Will it last? Learning Stable Features for Long-Term Visual Localization

Marcin Dymczyk, Elena Stumm, Juan Nieto, Roland Siegwart, and Igor Gilitschenski
Autonomous Systems Lab, ETH Zurich

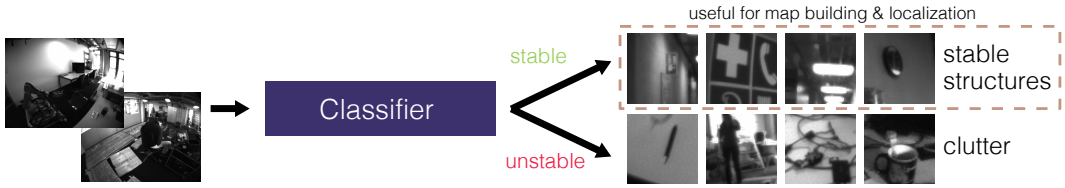


Figure 1: An overview of the proposed algorithm: A classifier is used to decide whether a particular visual feature is expected to be persistent or not. Our method uses full image information as input and helps to maintain compact stable-over-time maps that can be used for life-long localization.

Abstract

An increasing number of simultaneous localization and mapping (SLAM) systems are using appearance-based localization to improve the quality of pose estimates. However, with the growing time-spans and size of the areas we want to cover, appearance-based maps are often becoming too large to handle and are consisting of features that are not always reliable for localization purposes.

This paper presents a method for selecting map features that are persistent over time and thus suited for long-term localization. Our methodology relies on a CNN classifier based on image patches and depth maps for recognizing which features are suitable for life-long matchability. Thus, the classifier not only considers the appearance of a feature but also takes into account its expected lifetime. As a result, our feature selection approach produces more compact maps with a high fraction of temporally-stable features compared to the current state-of-the-art, while rejecting unstable features that typically harm localization. Our approach is validated on indoor and outdoor datasets, that span over a period of several months.

1. Introduction

Generating sparse 3D models from camera images is a key component of state-of-the-art simultaneous localization and mapping (SLAM) systems [3]. Usually, mapping relies on detecting image features, *i.e.* distinctive points in an image, tracking them, triangulating 3D landmarks and characterizing them using descriptors. Recent approaches tend to use such 3D models of a given area to perform visual localization and improve pose estimates [28] [26].

Localization systems rely on matching points from a query frame to the previously constructed 3D model [35]. As time passes and the map gets older, however, the matching becomes more and more difficult as the scenery is subject to appearance changes. In this paper, we consider life-long mapping scenarios, where the visual appearance can change due to lighting, seasonal, or visual and structural changes. Additionally, with the growing time-scale of the mapping scenario, the amount of data that needs to be stored, processed, or transferred becomes prohibitive. Figure 2 (a) presents a dataflow diagram of a mapping system where maps are transmitted from the frontend to the backend. We would like to reduce this stream of maps by means of online feature selection, as in Figure 2 (b).

Existing approaches to feature selection are not well suited for the life-long scenario we consider. Some approaches use the past data to find features which are reliably redetected [31], but ignore their actual semantic meaning. Other methods attempt to classify hand-crafted descriptors [9] to guarantee a high probability of matching. But even if a feature is well-recognizable and a good matching seems likely, it might be unreliable for life-long localization, *e.g.*, because it belongs to an object that is moved frequently or may even entirely disappear (such as a pen or a mug on a desk, see Figure 1).

This paper introduces a learning-based approach for the problem of life-long feature selection. Our method is based on a classifier of raw image patches and depth maps and tries to grasp not only the *matchability*, but also a semantic meaning of a particular 3D landmark. Specifically, our contributions are:

- We propose a classifier framework to select appearance-based map features for long-term localization purposes using raw image and depth information.

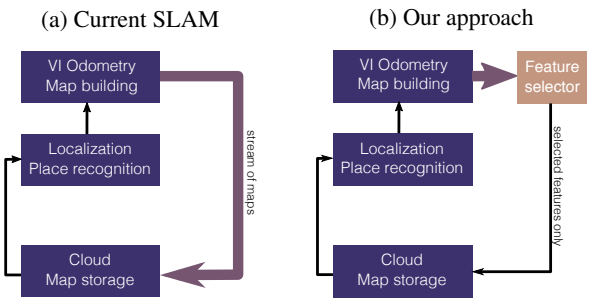


Figure 2: *Left:* The diagram presents the current flow in a typical mapping and localization system [28]. An agent streams mapping data (either raw image stream or visual-inertial odometry output) to the backend. The backend processes the maps and merges them into a single consistent global map which is then provided back to the agent(s) for localization. *Right:* The proposed data flow. Useful features can be selected on the agent's side which significantly reduces the amount of the data transmitted back to the backend.

