Egocentric Visitor Localization and Artwork Detection in Cultural Sites Using Synthetic Data

Santi Andrea Orlando\textsuperscript{a,b,2}, Antonino Furnari\textsuperscript{a,2}, Giovanni Maria Farinella\textsuperscript{a,c,2,**}

\textsuperscript{a}University of Catania, Department of Mathematics and Computer Science, Via Santa Sofia - 64, Catania 95125, Italy
\textsuperscript{b}DWORD - Xenia s.r.l., Acicastello, Catania, Italy
\textsuperscript{c}National Research Council, ICAR-CNR, Palermo, Italy

ABSTRACT

Computer vision and machine learning can be used in cultural heritage to augment the experience of visitors during the exploration of the cultural site, as well as to assist its management. To achieve such goals, two fundamental tasks should be addressed, i.e., localizing visitors and recognizing the observed artworks. Wearable cameras offer a convenient setting to address both tasks through the analysis of images acquired from the visitors’ points of view. However, the engineering of approaches to address such tasks generally requires large amounts of labeled data. We propose a tool which can be used to collect and automatically label synthetic visual data suitable to study image-based localization and artwork detection. The tool simulates a virtual agent navigating the 3D model of a real cultural site and automatically captures video frames along with the related ground truth camera poses and semantic masks indicating the position of artworks. We generate a dataset of synthetic images starting from the 3D model of a museum located in Siracusa, Italy. The experiments suggest that the proposed tool allows to drastically reduce the effort needed to collect and label data, providing a means to generate large-scale datasets suitable to study localization and artwork detection in cultural sites.

© 2020 Elsevier Ltd. All rights reserved.

1. Introduction

Visual information collected from the point of view of the visitor by means of a wearable camera enables new applications and services useful both for the visitors and the manager of a cultural site \cite{1, 2}. In particular, devices, such as Hololens, can be used to augment the fruition of the cultural sites through augmented reality and, at the same time, to collect visual information useful to understand their behavior. Examples of useful applications are, for instance, indoor localization and navigation to monitor the paths followed by visitors and guide them to reach a specific Point Of Interest (POI), such as an artwork or a location. Another application could be a recommendation system which suggests the next point of interest to see based on the history of already visited locations and observed objects. The deployment of most modern approaches to the aforementioned applications generally relies on large amounts of domain-specific data labeled for the task at hand, which can hinder the study and development of new methods. Moreover, egocentric data useful to design vision systems can be difficult to collect due to privacy concerns. To address this issue, we propose to generate synthetic data using the procedure illustrated in Fig. 1. This pipeline involves the acquisition of a 3D model of the real cultural site using a commercial tool such as Matterport 3D\textsuperscript{3}. We hence introduce a tool developed using the Unity game engine\textsuperscript{4}, which allows to add semantic labels to specific locations and objects (e.g., artworks) directly in the 3D model. The tool allows to simulate a virtual agent which navigates the environment observing the specified objects. During the navigation, the tool collects RGB frames automatically labeled with location- and object-based labels. It is worth noting that the whole procedure only requires a 3D scan of the considered site and the labeling of the 3D model to allow the auto-

\textsuperscript{**}Corresponding author: Tel.: +39 095 7337 219; fax: +39 095 330094; e-mail: gfarinella@dmi.unict.it (Giovanni Maria Farinella)

\textsuperscript{2}These authors are co-first authors and contributed equally to this work.

\textsuperscript{3}https://matterport.com/.

\textsuperscript{4}Unity game engine: https://unity.com
matic generation of an arbitrary number of labeled visual data. Moreover, if new labels need to be included in the pipeline, it is only required to update the labeling and repeat the automatic data generation procedure. This makes the proposed procedure much less expensive and much more scalable than manually collecting and labeling data, especially during the design of the egocentric vision algorithms. The proposed tool is available for research purposes at our web page\(^5\).

