2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

RISE: A Novel Indoor Visual Place Recogniser

Carlos Sánchez-Belenguer[1], Erik Wolfart[1] and Vítor Sequeira[1]

Abstract— **This paper presents a new technique to solve the Indoor Visual Place Recognition problem from the Deep Learning perspective. It consists on an image retrieval approach supported by a novel image similarity metric. Our work uses a 3D laser sensor mounted on a backpack with a calibrated spherical camera i) to generate the data for training the deep neural network and ii) to build a database of geo-referenced images for an environment. The data collection stage is fully automatic and requires no user intervention for labelling. Thanks to the 3D laser measurements and the spherical panoramas, we can efficiently survey large indoor areas in a very short time. The underlying 3D data associated to the map allows us to define the similarity between two training images as the geometric overlap between the observed pixels. We exploit this similarity metric to effectively train a CNN that maps images into compact embeddings. The goal of the training is to ensure that the L2 distance between the embeddings associated to two images is small when they are observing the same place and large when they are observing different places. After the training, similarities between a query image and the geo-referenced images in the database are efficiently retrieved by performing a nearest neighbour search in the embeddings space.**

I. INTRODUCTION

Recognizing places from visual information is a well known problem that has been present in the literature for a long time. In the last decade, visual place recognition has gained increasing attention due to the vast amount of geo-localized image datasets [1][2][3][4], the increase of portable acquisition devices (i.e. mobile phones) and the limitations of GPS localization systems in indoors and cluttered urban environments.

Visual place recognition is a challenging problem due to three major facts: (1) the appearance of a place can change drastically over time, (2) places are not always observed from the same viewing point and (3) cameras are strongly affected by light conditions. It can be used for a wide range of applications like loop detection in SLAM approaches [5] or autonomous navigation [6].

Traditional approaches facing this topic relied either on *global* methods (e.g. [7]), which process the image as a whole using global descriptors, like GIST [8], or on *local* methods that extract selectively parts of the image using local-invariant feature extractors (such as SURF [9], or SIFT [10]) and match them using Bag-Of-Words [11] or voting schemes. Even though both methods differ in the way they approach the problem, they share a common feature: descriptors used to characterize sets of pixels are always hand-crafted.

[1] Carlos Sánchez-Belenguer, Erik Wolfart and Vítor Sequeira with the European Commission, Joint Research Centre (JRC), Via Enrico Fermi 2749, Ispra (VA), Italy. [name.surname]@ec.europa.eu

In the last years Deep Learning has revolutionized many disciplines. Particularly, in the Computer Vision field, Convolutional Neural Networks (CNNs) have achieved state-of-the-art performance on several recognition and classification tasks. The key idea behind CNNs is their ability to automatically learn high-level global features, in contrast with hand-crafted ones like GIST. Similarly to the rest of machine learning approaches, CNN-based place recognisers require large sets of training data to perform properly. For this reason, researchers working in this field dedicate lots of time and resources to the data collection stage [12][13].

The first approach exploiting CNNs in visual place recognition problems was [14]. It combined the automatic learning of features together with other filters to effectively solve the localization problem. Subsequent works using CNNs for solving place recognition problems have mostly focused their efforts on urban outdoor environments [3][15][16][17]. The reason for that is probably related to the availability of training data: acquiring outdoor geo-referenced training images can be easily achieved with a consumer GPS-enabled camera. Additionally, the accuracy of the ground-truth pose estimation plays a critical role in narrow environments (e.g. $0.5\,\mathrm{m}$ error in the camera pose when facing a building can be negligible, whilst in an indoor environment it can mean a different room). There are plenty of public datasets available on-line with urban outdoors imagery [18][19][20][21] or natural landscapes [4][22] and most of the papers train and compare themselves using them.

However, when it comes to the visual indoor localization problem, both the number of papers and public datasets drops significantly. The most common dataset used to train and evaluate performance of indoor localization approaches is the 7-Scenes [23], collected using an RGB-D Kinect sensor. The main problem of using such a low-range sensor is that the area covered during the acquisition is limited. Similarly, works like [24] or [25] rely on their own RGB-D acquisitions facing the same problem. Alternatively, approaches like [26] collect larger amounts of data but rely only on high-level annotations (e.g. building, sub-building).

In general terms, almost all CNN-based visual place recognisers face the problem in an *indirect* manner: having a database of geo-localized images and a query image, the goal of the system is to retrieve similar images together with their associated poses (image retrieval). Assuming that the query was resolved correctly, the camera has to be nearby the reported poses. Alternatively [27][28][29] face the problem in a *direct* manner: given a query image, authors present a CNN that is able to regress the pose of the camera in 6 degrees of freedom.