- Resulting feature selection is shown to implicitly handle semantics, such as vegetation vs. buildings outdoors, or stable structures vs. clutter indoors.
- Maps obtained with our strategy allow for superior localization performance compared to state-of-the-art methods.
- We can significantly reduce the number of features in our map (*e.g.* by 80%) and still correctly register most of the query frames (75% of frames, about 20% better than state-of-the-art).
- We show that our method generalizes well, by deploying it in environments with significantly different appearances than the training environment and on a public long-term vision dataset.

The remainder of this paper is structured as follows. In Section 2, we discuss related work for appearance-based SLAM and feature selection approaches. In Section 3, we provide a problem definition. Our methodology is explained in Section 4 by describing the architecture of our classifier and discussing the general concept of the underlying ground truth training data. Setup of the experiments is discussed and their results are presented in Section 5. The work is concluded and discussed in Section 6.

2. Related Work

When extracting keypoints from images, SLAM systems typically rely on well-known algorithms for detecting distinctive points, such as the Harris corner detector [8] or a Difference of Gaussians (DoG) [24] based approach. These algorithms are crucial as they ensure that a potentially interesting point may be redetected in another image. One of the main goals of keypoint descriptors, such as SIFT [24], BRISK [21], or FREAK [30] is to obtain a description of the neighborhood of a keypoint that is viewpoint, scale and lighting invariant. Unfortunately, these algorithms do not consider suitability of a detected keypoint for use in long-term matching, and thus, too many partially irrelevant features are typically stored.

There are several approaches addressing this problem in current SLAM literature. In [2], it is proposed to store several maps of the same area covering different appearances. The advantage of this approach is that it contains very rich information about the environment. This comes at the cost of increased data quantity (including partially redundant information) and the need for introducing additional logic for handling different map variants [22]. A method for learning place-specific classifiers for identifying distinctive landmarks was proposed in [27] and subsequently improved in [23]. While these approaches improve localization quality between different appearance conditions, they require training location-specific classifiers and cannot be applied to previously unseen environments for reducing the amount of stored data at the time of initial recording.

Early consideration of life-long mapping can be found in [18], [19] where an incremental update scheme for an existing map is proposed. There, relocalization under different viewing conditions is achieved by storing redundant information and relying on particular types of map representation. Reducing map size by removing rarely observed features from the mapping backend was proposed in [31] and [5]. Even in a system that involves this type of map reduction, the present work serves as a useful additional component because it reduces the amount of data transferred to the backend and does not require multiple visits to the same environment for quantifying feature usefulness.

The work presented in [38] compares SIFT and SURF features, proposes the use of higher resolution images, and suggests introducing additional geometric constraints for localization. *Rosen et al.* [33] derive a stochastic filter that uses *Survival Analysis* [10] for modeling the lifetime of a feature. While that work is useful for map maintenance, it does not aim at providing an a priori feature quality criterion. In the work by *Hartmann et al.* [9], a random forest is trained to predict the matchability of hand-crafted SIFT descriptors. That is, the classifier predicts feature distinctiveness for better matching between consecutive frames but does not detect whether a feature (with a potentially high *matchability* score) might belong to a dynamic object or have an unstable appearance over time. In contrast, our approach captures these aspects as the training procedure learns directly on image patches and the groundtruth contains multiple visits to the same environment over an extended period of time.

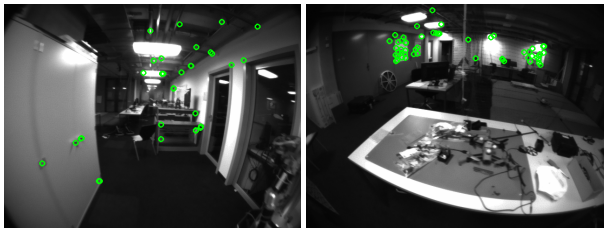


Figure 3: Two sample frames from our indoors dataset. The landmarks that were reliably redetected every time are marked using green circles. Those landmarks generally belong to constant elements of the environment, such as ceiling, door frames or heavy furniture. Our aim is to train a classifier that will be able to pick these useful features.

Furthermore, there are several approaches that assume existence of additional geo-location information for performing a feature quality assessment and a potential reduction of the number of features. Similarly to [9], the approach presented by *Kim et al.* [16] aims at detecting features that will have high matching score for obtaining good geo-localization. *Turcot and Lowe* [37] propose an approach that requires several different views of a scenery to be present in order to asses the quality of the features therein. *Knopp et al.* [17] propose the elimination of confusing features by counting the number of false matches within an image retrieval framework. The two latter approaches are thus not applicable to choosing stable features when there is no prior information about the current scenery.

3. Predicting feature reliability

Our goal is to obtain a subset of 3D landmarks of an appearance-based map that will provide as much information as possible for reliable localization and place recognition in a life-long scenario. The task of selecting those reliable, persistent localization features is not simple even for humans. We could claim that static objects in the environment can generally serve as useful landmarks, but it is not intuitively clear which of these will be redetected or matched correctly in the future, as illustrated in Figure 3.

The decision whether a feature is suitable for long timespan localization depends on three criteria:

- First, how well can this feature be matched in different images?
- Second, is this feature robust against changes in the environment?
- Third, will this feature remain in the environment for a longer period of time?

