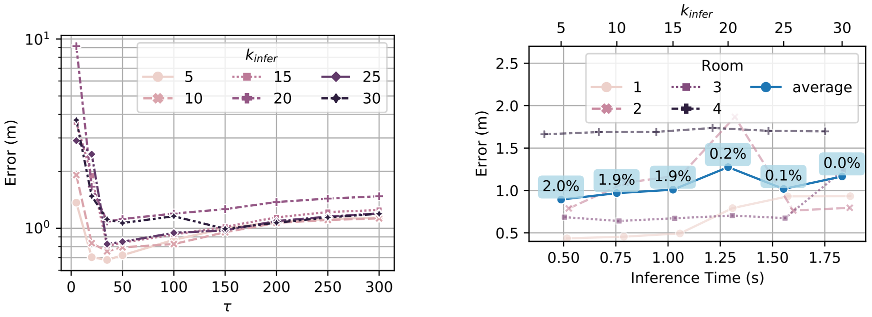
The dataset used in this research has been introduced in [11]. It has been recorded in the Bellomo Palace Regional Gallery, a museum located in Syracuse, Italy. To capture the visitors' experiences, the authors recorded 10 videos using a GoPro Hero 4 wearable camera and Matterport 3D to create a 3D scan of the museum's environment. They selected four rooms within the museum to collect data, as they contained a variety of items such as statues, paintings, and display cases, which provide a representative sample of what a museum typically offers. The videos were extracted into image sequences. The obtained images are divided into three sets for training, testing, and validation. Specifically, all frames from the first to sixth video are used as the training set, frames from the seventh and eighth videos as the test set, and frames from the ninth and tenth videos as the validation set. We consider all position estimations further away than 1000m from the accessible area (far beyond the boundaries of a museum) as a failure in localization and discard them before computing the average localization errors.

We run our method on a mid-end desktop computer equipped with an RTX-2080Ti graphic card and an AMD Ryzen 7 1700 Eight-Core Processor.

4.2 Parameters Study

We run a preliminary analysis on the validation set to fix some of the system's hyper-parameters.

First, we focus on the model-building procedure for analyzing hyperparameters like k_{build}, δ_{blur}, $\delta_{duplicate}$. The results from the exploration of different build parameter configurations are reported in Table 1. We derive meaningful values for δ_{blur} and $\delta_{duplicate}$ from their distribution on the validation set. The



(a) Average localization error varying the threshold τ, for different values of k_{infer}.

(b) Efficiency vs effectiveness, varying k_{infer}. In blue boxes, the avg. percentage of failure cases is reported.

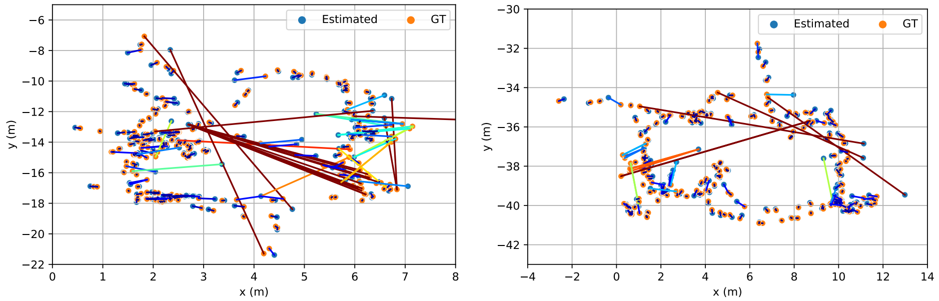
Fig. 3: Effect of hyper-parameters k_{infer} and τ on effectiveness and efficiency, on the validation set.

Table 1: Mean location error and build times. *Survived images* are the images that remained after the blur and near-duplicate filtration, while *registered images* are the ones that were registered by COLMAP on the 3D point cloud.