978-1-7281-7395-5/20/$31.00 ©2020 European Union

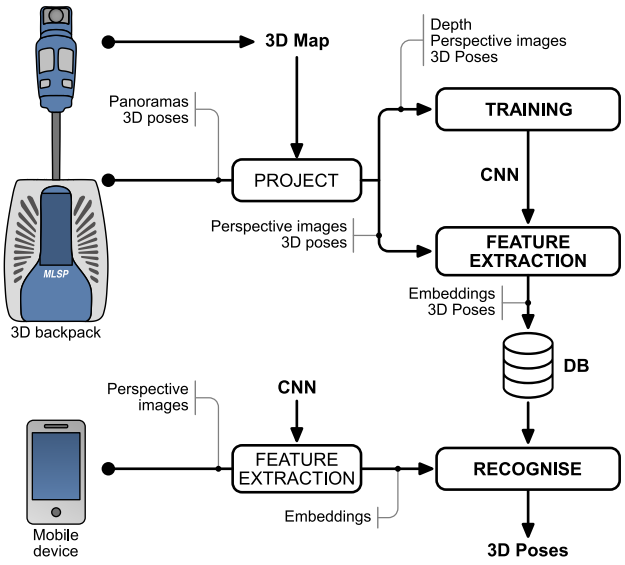

Fig. 1. Diagram of the proposed system

A. Overview

In this paper we cast the visual place recognition problem as a Deep Learning one. More specifically, we focus our efforts in the data collection stage and the training strategy, leaving the details of neural network architecture and fine tuning of parameters for future works.

Considering that the quality of the results provided by machine learning algorithms is closely related to the quantity and quality of the training data, we emphasize in this particular aspect. To do so, we present an indoor data collection pipeline where no user interaction is needed in order to retrieve and label training images.

Figure 1 shows our system. We use a laser sensor mounted on a backpack to generate a 3D reference map using SLAM techniques. Then, using the same backpack in tracking mode with a calibrated spherical camera on top, we survey the environment retrieving 360 panoramas with their associated poses. These data are used for two purposes: (1) to train a CNN that maps images into compact embeddings and (2) to build a database of ground truth images with their associated poses. In our database, images are not stored as bitmaps but as the embeddings extracted by our trained CNN. This way, when the user acquires a new image (with no 3D pose associated), its embeddings are extracted using the same CNN and queried to the database. The query is resolved by similarity in the embeddings space and the 3D poses associated to the most similar database images are returned.

This paper presents the foundations of a more ambitious research project that aims to solve the general indoor localization problem using only visual information (RISE: Robust Indoor Localisation in Complex Scenarios). In this sense, major contributions are: (1) an automatic acquisition pipeline that builds a 3D reference map and collects labelled images without the need of user interaction, (2) a loss function that allows defining the similarity between two images in a formal way.

II. METHOD

Localization is always performed w.r.t. an internal representation of the world: a *map*. This map provides a common reference frame for expressing the different poses of the sensor in time. For generality, we map the world in a 3D coordinate system, so poses are defined as 6 degrees of freedom variables. In order to create the map, we use a 3D laser sensor mounted on a backpack that, using SLAM techniques, generates a 3D point cloud of the environment.

The collection of training data is performed using the same backpack with a calibrated spherical camera. In this case, instead of using the laser sensor for mapping, we load the previously generated map and localize ourselves in real-time with centimetre accuracy. Once geo-localized panoramas are available, we use them to synthesize standard perspective images in order to perform the training.

The implementation of our place recognizer is inspired by FaceNet [30]. In the original work, a mapping function from face images to compact descriptors (embeddings) was learned. These compact representations could be compared using the Euclidian distance: two images from the same face were producing very similar embeddings (small distance) whilst two images from different faces had very different embeddings (large distance).

The main difference of our approach w.r.t. FaceNet is related to the loss function: when training a neural network to recognize faces, labels are discrete (two faces are either from the same person or from a different one). However, when it comes to places, two images can be partially observing the same space. This distinction makes necessary to formally define a similarity metric between images that allows evaluating how well the neural network is performing during training and to provide it meaningful feedback. Next sub-sections will cover these points in detail.

A. Data collection

Mapping and geo-localized data collection is performed using a backpack equipped with a 3D spinning laser sensor, an inertial measurement unit and a calibrated spherical camera on top. It can work in two different modes: (1) mapping, where a full SLAM pipeline is implemented [31] and a globally consistent 3D point cloud is produced and (2) tracking, where a reference map is loaded and an accurate real-time pose is provided in the map's reference frame [32].

Our backpack was originally developed for nuclear safeguards applications, e.g. design information verification and change monitoring, but it has been used in many other domains for the last two years. It was awarded with the first place in the 2015 Microsoft Indoor Localization Competition and used for refereeing in the 2016, 2017 and 2018 editions.

Once the 3D map of the environment has been produced, the same place gets re-visited several times with the backpack in tracking mode, using the camera to collect images of the same places in different conditions, e.g. lighting, distribution of furniture, people moving around etc.