To show the usefulness of the proposed pipeline and tool, we generated simulated data from the 3D models of two real environments. The first one includes the 1\(^{st}\) floor of the Galleria Regionale Palazzo Bellomo museum\(^6\). The second environment is an office area, previously introduced in \([3]\). The datasets generated from the two environments are different in terms of areas dimension, nature of the building, objects, and number of images. We use the collected data to benchmark the two aforementioned tasks of visitor localization and artwork detection. In particular, egocentric visitor localization is investigated using an approach based on image retrieval which relies on a metric learned using a triplet loss \([4]\). We perform experiments to assess the influence of the dataset size, the ability of a localization approach to generalize across datasets, as well as the effect of temporal smoothing on localization accuracy. The artwork detection problem is investigated by training three state-of-the-art object detectors on the generated data.

The results highlight that: i) the generated data can be effectively used to learn a suitable metric for localization based on image-retrieval; ii) the number of labeled images affects the performance of image-based localization; iii) temporal smoothing is an effective technique to improve the results of localization approaches; iv) state-of-the-art object detectors can be used to detect and localize artworks given enough labeled data.

In sum, the contributions of this work are as follows: 1) we propose and publicly release a tool to generate simulated egocentric visual data of a real cultural site. The generation/labeling process is inexpensive as the data is automatically labeled for tasks of image-based localization and artwork detection, 2) we generate and release a dataset of labeled egocentric images obtained by performing simulations in a real cultural site, 3) we perform a benchmark of egocentric visitor localization and artwork detection on the generated data. The benchmark is designed to assess the influence of the dataset size, the ability of a localization model to generalize across environments, and the potential of using temporal smoothing techniques to improve localization results.

2. Related Work

**Computer Vision for Augmented Fruition.** The use of wearable devices equipped with cameras for visit augmentation has been investigated in \([1]\). The authors of \([5]\) proposed a system for context-aware applications and tourist assistance. Convolutional Neural Networks (CNN) have been exploited for artwork classification in \([6]\). In \([7]\), the use of fully convolutional networks has been investigated to obtain image classification and object detection from egocentric cameras with the aim of enhancing audio-tour guides. The authors of \([8]\) proposed a system for the automatic detection of visual attention by using a head-mounted camera and computer vision techniques. In \([9]\), a room-based localization approach is designed to localize the visitors of a cultural site using wearable devices equipped with a camera. The authors of \([10]\) investigated the use of geo-referenced images collected from social media to understand visitors trends and patterns, as well as to highlight the most visited locations in a natural area. In this paper, we propose a general pipeline to generate egocentric visual data of a real cultural site which can support investigations akin the aforementioned ones.

**Image-Based Localization.** Approaches based on odometry \([11]\) and SLAM \([12, 13]\) can be used to perform localization in unknown buildings. However, these approaches are limited by the fact that the reference coordinate system used for localization is not generally specified in absolute terms, but, instead, it is referred to the first frame analyzed by the system. Other approaches assume the environment to be known a priori. For instance, in \([14]\), it is proposed to directly predict the 6DoF pose of cameras from the acquired images using a CNN. An alternative approach consists in the exploitation of image-retrieval techniques \([15]\). According to this scheme, a position is assigned to a query image by searching for the most similar image in a geo-tagged database \([16, 17, 18]\). In more general terms, recent works \([14, 19, 17, 9, 20]\) have shown that CNNs can be used to predict the position in which a query image has been captured. Unfortunately, in order to achieve good results, such methods require large labeled datasets, the collection of which can be demanding in terms of human effort and time. Inspired by the approaches based on image retrieval, we study the problem of image-based localization based on image retrieval mechanism by learning a suitable metric with a triplet loss.

**Datasets for Image-Based Indoor Localization.** As noted in \([15]\), the number of datasets for outdoor localization has increased over time thanks to the availability of geo-tagged data from social networks (e.g., Flicker) and research on autonomous driving \([21]\). Instead, datasets for indoor localization have been generally acquired in small environments, as in \([15]\).
While images of an outdoor environment can be in general labeled for image-based localization using Structure from Motion (SfM) techniques [14], this procedure is not trivial in indoor settings, as noted by different authors [22, 19]. This limits the size of current datasets and often requires the use of dedicated hardware to obtain ground truth poses. For example, the authors of [22] used a complex system called NavVis\(^7\) equipped with 6 cameras and 3 laser rangefinders to provide the 6 Degrees of Freedom (6DoF) pose of each acquired image. In this work, we aim to provide a useful tool to generate large quantities of labeled egocentric visual data which can be used to study the development of image-based indoor localization techniques.

