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

Hierarchical Multi-Process Fusion for Visual Place Recognition

Stephen Hausler and Michael Milford

***Abstract*— Combining multiple complementary techniques together has long been regarded as a way to improve performance. In visual localization, multi-sensor fusion, multi-process fusion of a single sensing modality, and even combinations of different localization techniques have been shown to result in improved performance. However, merely fusing together different localization techniques does not account for the varying performance characteristics of different localization techniques. In this paper we present a novel, hierarchical localization system that explicitly benefits from three varying characteristics of localization techniques: the distribution of their localization hypotheses, their appearance- and viewpoint-invariant properties, and the resulting differences in where in an environment each system works well and fails. We show how two techniques deployed hierarchically work better than in parallel fusion, how combining two different techniques works better than two levels of a single technique, even when the single technique has superior individual performance, and develop two and three-tier hierarchical structures that progressively improve localization performance. Finally, we develop a stacked hierarchical framework where localization hypotheses from techniques with complementary characteristics are concatenated at each layer, significantly improving retention of the correct hypothesis through to the final localization stage. Using two challenging datasets, we show the proposed system outperforming state-of-the-art techniques.**

I. Introduction

A hierarchical approach to localization is a well-established process with roots in computational efficiency and provides a method of improving spatial accuracy of localization. Hierarchies have also been discovered in the mammalian brain, both in the structure of grid cells in the Hippocampus [1], and in the visual pathway of the Visual Cortex [2]. In this research we ask the question: does a hierarchical approach to visual localization provide a direct improvement to the localization success rate when different image processing methods are used, and how should such a hierarchical approach be structured? To answer this question, we perform an extensive investigation into combining different combinations of local, global and deep learnt image descriptors in a hierarchical localization pipeline. We showcase our findings using the datasets Nordland and Berlin Kurfurstendamm. The Nordland dataset tests our hierarchical fusion under severe appearance change but no viewpoint change, while Berlin verifies these results under severe viewpoint change.

SH was supported by an Australia Postgraduate Award. MM was partially supported by ARC grants FT140101229, CE140100016, AOARD grant FA2386-19-1-4079, QUT Centre for Robotics and the Australian Government via grant AUSMURIB000001 associated with ONR MURI grant N00014-19-1-2571. SH and MM are with the QUT Centre of Robotics, School of Electrical Engineering and Robotics, Queensland University of Technology (stephen.hausler@hdr.qut.edu.au).

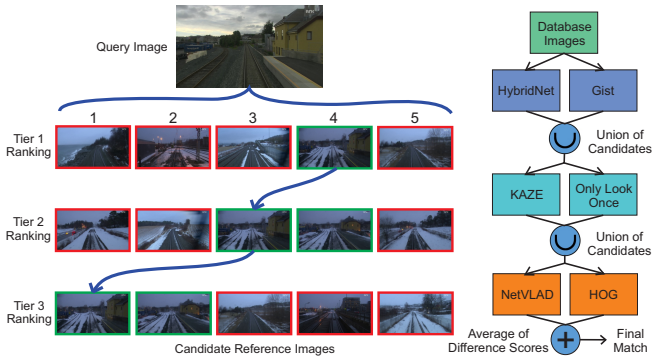

Fig. 1. Our approach selects best-matching candidate reference images for different tiers in a hierarchical approach. The top candidates in Tier 1 are passed to Tier 2, which finds a smaller number of top candidates to pass to Tier 3. The top ranked image in Tier 3 is the location we loop close to. As we move down the hierarchy, the ground-truth match is successfully shifted to the front of the ranked list, by virtue of the use of different but complementary image processing methods in each tier. The candidate ranking example shown is an experimental result from the Nordland dataset and a green border denotes a matching image that is within the ground-truth tolerance for our experiments.

In this paper we provide the following contributions:

- We show how merely fusing together multiple visual place recognition techniques in parallel is inferior to a hierarchical approach.
- We develop a hierarchical framework that can be configured to improve both computation speed and localization performance, and demonstrate its superior performance using both two and three tier architectures.
- We show how the individual performance of a place recognition method does not always directly predict its utility in a hierarchical system, and show how combining different techniques can result in superior performance compared to stacking a single, higher performing technique.
- We expand the system to enable concatenation of multiple place recognition techniques within a level of the hierarchy, leading to improved retention of the correct place recognition hypothesis that results in additional improvements in performance.

The source code for this paper is available online[1].