The spherical camera produces 360° equi-rectangular images in Full HD resolution. To avoid precision loss when

geo-referencing them two factors have to be considered: (1) the camera has to be time-calibrated with the lidar's internal clock and (2) the extrinsic calibration between the camera and the lidar sensor, $\mathbf{\Gamma}_c^l$, has to be estimated so:

$$\mathbf{\Gamma}_c^m = \mathbf{\Gamma}_l^m \mathbf{\Gamma}_c^l$$

where $\mathbf{\Gamma}_c^m$ is the pose of the camera in the map's reference frame and $\mathbf{\Gamma}_l^m$ is the pose reported by our system in the map's reference frame, located in the lidar's optical centre.

B. Image similarity

The final goal of our approach is to train a CNN that learns a mapping function between images of places and compact embeddings. Two images observing the same place have to produce similar embeddings, so the Euclidean distance between them is close to zero. Ideally, two images observing different places should produce different ones, so the Euclidean distance between both embeddings is large.

To train such a system, and given a pair of training images, we have to define a distance function that represents how similar they are. We express our similarity function in terms of geometric *overlap*. To do so, we proceed in three steps, illustrated in Figure 2:

1) Create a 3D map of the environment using the backpack in mapping mode (thick black L-shaped polygon in Figure 2) and voxelize the environment, assigning an identifier to each full voxel (squares in Figure 2).
2) For each spherical image that was acquired in tracking mode, synthesize multiple calibrated perspective images (I_1 and I_2 in Figure 2) and retrieve the set of full voxels that are visible from the reference map (red/green named squares, respectively).
3) Given two images, their overlap is the ratio between the number of common voxels (named dashed squares in Figure 2: $\{v_3 \dots v_7\}$) w.r.t. the total number of observed voxels (named squares in Figure 2: $\{v_1 \dots v_{10}\}$).

Given two images, I_1 and I_2, and their associated sets of visible voxels, $V^{(1)}$ and $V^{(2)}$ respectively, we define the overlap as:

$$\begin{aligned} V^{(1)} &= \left\{ v_1^{(1)}, v_2^{(1)} \dots v_n^{(1)} \right\} \\ V^{(2)} &= \left\{ v_1^{(2)}, v_2^{(2)} \dots v_m^{(2)} \right\} \\ V^{(1)} \cap V^{(2)} &= \{v_1, v_2 \dots v_p\} : v_i \in V^{(1)} \wedge v_i \in V^{(2)} \\ \text{overlap}\,(I_1, I_2) &= \frac{2p}{n+m} \end{aligned} \tag{1}$$

Synthesizing perspective images from spherical panoramas allows us to boost the coverage and the data collection rate of a survey: a single spherical view of the environment can produce plenty of perspective images just by using different projection parameters (yaw, pitch, roll and field of view). For each perspective image we synthesize, the colour and the depth components are efficiently calculated using the GPU.

The colour component calculation is a straight forward process: it re-maps the spherical coordinates of pixels in the equi-rectangular panorama into the image plane of a perspective projection. For the depth component calculation, we re-project the 3D map into the perspective image plane. If two 3D points fall in the same position, the closest one prevails. We can use the resulting depth map at a later step (together with the pose of the camera and the projection parameters) to compute the global 3D position of each pixel in the image and retrieve the voxel that contains it. Figure 3 shows an example of this process.

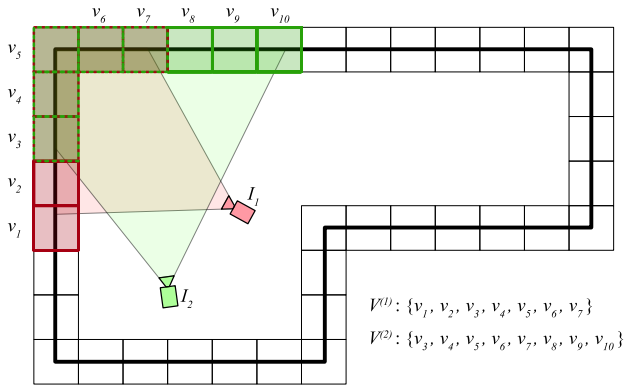

Fig. 2. Overlap between two images. Given a map (thick black line), its associated voxels (thin squares) and two geo-referenced images (I_1 and I_2), the visible voxels for each image are calculated (red/green squares). The overlap between I_1 and I_2 is the ratio of shared visible voxels (dashed squares) w.r.t. the total number of visible ones (named squares).