**Artwork Detection.** While some authors proposed customized approaches to artwork detection in a museum [6, 7], such problem can be in general addressed by relying on standard object detection algorithm, when enough training data is provided. Modern object detectors based on CNNs can be divided into two categories. Two-stage detectors [23, 24, 25] perform object detection in two passes. In the first pass, category-independent region proposals are generated from the input image. The content of each region is hence classified as containing one of the objects of interest or none of them (background class). One-stage object detection approaches [26, 27, 28] present a unified pipeline that directly predicts class probabilities and bounding box coordinates from images without using pre-computed region proposals. To exploit object detection algorithms in the considered context, it is necessary to label each frame with the bounding box of each artwork. However, this procedure is generally time consuming. The proposed tool can be used to effortlessly generate images automatically labeled for object detection. This is achieved by labeling the artworks directly in the 3D model of the environment only once prior to the generation of the data. To assess the usefulness of the tool, we benchmark three popular object detectors [24, 28, 25] on the generated data.

3. Tool

To allow automatic data generation as illustrated in Fig. 1, we developed a tool using the Unity game engine. The proposed tool allows to import a 3D model of a scanned environment and render images from any position and orientation of the model expressed as a set of 6DoF coordinates. The tool allows to label the extent of each room in the building by specifying polygonal structures on the top view of the 3D model. Additionally, the position of specific objects (e.g., artworks) can be labeled directly in the 3D model by adding cuboid shapes which are rendered in a separate view referred to as the “semantic mask”. An example of this labeling procedure is shown in Fig. 2. Each artwork is color-coded with an RGB ID. Fig. 3 shows an example of a collected RGB-Semantic Mask pair, along with the related coordinates in the 3D model. Once the model has been labeled, each generated frame is automatically labeled with: 1) the related 6DoF camera pose, 2) the ID of the current room, 3) the ID of the object appearing in the center of the scene and its distance from the agent, 4) the semantic mask indicating the position of objects.

**Navigation Process of the Proposed Tool.** In order to collect realistic data of a visitor navigating the environment, we built a component able to simulate a virtual agent navigating the cultural site. To allow the navigation, we defined a grid of 2D points on the walkable area of the model. The points of the grid were placed 1.5 meters apart along both axes. For each labeled artwork we randomly generated between 30 and 50 observation points. Each observation point looks at the artwork from a different point of view. The navigation starts from a random room and is performed as follows. Let \( T \) be the number of targets in the room and let \( A_j \) be the number of artworks in the current room \( j \). All the \( T_j \) targets located in current room \( j \) are reached in a random order. The targets to be reached are mixed with the observation points of the artwork in the room. Specifically, every \( \frac{T_j}{n} \) navigation steps, \( n = 5 \) observation points are randomly selected for one of the artworks located in the room. The selected artwork is visited only once when visiting the room. After all targets in the room are visited, the next room is reached, and the routine is repeated. In each simulation, the agent navigates the rooms first in clockwise order, then in counter-clockwise order. For instance, if the navigation starts in clockwise order, after all rooms are reached, they are navigated again in counter-clockwise order. Fig. 4 shows an example of a possible navigation path generated by the proposed tool. The overall number of targets reached during a nav-

---

\(^7\)NavVis: https://www.navvis.com/
igation is $2 \cdot \sum_j (T_j + n \cdot A_j)$, where the factor 2 is due to the clockwise/counter-clockwise navigation schemes. During the visit we performed standard animations that simulate a look-around behavior similarly to what done in [20], with the exception that we did not include roll rotation.

When each observation point is reached, a “look at” behavior is applied to the camera, which triggers the agent to look at the artwork. This allows to collect realistic images of artworks which can be useful to perform artwork detection and recognition.