II. Related Work

In solutions to solve the data association problem in localization (perform loop closure), a hierarchical approach is commonly used [3]–[9]. In many of these approaches, a

[1]https://github.com/StephenHausler/Hierarchical-Multi-Process-Fusion

coarse global search is performed across all stored database image representations before a second, fine-grained search is used to filter the set of candidates produced by the global search [4]. Often the second search will utilise computationally-intensive geometric approaches, such as Bundle Adjustment [10] and Co-visibility Clustering [5].

In these hierarchies, a wide-variety of image descriptors have been used. These descriptors fall into the categories of global [11], [12], local [13], [14] and deep learnt [15], [16]. Commonly a different category of descriptor is used in different stages of a hierarchical approach. Maohai et. al. performed a two-stage localization hierarchy, using a color histogram global image descriptor to provide a coarse localization and then evaluated the resultant candidates using SIFT feature matching [8]. A more advanced version of this was developed which uses the PHOG descriptor at the first stage of the hierarchy, then uses FAST corners with LDB binary descriptors [6]. Their work also included RANSAC verification and a Bayes Filter to further improve localization. Prior work has also investigated the combination of deep-learnt and local features in a hierarchy, showing that accurate 6-DoF pose estimates can be produced at real-time if the set of candidates is first filtered using a deep-learnt global descriptor [5].

A single image descriptor can also be used in a hierarchy. One approach is to sub-analyze the images, for example, using patch-verification and Sum-of-Absolute Differences [17]. Alternatively, a sequence of images can be interpreted across multiple spatial scales, providing a hierarchical set of multi-scale clustered descriptors of the current scene [3].

While many of the aforementioned localization approaches have used one or two image processing methods in their hierarchy, none of those approaches use an arbitrarily large number of methods. Fusing a large number of image descriptors has had some related investigation, such as SRAL [18], which simultaneously used six different types of visual features. In the author's previous work, four different image processing methods were fused in a temporal sequence [19]. Neither of these two approaches utilised a hierarchical framework, while this work showcases using multiple image processing methods in a hierarchy.

III. Proposed Approach

In this work we present a sequence of investigations into hierarchical fusion of place recognition techniques which inform subsequent design of a novel, high performing hierarchical place recognition framework. In a typical hierarchical localization system, loop closure candidates from a first, computationally cheap localization method are used to define a set of potential matches for evaluation by a second, computationally expensive sub-system. This pipeline allows for efficient real-time localization, even in long-term navigation trials. In our experiments we use a three-tier hierarchy, however, our proposal is customizable and can be applied to any arbitrary number of tiers. Each tier uses a different image processing method to evaluate the similarity between the currently viewed scene and the provided candidates. Additionally, we include the option of adding additional image processing methods within a particular tier [20], such that the selected candidates from that tier becomes the union of the best candidates from the multiple methods within that tier. In this section, we will begin by describing the configuration of each image processing method we use in our experiments.

A. Design and Configuration of Image Processing Methods

For this work, we selected a total of six different image processing methods (half hand-crafted, half deep learnt). Our proposed approach is equally applicable to methods not chosen and any of the following methods can be replaced with an alternate approach.

Histogram of Oriented Gradients (HOG) - we use Dalal and Trigg's HOG [11] with a cell size of 30 by 30 pixels. We also re-size the input images to 300 by 300 pixels, which assists in alleviating small appearance variations while also reducing the size of the feature vector produced by HOG.

Gist - uses Gabor filters to extract gradients from an image, for a range of spatial scales and frequencies [12]. Using the default settings, Gist outputs a 512 dimension feature vector from an input image.

KAZE - is a local feature detector and descriptor similar to SURF [14] or SIFT [21], except it has demonstrated improved feature quality but is also computationally expensive [13]. We match features between two images using MATLAB's built-in *matchFeatures* function, and specify a match filter with a *MatchThreshold* of 20 and a *MaxRatio* of 0.7. By applying the filter, we remove incoherent matches which fail Lowe's ratio test [21]. The distance between a query and a database image is then the sum of the residual distances between the twenty strongest matching features. The database images with the smallest distance are considered the best matching candidates.

NetVLAD - is a neural network designed for place recognition, inspired by the success of VLAD [15]. We use the network pre-trained on Pittsburgh 30k and re-size our images to fit the input size of the network. We match images using the computed NetVLAD feature vector, which has a dimensionality of 4096.

HybridNet - is a re-trained version of AlexNet, trained on images recorded by a collection of security cameras over an extended period of time [22]. In our use of this network, we extract a feature vector from the Conv5 layer and use an aggressive method of dimensionality reduction. We compose the feature vector by aggregating the spatial (W x H) position of non-zero maximum activations across all the feature maps. As W and H are both dimension 13 in Conv5, this method produces a feature of dimension 169.

