2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Reuse your features: unifying retrieval and feature-metric alignment

Javier Morlana and J.M.M. Montiel

Abstract— **We propose a compact pipeline to unify all the steps of Visual Localization: image retrieval, candidate re-ranking and initial pose estimation, and camera pose refinement. Our key assumption is that the deep features used for these individual tasks share common characteristics, so we should reuse them in all the procedures of the pipeline. Our DRAN (Deep Retrieval and image Alignment Network) is able to extract global descriptors for efficient image retrieval, use intermediate hierarchical features to re-rank the retrieval list and produce an initial pose guess, which is finally refined by means of a feature-metric optimization based on learned deep multi-scale dense features. DRAN is the first single network able to produce the features for the three steps of visual localization. DRAN achieves competitive performance in terms of robustness and accuracy under challenging conditions in public benchmarks, outperforming other unified approaches and consuming lower computational and memory cost than its counterparts using multiple networks. Code and models will be publicly available at github.com/jmorlana/DRAN.**

I. Introduction

Feature extraction is a relevant step in most computer vision tasks. Traditional approaches rely on image gradients to extract *sparse features*, for example, edges or keypoints in a fixed hand-crafted manner. In the last decade, deep learning has taken the spot in feature extraction, with Convolutional Neural Networks (CNNs) as the most successful method. CNNs apply a set of convolutional filters to the image, obtaining a *dense hierarchy of features*. Inherently, CNNs go deeper as the resolution decreases while obtaining higher semantics, obtaining a pyramidal representation of the image.

We focus on visual localization for Visual SLAM (Simultaneous Localization And Mapping), i.e, we are assuming that a 3D map of the scene is available, which has been built using SfM (Structure-from-Motion) or Visual SLAM. The map is composed of 3D points, $\mathbf{P}_i$, and keyframes. Keyframes, or reference frames, are a selected set of images from which the map geometry is computed by Bundle Adjustment. Per each keyframe, we have available its image $\mathbf{I}_j$ and its camera pose $\{\mathbf{R}_j, \mathbf{t}_j\}$. Given a query image $\mathbf{I}_q$ and the map, the camera location procedure efficiently retrieves the closest map keyframes $\mathbf{I}_k$ and estimates the 6-DoF pose of the query image with respect to the map, $\{\mathbf{R}_q, \mathbf{t}_q\}$.

First we apply a *keyframe retrieval step*, in which the typical deep learning retrieval algorithm usually takes three sub-steps: i) the query image is forwarded through the network encoder, obtaining dense hierarchical features, ii) the deepest feature map is pooled [1, 2] obtaining a compact global descriptor, and iii) the descriptor is compared against the other global descriptors of the keyframes in the database, ranking the keyframes by descriptor similarity. Images are translated into compact vectors based on the deepest features, which typically encodes the high-level features of the image. The comparison of global descriptors gives an initial list of candidate keyframes ranked by similarity.

For the *camera feature-metric pose estimation step*, we apply deep image alignment techniques [3–6] minimizing a feature-metric error of the dense hierarchical features that encode the image at different resolutions, estimating the relative pose between the query and a keyframe. These methods are robust against illumination and point of view changes, and achieve high precision as they can get subpixel accuracy, but they need a good initial pose guess to converge to the correct minimum.

Most of the candidate keyframes in the list depict the same place as the query and might provide a coarse pose initialization to the feature-metric optimization, but their sorting criteria does not take into account the conditions that most favour the pose optimization, i.e. point of view similarity. For this reason, the intermediate *re-ranking step* re-sort the keyframes in the initial list to prioritize the overlap and point of view similarity with the query. This step exploits the dense hierarchy of features to produce matches between the query and the keyframes, yielding 2D-3D matches between the query and the map. The matches are refined by a PnP+RANSAC, obtaining an initial pose guess which is closer to the true one, and the feature-metric optimization can successfully converge.

Other works in literature deal with Visual Localization employing different networks for each of the steps. In contrast to them, we use a unique dense hierarchy of features, what is efficient and elegant, providing competitive results of accuracy. Benefits of unified approaches are threefold. Firstly, unified learnt approaches will lead the way to deep direct SLAM algorithms able to use their own features for relocalization and loop closure. Direct SLAM methods use image intensities to perform tracking but they need additional features for relocalization and loop closure. Secondly, unified approaches could benefit from the image retrieval training data, which is typically easier to obtain as it is labelled only at the image level, using this same data for the training of their local descriptors. And lastly, it is obviously more efficient as the images are processed by a single network.

Our key insight is that it is beneficial to use a unique hierarchy of learned features for all the steps of the camera visual localization. Following these ideas, we propose DRAN

This work was supported by the EU-H2020 grant 863146: ENDOMAPPER, the Spanish government grants PGC2018-096367-B-I00, and by Aragón government grant DGA_T45-17R.

The authors are with the Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, María de Luna 1, 50018 Zaragoza, Spain. E-mail: {jmorlana, josemari}@unizar.es.