4. Datasets

**Bellomo Dataset.** By using the proposed tool, we collected a dataset for 6DoF image-based localization in a Cultural Site and artwork detection. The dataset has been generated considering as environment the 1st floor of Galleria Regionale Palazzo Bellomo of Syracuse, Italy (see Figure 5). The considered area has an extension of 760 m². We simulated 4 long navigations with a virtual agent 1.70 m tall. All frames have been captured using a virtual camera with a field of view of 60° at a frame rate of 5 fps. Since visitor localization in a map is already useful in cultural sites, we performed all our experiments in the framework of 3 Degrees of Freedom (3DoF) localization. To this aim, all 6DoF camera poses have been converted to 3DoF by discarding the z-axis and computing pose rotation by converting the Euler angle about the z-axis $\theta$ to a unit vector $(u, v) = (\cos \theta, \sin \theta)$. Fig. 5 shows an example of a 3DoF camera pose in the considered environment.

The dataset has been split into Training set, Validation set and Test set as follows:

- the Training set contains the frames of the 2nd and 3rd navigation, which accounts to 51,284 frames. For the experiments we have considered 4 random subsets containing: 25% of the training set (12,821 frames), 50% of the training set (25,642 frames), 75% of the training set (38,463 frames) and 100% of the training set;
- the Validation set contains the frames of 4th navigation, which accounts to 5,000 frames;
- the Test set contains the frames of 1st navigation which accounts to 24,525 frames.

Ideally, a good dataset for localization should be sampled densely enough to make sure that, for a given test image, an image with a similar pose is contained in the training set. To evaluate if this property is satisfied by the dataset, for each test sample we perform an optimal nearest neighbor search which associates it to the closest pose in the training set. Position and orientation errors are hence computed for each matched pair and averaged across the test set. The error computed using this ideal matching approach is a lower-bound for any localization method based on image retrieval. Since the concept of “closest pose” should be defined with respect to both position and orientation, we first define an orientation threshold $T_h$, and then match poses which have an orientation distance less than the threshold. A test sample $x_i$ is associated to the training sample $x_j^*$ using the following rule:

$$x_j^* = \arg \min_{x_j} \{ d_p(x_i, x_j) \mid d_o(x_i, x_j) < T_h, x_j \in \text{Train} \}$$

where $d_p(x_i, x_j)$ is the position distance (the Euclidean distance in our experiments) and $d_o(x_i, x_j)$ is the orientation distance. Table 1 reports four lower-bound errors computed considering different values for the threshold $T_h$. It is worth noting that, even for very small values of $T_h$, the dataset presents reasonably small position errors. Similarly, for a reasonable threshold of $10.825^\circ$, the lower-bound position error is as small as 0.165 m.

To benchmark artwork detection, we labeled 16 artworks in the 3D model. Each artwork in the generated images is hence associated a semantic segmentation mask, which is converted into a bounding box for experimental purposes.

**Stanford Dataset.** For the experiments on localization, we also considered the dataset proposed in [20]. This dataset has been generated from the 3D model of the S3DIS dataset in [3], an office area of about 330 m². To collect samples for this dataset, we have simulated 3 virtual agents with different heights (1.5 m, 1.6 m, 1.7 m) have been used to perform simulations. Each simulation is composed of 30 different navigations. In each navigation, the agent reaches 21 random target points. The target points are selected in such a way that the first and last
points to reach are the same. Following this procedure, we generated 90 navigations. This dataset contains a total of 886,823 images labeled with the related 3DoF camera poses.

5. Methods

This section describes the considered methods. In particular, Section 5.1 describes the pipeline employed to perform the image-based localization task. In this section, we describe the criterion used to train the deep learning and image-retrieval approaches. We also detail how we sample triplets for training purposes. Finally, we provide details about the temporal smoothing techniques implemented to improve the localization results. In Section 5.2, we illustrate the use of three different object detectors to perform artwork detection.

5.1. Image-Based Localization

To benchmark image-based localization, we considered an approach based on image-retrieval. The distance between two images is measured using a metric \( d_\theta \) induced by a representation function \( \Phi_\theta \) depending on parameters \( \theta \): \( d_\theta(x_i, x_j) = ||\Phi_\theta(x_i) - \Phi_\theta(x_j)||_2 \). In our experiments, we use an InceptionV3 [29] CNN\(^8\) as representation function \( \Phi_\theta \). We consider two versions of \( \Phi_\theta \). The first version is obtained by considering the CNN pre-trained on ImageNet, whereas the second version is obtained by fine-tuning the model pre-trained on ImageNet through a triplet architecture [4, 20].