Only Look Once - in this method, later convolutional layers are used to find spatial regions with the strongest activations [23]. Multiple region descriptors are then created from the activations within each of these spatial regions in an earlier convolutional layer. To calculate the similarity between two images, these region descriptors are cross matched. This image processing method can match images

across both viewpoint and appearance variations, but it is computationally expensive. We use the open-source version of Only Look Once[2] to calculate the image similarity score.

B. Computation of Normalized Difference Scores

Each aforementioned image processing method produces a difference score between the current query image and either every database image or the set of candidates in the previous tier of the hierarchy. These raw difference scores have a wide variation in their data spread. Therefore we use min-max normalization to normalize all difference scores to the range of 0 to 1, where 1 denotes the best matching database image and 0 the worst.

$$D_{mn}(i) = \frac{D(i) - \max(D)}{\min(D) - \max(D)} \tag{1}$$

C. Fusion of Multiple Methods in a Hierarchy

The first tier of our hierarchy performs a global search across all database images, and returns k_{t1} nearest neighbour candidates with respect to the current query image. If multiple image processing methods are used in the first tier, then the returned candidates are the union of the nearest neighbours from each method in tier 1:

$$C_{t1} = C_{m1} \cup C_{m2} .. \cup C_{mn} \quad C \in \mathbb{R}^{k_{t1}} \tag{2}$$

where $m_1..mn$ are the methods from tier 1, up to n methods in this tier. C denotes the set of candidates, with a number of candidates up to k.

Candidates C_{t1} are then passed to the second tier of the hierarchy, to be evaluated by a more fine-grained search across this smaller set of 'potentially good' candidates. The image processing methods in tier 2 can and likely should be different to those in tier 1, with characteristics that enable the differentiation of perceptually aliased candidates. Because tier 2 only has to analyze a small number of candidates, rather than the entire database, the methods used can be more computationally intensive. k_{t2} nearest neighbour candidates are selected from this tier, comparing each candidate to the current query image, with the formulation described in Equations 3 and 4.

$$k_{t2} < k_{t1} \tag{3}$$

$$C_{t2} = C_{m1} \cup C_{m2} .. \cup C_{mn} \quad C \in \mathbb{R}^{k_{t2}} \tag{4}$$

Methods m in tier 2 are different to the n methods in tier 1 and the values of n can be different or the same between tiers.

At this point, further tiers can be added as needed, continuing to pass a shrinking pool of candidates. However, once the final tier of the hierarchy is reached, a best match consensus is determined. As the best match is a singular value, the union operator can no longer be applied between different image processing methods in the one layer. Instead, we calculate the mean normalized difference score across these multiple methods. The largest mean scoring candidate is then selected as the best match from the final tier of the hierarchy, as described by Equations 5 and 6.

$$D_{t3}(i) = \frac{D_{m1}(i) + D_{m2}(i) + ..D_{mn}(i)}{n} \quad i \in \mathbb{R}^{k_{t2}} \tag{5}$$

$$bestCand = \text{argmax}(D_{t3}) \tag{6}$$

With a selection of complementary image processing methods, accurate localization can generally be achieved at this final tier. However, an edge case can exist where the earlier layers successfully identify the global best match candidate while the later layer, with a different image processing method, is unable to identify the correct match. To guard against this condition, we provide an extension which fuses the difference scores from the best matching candidates in earlier layers to the final layer decision process.

D. Enhancing Localization using Earlier Layers

To further improve localization, the mean normalized difference from the final tier is added to difference scores from those same candidates in earlier tiers. Assuming a three tier hierarchy, the final tier will produce k_{t2} normalised difference scores for a list of C_{t2} candidates. The difference scores from earlier layers are then extracted for the final tier candidate set, resulting in k_{t2} scores per layer. To make scoring equivalent and hence combinable across tiers, we re-normalize the extracted difference scores to fall in the range of 0 to 1. Because earlier tier methods may be worse-performing comparing to later tier methods, we include the option of biasing the summation (making it less fair) using pre-calibrated weight scalars for each tier:

$$D_{final} = D_{t1}W_1 + D_{t2}W_2 + D_{t3}W_3 \tag{7}$$

where D_{t2} and D_{t1} are the normalized subset of all difference scores, as described by Equations 8-10. For all our experiments, we set the weight scalers to the values 0.5, 0.75 and 1 respectively. We heuristically selected these weight values by evaluating the recall rate of a variety of method combinations with different selections of weight values and chose the weight scalers that, by consensus, gave the best localization performance.

