Unsupervised Metric Relocalization Using Transform Consistency Loss

Mike Kasper
University of Colorado, USA
michael.kasper@colorado.edu

Fernando Nobre
Amazon, USA
fnobre@amazon.com

Christoffer Heckman
University of Colorado, USA
christoffer.heckman@colorado.edu

Nima Keivan
Amazon, USA
keivan@amazon.com

Abstract: Training networks to perform metric relocalization traditionally requires accurate image correspondences. In practice, these are obtained by restricting domain coverage, employing additional sensors, or capturing large multi-view datasets. We instead propose a self-supervised solution, which exploits a key insight: localizing a query image within a map should yield the same absolute pose, regardless of the reference image used for registration. Guided by this intuition, we derive a novel *transform consistency loss*. Using this loss function, we train a deep neural network to infer dense feature and saliency maps to perform robust metric relocalization in dynamic environments. We evaluate our framework on synthetic and real-world data, showing our approach outperforms other supervised methods when a limited amount of ground-truth information is available.

Keywords: unsupervised learning, relocalization, deep features, saliency

1 Introduction

Visual relocalization refers to the task of registering a query image to an existing map, effectively seeking an answer to the question, "have I been here before?" This is a critical problem in autonomous robot navigation [1, 2, 3]. Relocalization may be limited to image retrieval, where the query image is assumed to share the same pose as the matched reference image [4, 5, 6]. More challenging still is the task of metrically refining this pose by estimating a 6 degree-of-freedom (DoF) offset between the two images [7, 8]. In this work, we focus only on the latter problem.

As the reference and query images may be separated by an arbitrary amount of time, large changes in visual appearance can inhibit accurate relocalization. Hand-engineered feature descriptors have proven insufficient in extremely dynamic environments [9, 10, 8]. Consequently, recent approaches have pursued the use of neural networks to learn more robust image representations [11, 12, 8]. However, these network architectures are typically trained with ground-truth image correspondences, requiring the scene geometry and camera poses to be known.

Acquiring these ground-truth data in the real world is non-trivial. Simpler solutions sacrifice domain coverage by restricting camera placement or synthetically rendering alternate views. Better coverage of the target domain is achieved via additional sensing or large image sets of the same scene. These requirements make one-time data capture burdensome and online capture with lightweight robot platforms nearly impossible, thus limiting our ability to adapt to novel environments.

To circumvent these problems, we instead propose a solution that does not require ground-truth image correspondences while still achieving full coverage of the target domain. This permits an unmodified version of our robot platform to collect training data while deployed. With our framework, training samples can be captured periodically, when deemed both prudent and convenient by the robot. To make correspondence-free training possible, we propose a novel loss function inspired by a key insight: when localizing a query image within a map, the same absolute pose should be obtained, regardless of the reference image used for registration. That is, given two reference images in the same world frame, aligning a query image to either one should yield the same global pose.

4th Conference on Robot Learning (CoRL 2020), Cambridge MA, USA.

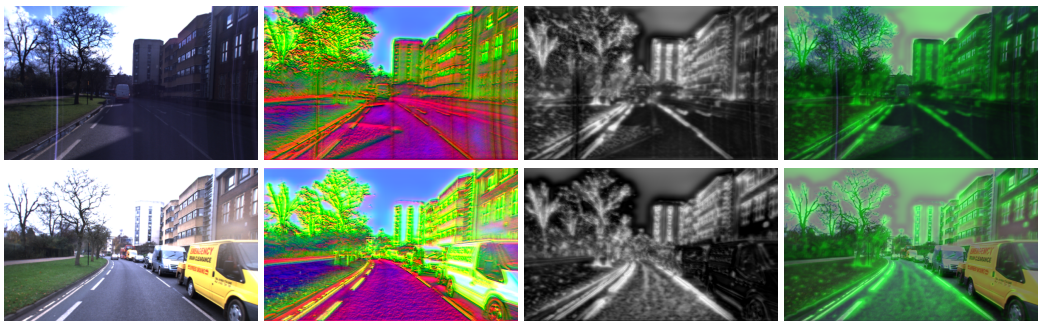

Figure 1: Images of the same location taken at different times and their resulting feature and saliency maps. For visualization purposes, we fuse all pyramid levels together using PCA for feature maps and a weighted average for saliency maps. Note the similarity of the feature maps, despite large discrepancies in the input images, and how the saliency maps successfully mask out dynamic objects.