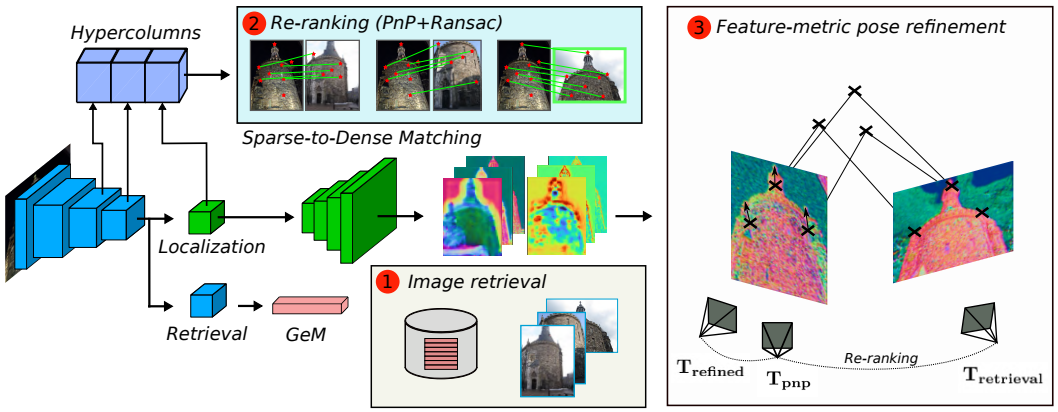

Fig. 1: **DRAN pipeline.** DRAN provides the features for the three steps of Visual Localization with a single network. The architecture follows a U-Net style with two heads: localization and retrieval. The green blocks are trained specifically for feature-metric alignment, while the blue ones are pretrained in a generic retrieval dataset and frozen during training.

(Deep Retrieval and image Alignment Network, Figure 1), a unified architecture that combines the knowledge from retrieval and camera pose stages. Our contributions are:

- DRAN, a multi-task single network providing, in a single forward pass, both an image global descriptor for retrieval, hypercolumns for re-ranking and initial pose estimation, and a multi-level dense feature hierarchy for feature-metric camera pose refinement.
- An evaluation under challenging conditions, showing that our system achieves better performance than other unified systems, and it is competitive against feature matching pipelines that combine different networks.

II. RELATED WORK

We review the relevant work in the related areas of this paper: direct and indirect SLAM, image retrieval, deep camera pose estimation and unified methods.

Direct and Indirect SLAM is the widely used classification for traditional SLAM techniques. Indirect SLAM [7–10], processes the image to detect, describe and match a sparse set of keypoints that are robust to illumination and viewpoint changes, these matches are fed in a geometric Bundle Adjustment (BA) to recover the scene geometry. In contrast, direct SLAM operates with raw image intensities, relying on brightness consistency [11–13] to also feed, in this case, a photometric BA. Direct alignment allows sub-pixel accuracy but is vulnerable to illumination changes and suffers from small convergence basin. We propose the use of learned features to overcome the challenges of direct methods when applied to camera pose estimation. One of the first works attempting this is [14], which proposes the integration of learned features for relocalization into DSO [11]. Differently, a SLAM system with our unified approach as a feature extractor would use its own features for every task, without relying on classical photometric alignment.

Image retrieval (IR) stands for the task of efficiently retrieving an image among a database of visited places, in our case the keyframes. To perform a quick search, a compact image representation is needed. Traditionally, the image embedding was obtained by the aggregation of hand-crafted local descriptors such as SIFT [15], ORB [16] in a Bag-of-Words [17, 18] representation. Nowadays, the field is dominated by CNN representations that aggregate feature maps into a global descriptor [1, 2, 19–21]. IR training only requires image-level labels, i.e. if two images depict the same place (positives) or not (negatives). This is much easier to obtain than precise pixel-level labels, which are typically needed for deep local features. The global descriptor allows two images to be compared efficiently with a single distance computation. Our work adopts the state-of-the-art pooling method Generalized-Mean (GeM) for the IR step[1].

Deep camera pose estimation mainly encompasses three approaches: pose regression, image matching and deep direct alignment. Pose regression [22, 23] learns to directly map the input to pose parameters without any 3D constraint. They require a lot of training data and do not generalize well to novel domains. Differently, image matching extracts local features [24–26] and performs data association by descriptor distance or learned matchers [27, 28]. The pose can be obtained with a PnP [29] algorithm if those local features are present in a 3D reference model. Deep direct alignment [3–6, 30] takes an approach similar to direct SLAM, trying to minimize a photometric error based on the deep features of a CNN, i.e. a feature-metric optimization. In this work, we build on top of the recent framework PixLoc [3], which extracts multi-scale dense descriptors that are aligned iteratively in a pyramidal approach. The feature-metric error is evaluated on the 2D projections of the corresponding 3D model built by SfM. S2DNet [31] is another approach that also takes advantage of the map projections to match sparse deep descriptors from the reference frame against the dense descriptors of the query. In contrast to us, S2DNet is trained to solve only the matching step between two images.