Training Procedure of Triplet Network. The goal of the training procedure based on the triplet architecture is to encourage \( \Phi_\theta \) to map images of nearby places close to each other in the feature space, while maximizing the distance between images captured from distant locations. Training samples are provided in the form of triplets \( (x_i, x_i^+, x_i^-) \), i.e., tuples of three images: an “anchor” image \( x_i \), a “similar” image \( x_i^+ \) and a “dissimilar” image \( x_i^- \). The model is trained using a Margin Ranking Loss defined as follows:

\[
L_\theta(x_i, x_i^+, x_i^-) = \max(0, d_\theta(x_i, x_i^+) - d_\theta(x_i, x_i^-) + m)
\]  

where the margin \( m \) is a hyper-parameter, which we set to \( m = 0.2 \). The considered loss encourages the distance \( d_\theta(x_i, x_i^+) \) between the anchor and the positive image to be smaller than the distance between the anchor and the negative image \( d_\theta(x_i, x_i^-) \) by at least the margin \( m \).

Triplets Sampling. We sample triplets using the criterion described in [20], i.e., for each “anchor” image \( x_i \) in the training set, we randomly sample two images \( x_i^+ \) and \( x_i^- \) such that:

\[
\begin{align*}
d_\theta(x_i, x_i^+) &< th_o \land d_\theta(x_i, x_i^-) < th_o \\
d_\theta(x_i, x_i^+) &> th_o \lor d_\theta(x_i, x_i^-) > th_p
\end{align*}
\]  

where \( th_o \) is a threshold on orientation distances between poses and \( th_p \) is a threshold on position distances between poses. The network is trained for 100 epochs, using Stochastic Gradient Descent (SGD) with a fixed learning rate of \( 10^{-3} \) and a momentum equal to 0.9.

Temporal Smoothing. Since localization in a cultural site can be performed in a sequential fashion, we also investigate the use of online post-processing techniques which can be used to impose a temporal smoothing prior over the predicted coordinates. Specifically, we consider a filter which replaces the currently predicted pose \( x_i \) with a statistic \( \hat{x}_i \) of the last \( K \) observed poses:

\[
\hat{x}_i = \Gamma((x_{i-k})_{k=0}^{K-1})
\]  

where \( K \) is the size of the temporal window, \( x_{i-k} \) is the history of the last \( K \) observed poses and \( \Gamma \) is a statistical function. We considered three different choices for \( \Gamma \): the mean, the median, and a trimmed mean, i.e., the mean of the second and third quartiles of the sample. While the smoothing filter can be applied to the whole pose, we didn’t notice any improvement on orientations. Therefore, we apply temporal smoothing only to positions.

5.2. Artwork Detection

We benchmark artwork detection on the generated dataset by using three state of the art object detectors. Specifically, we consider Faster R-CNN [24], a two-stage object detector based on ResNet-101, Mask R-CNN [25], an extension of Faster R-CNN which jointly performs object detection and instance segmentation, and RetinaNet [28], a single-stage object detector based on a focal loss.

Since an accurate localization can be used to estimate what a visitor is looking at in a cultural site, we also considered an artwork detection baseline based on localization. Specifically, given a test image, we consider as the detected artwork, the object appearing in the center of the training image matched by the nearest neighbor search. We hence compare this object to the one actually appearing in the center of the test image.

6. Experimental Settings and Results

We trained two triplet networks on the Bellomo and Stanford datasets sampling triplets using the criterion in (3). We used as thresholds \( th_o = 45^\circ \) and \( th_p = 0.5 \) m, aligned with the results in [20].

Influence of the Size of the Training Set. We performed experiments to assess the influence of the size of the Training Set during the training and retrieval phases. Position and orientation errors are reported in the format \( \mu \pm \sigma \), where \( \mu \) is the mean error and \( \sigma \) is the standard deviation. Table 2(a) reports position and orientation errors obtained on the Bellomo dataset training the model with different percentages of training data. The training subsets have been obtained sub-sampling

---

8We remove the last layer of the CNN and use it as a feature extractor.
Table 2. Average position and orientation errors obtained using 25%, 50%, 75% and 100% training samples for training on the Bellomo dataset (a)-(c), and 10, 000, 20, 000, 30, 000, 40, 000 training samples on the Stanford dataset (d)-(f). In (a) and (d), the same training subset is used both to train the model and perform the nearest neighbor search. In (b) and (e), the training subset is used to train the model, whereas the nearest neighbor search is always performed on the whole training set. Tables (c) and (f) show the results of baselines obtained by considering a model pre-trained on ImageNet.