Guided by this intuition, we develop a novel *transform consistency loss*, which operates over three images: two reference images, whose relative pose is known, and a single query image, whose relative pose is unknown. Employing this loss function, we train a network to infer dense feature and saliency maps, as seen in Figure 1. These can then be used to perform direct image registration that is robust to dynamic objects and illumination. The novel contributions of this work are:

- A transform consistency loss function for unsupervised learning.
- A network architecture for joint feature and saliency map inference.

By removing many of the requirements on data acquisition, we can learn from a much wider range of real-world images. Consequently, our framework significantly outperforms other state-of-the-art, supervised methods when trained on datasets of limited size and scope.

2 Related Work

Traditionally, hand-engineered features have been used for metric relocalization [13, 14, 15, 16]. They are designed to be invariant to changes in scale, orientation, and illumination. ORB-SLAM [1] is an example framework that employs such features. It first retrieves image candidates via a place recognition system based on bags of binary words [17], and then estimates a 6-DoF offset by minimizing the reprojection error between the image coordinates matched using ORB descriptors [15]. While hand-engineered features work well for visual odometry, they have proven inadequate for relocalization in extremely dynamic environments [10, 8].

To address the deficiencies of hand-engineered features, approaches relying on alternative image representations generated by neural networks have risen in popularity. These frameworks have been used to perform sparse keypoint detection and description [9, 10, 18, 19] as well as dense image registration [20, 21, 22, 8]. However, current solutions assume ground-truth correspondences are available at training time in order to compare multiple observations of the same point. Example loss functions include *contrastive loss* [23] and *cosine similarity* [19].

Several methods for obtaining these ground-truth correspondences have been proposed. Perhaps the simplest approach of all is to use simulated data, as we have perfect knowledge of scene geometry and camera poses [18, 8]. However, the viability of *sim2real* transfer learning greatly depends on the target environment. In general, we cannot enumerate an exhaustive list of the expected visual phenomena, let alone render them with a sufficient level of fidelity. Consequently, most approaches employ real-world data and therefore must directly tackle the correspondence problem.

One solution is to apply known transformations (*e.g.* color shifts, homographic warps) to real-world images [18, 19]. However, this still relies on synthetic data and therefore runs the risk of insufficiently sampling the target domain. Alternatively, Verdie *et al.* propose using a stationary camera to observe the same scene under varying lighting and weather conditions [9]. This makes the correspondence problem trivial, as the same pixel corresponds to the same world point in every image,

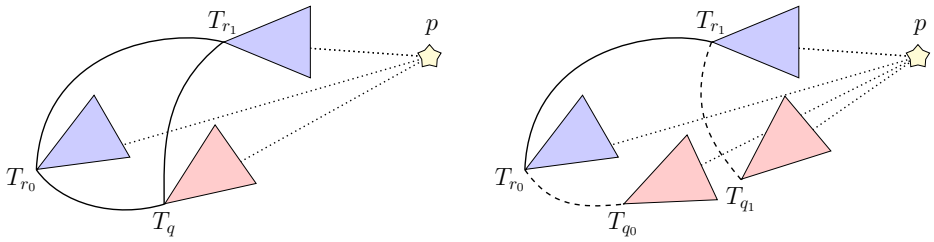

Figure 2: A visualization of transform consistency. On the left, we see the ground-truth configuration of three frames r_0, r_1, q. On the right, we see the result of *independently* aligning the query image q with each reference image r_0, r_1. We seek a robust representation of the mutually visible points p as to minimize the distance between the estimated transforms T_{r_0,q_0} and T_{r_0,q_1}.

but fails to capture visual artifacts produced by camera motion. In a similar vein, Liang *et al*. employ RTK positioning to obtain images captured from the same viewpoint [24]. Not only does this require additional hardware, it also constrains the relative camera poses to be identity. In [8], the authors employ a visual-odometry framework to automate keypoint selection and depth estimation, but require expansive 3D pointclouds or a motion-capture system to obtain the inter-sequence transforms.

A popular alternative to sensor-based solutions is Structure-from-Motion (SfM) [10, 8, 25, 19]. This involves processing large image sets of the *same* scene to build a cohesive model of the surface geometry and camera poses. This is arguably the most general solution to the correspondence problem, as it even works on random image collections scraped from internet. However, SfM cannot be used to extract correspondences from sparsely sampled scenes. It also assumes that the images are similar enough that existing methods can perform registration.

More recently, Schmidt *et al*. propose a self-supervised approach for learning robust feature descriptors [12]. Training data is sourced from multiple, independent 3D reconstructions of a single subject, built using a 3D reconstruction pipeline (*i.e*. DynamicFusion [26]). They then train solely on *intra*-sequence correspondences. Despite not training on *inter*-sequence correspondences, the authors empirically find the inferred features map well across each reconstruction. However, their trained model only serves for a single subject. It also requires large amounts of data to be captured and processed, which is problematic if we expect this to be done *in situ* by a deployed mobile agent.