Unified approaches try to join the local and global descriptor extraction into a single system. DELG [32] obtains a GeM global descriptor [1] and applies an attention mechanism [33] to obtain local descriptors for re-ranking the initial candidates. HF-Net [26] proposed a distillation framework in a teacher-student approach to learn from NetVLAD [2] and SuperPoint [24], being able to perform retrieval and camera localization. UR2KiD [34] also performs retrieval

and matching, with the benefit of being trained only with image labels. S2DHM [35] uses an encoder pretrained for retrieval to perform re-ranking and extract an initial pose under challenging conditions. We use the ideas of S2DHM but applying them to networks trained in generic retrieval datasets (SfM120k [1], MSLS [36]). Besides, differently from them, our work unifies the tasks of image retrieval, local matching and feature-metric image alignment, introducing, to the best of our knowledge, the first system to combine them in the same network.

III. Features for Retrieval and Alignment

We propose the DRAN (Deep Retrieval and image Alignment Network) architecture, whose main steps are described in Fig. 1. We argue that most of the meaningful features are already extracted by IR networks, so we do not need to train another full pipeline end-to-end to detect features for the camera pose estimation. We adopt a shared encoder architecture with two heads: retrieval and localization. The encoder is a VGG16 net pretrained for IR in the SfM120k/MSLS dataset, which is splitted after the `conv4` stage. We use retrieval datasets to pretrain our encoder, as this data is easier to obtain than accurate 3D models and allows to acquire invariance against challenging conditions.

The retrieval head incorporates the last stage of the truncated encoder (`conv5`) and a GeM pooling layer that aggregates the last feature map into a compact vector. As we do not want to affect the retrieval performance, both the encoder and the retrieval head remain frozen during training, retaining their original weights fine-tuned for retrieval.

The localization head has identical structure as the retrieval head (`conv5` of VGG16), with the difference that it is optimized during training. It connects to a decoder network with skip connections in a U-Net [37] style. Following [3], we optimize all the weights involved in order to learn features for accurate camera localization (III-C). The output of the decoder is the hierarchy of features, along with its uncertainty. The representation of an image for each scale level l is a feature map $\mathbf{F}^l = \mathbb{R}^{W_l \times H_l \times D_l}$ and its per pixel uncertainty $\mathbf{U}^l = \mathbb{R}^{W_l \times H_l}$. The Levenberg-Marquardt optimization will minimize the feature-metric error in these features, as explained in III-C.

To provide a good initial guess for the feature-metric camera pose optimization, we extract hypercolumns [35] using the features computed by the shared encoder and the localization head. We filter the retrieval candidates and obtain an initial pose by means of a PnP+RANSAC. This pose is better than the coarse pose obtained by IR, boosting the localization performance of the subsequent optimization.

A. Compact Global Image Descriptor

We adopt the Generalized-Mean (GeM) [1] pooling layer as the method to aggregate the activations from the last layer of the retrieval head. The GeM operation for each channel of the $C \times H \times W$ activation maps is described in Eq. 1. $\mathcal{X}_k$ is the set of HW activations of each channel and p_k is the learnt parameter that controls the pooling operation.

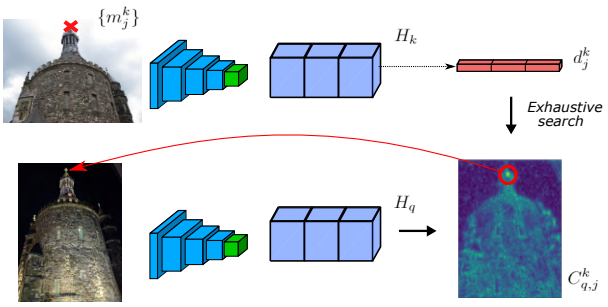

Fig. 2: Exhaustive matching for re-ranking and initial pose. Only points in the 3D map are matched, obtaining a sparse set of descriptors from the dense hypercolumn H_k. Each descriptor d_j^k is searched densely in the query hypercolumn H_q, identifying the match as the global maxima of $C_{q,j}^k$.

$$\mathbf{f} = [f_1 \dots f_k \dots f_K]^\top \in \mathbb{R}^{512}, \quad f_k = \left(\frac{1}{|\mathcal{X}_k|} \sum_{x \in \mathcal{X}_k} x^{p_k} \right)^{\frac{1}{p_k}} \tag{1}$$

Both the encoder and p_k are trained on the SfM120k [1] dataset, which depicts popular landmarks, or in MSLS [36], composed of urban and suburban scenes. It is trained using a siamese architecture and the contrastive loss [1, 38]. The contrastive loss takes as input a tuple containing 1 query, 1 positive example and 5 negative examples. Here the objective is to obtain similar descriptors for images depicting the same place, where the distance is computed by the dot product of two L2-normalized global descriptors. For testing, multiscale and whitening is applied as in [1].

B. Re-ranking and initial pose estimation

The IR module focuses on finding keyframes depicting the same place than the query, but in some cases, the first candidate is not the best initialization for feature-metric alignment. Images can suffer from big changes in scale or in point of view, or little overlap, resulting in a bad pose initialization that hinders convergence to the optimal pose.