Because the candidates passed to later layers are the concatenation of candidates from each method in the same tier, we want to use the maximum difference scores to guarantee that only the best performing method in a layer is being used in the final calculation. We begin by finding the maximum difference score for each candidate id across the different methods:

$$D_{t2}(i) = \max_n(D_{m1}(i), D_{m2}(i), .., D_{mn}(i)) \tag{8}$$

$$D_{t2}(j) = D_{t2}(i \in C_{t2}) \quad i \in \mathbb{R}^{k_{t1}} \tag{9}$$

$$D_{t2}(j) = \frac{D_{t2}(j) - \min(D_{t2})}{\max(D_{t2}) - \min(D_{t2})} \quad j \in \mathbb{R}^{k_{t2}} \tag{10}$$

Equations 8-10 are repeated for D_{t1} and any other tiers prior to the last tier. The best matching candidate is the maximum score in D_{final}. In our results, we call this the *Multi-tier* recall.

[2]https://github.com/scutzetao/IROS2017_OnlyLookOnce

IV. Results

A. Dataset Configuration

We evaluate our proposal using the publicly available and widely used datasets Nordland [24] and Berlin Kurfurstendamm [25]. These two datasets capture a range of relevant place recognition challenges with significant appearance variation on Nordland and large viewpoint shifts on Berlin. We split each dataset into train and test sets and we use the training set to evaluate different combinations of image processing methods in our hierarchical approach.

The Nordland dataset consists of a 728 km train trip through Norway, across four different seasons. In our experiments we use the Summer and Winter seasons, where our database contains Winter images while our query set is from Summer [26]. To generate our training and test sets, we extracted frames at 1 FPS from the original videos, omitting sections where the train is either stopped or in a tunnel [16]. Our training set contains 1000 frames extracted from the start of the Nordland train trip. Our test set also contains 1000 frames, except these frames are taken from a later section of the train route. For all experimental results on Nordland we use a ground-truth tolerance of 10 frames.

The Berlin Kurfurstendamm dataset contains a collection of images downloaded from Mapillary [27], captured in the city of Berlin along the road Kurfurstendamm. For our training set, we use 280 images recorded from a bicycle as our query set. Our reference set contains 314 images captured by both a car and a bus driving on the same road. Our test set is similar, except the query images are recorded by a different bicycle, travelling on the same road on a different date with several years time gap between the two query sets. We use a ground-truth tolerance of 50 meters, since there is a large real-world distance between successive frames.

B. Evaluate Individual Methods on Train Set

To begin our experiments, we analyze the performance of each individual image processing method on the two training datasets (see Figures 2 and 3). We display the localization performance using the Recall @ N metric, where N is the number of top candidates. As more top candidates are used, the recall approaches one as the ground-truth matching candidate is more likely to exist in the larger set of candidates. We limit N to 50 for Berlin and 100 for Nordland, since the Berlin dataset is significantly smaller than Nordland. Different visual features exhibit substantially different Recall @ N characterizations. Additionally, some methods are more suited to a particular dataset. For example, HOG has outstanding performance on Nordland, where there is no viewpoint changes, but localizes poorly on Berlin (which has very large viewpoint shifts).

C. Combine Two Methods in a Hierarchy

In this experiment, we combine two different localization techniques in a hierarchy, where the first method passes the top 50 (Berlin) or 100 (Nordland) candidates to the second method. The second method only has to select the best candidate out of 50 or 100 potentially good candidates,

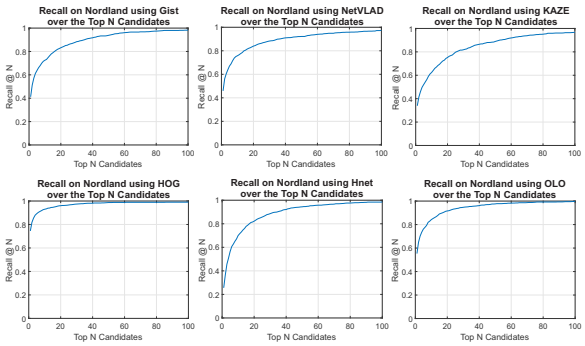

Fig. 2. Recall @ N curves for Gist, NetVLAD, KAZE, HOG, HybridNet and Only Look Once (OLO) on the Nordland Trainset. The recall @ 1 performance varies significantly between methods, however the recall @ 10 tends to be more consistent between methods.