To circumvent the aforementioned challenges for obtaining ground-truth image correspondences, we propose dropping the requirement altogether. In the next section, we present our novel loss function, which assesses the consistency of estimated 6-DoF camera poses. It leverages data already available in most visual localization frameworks, making it suitable for a much broader range of applications.

3 Method

Our problem formulation is largely inspired by *photometric consistency*. In short, photometric consistency uses the inferred model of the scene to render a synthetic image under different conditions (*e.g*. camera pose). The fidelity of the model can then assessed by comparing this rendering with an actual image captured under those same conditions. Photometric consistency has been used to learn depth from both monocular [27, 28] and stereo images [29].

Employing photometric consistency requires a static scene, where the brightness constancy assumption holds (although subtle illumination changes may be handled [30]). Such assumptions, however, cannot be made for relocalization. We therefore substitute photometric consistency for what we refer to as *transform consistency*. Here we use the inferred image features to *independently* register a query image with two reference images, whose relative pose is already known. We then evaluate our network by comparing the resulting transforms in a shared world frame. This concept is illustrated in Figure 2. To employ this method, we make the following assumptions:

- Dense depth maps are provided for both reference images.
- An accurate relative transform between the reference images is provided.
- The query image's visual-field sufficiently overlaps with both reference images.

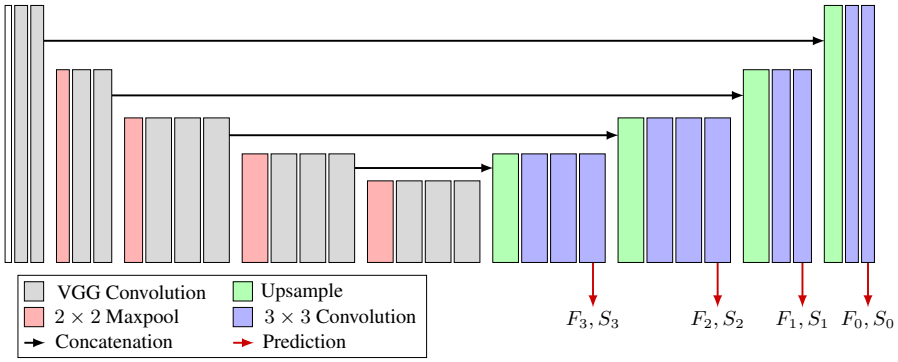

Figure 3: An overview of our network architecture. The contractive part of the network leverages the pretrained convolutional layers of VGG-16 [34], which remain constant. We independently pass each input image through the network to obtain a 16-channel feature map F and a single-channel saliency map S at each level of the image pyramid. The 3×3 convolutions are followed by batch-normalization and ReLU activation. Upsampling is achieved via bilinear interpolation. The final predictions consist of a single 1×1 convolution followed by sigmoid activation.

In practice, we obtain depth maps using stereo-vision [31], although an RGB-D sensor would also suffice. To ensure the accuracy of the relative transforms, we sample both references frames from a small spatio-temporal window of a captured video sequence. Off-the-shelf visual odometry pipelines can therefore reliably attain translation errors within 1% of the distance traveled [1, 2]. Finally, we uphold the *overlapping visual-field* assumption via a GPS. We can also leverage an image-retrieval system or crude registrations to a topological map [32]. Most vision-based, autonomous mobile platforms already satisfy these three requirements. Consequently, they can capture training data in the field, without additional sensing or significant computational overhead. In the sections that follow, we present our network architecture and formally define our objective function.

3.1 Network Architecture

In this work, we adopt a network architecture similar to DispNet [33] as it generates side predictions for constructing a multi-level image pyramid and has proven sufficient for complex, single-view inference tasks [27]. In the contractive part of the network, we employ the learned convolutional weights of the VGG-16 network [34], which are not updated during training. Our second modification involves the output predictions themselves. For a given input image I, we infer a set of 16-channel feature maps $[F^0, F^1, F^2, F^3]$ and single-channel saliency maps $[S^0, S^1, S^2, S^3]$. Here, F^0 denotes the finest resolution feature map and F^3 the coarsest. The same relationship holds for saliency maps. An overview of our network architecture is shown in Figure 3. How these predictions are integrated into our relocalization framework is detailed in the next section.