For this reason, we employed a re-ranking method that minimize this issue by selecting the best candidate to initialize with, based on the number of inliers. We brought the method S2DHM (Sparse-to-Dense Hypercolumn Matching) proposed by Germain et al. [35] to our features. The goal is to obtain local matches between the query and the keyframes with the encoder features, in order to filter the image retrieval candidates and estimate an initial 6DoF pose guess.

Given a query I_q and a keyframe I_k, we extract features from layers `conv_3_3`, `conv_4_1`, `conv_4_3`, `head_1` and `head_3`, using our trained localization head instead of the pretrained for retrieval. These features are upsampled using bilinear interpolation to match with the resolution of the earliest layer (`conv_3_3`, resolution is 1/4 of the input image) and concatenated. The result is the so-called *hypercolumns* H_q and H_k, a dense descriptor that encodes information from different levels of the network.

As we already have a 3D map, we will only try to match the 3D points P_k that were already detected in 2D locations $\{m_j^k\}_{j=1\dots P_k}$ in the keyframe. We interpolate the hypercolumn H_k at each location $\{m_j^k\}$, obtaining a sparse set of de-

scriptors $\{d_j^k\}_{j=1...P_k}$. To find the matches between the dense query hypercolumn and the sparse keyframe descriptors, a dot product is computed between every sparse descriptor d_j^k and the dense hypercolumn, obtaining a cross-correlation map $C_{q,j}^k = H_q \cdot d_j^k$. The tentative matches corresponds to the global maximum of the cross-correlation map for each of the points p_k considered. To avoid outliers due to repetitive patterns or occlusions, a ratio test is conducted.

The resulting P_k 2D-3D matches are fed into a PnP+RANSAC scheme, which outputs the final inliers and the initial pose estimation. The keyframe with the most inliers is chosen, along with the PnP pose estimated for the query. This pose is the seed given to the final optimization.

C. Deep direct feature-metric image alignment

We employ the method proposed in [3] to align our hierarchy of deep features. As other recent works [4–6, 30], they treat deep image alignment as a direct feature-metric optimization problem, in a similar way as DSO [11] minimizes the photometric error. We describe it briefly, for further details please refer to [3].

As PixLoc proposes, we extract L=3 feature maps from the U-Net decoder, with strides 1, 4 and 16. The shallow levels encode low texture cues while deeper levels capture high level features and semantic content. This pyramidal representation is similar to traditional photometric alignment. Learning this representation instead of relying on the raw photometric values allows to overcome with the known limitations of direct image alignment: illumination changes and small convergence basin. For a feature level l, the feature-metric residual is defined as weighted addition of the feature-metric error between the query and the retrieved images for the 3D map points detected in the query image:

$$E_l(\mathbf{R}_q, \mathbf{t}_q) = \sum_{i,k} w_k^i \rho(\left\|\mathbf{r}_k^i\right\|_2^2) \quad (2)$$

$$\mathbf{r}_k^i = \mathbf{F}_q^l\left[\mathbf{p}_q^i\right] - \mathbf{F}_k^l\left[\mathbf{p}_k^i\right] \in \mathbb{R}^D \quad (3)$$

$$w_k^i = \frac{1}{1+\mathbf{U}_q^l\left[\mathbf{p}_q^i\right]} \frac{1}{1+\mathbf{U}_k^l\left[\mathbf{p}_k^i\right]} \in [0,1] \quad (4)$$

Where $\mathbf{F}_q^l$ and $\mathbf{F}_k^l$ are the feature maps for the query and the keyframe at a certain level l, $\left[\mathbf{p}_q^i\right]$ and $\left[\mathbf{p}_k^i\right]$ are the projections of the point $\mathbf{P}_i$ with subpixel accuracy, and $\mathbf{U}_q^l$ and $\mathbf{U}_k^l$ are the predicted uncertainty maps. w_k^i learns to determine if the location of a 3D point is good for localization or not. If the point projection has low uncertainty in both the query and the reference image, w_k^i will tend to 1. Otherwise, w_k^i will tend to 0, weighting down the residual in the optimization. For N points, (2) defines the goal function to be minimized with the Levenberg-Marquadt (LM) algorithm. The network learns to find good features to localize and whether a point is reliable or not.

The initial guess for the iterative optimization is the one selected by the Sparse-to-Dense matching. The algorithm starts optimizing the coarsest level (stride 16), the feature maps with lower resolution but higher depth, and successively optimizes the finer levels, going from a coarse estimation to a finer one. As the finest layer has the same resolution as the input image, feature-metric optimization can achieve subpixel accuracy, just like classic photometric alignment. The LM optimization for the camera pose comes down to solve the linear system $-\left(\mathbf{H}+\boldsymbol{\lambda}\,\mathrm{diag}\left(\mathbf{H}\right)\right)\boldsymbol{\delta} = \mathbf{J}^\top\mathbf{W}\mathbf{r}$, where $\mathbf{H} = \mathbf{J}^\top\mathbf{W}\mathbf{J}$. $\mathbf{J}$ is the Jacobian, $\mathbf{W}$ is weighting matrix depending on (4) and $\boldsymbol{\delta} \in \mathbf{SE}(3)$ is the pose update parameterized by its Lie algebra. The damping parameter $\boldsymbol{\lambda}$ is formulated as a fixed and learned 6×6 diagonal matrix coding the damping independently in each of the 6 DoF of the camera pose and for each of the levels.