These criteria can be formulated as a classification problem in which detected features have to be classified as "stable" or "unstable" for life-long relocalization.

The classification task we have formulated above is extremely challenging to model due to the number of variables that influence changes in objects. Therefore we have opted for a data-driven approach by proposing a method that is based on recent advances in deep learning techniques. Particularly, convolutional neural networks (CNNs) have proven to be highly-successful in solving various computer vision challenges, such as object recognition [34] or video classification [15].

Our approach to select reliable localization features is based directly on the raw image information instead of hand-crafted detectors or descriptors that were previosuly used. This makes it possible to detect whether a feature belongs to a potentially dynamic object. Additionally, we do not have an off-the-shelf model and would like to learn what constitutes a stable landmark from the past experiences. We propose the use of a CNN-based classifier that uses a local keypoint neighborhood and optionally depth information as inputs and outputs a probability that the feature is "stable". We expect the depth maps to provide additional information about the 3D geometry of the scene. This might help to avoid *e.g.* occlusion corners, which yield unstable descriptors over viewpoint changes. Ground truth labeling of features in training data is based upon the redetection statistics that were collected over an extended period of time. This ensures potentially good matchable descriptors to be classified as unsuitable if the underlying feature belongs to an object that may move or disappear over time.

4. Methodology

The proposed CNN is trained using a set of labeled data pairs $\{image, label\}$ or triplets $\{image, depth\ \ map, label\}$. We consider the image patch around the keypoint as a main source of information, providing cues of how the keypoint looks like and what type of object it potentially belongs to.

The details of the entire process are presented in the subsections below, that cover the training set formulation 4.1, network architecture 4.2 and the training process 4.3.

4.1. Obtaining Training Datasets

Raw image patches: In this paper, we assume we are dealing with a visual-landmark-based mapping and localization system. In such a system, sparse point features are tracked over consecutive frames and used to triangulate 3D landmarks. We can therefore extract a patch with the local neighborhood of a feature and use it as an input to the network. The actual neighborhood size should depend on the camera resolution, focal length of the lens, feature scale and the environment properties.

Depth maps: The depth maps are extracted from the clusters of camera frames using block matching and planar [6] or polar [32] rectification, depending on the type

of motion. An additional bilateral filtering step [36] with the similarity function defined on the raw image is used to densify the depth maps. This method fills the gaps in the depth map while preserving the edges. Similarly as for image patches, a local neighborhood of the feature is extracted from the depth map.

Ground-truth labeling: We have decided to use binary labeling ("stable" and "unstable") of the features based on the empirical probability of reobserving them. The probability can be estimated from multiple co-registered datasets with the same trajectory, similarly to [13]. In general, we can say that the features that get merged across most of the datasets:

- are most likely stable objects as they are consistently re-detected over a long period of time,
- maintain a stable appearance under lighting and viewpoint changes as the datasets are recorded during different times of the day and features are observed from multiple angles,
- have beneficial descriptor pattern that gets easily and uniquely matched.

For each landmark, we compute a score

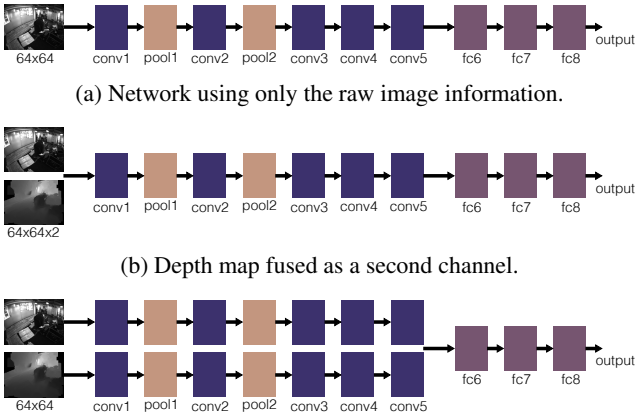
$$s = \frac{\text{number of datasets the landmark was observed in}}{\text{total number of datasets in the databse}}$$

and then label the landmark as "stable" if $s \geq 0.5$ (and as "unstable" otherwise).

4.2. Convolutional Neural Network Architecture

The proposed network architecture is based on the popular AlexNet network [20], which demonstrated very good classification performance while being computationally tractable on a desktop PC with a modern GPU (both in terms of training time and GPU memory requirements). There, we chose the last fully-connected layer to have 2 outputs, where each denotes a probability that the input belongs to either the "stable" or "unstable" class. This network is using a Softmax loss function, suitable for classification tasks. The additional depth information can be fused either as a second image channel or in a separate convolutional layer pipeline. The proposed network architectures are presented in Figure 4.

To classify a 3D landmark in the map that should be compressed, we calculate an output of the network for each of its observations (so for all the image patches observing this landmark) and then calculate a median probability of belonging to the "stable" class. If we need to select n best landmarks, we can simply use the "stable" class probability as the scoring function.