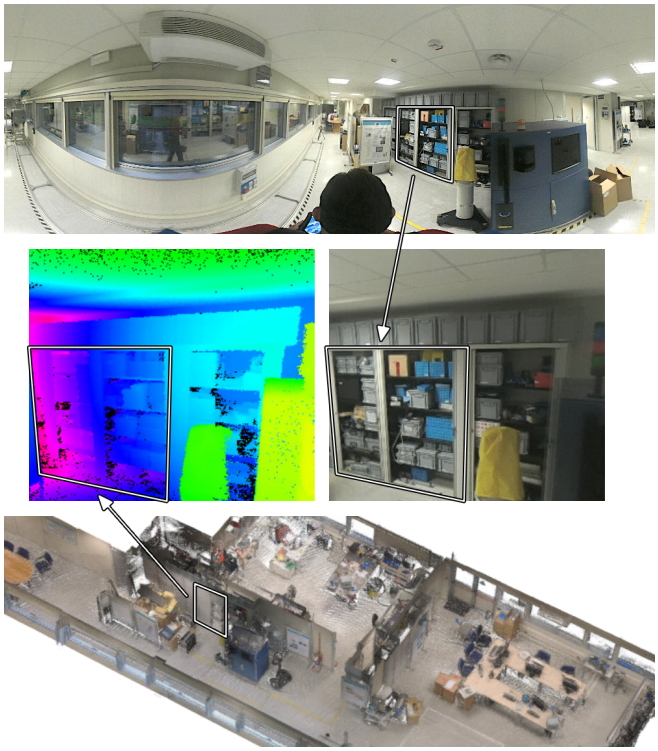

Fig. 3. Perspective image synthesis from equi-rectangular panoramas (top) and re-projection of the map (bottom). (middle-right) color component of a perspective image extracted from the panorama. (middle-left) depth component of each pixel calculated by re-projecting the map into the image plane (higher hue values represent more distant points).

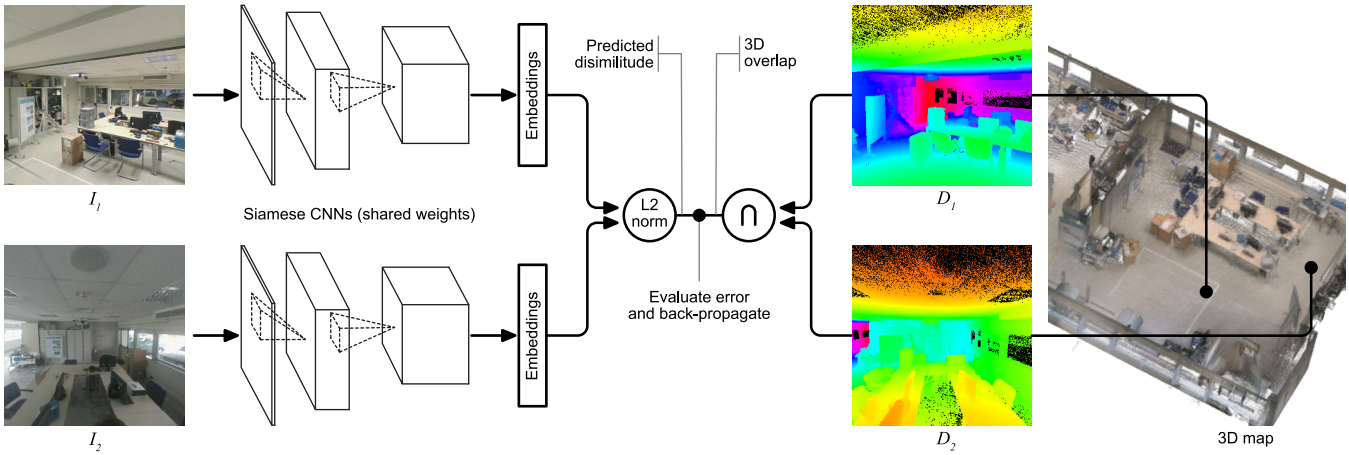

Fig. 4. Training overview. Given two images (I_1 and I_2 on the left), a 3D map (right), the 6 DoF poses of the images and the projection parameters, we compute the depth of each pixel by re-projecting the map into both image planes (D_1 and D_2, respectively). Then, we compute the ratio of overlapping points (labeled as *3D overlap*). The goal of the training stage is to optimize the weights of a CNN (with two instances in a siamese configuration) that maps each image into a compact vector (embedding). The L2 norm of the two embeddings (labeled as *predicted disimilitude*) has to be equal to $(1 - \text{overlap})$.

C. Training and place recogniser

We propose a deep CNN that maps images into compact embeddings. Given two RGB images, I_1 and I_2, and their associated embeddings, e_1 and e_2, respectively, we define the dissimilitude between the two images, $d(I_1, I_2)$, as:

$$d(I_1, I_2) = \|e_1 - e_2\| \tag{2}$$

During the training stage we aim to learn the weights that minimize the following error function:

$$E = [d(I_1, I_2) - (1 - \text{overlap}\,(I_1, I_2))]^2 \tag{3}$$

where $\text{overlap}\,(I_1, I_2)$ is calculated according to (1).