<table>
<thead>
<tr>
<th>Training and Search %</th>
<th>Position error (m)</th>
<th>Orientation Error (◦)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25%</td>
<td>1.133 ± 2.847</td>
<td>14.443 ± 22.621</td>
</tr>
<tr>
<td>50%</td>
<td>0.886 ± 2.286</td>
<td>11.961 ± 19.141</td>
</tr>
<tr>
<td>75%</td>
<td>0.785 ± 1.885</td>
<td>11.128 ± 18.595</td>
</tr>
<tr>
<td>100%</td>
<td>0.759 ± 2.019</td>
<td>10.362 ± 17.095</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training %</th>
<th>Search%</th>
<th>Position error (m)</th>
<th>Orientation Error (◦)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25%</td>
<td>100%</td>
<td>0.778 ± 2.139</td>
<td>10.362 ± 17.095</td>
</tr>
<tr>
<td>50%</td>
<td>100%</td>
<td>0.775 ± 2.119</td>
<td>10.316 ± 17.203</td>
</tr>
<tr>
<td>75%</td>
<td>100%</td>
<td>0.735 ± 1.789</td>
<td>10.377 ± 17.717</td>
</tr>
<tr>
<td>100%</td>
<td>100%</td>
<td>0.759 ± 2.019</td>
<td>10.362 ± 17.095</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Training samples</th>
<th>Search samples</th>
<th>Position error (m)</th>
<th>Orientation Error (◦)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10, 000</td>
<td>100, 000</td>
<td>0.321 ± 0.953</td>
<td>6.205 ± 13.199</td>
</tr>
<tr>
<td>20, 000</td>
<td>100, 000</td>
<td>0.315 ± 0.948</td>
<td>5.968 ± 12.408</td>
</tr>
<tr>
<td>30, 000</td>
<td>100, 000</td>
<td>0.318 ± 0.858</td>
<td>6.249 ± 12.645</td>
</tr>
<tr>
<td>40, 000</td>
<td>100, 000</td>
<td>0.344 ± 1.078</td>
<td>6.456 ± 13.044</td>
</tr>
</tbody>
</table>

Table 3. Percentages of correctly localized test images for different pairs of tolerance thresholds for the position and orientation error.

<table>
<thead>
<tr>
<th>Position Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.25m</td>
</tr>
<tr>
<td>Orientation error</td>
</tr>
<tr>
<td>30°</td>
</tr>
<tr>
<td>45°</td>
</tr>
</tbody>
</table>

Table 2(f) shows the results obtained using the baseline which considers a CNN pre-trained on ImageNet. Trends similar to the ones observed in the case of the Bellomo dataset apply to the results shown in Table 2(d)-(f), with the best results obtained when only part of the dataset is used to learn the feature space and the whole dataset is used for the nearest neighbor search. Also in this case, the advantage of using triplet to learn a suitable feature space is clear. Given the smaller scale of the Stanford environment, the localization method benefits more from triplet training, i.e., the mean position error of 1.341 m obtained using the ImageNet-pretrained baseline, drops down to 0.313 m after training the model using the triplet network.

Localization Accuracy. To evaluate the localization results, we evaluate the percentage of correctly localized test images when a set of tolerance thresholds is considered. Specifically, a test image is deemed to be correctly localized when the relative localization error is below the pre-specified set of position and orientation tolerance thresholds. Table 3 reports the results for different position error thresholds: [0.25m, 0.5m, 0.75m, 1m]; while for the orientation error, we consider the thresholds: [15°, 30°, 45°]. In the case of the Bellomo dataset, a high accuracy of 88.63% is achieved for tolerance thresholds of 1 m and 45°, which is reasonable for localization purposes in a cultural size. A similar accuracy is obtained for the Stanford dataset.
in the case of the Stanford dataset using smaller thresholds for the position error: 0.5 m and 45°.