(a) Network using only the raw image information.

(b) Depth map fused as a second channel.

(c) Depth map with separate convolutional layers.

Figure 4: Three CNN configurations that were evaluated. In b) and c), the depth map information in included.

4.3. Training process

As our training set labeling might contain errors (e.g. a potentially "stable" landmark could have been classified as "unstable" one due to temporary occlusions) and the appearance-based landmark quality in general cannot be rigorously defined, we decided to use fine-tuning [4]. This technique suggests to pre-train the network on a well-labeled, large dataset (e.g. ImageNet) and only later, after reducing the network learning rate, try to fine-tune the network to the actual task. *Karayev et al.* [14] state that pre-training a network allows us to reuse the mid-level features learned from the object classification datasets and is generally superior to hand-tuned features.

While pre-trained models for ImageNet object classification are publicly available (e.g. Caffe's Model Zoo [12]), there seems to be no existing pre-trained model corresponding to our input parameters (small input image size, grayscale). Our approach is to train a network on the ImageNet dataset adapted to our needs, by scaling down the original ImageNet images and converting them to grayscale. The training performed on this small ImageNet dataset then provides us with a pre-trained network. In the final step, we fine-tune this network by learning on the labeled patch data from the mapping system.

5. Experiments

Our experimental setup and the datasets used for evaluations are presented in 5.1 and 5.2, respectively. In 5.3, we evaluate the network's output on the test set of "stable" and "unstable" labeled patches. We also verify the consistency of network predictions for patches of the same landmark (*i.e.* for multiple observations along a track of the same real-world feature). The second part of the evaluation is aimed at verifying the influence of our proposed method of selecting a subset of map features on the place recognition re-

sults (see 5.4). We compare our place recognition precision-recall results with the results of using other appearance-based methods for feature selection and map compression. Additionally, in Subsection 5.5, we measured the network's forward propagation time on a CPU and a GPU.

5.1. Experimental Setup

The **LAB** and **HALL** datasets were recorded with a visual-inertial sensor incorporating a low-cost IMU and a monocular, grayscale camera with a resolution of 720x480 pixels and a wide field-of-view. Visual-inertial odometry with a translational error in the range of 1% of the distance traveled (similar to [29]) was used to construct the maps. Keypoints were detected using a Difference of Gaussians (DoG) keypoint detector. These keypoints were characterized using FREAK binary descriptors [30] to co-register the maps. The keypoint locations were used to extract the raw image patches and depth map patches. Random cropping of patches (the network takes 64×64 patches cropped out of 68×68 keypoint patches) and mirroring them helped to augment the training set and improve the robustness towards viewpoint and orientation changes.

The Caffe [12] deep learning framework was used for CNN training and image patch classification. Both learning and classification was performed on a desktop PC with an Intel i7-920 (@ 2.66 GHz) CPU and an NVIDIA GeForce GTX980 GPU.

5.2. The Datasets

For evaluation we use 3 different sets of datasets with different scene characteristics, but all containing highly dynamic parts of the scene.

LAB datasets consist of 27 trajectories along the same path, of about 150m, in an indoor research lab environment with rather confined spaces (see Figure 3). The datasets were recorded over a period of 3 months. A total of 25 trajectories were used to construct the training set while 2 (recorded 2 months apart) were used for evaluation.

HALL datasets consist of 14 trajectories walking the same path, of about 100m, in a large university hall. The space is usually crowded, it also contains student working places with many dynamic objects. The datasets were recorded over a period of 2 weeks. A total of 12 trajectories were used for training and the remaining 2 (recorded 10 days apart) for evaluation.

We have also used the public University of Michigan **NCLT** datasets [1]. We have selected an area that was covered by 19 datasets in total (from $(x, y) = (-301, -448)$ to $(-183, -432)$ according to the ground-truth annotation), collected over 15 months. We have used only the front-facing camera of the Ladybug sensor. A total of 17 of the trajectories were used for training and the remaining 2 for evaluation (2012-08-20 and 2012-03-17).

The training datasets were optimized by applying a visual-inertial weighted least-squares optimization to each run individually, minimizing the reprojection errors and inertial residuals. Positions of the landmarks were included in the optimization procedure. Afterwards loop-closure was performed using a place recognition algorithm similar to [25][11]. As a result of this procedure, we obtained a map with co-registered trajectories. Based on the output of the loop-closure engine, we also partially merged features. The final map contains information on the number of runs and the number of frames per run in which a feature was visible.

The two evaluation datasets were used to verify the place recognition precision and recall (see Section 5.4). From the first dataset, a reference map was created and reduced (in terms of number of features) to a desired degree using the proposed method. The second dataset was used as a query dataset, *i.e.* we attempted to localize each of its keyframes against the reduced map dataset.

The localization evaluation requires accurate ground-truth poses so we needed to precisely align the pairs of datasets in a common global frame. We used our loop-closure system to find candidate 2D-3D matches between the two evaluation datasets. The matches were then filtered using a geometric-verification method (P3P pose solver [7]). Using the feature correspondences, the maps were co-registered. To further refine the map quality, a joint visual-inertial weighted least squared optimization was performed.