To train the CNN we proceed similarly to [30] and as illustrated in Figure 4: we define two instances of the same CNN in a siamese configuration (forcing weights to be equal in both instances). Then, we feed the system with two different training images and compute the estimated dissimilarity between their embeddings using (2) (left side of Figure 4). Simultaneously, we compute the ground truth overlap between both images using (1) (right side of Figure 4). Finally, we evaluate the error function using (3) and back-propagate to optimize the weights of both instances of the CNN simultaneously.

Once the CNN is trained, to use it for place recognition purposes, we populate a database with the set of geo-referenced images. To do so, instead of storing all pixel intensities, we feed the images into the trained CNN and compute their associated embeddings. Each entry of the database consists on a pair of values $\langle e_i, \mathbf{\Gamma}_i \rangle$, where e_i is the embedding and $\mathbf{\Gamma}_i$ is the associated pose.

During the localisation phase, as the camera moves inside the environment, new images are acquired with no pose information. For each of them the corresponding embedding is calculated using the pre-trained CNN. To retrieve similar images from the database (e.g. observing the same place) and their associated poses in the map a nearest neighbour radius search is performed in the high-dimensional embedding space. This is efficiently resolved by using a kd-tree pre-allocated with all the embeddings present in the database.

III. RESULTS

To validate our approach in a realistic way, we selected the JRC Visitors Centre as a testbed. It has more than $1000\,\text{m}^2$ with a large exhibition area spread over multiple spaces together with a restaurant area and a meeting room.

Our experiments target two specific aspects of our approach: on the one side, we validate our data acquisition and processing pipeline comparing it with other approaches and existing databases. On the other side, we asses the performance of our training procedure (relying on our similarity metric) and compare it against [24].

A. Data collection

Figure 5 shows the reference map generated with the backpack. The acquisition took only 15:13 minutes in which $926\,\text{m}$ were walked (orange line in Figure 5-top). Along the trajectory, 13,695 panoramas were acquired at full HD resolution (1920×1080 pixels per image).

To evaluate the camera calibration (both in time and extrinsically), we projected the panoramas over the point cloud. Figure 5-bottom shows the result of this test. Notice how final colors match accurately their corresponding 3D points: paintings in the walls are clearly delimited and furniture does not create artifacts in the edges due to calibration issues. Additionally, Figure 4 and Figure 3 show a different map coloured in the same way, with no visible artifacts.

Once a reference map was available, we performed 4 additional acquisitions in order to collect training data (sequences 1, 2, and 3) and evaluation data (sequence 4). We spaced those acquisitions over time and ensured to capture the environment under different lightning conditions.

Perspective images, together with their associated depth component, were synthesized in the following manner: we

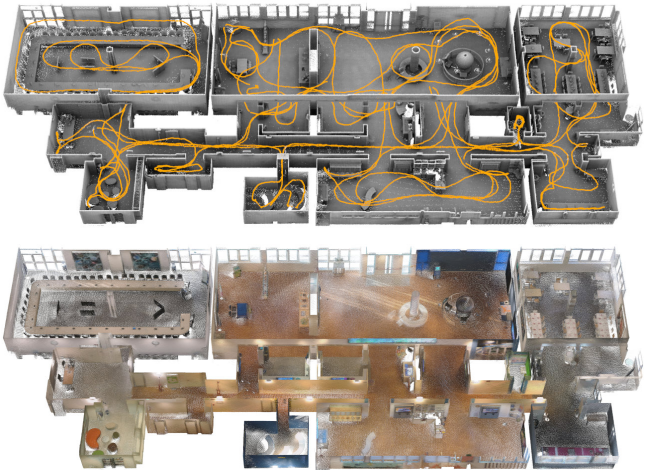

Fig. 5. Map generation and camera calibration. (top) 3D map generated, together with the estimated trajectory for the camera (orange line). (bottom) projection of the camera panoramas over the 3D map to validate calibration.

divided each trajectory into segments of $50\,\mathrm{cm}$. For each segment we selected only the panorama corresponding to the most static pose of the sensor, in order to reduce image blurriness. Selected panoramas were then used to produce 64 perspective images each, setting randomly the projection parameters in the following ranges:

$$\theta_x \in [-5°, 5°] \quad \theta_y \in [-10°, 20°] \quad \theta_z \in [0°, 360°]$$
$$f \in [60°, 70°]$$

where f is the field of view of the perspective camera, roll, θ_x, and pitch, θ_y, angular ranges were small because we consider a camera placed coarsely vertical and yaw, θ_z, covers the full 360 degrees.

The following table provides an overview of the data collection results showing, for each sequence, the lightning conditions, the acquisition duration, trajectory length and number of perspective images synthesized for each sequence.