Loss function The only supervision for learning is the ground truth pose for the query images, $\left\{\bar{\mathbf{R}}_q, \bar{\mathbf{t}}_q\right\}$. The loss function (5) penalizes the Huber cost of distance in pixels the between the reprojection of the map points in the ground truth camera pose, and the reprojection of the same points in the feature-metric optimized camera pose,$\{\mathbf{R}_{l,q}, \mathbf{t}_{l,q}\}$, averaging among the different scales.

$$\mathcal{L} = \frac{1}{L}\sum_l \sum_i \left\|\Pi(\mathbf{R}_{l,q}\mathbf{P}_i + \mathbf{t}_{l,q}) - \Pi(\bar{\mathbf{R}}_q\mathbf{P}_i + \bar{\mathbf{t}}_q)\right\|_\gamma \quad (5)$$

Where Π represents the reprojection transformation and γ is the Huber cost. This formulation is not affected by the geometric scale of the scene, as it only works with the reprojection of the points and not directly with camera poses.

IV. Experiments

In this section we evaluate the advantages of our *retrieval deep alignment*, comparing it against other learned approaches. In section IV-A, we explain the datasets used for training and evaluation, and the baselines used to compare. We show the accuracy of the camera pose estimated by our system in large-scale visual localization in section IV-B, performing better than the other unified approaches and competitively against feature matching approaches. Finally, in section IV-C, we perform an ablation study, showing the benefits of each of the elements of the pipeline, and the run time and memory efficiency of the DRAN.

A. Datasets and baselines

Training As other works [3, 35], we train two models in different datasets to learn specific domains. For the first one, our retrieval encoder is pretrained on SfM120k [1], which depicts common landmarks around the world and clustered with COLMAP [39]. Training uses the contrastive loss and hard-negative mining. We experimented with the original version given by the authors, which initializes their training with weights from custom ImageNet pretraining on Caffe [40], but we found better convergence when initializing with PyTorch [41] ImageNet pretraining [42]. We use MegaDepth [43] dataset for training the feature-metric alignment procedure. It contains about 1 million images depicting popular landmarks, grouped into 196 scenes and reconstructed by COLMAP [39]. MegaDepth provides depth maps and pose information for every camera. We use the same split as D2-Net [25] for training and validation. The training is performed for 20k iterations with the Adam optimizer, using a constant learning rate of 5×10^{-6} and a batch size of 6. This first model will be evaluated in Aachen Day-Night.

	Method	Aachen Day-Night		RobotCar Seasons		Extended CMU Seasons		
		Day	Night	Day	Night	Urban	Suburban	Park
FM	Pixloc	64.3 / 69.3 / 77.4	51.0 / 55.1 / 67.3	52.7 / 77.5 / 93.9	12.0 / 20.7 / 45.4	88.3 / 90.4 / 93.7	79.6 / 81.1 / 85.2	61.0 / 62.5 / 69.4
	S2DNet	84.3 / 90.9 / 95.9	46.9 / 69.4 / 86.7	53.9 / 80.6 / 95.8	14.5 / 40.2 / 69.7	-	-	-
	D2-Net	84.3 / 91.9 / 96.2	75.5 / 87.8 / 95.9	54.5 / 80.0 / 95.3	20.4 / 40.1 / 55.0	94.0 / 97.7 / 99.1	**93.0** / **95.7** / **98.3**	**89.2** / **93.2** / **95.0**
	hloc	**89.6** / **95.4** / **98.8**	**86.7** / **93.9** / **100**	**56.9** / **81.7** / **98.1**	**33.3** / **65.9** / **88.8**	**95.5** / **98.6** / **99.3**	90.9 / 94.2 / 97.1	85.7 / 89.0 / 91.6
Unified	S2DHM	56.3 / 72.9 / 90.9	30.6 / 56.1 / 78.6	45.7 / 78.0 / **95.1**	**22.3** / **61.8** / **94.5**	65.7 / 82.7 / 91.0	66.5 / 82.6 / **92.9**	54.3 / **71.6** / **84.1**
	HF-Net	**79.9** / 88.0 / 93.4	40.8 / 56.1 / 74.5	53.0 / 79.3 / 95.0	5.9 / 17.1 / 29.4	**89.5** / **94.2** / **97.9**	76.5 / 82.7 / 92.7	57.4 / 64.4 / 80.4
	UR2KiD	**79.9** / **88.6** / **93.6**	45.9 / 64.3 / 83.7	-	-	-	-	-
	DRAN (ours)	76.9 / 86.2 / 90.8	**65.3** / **78.6** / **85.7**	**55.9** / **80.7** / **95.1**	19.8 / 36.7 / 54.8	88.7 / 91.4 / 93.8	**85.4** / **87.5** / 90.0	**67.3** / 69.7 / 72.0

TABLE I: Results for Large-scale localization in Aachen Day-Night, RobotCar Seasons and Extended CMU Seasons. We highlight in red the best result for the **Feature Matching** alternatives, while in blue the best result for the **Unified** algorithms.