5.3. Evaluating Network Output

As a first step of evaluating our method, we have verified the binary classification accuracy of the proposed method and the three network architectures introduced in Section 4.2. The results are presented in Table 1. For both the LAB and NCLT datasets, depth information provides significant improvement over using solely the raw image patches. Additionally, fusing the depth map information in a fully-connected layer level yields the highest accuracy, which might be contributed to the fact that it is easier to optimize the convolutional filters separately. All the subsequent evaluations will assume this network architecture.

Table 1: Comparison of validation set binary classification accuracy for LAB and NCLT datasets and three network configurations: 1) using only a raw image patch 2) raw image patch stacked with depth as a two-channel image 3) raw image and depth propagated through convolutional layers separately and combined in fully-connected layers.

	LAB dataset	NCLT dataset
raw image patch	65.2%	69.5%
+ depth as a 2nd channel	67.4%	71.7%
+ depth in a FC-layer	71.2%	73.1%

We have also verified the relationship of the network predictions with the ground-truth labeling of the test set. The results are visualized in Figure 5. It shows the relationship between the ground truth labeling of our training set (given by the number of runs in which a landmark has been viewed) and the resulting classification scores on the evaluation set. All landmarks that have a score above 0.5 are classified as suitable for life-long localization. From this plot, it can be seen how, *on average*, image patches that contain "stable" landmarks (*i.e.*, landmarks visible in more than 12 runs) are classified as good and vice versa.

We have furthermore evaluated how consistent the network output is for multiple observations over different runs of a single real-world landmark. That is, we want to know how the scores of our classifier differ for different observations of the same feature. Ideally, all patches of a single landmark should yield scores which are close to each other. Figure 6 shows the distribution of zero-mean landmark scores for a map with 100,000 features. The results show that the output of the network is consistent for a single landmark, most often associating it with the same class.

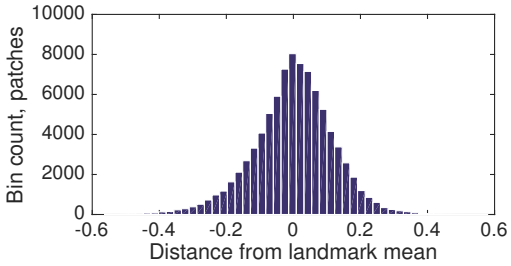


Figure 6: Distribution of classifier outputs after subtracting the feature-dependent mean from the classifier output. Most of the values are in the range of $[-0.2, 0.2]$ which suggests that our classifier yields reproducible results under different viewing conditions of a landmark.

A further evaluation provides a more intuitive understanding of the network output. Figure 7 shows an example of a frame with classified keypoints from the LAB dataset. Figure 8 presents the best and the worst features in NCLT and LAB datasets according to our proposed classifier. We can state that the network implicitly learns the semantic meaning of the patches, corresponding to which objects tend to be "stable" and "unstable". This matches our initial proposal to use the semantic information and also means that our training set actually conveyed the notion of stability.

5.4. Evaluating Localization Quality

In the previous section, we demonstrated that the network output is correlated with our ground-truth labeling and that the patch scores correspond to our expectations. This section evaluates how the localization quality depends on the selected subset of features and its size.

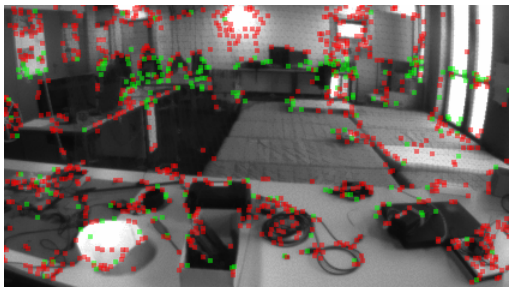


Figure 7: A single frame and a list of detected keypoints processed by our method. Features in the foreground belong mostly to tools and hardware pieces that were displaced almost every day and do not provide reliable place recognition clues. Features classified as "stable" are most often parts of stable objects (windows, furniture, poster on the left). It is worth noting that features belonging to the saturated region on the wall opposite (a reflection of a ceiling lamp) are also rejected.