	Lightning	Duration (min)	Length (m)	Images (#)
Sequence 1	morning	6:06	310	36,479
Sequence 2	afternoon	6:30	383	44,415
Sequence 3	evening	12:00	668	77,567
Sequence 4	noon	4:13	294	33,983

It is important to note that, during the entire process, there was no need for user intervention (only carrying the backpack during the acquisitions). The data processing was performed automatically and took around two hours in total.

We can compare the performance of our data acquisition pipeline with [13] and [12]. In [13] authors collected training data by dividing very large buildings into a grid of $60\,\mathrm{cm} \times 60\,\mathrm{cm}$. Then, they manually collected images by taking 10 pictures with a phone inside the delimited area of each cell pointing in a fixed direction. Alternatively, in [12] authors built a database of images for training place recognizers named *Places Database*. To do so, after downloading enormous amounts of images from on-line search engines, manual annotations were produced using Amazon Mechanical Turk. This service provides human workforce on-demand for well-defined tasks. In the original paper, authors explain the strategies that were designed to assess the correction of annotations in order to ensure the final data quality.

From these approaches, two main differences w.r.t. ours have to be highlighted: (1) our system produced 192,444 geo-referenced images in a GPS-denied environment with less than 29 minutes of user interaction. (2) Given the proven robustness of our backpack when tracking the sensor pose and the formally defined overlap between images, the training data we generate ensures the absence of labeling errors without the need of human supervision.

If we compare our training dataset against 7-Scenes [23] two major differences arise: (1) the area we are able to cover is orders of magnitude larger and (2) the performance collecting images is considerably boosted. The first point is related to the laser sensor used: ours has a longer range ($100\,\mathrm{m}$ vs $10\,\mathrm{m}$) and a full $360°$ field of view, enabling the user to move freely inside large environments without compromising the SLAM/tracking processing pipelines. The second point is related to the camera: using a spherical one enables our pipeline to synthesize many perspective images from a single geo-referenced panorama.

B. Training

We start by pre-computing pairs of training images and their associated overlap, using a voxel size of $20\,\mathrm{cm}$ and sequences 1 to 3, leaving sequence 4 for evaluation purposes.

Since we want to ensure that training data is visually meaningful for localization purposes, selected images for the training should be descriptive enough (e.g. an image staring at a white wall does not describe the environment). To do so, we automatically selected a total of 7,922 *source* images from all three sequences ensuring that the average depth of the pixels is always above $3.5\,\mathrm{m}$. This way, only relatively general viewpoints passed the test. For each *source* image we created 250 pairs with the most overlapping images in the full set and another 250 pairs with no overlapping images. By doing so we pre-computed 3,961,000 training samples.

The CNN we used for our experiments is based on the VGG-16 architecture pre-trained with the imageNet database [33]. In our architecture, the last 4 layers (softmax + 3 fully connected) are replaced by a single 1x1x128 fully connected layer with L2 normalization (embeddings output layer). The training was performed in mini-batches of 16 samples, where each batch included 8 training pairs with high overlap and another 8 training pairs with no overlap.

We trained our system for 48 hours on a nVidia GeForce GTX 1070 GPU. After the process was completed, our system was predicting overlaps with an error below $1\,\%$ in a small evaluation set that was left out of the training data for feedback purposes. Then, we populated a database with the embeddings extracted from all three training sequences and used sequence 4 for evaluation (33,983 images).

Using so many images for the database ensures a very rich characterization of the environment: for each image in sequence 4 we computed the ground-truth overlap with the

images of the database. On average, for each query image there is a counterpart in the database with $90\,\%$ overlap or more, 23 images with $80\,\%$ overlap or more, 347 images with $60\,\%$ overlap or more and more than 1,500 images with $40\,\%$ overlap or more. Figure 6 illustrates overlap values.

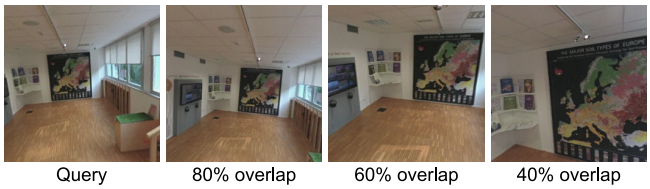

Fig. 6. Given a query image (left-most), example of various counterparts in the database with their associated ground-truth overlap.

After evaluating the complete sequence 4 to assess the behaviour of our trained CNN, we achieved an average error of $64\,\mathrm{cm}$ with the first image returned by the system. This error increases as we retrieve more candidates, but we visually observed that candidate poses tend to be distributed around the ground truth pose of the sensor. Figure 7 (orange) shows the average positional error of our system depending on how many images are retrieved.