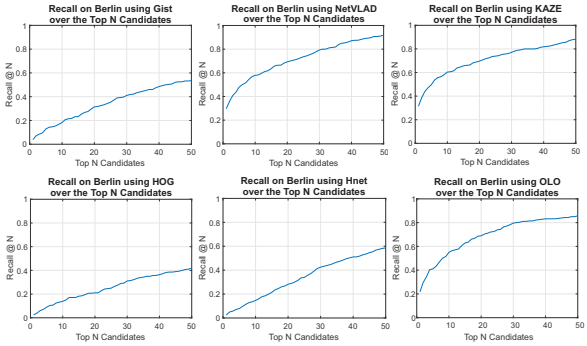

Fig. 3. Recall @ N curves for Gist, NetVLAD, KAZE, HOG, HybridNet and Only Look Once (OLO) on the Berlin Trainset. NetVLAD, KAZE and OLO perform consistently well compared to HOG, Gist and HybridNet, which cannot handle the large viewpoint changes.

rather than selecting the best candidate out of the full reference database. For the Nordland train set, we select two methods out of Gist, NetVLAD and KAZE and exhaustively evaluate all combinations of these methods (Figure 4). We chose Gist, NetVLAD and KAZE in order to combine a global descriptor, a deep-learnt approach, and a local feature detector. Additionally, these three methods exhibited similar Recall @ N characterizations, thus providing the fairest analysis of the real benefits of combining multiple methods in a hierarchy.

We use the methods NetVLAD, KAZE and OLO when evaluating the Berlin training set (Figure 5), because of the poor performance of the other three methods (as determined in Section IV. B.). We show our results using the recall at the best candidate for each method individually and for the final combination (using the algorithm described in Section III. D.). The reduced performance of Multi-tier in the rightmost two method combinations suggests that the arbitrarily assigned weight scalers need to be adjusted for these orders of methods.

D. Combine Three Methods in a Hierarchy

By adding a third tier to the hierarchy, we can pass a small number of potential best candidates to a method which can distinguish the best match from a small number

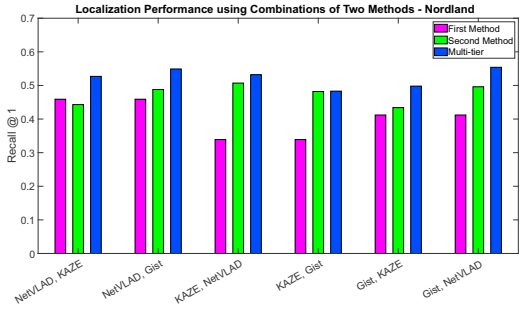

Fig. 4. For the Nordland training set, we combine different sets of two methods and show the recall at the top candidate. For all three methods, if the same method is used in the second tier of the hierarchy, the recall @ 1 improves. Any order of methods in the hierarchy improves the Multi-tier localization rate.

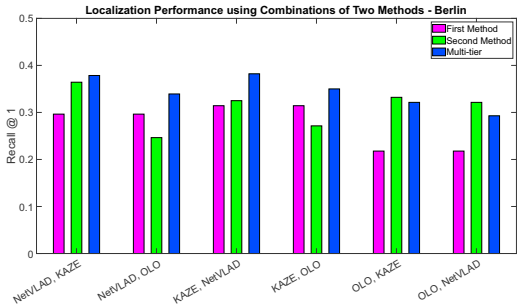

Fig. 5. For the Berlin dataset, again we combine different sets of two methods and show the recall at the top candidate. Only Look Once is the worst performing method and cannot compete with the combination of NetVLAD and KAZE.

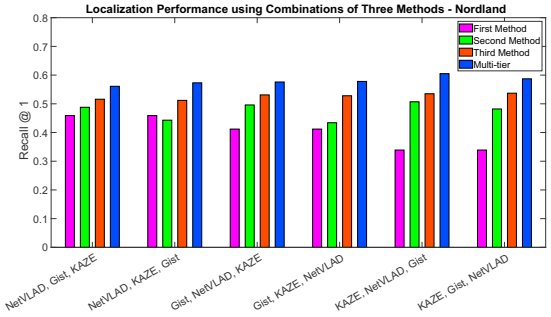

Fig. 6. We combine three different methods and show the recall at the top candidate on the Nordland train set. An increasing trend exists, where the recall @ 1 improves as the hierarchy is progressed and the multi-tier recall produces the best result.