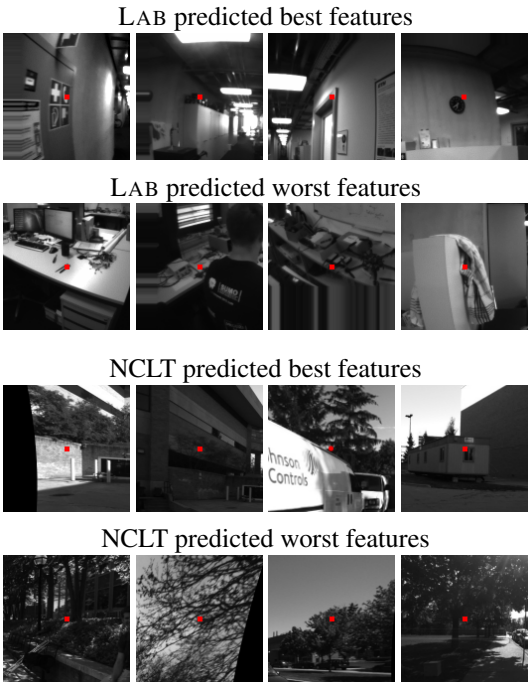


Figure 8: Image patches from the LAB and NCLT datasets that were classified as the best/worst by our method. The selection corresponds with human intuition: permanent structures with strong corners provide reliable localization cues while the clutter of dynamic objects (indoors) or vegetation (outdoors) should be rather rejected.

The key idea is to use one of the evaluation datasets (*map dataset*) for creating a reduced map using the proposed methodology. Then, we try to register each keyframe of the second evaluation set (*query dataset*) against the map dataset. We compare the resulting 6 DoF pose with ground-truth information. If the pose error is within the tolerance limits (0.07 m for position, 1.0 deg for rotation), then the localization is considered as successful. This is used to ver-

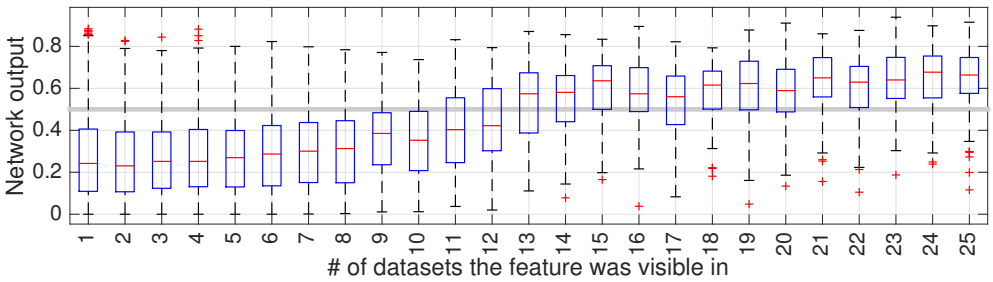


Figure 5: Illustration of the relationship between the statistics of the classifer output and the number of datasets a particular feature was observed in (LAB). There exists a clear positive correlation of the network output with number of datasets observing a landmark. The median output crosses the 0.5 level between 12 and 13 datasets which matches the labels used for learning.

ify how the results of the place recognition engine change when we reduce the number of features in the map (while keeping features with higher classifier score).

We compare the results of our method to three other approaches that assign a score to a keypoint only based on the visual information of its neighborhood. These methods are briefly described below.

Random selection: Landmarks are assigned a score that is obtained by random sampling from a uniform distribution in range $[0,\ 1]$. Random selection provides a very good baseline to evaluate feature selection methods. If a certain method performs better than random, then we can claim that the criteria it uses are indeed correlated with the phenomenon we try to model.

DoG score: Our mapping system uses Difference of Gaussians (DoG) as a keypoint detector. Therefore it makes sense to use DoG response as a very basic feature selection method. Intuitively, the more salient the feature, the more probable it is that we will redetect it in the next run over the same area. We need to consider the limitations of this method: it cannot grasp any life-long feature behavior or actual feature matchability.

Predicting Matchability score: We also used a method that was proposed in [9]. It is expected to select well-matchable and uniquely-matchable features. As already mentioned when discussing related work, this method is expected to perform worse in comparison to our approach as it does not consider long-term feature visibility.

The results of the comparison on the LAB dataset are visualized in Figure 9. In this visualization, we used F_1 score, a popular data retrieval metric. It combines the values of precision and recall by computing a harmonic mean of the two:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Our approach outperforms the comparison methods and still yields a good trade-off between precision and recall even after removing 80% of the worst (in terms of classifier score) features (F_1 score of 0.75 for our method, 0.55 for state-of-the-art [9]). A top-down view of the localization results is presented in Figure 10.

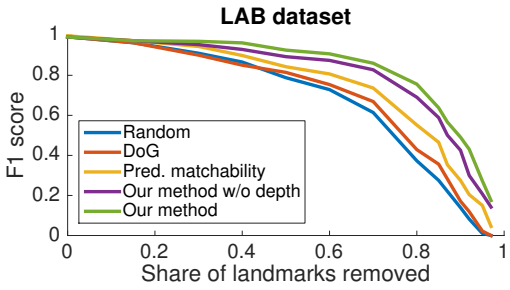


Figure 9: F_1 score curves comparing our proposed method with other baseline methods described in 5.4. The higher the F_1 score value for a given feature reduction number, the more keyframes were successfully localized (with fixed tolerance values). We can see that using the DoG response brings only a slight advantage over random feature selection, as it does not include descriptor uniqueness and long term feature stability. The plot also shows that our method performs best, especially at high map reduction values. We can reduce map by about 80% and still be able to register about 75% of query keyframes against it. Including the depth information brings a benefit over using the raw image only.