δ_{blur}	$\delta_{\text{duplicate}}$	k_{build}	Error (m)	Time (s)	survived images (%)	registered images (%)
70	0.45	10	1.96	60.1	35.2	33.1
70	0.45	15	1.48	75.7	35.2	33.5
90	0.45	10	1.28	48.8	30.6	28.7
90	0.45	15	1.54	60.0	30.6	29.2
90	0.55	10	1.19	72.5	43.3	41.3
90	0.55	15	2.23	118.0	43.3	41.7
0	1.0	10	0.78	1524.1	100.0	98.8

localization error, measured in meters, is averaged through a selected range of values for the threshold τ and k_{infer} to give an overall estimate of the model's performance without a-priori setting any inference hyper-parameters. We leave the results obtained without any filtering ($\delta_{\text{blur}} = 0$, $\delta_{duplicate} = 1.0$) as the last row of the table. We can see how, using blur and near-duplicate filtering, we can obtain overall comparable error values with this original approach, in turn decreasing the build times with a speedup of more than 20x. Given its effectiveness-efficiency ratio, we consider the model built with $\delta_{\text{blur}} = 90$ and $\delta_{duplicate} = 0.55$ for further experiments on the inference parameters.

Next, we proceed by studying the hyper-parameters k_{infer} and τ. As previously hypothesized, while the advantage of fine-grained localization is the potential high accuracy, there are some strong outliers due to geometric estimation failures. This behavior is shown in Figure 2a. If we apply the thresholding for deciding if either using fine-grained or coarse-grained localization outputs, we notice in Figure 3b how we are able to diminish the number of outliers when the



(a) Room 2.

(b) Room 3.

Fig. 4: Visualization of predicted with respect to ground-truth poses, projected on the XY plane. Lines represent corresponding ground truth and predicted poses. Pairs are colored with a gradient indicating the localization error (from blue – small error – to red – large error).

threshold τ is increased. This, in turn, validates our hypothesis that a higher number of inliers contributes to a better fine-grained localization. In Figure 3a, we study how the localization error varies depending on the number of nearest neighbors. In particular, the lowest error is achieved for $k_{\text{infer}} \in [5, 10]$, for a relative small τ. This is reasonable since (i) too few or too many nearest neighbor registered images can provide noisy local matches that degrade the fine-grained geometric localization, and (ii) high thresholds τ inhibit the advantages introduced by the fine-grained localization. Analyzing the plots, we decide to fix $k_{\text{infer}} = 5$ and $\tau = 50$ in the rest of the experiments.

In Figure 3b, we show how, varying k_{infer}, we also obtain different system latencies, as the process of geometric position estimation becomes more and more expensive with an increasing number of local feature matches. The choice of $k_{\text{infer}} = 10$ keeps the response time below 0.8 seconds, enabling a sufficient frame rate for localizing the user in real-time. The only drawback of keeping k_{infer} low is that we have, on average, 2% of query images that cannot be localized due to either (i) failure of coarse-grained localization – there are no images registered in the 3D point cloud among the first k_{infer} found – (ii) failure of fine-grained localization – there are no local features matches among the registered images found among the first k_{infer} ones – or (iii) the estimated location is beyond 1000m from the walkable area, which we consider a failure as well.

4.3 Results

We compare our method with the following state-of-the-art visual localization approaches: i) a SIFT-based image retrieval approach, called *Vote And Verify* [19], which tackles primarily image retrieval but enforces geometric verification constraints; ii) the PoseNet approach [9], which directly regress pose using a deep convolution network; iii) PAM-CAM [11], which also regresses the camera pose but using a more advanced deep network embodied with attention modules and trained by employing a self-supervision approach.

We report the results using the hyper-parameters set as explained in Section 4.2. We report four different variants: i) *only FG* is the model only employing fine-grained features matching, which downcasts the inference method to the one proposed in the HLOC framework [15]; ii) *only CG 1-nn* and iii) *only CG 5-nn* are the models employing only coarse localization – i.e., the position of the most-likely image registered in the 3D point cloud, using one nearest neighbor and the centroid among the five nearest neighbors respectively; iv) *FG + CG* is the final method employing both coarse-grained and fine-grained localization, using the thresholding method explained in Section 3.2.