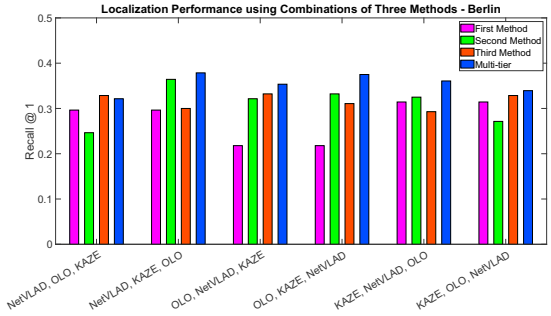

Fig. 7. For the Berlin training dataset, we combine three different methods and show the recall at the top candidate. Because Only Look Once is less suited to this training dataset, using the two methods NetVLAD and KAZE has the same Recall @ 1 as using three methods.

of perceptually aliased images. We again pass 50 or 100 candidates from the first tier to the second method, then we pass 10 candidates from the second tier to the final method. In Figure 6 for the Nordland dataset, the Recall @ 1 consistently improves as the number of candidates is reduced, irrespective of the order of the methods. The mean Recall @ 1 for the First, Second and Third Methods is 40.3%, 47.5% and 52.7% respectively, and 58.0% when using multiple tiers.

Figure 7 (Berlin trainset) reveals that the benefits of the hierarchical approach are not just because of inherent benefits of a shrinking candidate pool, but also the complementary interactions between different methods. For example, the Recall @ 1 for KAZE in the Second Method position is 36.4% after receiving candidates from NetVLAD, while the Recall @ 1 for KAZE after receiving candidates from Only Look Once is 33.2%. The mean Recall @ 1 for the First, Second and Third Methods is 27.6%, 31.0% and 31.6% respectively, and 35.5% for Multi-tier.

E. Combine Multiple Methods in a Single Tier

In this experiment, we maintain the same number of tiers except now we have two methods within each tier. Using two methods per tier increases the retention of the correct place recognition hypothesis if environment variations cause a particular method to perform poorly. We use the original four methods showcased in the previous experiments, plus the additional methods of HybridNet and HOG. Out of these six methods, we paired methods together based on both the length of the feature vector produced and the type of algorithm. For example, HybridNet (Hnet) and Gist have the smallest feature vectors (169 and 512 respectively), while NetVLAD and HOG use vectors of size 4096 and 2916. Only Look Once and KAZE both do not produce a feature vector and instead have unique image comparison algorithms.

In Table I, we detail the hierarchy of methods used in each experimental combination for this section. We also provide the computation time to run each order of the six methods on the Nordland dataset, since some methods are more computationally intensive than others. In Figures 8 and 9, the Recall @ 1 for each combination is displayed. We found that HOG was particularly suited to localizing on Nordland, irrespective of the tiered position of the method. This is why in Experiments 1 and 2 the Recall @ 1 is higher in Tier 1 than Tier 3. Nonetheless, moving HOG and NetVLAD to a later tier still provides a localization improvement from 71.3% in Tier 1 of Exp1 to 75.8% in Tier 3 of Exp4. Using six methods on Berlin provides an interesting failure case scenario: the extremely poor performance of HybridNet, Gist and HOG often causes the ground-truth matching candidate to be rejected in the early tiers, even when returning the top 50 candidates from each method in the first tier.

F. Study on Varying Number of Candidates Passed Between Tiers

We conclude our experiments on the training datasets by performing an investigation into a varying candidate count passed between hierarchies (Figure 10). In the left-most set

TABLE I

SIX METHOD COMBINATIONS FOR EXPERIMENT

Experiment Number	Methods Tier 1	Methods Tier 2	Methods Tier 3	Compute Time per Frame (s)
Exp1	NetVLAD, HOG	Hnet, Gist	KAZE, OLO	0.39
Exp2	NetVLAD, HOG	KAZE, OLO	Hnet, Gist	4.02
Exp3	Hnet, Gist	NetVLAD, HOG	KAZE, OLO	0.39
Exp4	Hnet, Gist	KAZE, OLO	NetVLAD, HOG	4.19
Exp5	KAZE, OLO	Hnet, Gist	NetVLAD, HOG	21.3
Exp6	KAZE, OLO	NetVLAD, HOG	Hnet, Gist	18.3

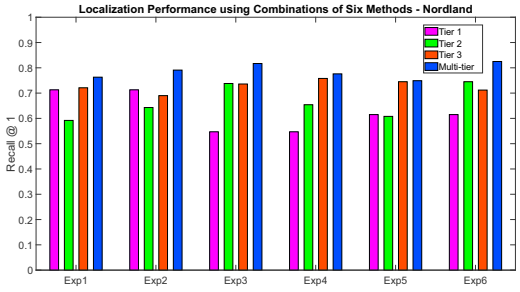

Fig. 8. Results for 6 methods on Nordland, where there are two methods per tier of the hierarchy. When 6 methods are used, because the first two tiers concatenate the candidates from each method, 200 and 20 candidates are passed from their respective tiers.

of bars, each subsequent method is given almost all the reference images. The number of passed candidates reduces towards the right. The peak recall @ 1 for the third method is at 50, 10 and 1 candidates, which demonstrates that hierarchical multi-process fusion is superior to simply fusing multiple methods in parallel. When difference scores from earlier tiers are considered, the utilization of a hierarchy (reducing the number of candidates per tier) is more important than the absolute candidate count.