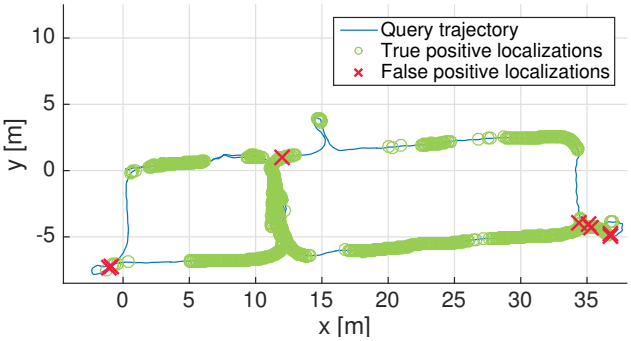


Figure 10: Top down plot of the LAB dataset presenting the localization results using a map reduced by the proposed method. The 3D features of the map were reduced by 80%, leaving about 35,000 features out of 180,000. The localization system returned 792 correct localizations and only 8 false positives.

So far, we have proven that we are able to learn the environment characteristics and then successfully predict the

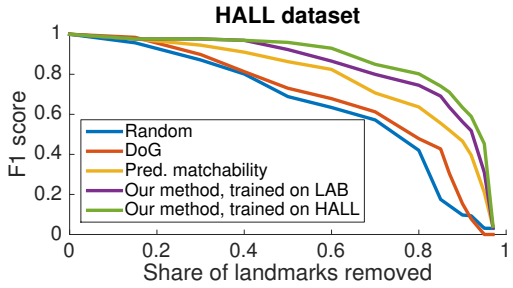


Figure 11: F_1 score curves comparing our proposed method with other baseline methods on an indoor HALL dataset. The results show that our approach generalizes well to other (but still indoor) environments – the classifier trained on the LAB dataset still works better than other state-of-the-art methods on the HALL data.

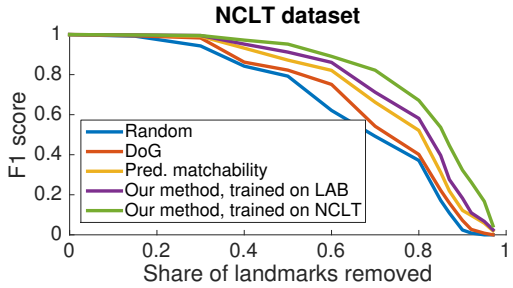


Figure 12: F_1 score curves comparing our proposed method with other baseline methods on an outdoor dataset from NCLT. Our method trained on the same dataset yields best results. Using the suggested classifier trained on the indoor LAB datasets still outperforms the other baseline methods.

stability and matchability of visual features for new maps of the area. While this result seems to be already useful, proving a certain level of generalization would be advantageous. We have therefore evaluated the performance of the suggested approach on the other datasets, HALL and NCLT. We have compared the classifier trained on the LAB dataset with the classifiers trained on the evaluated environments.

Figure 11 and Figure 12 present the localization F_1 scores of this evaluation. Even when trained in another environment, the proposed method still works better than the second-best Predicting Matchability [9]. In Figure 11, our approach shows only a small retrieval loss when trained on another indoor dataset, which means it can generalize with-ing similar environments. In Figure 12, one can notice that the gap between Predicting Matchability and our classifier trained on the LAB dataset is relatively small. This is in line with intuition as, due to the change of the environment from indoors to outdoors, our method can no longer predict long-term stable features as well as in the area it was trained on.

5.5. Runtime evaluation

The computation time for evaluating the quality of a single landmark patch is dependent on whether the network is evaluated on a CPU or a GPU. On our CPU, the mean duration of classification was 17.2 ms (with a standard deviation of 4.7 ms). When using a GPU this time goes down to 2.7 ms (with a standard deviation of 1.7 ms). Our results show that the approach is applicable for real-time feature classification at interactive frame rates.

6. Conclusion

This paper presents a method for assessing the suitability of features for life-long localization. Our proposed approach uses a CNN classifier that is trained on local image patches and depth images extracted from image features over multiple runs through the same environement, at different times, over a period of several months. Thus, the classifier evaluates the quality of a feature by considering both its distinctiveness and its long-term visibility, discarding seemingly good features that stem from non-persistent objects.

By using the proposed method, we were able to reduce the map size by over 70% while still being able to register around 80% of the relevant keyframes. Therefore, our classifier can significantly limit the map size, reducing the storage and data transfer requirements in typical mapping scenarios.

Our method provides better localization results than other feature selection methods, even for a completely new environment that was not present in the training set. An inspection of the classifier output suggests that our training data has captured a notion of feature stability, intuitively understood by object semantics. In future work, we find it would be interesting to exploit environment semantics in more detail, evaluate different network architectures or leverage the 3D context for better classification.

Acknowledgments

The research leading to these results has received funding from Google's project Tango.