Final results are reported in Table 2a. All the methods we use for comparison and reported in the table have been fine-tuned on the Bellomo dataset by the authors in [11]. The proposed method outperforms all the other ones on this challenging benchmark. Specifically, although either the FG or CG methods alone cannot improve over the state-of-the-art, we obtained the best results when employing both approaches jointly. These results prove that the non-regression-based approaches relying on geometric verification can obtain the best results by keeping the build (Table 1) and inference times (Figure 3b) bounded for enabling real-time visitor localization. It is also interesting to note that 1-nn in the CG configuration obtains the best results over the 5-nn one, probably because the NetVLAD global features can retrieve with a high likelihood the most relevant registered images as the first result. In Figure 2b, we also report localization errors for all four rooms. The highest contribution to the error comes from *Room 4*, probably due to scarce lighting and a big glass case in the middle, which creates false positive matching among local features. However, the median is far lower than the mean, suggesting that the relatively few outliers still have a strong impact which should be further mitigated in future works.

In Figure 4, we show qualitative results of the estimated and ground-truth position pairs in two rooms of the Bellomo dataset. Lines indicate corresponding estimated and ground-truth positions. Some failure cases are particularly visible in Figure 4a, where – due to failure in geometric estimation and retrieval of the correct 1-nn image – some query images are associated with the wrong registered camera. Apart from these edge cases, we can notice that the location is generally estimated with good accuracy.

5 Conclusions

This paper proposes an efficient pipeline based on the Hierarchical Localization (HLOC) framework for localizing egocentric video streams in interior cultural heritage sites, such as museums. The proposed method overcame some of the drawbacks of the original HLOC framework by filtering out uninformative visual inputs – near-duplicated or blurred images – and proposing a smart aggregation of localization information from both fine- and coarse-grained modules to mitigate the effect that strong outliers have in the geometric estimation. Thanks to

(a) Results on the test sets, averaged among the four different rooms.

Method	**Loc. error (m)**
PoseNet (beta 100)	1.43
Vote-and-Verify	0.82
PAM-CAM	1.26
Our (only FG)	1.65
Our (only CG 1-nn)	1.16
Our (only CG 5-nn)	1.24
Our (FG + CG 1-nn)	**0.62**

(b) Localization error (mean and median, in meters) for each of the four rooms.

	R1	**R2**	**R3**	**R4**
mean	0.48	0.66	0.42	1.66
median	0.09	0.10	0.11	0.19

Table 2: Final localization results on the Bellomo dataset.

extensive experimentation on the challenging Bellomo dataset, we were able to characterize the most influential factors affecting both fine- and coarse-grained localization outcomes, obtaining state-of-the-art results with respect to other approaches on the same dataset.

In future works, we plan to address 6D localization by including orientation estimation, and we plan to substitute the SfM construction pipeline by employing Matterport 3D scans to estimate the position using the depth information without relying on a 3D point cloud. This would further increase the efficiency and the overall usability of the proposed localization framework.

Acknowledgements This research was supported by AI4Media – A European Excellence Centre for Media, Society, and Democracy (EC, H2020 No. 951911), by SUN – Social and hUman ceNtered XR (EC, Horizon Europe No. 101092612), and by VALUE – Visual Analysis For Location And Understanding Of Enviroments (PO FESR 2014/2020 program, action 1.1.5).