G. Evaluate Optimal Method Set on Test Sets

We conclude our results by evaluating a set of selected methods on the two test datasets for Nordland and Berlin. We selected the set of methods with the highest Multi-tier Recall @ 1 on the training sets. Therefore we selected Exp6 as the set of methods to evaluate on the Nordland test set, and the three method combination of NetVLAD, KAZE and OLO for Berlin. While we could have chosen the two method combination of NetVLAD and KAZE, which had equally good performance, the use of an additional method improves the robustness to dataset challenges introduced in the test set.

In the Nordland test set (Figure 11), the multi-tier algorithm produces the highest Recall @ 1 of 77.2%. For the Berlin test set, the multi-tier algorithm had a Recall @ 1 of 41.0%. In both cases we improve upon the recall performance of the state-of-the-art algorithm NetVLAD.

V. DISCUSSION AND CONCLUSION

In this paper we investigated the combination of multiple different image processing methods in a hierarchical structure, for the visual place recognition task. From our insights, we contribute a novel and high performing hierarchical framework for the localization task. Our results show that the combination of complementary methods in a hierarchy

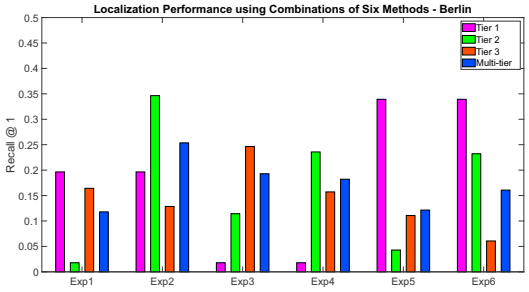

Fig. 9. Results for 6 methods on the Berlin train set. The Recall @ 1 is disjointed because of the failure of HOG, Hnet and Gist in this challenging environment.

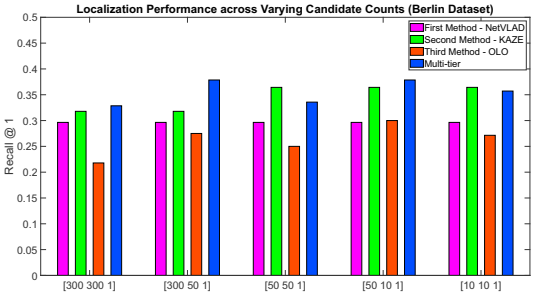

Fig. 10. Plot of Recall @ 1 across a varying candidate count, on the Berlin training set. We use the three methods NetVLAD, KAZE and OLO.

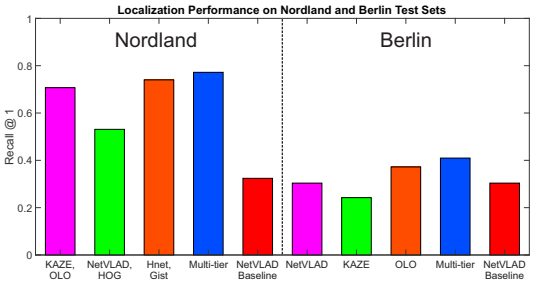

Fig. 11. In the five bars to the left, we evaluate our chosen set of six methods on the Nordland test set. In the right-hand bars, we evaluate the three best methods for Berlin using the test set.

improves localization beyond any individual method, and the hierarchy, rather than a flat parallel structure, is key to this improvement. This can be observed by comparing the Recall @ 1 for a method in the first tier versus the second or third tiers in the hierarchy. We hypothesize that our approach works because each image processing method has its own varying criteria for which images are perceptually aliased with respect to the query image. By combining multiple methods in a hierarchy, an early tier method can filter out candidate images which would appear perceptually aliased to a later tier method.

By using a calibration training set we can remove or re-weight methods which perform poorly in the current environment. An avenue of future work would be to add a source of weak ground-truth data or a second sensing modality, to decide on-the-fly whether a particular method needs to be omitted from the place match decision process.

REFERENCES

[1] H. Stensola, T. Stensola, T. Solstad, K. FrØland, M. B. Moser, and E. I. Moser, "The entorhinal grid map is discretized - Supplementary Information 2," *Nature*, vol. 492, no. 7427, pp. 72–78, 2012.