We compared our approach against RelocNet [24], a state-of-the-art technique that improved previous approaches like PoseNet [27] and its subsequent geometric extensions [28][29]. RelocNet faces the place recognition problem from the same perspective of our paper (CNN based image retrieval approach), but the cost function that authors present differs from ours: they define the overlap as the intersection between the frusta of the two images being compared.

We find our approach more complete in the sense that we take into consideration the geometrical component. Also, our approach is parameter-free, whilst in RelocNet the maximum distance of the frustum has to be set in order to perform the intersection between two finite volumes.

In order to confirm this hypothesis, we set the frustum maximum distance to $8\,\mathrm{m}$ and repeated the same training changing only the cost function. As Figure 7 shows, the first result returned by RelocNet is, on average, $89\,\mathrm{cm}$ away from the real pose of the camera and, as we retrieve more results, the positional error increases much faster than the same system trained using our cost function. This is a clear consequence of the generalization problems of the CNN due to the presence of miss-labeled training samples.

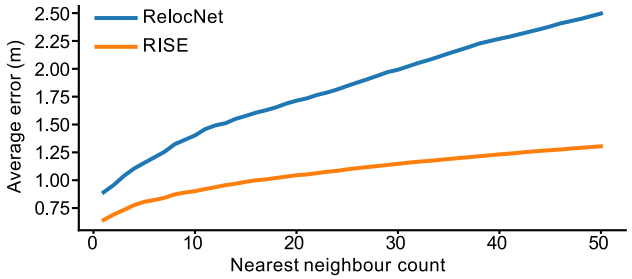

Fig. 7. Comparison between our approach (RISE) and RelocNet. Average positional error in the pose estimation w.r.t. the number of images retrieved.

Figure 8 shows the implications of the two cost functions using the same query image: in our case (top), since we consider geometry of the environment, all our training samples gather around the query image and in the same room. In the case of RelocNet (bottom), since they only consider the volume of the camera frustum, several wrong associations are created between images of two different rooms and labeled with high overlap values. Obviously, these wrong associations are expected to have an impact in the effectiveness of the training process, as Figure 7 shows.

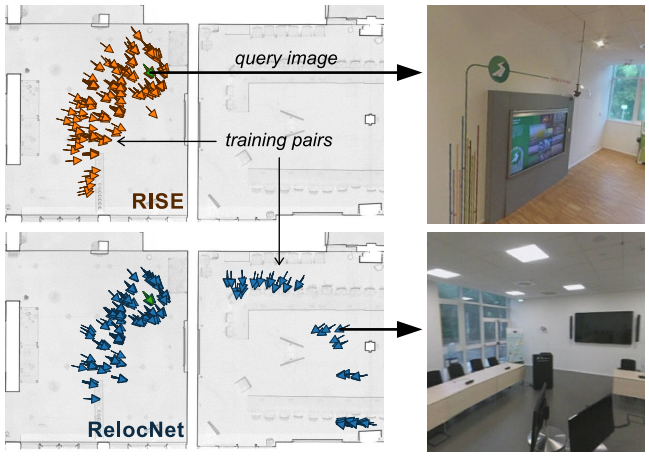

Fig. 8. Training pairs computed with the same query image using our cost function (top) and the one presented in RelocNet (bottom).

IV. CONCLUSION AND FUTURE WORKS

We have presented an effective pipeline that exploits 3D information to train an indoor visual place recogniser. Using a 3D laser sensor and a spherical camera, our system is able to automatically map an environment and retrieve large amounts of geo-referenced panoramas. We have shown that our system is capable of labelling data automatically and synthesize training images without the need of user intervention. Thanks to a novel image similarity metric that takes into consideration the geometry of the environment, we have presented a full training pipeline that covers the end-to-end learning of an indoor place recognition system. Using only images (i.e. no 3D data), our system can then localise the camera inside a large building with errors below $1\,\mathrm{m}$.

We have shown that, in comparison with other approaches, the data acquisition rate of our system is orders of magnitude faster. Also, given the robustness of our 3D tracking algorithm, the quality of the automatic labeling is ensured. Compared to current indoor datasets, our system has proven to be more scalable, allowing to acquire large indoor environments in a robust manner. Results have demonstrated the benefits of considering the geometric component in the cost function, improving the accuracy of state-of-the-art techniques.

Future works will aim to (1) refine the pose estimation by using, for example, classic computer vision approaches and (2) increase robustness of the place recogniser by disambiguating the pose of the camera over time using, for example, particle filters or recurrent neural networks.

REFERENCES

[1] M. T. Islam, C. Greenwell, R. Souvenir, and N. Jacobs, "Large-Scale Geo-Facial Image Analysis," *EURASIP Journal on Image and Video Processing (JIVP)*, vol. 2015, no. 1, p. 14, 2015.

[2] T. Weyand, I. Kostrikov, and J. Philbin, "Planet - photo geolocation with convolutional neural networks," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 37–55.