3.2 Loss Function

Our training loss comprises multiple functions over one or more $\mathbb{SE}(3)$ transforms, which are obtained through 3D registration of the inferred feature maps. Using a simplified implementation of the inverse-composition algorithm presented in [22], we ensure that this registration process permits auto-differentiation. In contrast with [22], inference is performed over each image individually. Similar to their convolutional M-estimator, our saliency maps re-weight the residuals of the image-registration problem. However, our saliency maps are generated *once* for each image. Finally, we adopt a pure Gauss-Newton implementation, dropping their Levenberg-Marqaurdt damping scalar.

3.2.1 Transform Consistency

As previously stated, training samples consist of two references frames r_0, r_1 and a query frame q. Each frame is represented by a rectified RGB image and a camera projection matrix $K \in \mathbb{R}^{3\times3}$. Reference frames are also accompanied by their respective depth maps D_{r_0}, D_{r_1}, and relative transform $\hat{T}_{r_0,r_1} \in \mathbb{SE}(3)$. Using the inferred feature and saliency maps, we invoke two instances of

direct 3D image registration. We *independently* align the query feature map F_q to each reference feature map F_{r_0}, F_{r_1} to obtain the relative transforms T_{q,r_0} and T_{q,r_1} by minimizing

$$T_{q,r} = \underset{T_{q,r}}{\operatorname{argmin}} \sum_{u \in I_r} S_r(u) S_q(u') \left\| F_r(u) - F_q(u') \right\|_\gamma . \tag{1}$$

Here u' is where the coordinates u in the reference image project onto the query image, given the current estimate $T_{q,r}$, depth map D_r, and camera projection matrices K_r, K_q. The Huber norm $\|\cdot\|_\gamma$ makes this optimization more robust to outliers. Each iteration k of the multi-level image registration yields new estimates T^k_{q,r_0}, T^k_{q,r_1}, until we obtain the final estimates T^*_{q,r_0}, T^*_{q,r_1}. These final estimates serve as ground-truth for the opposing alignment in the transform consistency loss

$$L_{\mathrm{c}}\big(T^*_{q,r_0}, T^k_{q,r_1}\big) = \left\| \log\big(\hat{T}_{r_0,r_1} \big(T^k_{q,r_1}\big)^{-1} T^*_{q,r_0}\big) \right\|_1 . \tag{2}$$

Intuitively, in Eq. (2) we are trying to compute the difference between two estimates of the same pose. However, this first requires moving both poses to a shared reference frame. In this case, we move T^k_{q,r_1} to reference frame r_0 using the provided transform $\hat{T}_{r_0,r_1}$. We then compute the relative transform error in $\mathfrak{se}(3)$ tangent space, and finally take the L1-norm of the resulting residual vector.

Transform consistency eschews the need for ground-truth poses by maximizing the agreement of two independent estimates. While there are an infinite number of *incorrect* solutions that would minimize this loss, we argue that the solution space is largely constrained by the iterative image registration process and a sufficiently large number of training examples. However, we alleviate this problem further with an additional regularization term, as described in the next section.

3.2.2 Transform Accuracy

As we assume knowledge of the relative transform $\hat{T}_{r_0,r_1}$, we can additionally perform image registration between the two reference frames and evaluate each intermediate transform $T^k_{r_1,r_0}$

$$L_{\mathrm{a}}\big(T^k_{r_1,r_0}\big) = \left\| \log\big(\hat{T}_{r_0,r_1} T^k_{r_1,r_0}\big) \right\|_1 . \tag{3}$$

The main benefit of this term is to provide stability during initialization. Empirically, we find the network may diverge when trained with the consistency loss alone. This is not the only benefit, however. Eq. (3) also promotes saliency maps that mask out fast moving objects and feature maps that compensate for large camera baselines, as these challenges are still present in a pure visual-odometry setting. Our final loss function becomes the sum of these two terms, aggregated over each iteration of image registration, using a constant scalar λ to modulate the influence of each term

$$\sum_k L_{\mathrm{c}}\big(T^*_{q,r_0}, T^k_{q,r_1}\big) + L_{\mathrm{c}}\big(T^k_{q,r_0}, T^*_{q,r_1}\big) + \lambda L_{\mathrm{a}}\big(T^k_{r_1,r_0}\big). \tag{4}$$

4 Experimental Results

We evaluate our framework using both synthetic and real-world data. During training and evaluation our network uses a fixed number of image-registration iterations, with all relative transforms initialized with identity. From the coarsest level of the image pyramid to the finest, we perform 16, 12, 8, & 4 iterations. The input image resolution for both datasets is 640×384. We train the network using the Adam Optimizer [35] with a learning rate of 10^{-4} during initialization and 10^{-5} during general training. Gradient accumulation is also used to achieve an effective mini-batch size of 16.