For our second model, our retrieval encoder is pretrained on MSLS [36], a large dataset for urban and suburban place recognition from images sequences. We used the model provided by [38], which was trained in MSLS using their Generalized Contrastive Loss. As PixLoc [3], we use the training set of the Extended CMU dataset for the feature-metric training. The training is performed for 40k iterations with the Adam optimizer, using a constant learning rate of 1×10^{-5} and a batch size of 3. This model will be evaluated in RobotCar and the Extended CMU seasons datasets.

Baselines We compare against two groups of methods: Feature Matching (FM) and Unified methods.

For Feature Matching, we consider methods for matching between two images, where the retrieval is given by an external IR network, typically NetVLAD. In FM, we find the state-of-the-art methods for local matching as D2-Net [25] or the toolbox hloc [26, 27], which uses SuperPoint [24] and SuperGlue [27]. We also compare against PixLoc [3], which performs feature-metric optimization, and S2DNet [31], a learned matching system for sparse-to-dense matching.

In the Unified methods, we consider methods able to perform retrieval and camera pose estimation with a single network. Here we found S2DHM [35], which performs sparse-to-dense matching with an encoder trained for retrieval in a subset of RobotCar Seasons or Extended CMU. HF-Net [26] proposes a pipeline that distilles knowledge from NetVLAD [2] and SuperPoint [24] in a compact network. UR2KiD[34] is able to perform retrieval and local matching while only being trained with image-level labels (no keypoints needed).

B. Large-scale localization

Aachen Day-Night objective is to evaluate visual localization under challenging conditions. It is composed of 4,328 reference images depicting the city of Aachen, taken during daytime with hand-held devices. A 3D model is reconstructed with these images, and 922 queries (824 daytime, 98 nighttime) along with its 6DoF are provided. The benchmark protocol from [44] reports the percentage of queries localized for different thresholds for the camera position and orientation error. Localization recall is provided for three threshold levels: (25 cm, 2º), (50 cm, 5º) and (5 m, 10º), which can be seen in Table I. For obtaining the retrieval candidates with DRAN, we use multiscale and learned whitening, for scales $1, \frac{1}{\sqrt{2}}$ and $\frac{1}{2}$, as proposed by [1].

Among the unified approaches, DRAN performs the best by a great margin in the night-queries, while performing comparably as UR2KiD in the day ones. Feature matching pipelines as hloc, that uses NetVLAD, SuperPoint and SuperGlue, perform better than ours, with the drawback that they use several networks to perform localization, while we only need one. DRAN obtains the camera pose estimation using $N = 5$ retrieval candidates. hloc, for example, reports results using 50 candidates.

Aachen Day-Night reference poses are really sparse, providing a extremely coarse initialization that makes difficult to pure feature-metric methods to converge. Here is where the sparse-to-dense method shines, finding matches in difficult situations (Figure 3) and providing a much more precise initial pose to the feature-metric optimization. In contrast to PixLoc, which only performs the optimization with a given retrieval, we are able to perform our own retrieval and estimate more accurate and robust camera poses.

RobotCar Seasons and Extended CMU Seasons show two driving scenarios. RobotCar is comprised of several sequences under different weather conditions that are classified under day and night groups. CMU is comprised by three kind of scenarios: urban, suburban and park, which are depicted under seasonal changes. We use multiscale and PCA whitening to obtain the retrieval candidates.

For RobotCar, our method outperforms all the other Unified approaches in the Day condition and obtains similar results as hloc, the best Feature Matching method. In the Night condition, we perform similarly as S2DHM in the finest threshold, but they outperform us in the others. The high night-time performance of S2DHM could be explained because it is the only method that has been trained in a subset of RobotCar Seasons, including night images.

For Extended CMU, the results are heterogeneous. Complex Feature Matching pipelines as hloc and D2-Net perform better than Unified approaches, with the drawback of needing an external retrieval. We outperform the other Unified frameworks at the finest level in Suburban and Park scenes, and perform comparably to HF-Net in the Urban set. This higher performance in the finest threshold arguably comes from the feature-metric optimization, being DRAN the only Unified approach that applies this step. We outperform PixLoc, the other pure feature-metric approach, in all of the scenes.