References

1. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: Netvlad: Cnn architecture for weakly supervised place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5297–5307 (2016)
2. Arrighi, G., See, Z.S., Jones, D.: Victoria theatre virtual reality: A digital heritage case study and user experience design. Digital Applications in Archaeology and Cultural Heritage **21**, e00176 (2021)
3. Balntas, V., Li, S., Prisacariu, V.: Relocnet: Continuous metric learning relocalisation using neural nets. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 751–767 (2018)
4. Cao, S., Snavely, N.: Graph-based discriminative learning for location recognition. In: Proceedings of the ieee conference on computer vision and pattern recognition. pp. 700–707 (2013)

5. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 224–236 (2018)
6. Durrant-Whyte, H., Bailey, T.: Simultaneous localization and mapping: part i. IEEE robotics & automation magazine **13**(2), 99–110 (2006)
7. Furnari, A., Farinella, G.M., Battiato, S.: Recognizing personal locations from egocentric videos. IEEE Transactions on Human-Machine Systems **47**(1), 6–18 (2016)
8. Ishihara, T., Vongkulbhisal, J., Kitani, K.M., Asakawa, C.: Beacon-guided structure from motion for smartphone-based navigation. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 769–777. IEEE (2017)
9. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: Proceedings of the IEEE international conference on computer vision. pp. 2938–2946 (2015)
10. Laskar, Z., Melekhov, I., Kalia, S., Kannala, J.: Camera relocalization by computing pairwise relative poses using convolutional neural network. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 929–938 (2017)
11. Mauro, D.D., Furnari, A., Signorello, G., Farinella, G.M.: Unsupervised domain adaptation for 6dof indoor localization. In: International Conference on Computer Vision Theory and Applications - VISAPP (2021), https://iplab.dmi.unict.it/EGO-CH-LOC-UDA/
12. Milosz, M., Skulimowski, S., Kęsik, J., Montusiewicz, J.: Virtual and interactive museum of archaeological artefacts from afrasiyab–an ancient city on the silk road. Digital Applications in Archaeology and Cultural Heritage **18**, e00155 (2020)
13. Ragusa, F., Furnari, A., Battiato, S., Signorello, G., Farinella, G.M.: Egocentric visitors localization in cultural sites. Journal on Computing and Cultural Heritage (JOCCH) **12**(2), 11 (2019)
14. Ragusa, F., Furnari, A., Battiato, S., Signorello, G., Farinella, G.M.: Ego-ch: Dataset and fundamental tasks for visitors behavioral understanding using egocentric vision. Pattern Recognition Letters **131**, 150–157 (2020)
15. Sarlin, P.E., Cadena, C., Siegwart, R., Dymczyk, M.: From coarse to fine: Robust hierarchical localization at large scale. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12716–12725 (2019)
16. Sarlin, P.E., Debraine, F., Dymczyk, M., Siegwart, R., Cadena, C.: Leveraging deep visual descriptors for hierarchical efficient localization. In: Conference on Robot Learning. pp. 456–465. PMLR (2018)
17. Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A.: Superglue: Learning feature matching with graph neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4938–4947 (2020)
18. Sarlin, P.E., Unagar, A., Larsson, M., Germain, H., Toft, C., Larsson, V., Pollefeys, M., Lepetit, V., Hammarstrand, L., Kahl, F., et al.: Back to the feature: Learning robust camera localization from pixels to pose. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3247–3257 (2021)
19. Schönberger, J.L., Price, T., Sattler, T., Frahm, J.M., Pollefeys, M.: A vote-and-verify strategy for fast spatial verification in image retrieval. In: Asian Conference on Computer Vision (ACCV) (2016)
20. Schönberger, J.L., Zheng, E., Pollefeys, M., Frahm, J.M.: Pixelwise view selection for unstructured multi-view stereo. In: European Conference on Computer Vision (ECCV) (2016)

21. Starner, T., Schiele, B., Pentland, A.: Visual contextual awareness in wearable computing. In: Digest of Papers. Second International Symposium on Wearable Computers (Cat. No. 98EX215). pp. 50–57. IEEE (1998)
22. Walch, F., Hazirbas, C., Leal-Taixe, L., Sattler, T., Hilsenbeck, S., Cremers, D.: Image-based localization using lstms for structured feature correlation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 627–637 (2017)
23. Weyand, T., Kostrikov, I., Philbin, J.: Planet-photo geolocation with convolutional neural networks. In: European Conference on Computer Vision. pp. 37–55. Springer (2016)