[3] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[4] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? challenging seqslam on a 3000 km journey across all four seasons," *Proc. of Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation (ICRA)*, p. 2013, 01 2013.

[5] X. Zhang, Y. Su, and X. Zhu, "Loop closure detection for visual slam systems using convolutional neural network," in *2017 23rd International Conference on Automation and Computing (ICAC)*, Sep. 2017, pp. 1–6.

[6] Y. N. Kim, D. W. Ko, and I. H. Suh, "Visual navigation using place recognition with visual line words," in *2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Nov 2014, pp. 676–676.

[7] P. Taddei, C. Sánchez, A. L. Rodríguez, S. Ceriani, and V. Sequeira, "Detecting ambiguity in localization problems using depth sensors," in *3DV*, 2014.

[8] A. Oliva and A. Torralba, "Building the gist of a scene: The role of global image features in recognition," *Progress in brain research*, vol. 155, pp. 23–36, 2006.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[10] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, IEEE International Conference on*, vol. 2. Los Alamitos, CA, USA: IEEE Computer Society, sep 1999, p. 1150. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/ICCV.1999.790410

[11] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *null*. IEEE, 2003, p. 1470.

[12] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 487–495. [Online]. Available: http://dl.acm.org/citation.cfm?id=2968826.2968881

[13] Z. Liu, L. Zhang, Q. Liu, Y. Yin, L. Cheng, and R. Zimmermann, "Fusion of magnetic and visual sensors for indoor localization: Infrastructure-free and more effective," *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 874–888, April 2017.

[14] Z. Chen, O. Lam, A. Jacobson, and M. Milford, "Convolutional neural network-based place recognition," *CoRR*, vol. abs/1411.1509, 2014. [Online]. Available: http://arxiv.org/abs/1411.1509

[15] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *International Journal of Computer Vision*, vol. 124, no. 2, pp. 237–254, 2017.

[16] H. J. Kim, E. Dunn, and J.-M. Frahm, "Learned contextual feature reweighting for image geo-localization," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 3251–3260.

[17] F. Radenović, G. Tolias, and O. Chum, "Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples," in *European conference on computer vision*. Springer, 2016, pp. 3–20.

[18] A. Babenko and V. S. Lempitsky, "Aggregating deep convolutional features for image retrieval," *CoRR*, vol. abs/1510.07493, 2015. [Online]. Available: http://arxiv.org/abs/1510.07493

[19] A. Torii, J. Sivic, M. Okutomi, and T. Pajdla, "Visual place recognition with repetitive structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 11, pp. 2346–2359, Nov 2015.

[20] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 257–271, Feb 2018.

[21] J.-L. Blanco, F.-A. Moreno, and J. González, "A collection of outdoor robotic datasets with centimeter-accuracy ground truth," *Autonomous Robots*, vol. 27, no. 4, pp. 327–351, November 2009. [Online]. Available: http://www.mrpt.org/Paper:Malaga_Dataset_2009

[22] J. Brejcha and M. Čadík, "Geopose3k: Mountain landscape dataset for camera pose estimation in outdoor environments," *Image and Vision Computing*, vol. 66, pp. 1 – 14, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885617300963

[23] B. Glocker, S. Izadi, J. Shotton, and A. Criminisi, "Real-time rgb-d camera relocalization," in *International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, October 2013. [Online]. Available: https://www.microsoft.com/en-us/research/publication/real-time-rgb-d-camera-relocalization/

[24] V. Balntas, S. Li, and V. A. Prisacariu, "Relocnet: Continuous metric learning relocalisation using neural nets," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[25] Y. Chen, R. Chen, M. Liu, A. Xiao, D. Wu, and S. Zhao, "Indoor visual positioning aided by cnn-based image retrieval: Training-free, 3d modeling-free," in *Sensors*, 2018.

[26] F. Zhang, F. Duarte, R. Ma, D. Milioris, H. Lin, and C. Ratti, "Indoor space recognition using deep convolutional neural network: A case study at mit campus," 10 2016.

[27] A. Kendall, M. Grimes, and R. Cipolla, "Posenet: A convolutional network for real-time 6-dof camera relocalization," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2938–2946.

[28] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," 07 2017, pp. 6555–6564.

[29] ——, "Modelling uncertainty in deep learning for camera relocalization," 05 2016, pp. 4762–4769.

[30] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, 2015.

[31] S. Ceriani, C. Sánchez, P. Taddei, E. Wolfart, and V. Sequeira, "Pose interpolation slam for large maps using moving 3d sensors," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, Sept 2015, pp. 750–757.

[32] C. Sánchez, P. Taddei, S. Ceriani, E. Wolfart, and V. Sequeira, "Localization and tracking in known large environments using portable real-time 3d sensors," *Comput. Vis. Image Underst.*, vol. 149, no. C, pp. 197–208, Aug. 2016.

[33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.