To ensure the network converges reliably, we initialize the network using a substantially higher relative transform accuracy weight $\lambda = 10$. After training for a single epoch we lower it to $\lambda = 1$. While the transform accuracy loss provides stability, it is the transform consistency loss that permits us to learn robust image representations that map well across different video sequences. To illustrate this benefit, we also evaluate our network trained using only the transform accuracy loss, defined in Eq. (3). We will refer to this method as "Ours (VO)".

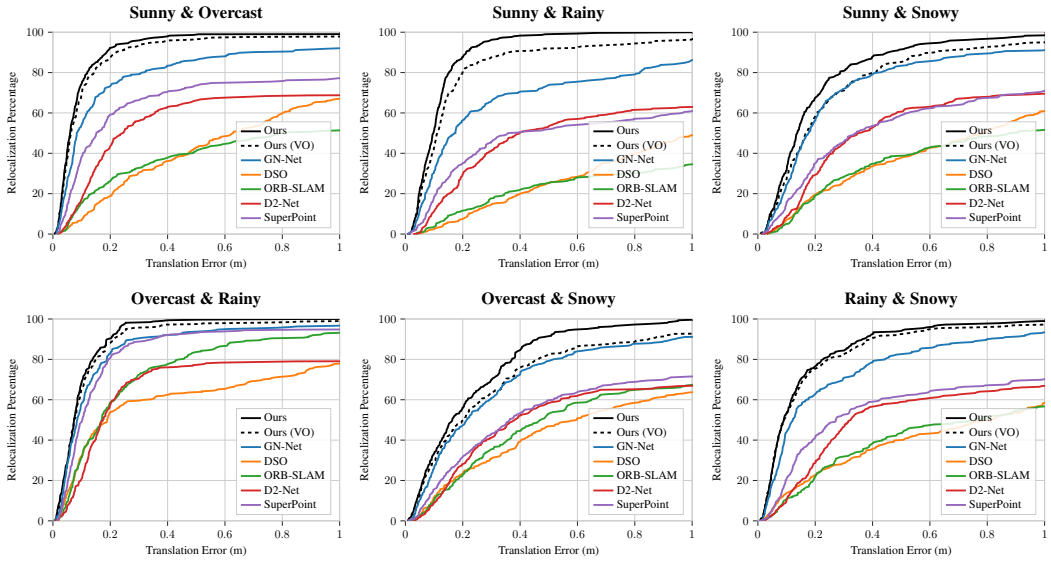

Figure 4: Cumulative relocalization accuracy compared to other leading methods on the GN-Net Relocalization Benchmark [8]. Each test consists of image pairs from the Robotcar dataset [36] that exhibit different weather conditions. We additionally compare the proposed system, *Ours*, with the same framework trained only using transform accuracy loss, *Ours (VO)*.

4.1 Training Datasets

For our experiments on synthetic data we employ the CARLA simulator [37]. We simulate a car driving on rural and urban roads, at different times of day and under different weather conditions. Each trajectory is randomly populated with dynamic cars and pedestrians. CARLA directly provides us with colors images, camera poses, and semantic segmentations. However, we compute depthmaps using Semi-Global Matching [31] on the stereo pair, rendered with added noise. In total, our training dataset consists of 8K images and 20K unique frame triplets $\{r_0, r_1, q\}$, which exhibit a median camera baseline of 1.25m. We use the same pipeline to generate a test dataset of 500 frame triplets, sampling a completely distinct set of random trajectories.

For experiments on real-world data, we employ the Oxford Robotcar dataset [36]. This dataset was captured by an autonomous car over the course of two years, using several cameras, lidars, GPS, and IMU. Given the extended duration of data acquisition, the captured images exhibit large changes in visual appearance due to varying time of day, weather, and season. We select training triplets using the global poses obtained through GPS-inertial positioning. Again, the median camera baseline is approximately 1.25m, but can be as high as 6m. For computing depth maps, we employ Semi-Global Matching [31] on the wide stereo pair. In total, our training dataset consists of approximately 7.5K images and 15.5K unique frame triplets, drawn from 8 Robotcar sequences.