C. Ablation study and Runtime

We perform an ablation study in the Aachen Day-Night dataset of the different modules of our algorithm in Table II. (R+A) uses our top scored keyframe of our retrieval network

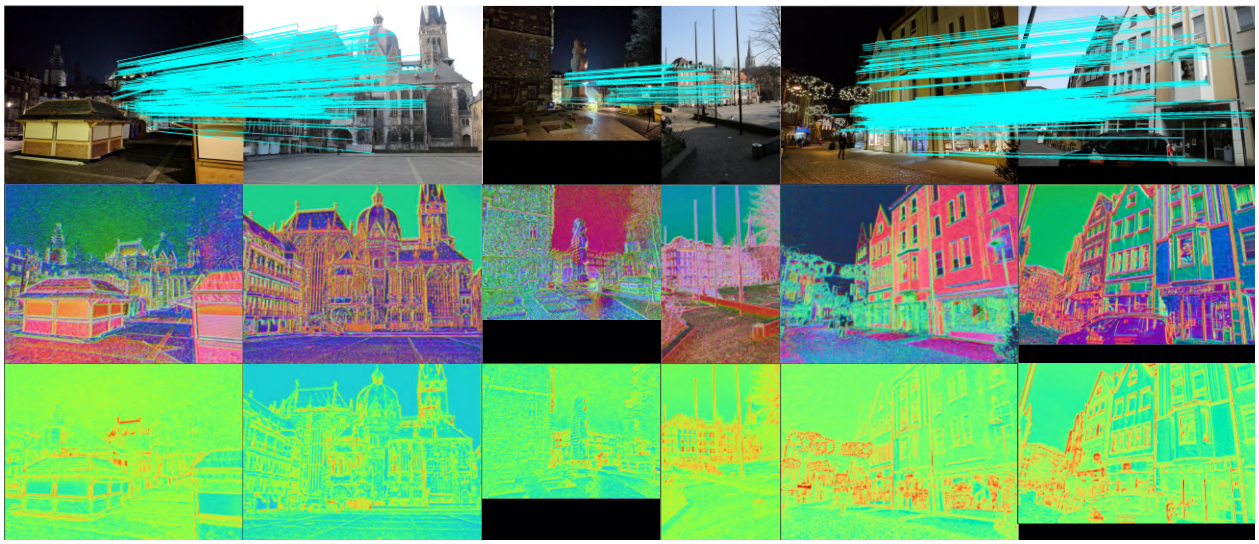

Fig. 3: Examples from Aachen Day-Night. Each two columns depicts a query-reference pair. The first row shows the Sparse-to-Dense matching of our approach, showing impressive results in night conditions under severe viewpoint change or occlusions. The second row shows the features **F** from the finest resolution learned in the decoder. The last row shows the also learned uncertainty maps **U**, where red means that points are more reliable, and blue that points are ignored.

Method	Aachen Day-Night	
	Day	Night
DRAN (R+A)	58.6 / 64.6 / 74.2	42.9 / 44.9 / 51.0
DRAN (R+P)	73.1 / 84.0 / 90.8	58.2 / 70.4 / 85.7
DRAN (R+P+A)	76.9 / 86.2 / 90.8	65.3 / 78.6 / 85.7
DRAN - light	76.1 / 85.6 / 90.3	63.3 / 74.5 / 83.7
GeM+S2DHM+PixLoc	78.0 / 86.7 / 91.9	66.3 / 75.5 / 89.8

TABLE II: Ablation study in Aachen Day-Night.

(R) as the intial guess for deep image alignment (A), gives similar results on the day to Pixloc, the most comparable method, but lags in the night queries because the system can not overcome bad retrieval initializations.

Performing retrieval and only PnP with hypercolumns (R+P) allows to produce an accurate pose by itself giving a huge boost to the performance. Using all the above modules (R+P+A), where the feature-metric optimzation uses R+P as initial guess conforms a unified pipeline that beats the state-of-the-art of multitasks methods in Aachen night queries, while being competitive on the day condition. We show an example in the supplementary video where DRAN can not converge using (R+A), but successfully localizes with the full method (R+P+A), showing the increase in robustness. We can conclude that the final feature-metric optimization is able to refine the pose when the sparse-to-dense matching works well. The sparse-to-dense step allows convergence for wide baselines poses and filters out bad retrieval candidates.

Additionally, we experimented with a light version of DRAN (DRAN - light), and a hierarchical approach (GeM [20] + S2DHM [35] + PixLoc [3]) which uses three different networks to perform retrieval, matching and pose refinement. This hierarchical approach gives similar results as our DRAN, while being slower (Table III) and consuming more memory, having to load three networks. DRAN - light uses less layers for hypercolumns, limits points to 512 in matching and only refines the medium and fine features for

Method	IR	Features	Match	Refine	*N*	Total
DRAN (R+P+A)	0	265	1109	295	5	6.10 s
DRAN - light	0	160	321	108	5	1.87 s
GeM+S2DHM+PixLoc	66	302	1233	280	5	6.81 s

TABLE III: Runtime. Times shown in ms, except for Total.

the optimization, achieving a huge speed up with a very little drop in performance. That indicates that our approach could be further miniaturized [26] and optimized to allow single network feature extraction in SLAM.

Experiments were performed with an Intel® Core™ i7-10700K (3.80 GHz) CPU and an Nvidia GeForce RTX 2080 Ti. We assume the database is already built, so we only extract features for the query image, match against *N* reference images and refine the best candidate, so the total time is given by $t_{total} = t_{IR} + t_{features} + t_{match} \times N + t_{refine}$.