**Temporal Smoothing.** Fig. 6 reports the average position errors obtained when applying the three temporal smoothing filters based on mean, median and trimmed mean on Bellomo (a) and Stanford (b). Note that temporal smoothing is applied to the methods obtaining the best performance in Table 2. The errors achieved with a window size equal to 0 denote results obtained when no temporal smoothing is used. It is worth noting that, on both datasets, temporal smoothing allows to improve over the baseline by significant margins. In both cases, the trimmed mean achieves the most robust results. In particular, we obtain an average position error of 0.599 m with a window size of 5 frames on the Bellomo dataset and an average position error of 0.327 m with a window size of 9 frames on the Stanford dataset.

**Generalization Results.** We investigate whether a model trained on an source environment can generalize to a target environment, given a small set of labeled samples from the target environment. To assess such ability, we fine-tuned the best model trained on Stanford using a subset of 25% training samples from Bellomo. Table 4(a) reports the results of this experiment. The fine-tuned model (last row of Table 4(a)) is compared to using the model trained on Stanford without any fine-tuning (first row) and to a baseline obtained fine-tuning the model from ImageNet pre-training weights (middle row). The results show that fine-tuning the model trained on Stanford allows to obtain a lower error (0.987 m and 11.701°) than fine-tuning from ImageNet (1.133 m and 14.443°), which suggests that generalization to a new environment can be obtained to some extent. Similarly, we fine-tuned the best model trained on Bellomo using 10,000 samples from Stanford. Table 4(b) reports the results. Similar considerations apply to this case as well, albeit with a smaller advantage (0.619m vs 0.691m and 11.137° vs 11.606°), which is probably due to the smaller scale of Stanford.

**Artwork Detection Results.** In Table 5, are reported the mean Average Precision (mAP) results for each of the considered object detectors. Faster R-CNN achieves good performance thanks to the large amount of labeled frames generated using the proposed tool. Mask R-CNN obtains a slight improvement thanks to the inclusion of segmentation masks, whereas RetinaNet achieves a smaller mAP due to its single-shot nature, which also makes it a faster detector.

Table 6 finally reports the accuracy obtained when predicting the observed artworks by using image-based localization approaches on the Bellomo dataset. Coherently with localization results, the best performance is obtained when using 75% of the training set to learn the feature space, in which case we achieve an accuracy of 0.739. This result suggest that an accurate localization can give insights on what the visitor is paying attention to in a museum.

7. **Conclusions**

We proposed a tool to generate and automatically label egocentric visual data starting from a 3D scan of real cultural site. The proposed tool can be useful to produce arbitrarily sized datasets to study image-based localization and artwork detection, which are two fundamental problems in visit augmentation and cultural site management. To assess the usefulness of the tool, we generated and released a dataset from a 3D scan of a real cultural site. The generated data is used to benchmark the tasks of image-based localization and artwork detection. Results highlight that the proposed pipeline can be used to study vision problems related to cultural heritage. In order to replicate the proposed pipeline in a new environment, the following steps have to be performed: 1) scan the target environment; 2) import the 3D model in the tool and label it for navigation and artwork detection; 3) prepare the training samples and generate

---

9For qualitative results, please refer to the supplementary material.
the triplets for the training of the triplet network; 4) train the triplet network; 5) extract features from each training image using the trained network; 6) localize new images performing a nearest neighbor search in the learned representation space.

Future works will investigate the use of synthetic data to improve the performance of algorithms dealing with real data.

Acknowledgments

This research is part of the project VALUE - Visual Analysis for Localization and Understanding of Environments (N. 08CT6209090207 - CUP G69J18001060007) supported by PO FESR 2014/2020 - Azione 1.1.5 - “Sostegno all'avanzamento tecnologico delle imprese attraverso il finanziamento di linee pilota e azioni di validazione precoce dei prodotti e di dimostrazioni su larga scala” del PO FESR Sicilia 2014/2020, and Piano della Ricerca 2016-2018 linea di Intervento 2 di DMI, University of Catania. The authors would like to thank Regione Siciliana Assessorato dei Beni Culturali dell’Identità Siciliana - Dipartimento dei Beni Culturali e dell’Identità Siciliana and Polo regionale di Siracusa per i siti culturali - Galleria Regionale di Palazzo Bellomo.

References