4.2 Experiments

We first look at performance in terms of translation error. In Figure 4 we see how our framework performs on the relocalization benchmark recently published with GN-Net [8]. This benchmark is separated into six different Robotcar sequence-pairs, exhibiting two distinct weather conditions (*e.g.* sunny and cloudy). Suitable image candidates are provided by the benchmark, allowing frameworks to focus solely on metric pose refinement. Here we can see how our framework compares with other leading methods including GN-Net [8], ORB-SLAM [1], DSO [38] SuperPoint [18], and D2-Net [25]. The plots in Figure 4 represent cumulative relocalization accuracy. This can be interpreted as the percentage of relocalizations (vertical axis) that are performed within a given translation threshold (horizontal axis). A theoretically perfect system would result in a horizontal line at the top of each plot, indicating that 100% of relocalizations were performed with zero error.

Next, we employ the semantic segmentations obtained using the CARLA simulator [37] to directly evaluate our saliency maps. Of particular interest is how well our saliency maps mask out dynamic

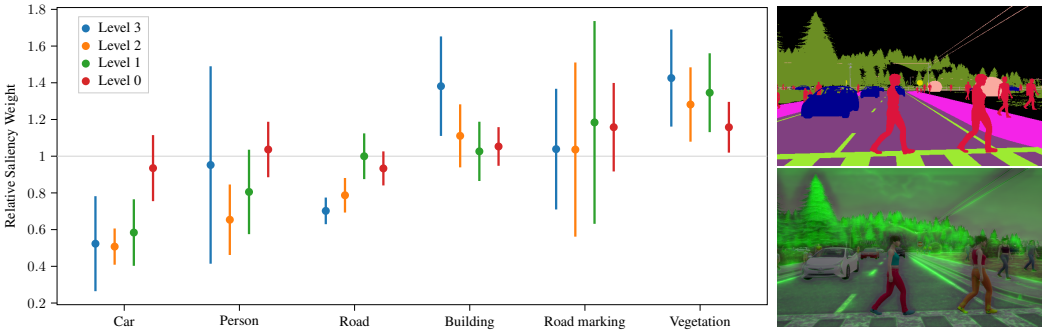

Figure 5: Using the CARLA simulator [37], we generate a dataset of images with their respective semantic labels (*top right*). We then evaluate our saliency maps (*bottom right*) for each class in the Cityscape taxonomy, comparing the resulting weights with those obtained using a uniform weighting strategy (*left*). The mean and standard-deviation of each pyramid level is plotted. Values above the gray line suggest additional focus while those below suggest the class is being ignored.

objects (*e.g.* cars and pedestrians). For each image in our test dataset, we compare the inferred saliency weights with those obtained using a uniform weighting strategy. As an example, if a instance segmentation occupies 50% of the image, but only accounts for 25% of the total saliency weight, its *relative saliency weight* is 0.5. Values above 1.0 suggest additional focus is being directed towards the class, while values below 1.0 suggest the class is being ignored. We perform this analysis independently on each pyramid level using the Cityscape taxonomy [39]. The observed relative saliency weight means and standard-deviations are plotted in Figure 5.

Image pairs in the GN-Net Relocalization Benchmark primarily exhibit changes in weather with moderate initial camera baselines. We evaluate our framework further with a more challenging dataset comprising day-night and seasonal changes, with significantly larger camera baselines. Aligning lidar pointclouds to obtain ground-truth, inter-sequence transforms, as performed in [8], is not only labor intensive but also sensitive to the hyperparameters and initial pose. We circumvent these problems by instead adopting a metric similar to relative pose error [40, 41]

$$E = \left\| \text{translation}\left(\hat{T}_{r_0,r_1} (T^*_{q,r_1})^{-1} T^*_{q,r_0} \right) \right\|. \tag{5}$$

Intuitively, we are comparing the relative translation computed via intra-sequence visual-odometry, with that computed from two independent map registrations. Our test dataset contains 40K frame triplets, with 100 samples drawn from 400 Robotcar sequence-pairs. Sampling in this fashion allows us to build a "confusion matrix" indicating how well reference frames from one sequence align the query frames of another. For this experiment, we retrain our network with a dataset of similar size and scope, created from a distinct set of Robotcar sequences. Results are shown in Figure 6.

5 Discussion

As shown in Figure 4, our approach consistently outperforms other leading methods. Compared to the next top performer, we successfully relocalize within 25cm for 15.31% more of the benchmark. We attribute the observed performance gains to our network architecture and image registration framework, as even the *Ours (VO)* approach proves to be quite accurate. By training directly on the 3D registration problem and employing multi-level saliency maps, we not only learn to ignore dynamic objects but also to down-weight objects in the foreground. In general these will exhibit larger optical flow vectors, which are problematic in direct image registration. By giving these objects less significance in the coarser saliency maps we can overcome large initial camera baselines.