[2] N. Kruger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. Piater, A. J. Rodriguez-Sanchez, and L. Wiskott, "Deep Hierarchies in the Primate Visual Cortex: What Can We Learn for Computer Vision?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1847–1871, Aug 2013.

[3] Z. Chen, S. Lowry, A. Jacobson, M. E. Hasselmo, and M. Milford, "Bio-inspired homogeneous multi-scale place recognition," *Neural Networks*, vol. 72, pp. 48–61, 2015.

[4] S. Garg, N. Suenderhauf, and M. Milford, "LoST? Appearance-Invariant Place Recognition for Opposite Viewpoints using Visual Semantics," *Proceedings of Robotics: Science and Systems XIV*, 2018. [Online]. Available: http://arxiv.org/abs/1804.05526

[5] P.-E. Sarlin, F. Debraine, M. Dymczyk, R. Siegwart, and C. Cadena, "Leveraging Deep Visual Descriptors for Hierarchical Efficient Localization," Tech. Rep., 2018.

[6] E. Garcia-Fidalgo and A. Ortiz, "Hierarchical Place Recognition for Topological Mapping," *IEEE Transactions on Robotics*, 2017.

[7] H. Korrapati and Y. Mezouar, "Vision-based sparse topological mapping," *Robotics and Autonomous Systems*, vol. 62, no. 9, pp. 1259–1270, sep 2014.

[8] L. Maohai, S. Lining, H. Qingcheng, C. Zesu, and P. Songhao, "Robust Omnidirectional Vision based Mobile Robot Hierarchical Localization and Autonomous Navigation," *Information Technology Journal*, vol. 10, no. 1, pp. 29–39, Jan 2011.

[9] M. Mohan, D. Galvez-Lopez, C. Monteleoni, and G. Sibley, "Environment selection and hierarchical place recognition," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2015, pp. 5487–5494.

[10] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle Adjustment — A Modern Synthesis," 2000, pp. 298–372.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. I, 2005, pp. 886–893.

[12] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[13] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE Features." Springer, Berlin, Heidelberg, 2012, pp. 214–227.

[14] 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.

[15] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, 2018.

[16] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of ConvNet features for place recognition," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-Decem. IEEE, 2015, pp. 4297–4304.

[17] M. Milford, W. Scheirer, E. Vig, A. Glover, O. Baumann, J. Mattingley, and D. Cox, "Condition-invariant, Top-down visual place recognition," in *Proceedings - IEEE International Conference on Robotics and Automation*. IEEE, 2014, pp. 5571–5577.

[18] F. Han, X. Yang, Y. Deng, M. Rentschler, D. Yang, and H. Zhang, "SRAL: Shared Representative Appearance Learning for Long-Term Visual Place Recognition," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1172–1179, 2017.

[19] S. Hausler, A. Jacobson, and M. J. Milford, "Multi-Process Fusion: Visual Place Recognition Using Multiple Image Processing Methods," *IEEE Robotics and Automation Letters*, pp. 1–1, 2019.

[20] S. Garg, N. Suenderhauf, and M. Milford, "Semantic–geometric visual place recognition: a new perspective for reconciling opposing views," *The International Journal of Robotics Research*, p. 027836491983976, apr 2019.

[21] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov 2004.

[22] Z. Chen, A. Jacobson, N. Sünderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford, "Deep learning features at scale for visual place recognition," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3223–3230.

[23] Z. Chen, F. Maffra, I. Sa, and M. Chli, "Only look once, mining distinctive landmarks from ConvNet for visual place recognition," in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe, 2017, pp. 9–16.

[24] N. Sünderhauf, P. Neubert, and P. Protzel, "Are we there yet? challenging seqslam on a 3000 km journey across all four seasons," in *Workshop on Long-Term Autonomy at the International Conference on Robotics and Automation (ICRA)*. IEEE, 2013, pp. 1–3.

[25] M. Milford, B. Upcroft, S. Shirazi, N. Suenderhauf, E. Pepperell, A. Jacobson, F. Dayoub, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, "Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free," in *Robotics: Science and Systems XI*, vol. 11. IEEE, 2015.

[26] S. Hausler, A. Jacobson, and M. Milford, "Filter Early, Match Late: Improving Network-Based Visual Place Recognition." Institute of Electrical and Electronics Engineers (IEEE), Jan 2020, pp. 3268–3275.

[27] N. Sunderhauf, F. Dayoub, P. Corke, B. Upcroft, and M. Milford, "Transerable Place Categorization and Semantic Mapping on a Mobile Robot using Convolution Networks," pp. 5729–5736, 2015.