V. Conclusions

We have presented the first unified pipeline that performs all the tasks concerning Visual Localization under challenging conditions. DRAN can beat in performance the other Unified approaches in several datasets and conditions, using lower run time and memory budget than its counterpart hierarchical approach which employs different networks, while being competitive against complex Feature Matching pipelines. Convergence and robustness have increased due to the re-ranking step, while the feature-metric optimization is responsible for the final accuracy. We see DRAN as a first step towards the paradigm of a unique feature extractor, able to provide good features for feature-metric tracking and relocalization in SLAM. Despite its benefits, we are aware of the current performance gap between complex Feature Matching pipelines and Unified approaches. We believe that the joint training of retrieval and matching of unified features and the inclusion of more powerful matching methods in the initial pose estimate will close this gap.

References

[1] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning CNN image retrieval with no human annotation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 7, pp. 1655–1668, 2018.

[2] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[3] P.-E. Sarlin *et al.*, "Back to the feature: Learning robust camera localization from pixels to pose," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3247–3257.

[4] L. Von Stumberg, P. Wenzel, Q. Khan, and D. Cremers, "GN-Net: The gauss-newton loss for multi-weather relocalization," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 890–897, 2020.

[5] L. Von Stumberg, P. Wenzel, N. Yang, and D. Cremers, "LM-Reloc: Levenberg-marquardt based direct visual relocalization," in *2020 International Conference on 3D Vision (3DV)*, IEEE, 2020, pp. 968–977.

[6] Z. Lv, F. Dellaert, J. M. Rehg, and A. Geiger, "Taking a deeper look at the inverse compositional algorithm," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4581–4590.

[7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[8] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, 2021.

[9] 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*, vol. 29, no. 6, pp. 1052–1067, 2007.

[10] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, IEEE, 2007, pp. 225–234.

[11] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[12] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular slam," in *European conference on computer vision*, Springer, 2014, pp. 834–849.

[13] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*, IEEE, 2011, pp. 2320–2327.

[14] M. Gladkova, R. Wang, N. Zeller, and D. Cremers, "Tight integration of feature-based relocalization in monocular direct visual odometry," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 9608–9614.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International conference on computer vision*, Ieee, 2011, pp. 2564–2571.

[17] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, 2006, pp. 2161–2168. DOI: 10.1109/CVPR.2006.264.

[18] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012, ISSN: 1552-3098. DOI: 10.1109/TRO.2012.2197158.

[19] J. Revaud, J. Almazán, R. S. Rezende, and C. R. d. Souza, "Learning with average precision: Training image retrieval with a listwise loss," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5107–5116.

[20] 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.

[21] M. Teichmann, A. Araujo, M. Zhu, and J. Sim, "Detect-to-retrieve: Efficient regional aggregation for image search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5109–5118.

[22] A. Kendall and R. Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5974–5983.

[23] T. Naseer and W. Burgard, "Deep regression for monocular camera-based 6-dof global localization in outdoor environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 1525–1530.

[24] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[25] M. Dusmanu *et al.*, "D2-net: A trainable cnn for joint description and detection of local features," in *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 2019, pp. 8092–8101.

[26] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12 716–12 725.

[27] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[28] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-free local feature matching with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8922–8931.

[29] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate o (n) solution to the PnP problem," *International journal of computer vision*, vol. 81, no. 2, p. 155, 2009.

[30] B. Xu, A. J. Davison, and S. Leutenegger, "Deep probabilistic feature-metric tracking," *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 223–230, 2020.

[31] H. Germain, G. Bourmaud, and V. Lepetit, "S2Dnet: Learning image features for accurate sparse-to-dense matching," in *European Conference on Computer Vision*, Springer, 2020, pp. 626–643.

[32] B. Cao, A. Araujo, and J. Sim, "Unifying deep local and global features for image search," in *European Conference on Computer Vision*, Springer, 2020, pp. 726–743.

[33] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3456–3465.

[34] T.-Y. Yang, D.-K. Nguyen, H. Heijnen, and V. Balntas, "UR2KiD: Unifying retrieval, keypoint detection, and keypoint description without local correspondence supervision," *arXiv preprint arXiv:2001.07252*, 2020.

[35] H. Germain, G. Bourmaud, and V. Lepetit, "Sparse-to-dense hypercolumn matching for long-term visual localization," in *2019 International Conference on 3D Vision (3DV)*, IEEE, 2019, pp. 513–523.

[36] F. Warburg, S. Hauberg, M. Lopez-Antequera, P. Gargallo, Y. Kuang, and J. Civera, "Mapillary street-level sequences: A dataset for lifelong place recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2626–2635.

[37] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[38] M. Leyva-Vallina, N. Strisciuglio, and N. Petkov, "Generalized contrastive optimization of siamese networks for place recognition," *arXiv preprint arXiv:2103.06638*, 2021. [Online]. Available: https://arxiv.org/abs/2103.06638.

[39] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[40] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[41] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[43] Z. Li and N. Snavely, "Megadepth: Learning single-view depth prediction from internet photos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2041–2050.

[44] T. Sattler *et al.*, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8601–8610.