In this first experiment, our network is trained with eight sequences of the Robotcar dataset. The GN-Net Relocalization Benchmark [8], however, publishes only two such training sequences. We argue this remains a valid comparison, as our solution does not leverage *any* ground-truth, inter-sequence transforms. In fact, the main benefit of this approach is the ease in which training data can be obtained. Given the minimal requirements outlined in Section 3, we can incorporate additional sequences only knowing the car's global position within 6m and heading within 15 degrees.

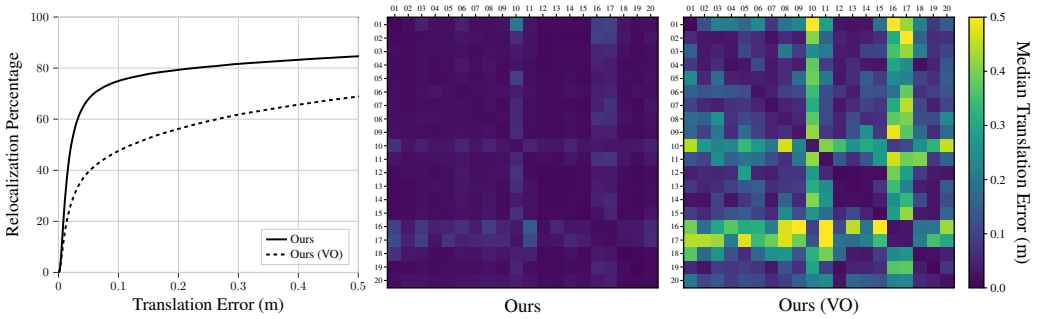

Figure 6: Cumulative relocalization accuracy (*left*) and "confusion matrices" (*right*) comparing the relative translation errors of the propose method, *Ours*, with the same framework trained only using transform accuracy loss, *Ours (VO)*. Each column of a matrix represents the Robotcar sequence of the reference frames, while each row represents the sequence of the query frames. The color of a cell indicates the median translation error, computed over 100 triplet samples.

Figure 5 shows that our network learns to ignore dynamic objects, consistently down-weighting the classes "car" and "person". In contrast, we see that larger, static objects receive additional attention, which aligns well with our intuition. As CARLA does not simulate seasonal changes in foliage, we also see the network safely learns to rely on vegetation. We do note, however, that higher resolution saliency maps trend towards a uniform weighting strategy (indicated by the gray line in Figure 5). We suspect that this is due to the convergence basin becoming narrower at each subsequent level, such that dynamic objects have significantly less impact on the refinement of the final pose, and are likely handled by the Huber norm used in Eq. (1).

In Figure 4, training our network with transform consistency loss appears to result in moderate performance gains, as compared to training with transform accuracy loss alone. However, the true benefit of using transform consistency is revealed when evaluating on a more challenging relocalization dataset, as seen in Figure 6. Most notably, we see that the *Ours (VO)* method fails to register nighttime sequences (10, 16, and 17) with daytime sequences. In fact, the majority of off-diagonal entries in its confusion matrix indicate significantly higher translation error. This can be expected, as the network does not learn from inter-sequence training examples. Evaluating with the relative pose error (5) also shows our framework is more accurate than the plots in Figure 4 would suggest.

In the most extreme conditions, we suspect that our static saliency maps, inferred once from a single input image, will not perform as well as those iteratively updated during image registration, as proposed in [22]. However, the benefit of our approach is that the saliency maps are not dependent on the current image pair. Consequently, they may be evaluated in isolation to determine which frames are suitable map keyframes. For example, we could reject frames dominated by dynamic objects and other features the network has learned are unreliable, as indicated by low-activation in the saliency maps. This is a potential direction for future research.

6 Conclusion

We have presented a novel training loss, which permits unsupervised learning of dense feature and saliency maps for robust metric relocalization. This greatly reduces the constraints on dataset acquisition, allowing data to be be captured *in situ* by mobile agents with little additional overhead. Our framework attains state-of-the-art results on the GN-Net Relocalization Tracking Benchmark, significantly outperforming other leading methods. These performance gains are attributed to our multi-resolution saliency maps and the larger training datasets that unsupervised learning affords.

References

[1] R. Mur-Artal and J. Tardos. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *Transactions on Robotics*, 2017.

[2] T. Qin, P. Li, and S. Shen. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *Transactions on Robotics*, 2018.

[3] X. Gao, R. Wang, N. Demmel, and D. Cremers. LDSO: Direct Sparse Odometry with Loop Closure. In *Intelligent Robots and Systems (IROS)*, 2018.

[4] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han. Large-Scale Image Retrieval with Attentive Deep Local Features. In *International Conference on Computer Vision (ICCV)*, 2017.