References

[1] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice. University of michigan north campus long-term vision and lidar dataset. *The International Journal of Robotics Research*, page 0278364915614638, 2015. 5

[2] W. Churchill and P. Newman. Experience-based navigation for long-term localisation. *The International Journal of Robotics Research*, 32(14):1645–1661, sep 2013. 2

[3] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: real-time single camera SLAM. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–67, jun 2007. 1

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional acti-

vation feature for generic visual recognition. In *ICML*, pages 647–655, 2014. 4

[5] M. Dymczyk, S. Lynen, M. Bosse, and R. Siegwart. Keep it brief: Scalable creation of compressed localization maps. 2015. 2

[6] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000. 3

[7] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003. 5

[8] C. Harris and M. Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151, 1988. 2

[9] W. Hartmann, M. Havlena, and K. Schindler. Predicting Matchability. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 9–16. IEEE, jun 2014. 1, 2, 3, 7, 8

[10] J. G. Ibrahim, M.-H. Chen, and D. Sinha. *Bayesian Survival Analysis*. Springer, 2001. 2

[11] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European conference on computer vision*, pages 304–317. Springer, 2008. 5

[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 4, 5

[13] E. Johns and G.-Z. Yang. Generative methods for long-term place recognition in dynamic scenes. *International Journal of Computer Vision*, 106(3):297–314, 2014. 4

[14] S. Karayev, A. Hertzmann, H. Winnemoeller, A. Agarwala, and T. Darrell. Recognizing image style. *CoRR*, abs/1311.3715, 2013. 4

[15] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 3

[16] H. J. Kim, E. Dunn, and J.-M. Frahm. Predicting Good Features for Image Geo-Localization Using Per-Bundle VLAD. In *ICCV*, 2015. 3

[17] J. Knopp, J. Sivic, and T. Pajdla. Avoiding Confusing Features in Place Recognition. In *ECCV*, 2010. 3

[18] K. Konolige and J. Bowman. Towards lifelong visual maps. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1156–1163. IEEE, oct 2009. 2

[19] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based Maps. *The International Journal of Robotics Research*, 29(8):941–957, jul 2010. 2

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. 4

[21] S. Leutenegger, M. Chli, and R. Y. Siegwart. BRISK: Binary Robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555. IEEE, nov 2011. 2

[22] C. Linegar, W. Churchill, and P. Newman. Work Smart, Not Hard: Recalling Relevant Experiences for Vast-Scale but Time-Constrained Localisation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, USA, 2015. 2

[23] C. Linegar, W. Churchill, and P. Newman. Made to Measure: Bespoke Landmarks for 24-Hour, All-Weather Localisation with a Camera. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 787–794. IEEE, may 2016. 2

[24] D. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1150–1157 vol.2. IEEE, 1999. 2

[25] S. Lynen, M. Bosse, P. Furgale, and R. Siegwart. Placeless Place-Recognition. In *2nd International Conference on 3D Vision*, volume 1, pages 303–310. IEEE, dec 2014. 5

[26] S. Lynen, T. Sattler, M. Bosse, J. Hesch, M. Pollefeys, and R. Siegwart. Get out of my lab: Large-scale, real-time visual-inertial localization. In *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015. 1

[27] C. McManus, B. Upcroft, and P. Newman. Learning place-dependant features for long-term vision-based localisation. *Autonomous Robots*, 39(3):363–387, jul 2015. 2

[28] S. Middelberg, T. Sattler, O. Untzelmann, and L. Kobbelt. Scalable 6-dof localization on mobile devices. In *ECCV*, 2014. 1, 2

[29] A. I. Mourikis and S. I. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3565–3572. IEEE, 2007. 5

[30] R. Ortiz. Freak: Fast retina keypoint. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 510–517, Washington, DC, USA, 2012. IEEE Computer Society. 2, 5

[31] H. S. Park, Y. Wang, E. Nurvitadhi, J. C. Hoe, Y. Sheikh, and M. Chen. 3d point cloud reduction using mixed-integer quadratic programming. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 229–236. IEEE, 2013. 1, 2

[32] M. Pollefeys, R. Koch, and L. Van Gool. A simple and efficient rectification method for general motion. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 496–501. IEEE, 1999. 3

[33] D. M. Rosen, J. Mason, and J. J. Leonard. Towards lifelong feature-based mapping in semi-static environments. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1063–1070, may 2016. 2

[34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, pages 1–42, April 2015. 3

[35] T. Sattler, B. Leibe, and L. Kobbelt. Fast image-based localization using direct 2d-to-3d matching. In *2011 International Conference on Computer Vision*, pages 667–674. IEEE, 2011. 1

[36] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998. 4

[37] P. Turcot and D. G. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problems. In *2009 IEEE 12th International Conference on Computer Vision Workshops*. IEEE, 2009. 3

[38] C. Valgren and A. J. Lilienthal. SIFT, SURF & Seasons: Appearance-based Long-Term Localization in Outdoor Environments. *Robotics and Autonomous Systems*, 58(2):149–156, 2010. 2