[5] S. Garg, N. Suenderhauf, and M. Milford. LoST? Appearance-Invariant Place Recognition for Opposite Viewpoints using Visual Semantics. *arXiv preprint: 1804.05526*, 2018.

[6] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.

[7] R. Gomez-Ojeda, Z. Zhang, J. Gonzalez-Jimenez, and D. Scaramuzza. Learning-Based Image Enhancement for Visual Odometry in Challenging HDR Environments. *International Conference on Robotics and Automation (ICRA)*, 2018.

[8] L. von Stumberg, P. Wenzel, Q. Khan, and D. Cremers. GN-Net: The Gauss-Newton Loss for Deep Direct SLAM. *Robotics and Automation Letters (RA-L)*, 2020.

[9] Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit. TILDE: A Temporally Invariant Learned DEtector. *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[10] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned Invariant Feature Transform. *European Conference on Computer Vision (ECCV)*, 2016.

[11] C. B. Choy and S. Savarese. Universal Correspondence Network. *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[12] T. Schmidt, R. Newcombe, and D. Fox. Self-Supervised Visual Descriptor Learning for Dense Correspondence. *Robotics and Automation Letters (RA-L)*, 2017.

[13] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints David. *International Journal on Computer Vision (IJCV)*, 2004.

[14] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. *European Conference on Computer Vision (ECCV)*, 2006.

[15] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. *International Conference on Computer Vision (ICCV)*, 2011.

[16] N. Dalal and B. Triggs. Histogram of Oriented Gradients for Human Detection. 2005.

[17] D. Gálvez-López and J. D. Tardós. Bags of binary words for fast place recognition in image sequences. *Transactions on Robotics (TR)*, 2012.

[18] D. Detone, T. Malisiewicz, and A. Rabinovich. SuperPoint: Self-supervised interest point detection and description. *Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018.

[19] J. Revaud, P. Weinzaepfel, C. De Souza, N. Pion, G. Csurka, Y. Cabon, and M. Humenberger. R2D2: Repeatable and Reliable Detector and Descriptor. *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[20] C. Tang and P. Tan. BA-Net: Dense Bundle Adjustment Network. *arXiv preprint: 1806.04807*, 2018.

[21] L. Han, M. Ji, L. Fang, and M. Nießner. RegNet: Learning the Optimization of Direct Image-to-Image Pose Registration. *arXiv preprint: 1812.10212*, 2018.

[22] Z. Lv, F. Dellaert, J. M. Rehg, and A. Geiger. Taking a Deeper Look at the Inverse Compositional Algorithm. *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[23] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. *Computer Vision and Pattern Recognition (CVPR)*, 2006.

[24] H.-J. Liang, N. J. Sanket, C. Fermuller, and Y. Aloimonos. SalientDSO: Bringing Attention to Direct Sparse Odometry. *IEEE Transactions on Automation Science and Engineering*, 2019.

[25] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler. D2-Net: A Trainable CNN for Joint Detection and Description of Local Features. In *Computer Vision and Pattern Recognition (CVPR)*, 2019.

[26] R. Newcombe, D. Fox, and S. Seitz. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[27] T. Zhou, M. Brown, N. Snavely, and D. Lowe. Unsupervised learning of depth and ego-motion from video. *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[28] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

[29] R. Garg, V. Kumar, G. Carneiro, and I. Reid. Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. *European Conference on Computer Vision*, 2016.

[30] T. Khot, S. Agrawal, S. Tulsiani, C. Mertz, S. Lucey, and M. Hebert. Learning Unsupervised Multi-View Stereopsis via Robust Photometric Consistency. *arXiv preprint: 1905.02706*, 2019.

[31] H. Hirschmüller. Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information. In *Computer Vision and Pattern Recognition (CVPR)*, 2005.

[32] W. Churchill and P. Newman. Practice makes perfect? Managing and leveraging visual experiences for lifelong navigation. *International Conference on Robotics and Automation*, 2012.

[33] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[34] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint: 1409.1556v6*, 2015.

[35] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[36] W. Maddern, G. Pascoe, C. Linegar, and P. Newman. 1 Year, 1000km : The Oxford RobotCar Dataset. *International Journal of Robotics Research (IJRR)*, 2017.

[37] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Conference on Robot Learning (CRL)*, 2017.

[38] J. Engel, V. Koltun, and D. Cremers. Direct Sparse Odometry. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.

[39] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

[40] K. Konolige, M. Agrawal, and J. Sola. Large-scale visual odometry for rough terrain. 2010.

[41] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR)*, 2012.