6-DoF Pose Refinement via Sparse-to-Dense Feature-Metric Optimization

Ajaykumar Unagar* Philipp Lindenberger* Nikolaos Tselepidis* Paul-Edouard Sarlin
ETH Zurich

Abstract

In this report, we detail our submission to the CVPR 2020 visual localization challenge. Previous work on sparse-to-dense matching showed outstanding robustness to extreme conditions, but lacks precision due to the low resolution of deep feature maps. We introduce a simple algorithm to refine the estimated pose based on the feature-metric error, and demonstrate improved localization accuracy. This, combined with better feature selection, results in state-of-art night localization on the RobotCar dataset.

1. Introduction

Visual localization is traditionally performed by matching local features across images [18, 17, 15, 16]. This assumes that such local features can be reliably detected [8, 3, 4, 13, 14] across conditions, and can be subsequently described with invariant descriptors. Obtaining repeatable features is however very difficult in extreme changing conditions, often involving heavy motion blur, specularities, noise, and lighting variations [18, 11, 1].

Germain *et al*. [5, 6] break from this paradigm by exhaustively matching features of sparse 3D points to all pixels of a dense feature map extracted from the query image. The corresponding 2D point is thereby selected as the location with maximum similarity. This *detection-by-description* scheme, which bears similarity with the training loss of recent keypoint detectors [4, 9], produces robust correspondences even in extreme conditions.

Sparse-to-dense hypercolumn matching (S2DHM) however can suffer from poor localization accuracy, since the resolution of the feature maps is limited by the high computational cost of the exhaustive matching, and an offset of a few pixels can shift the estimated pose by a few meters. S2DHM is at best capable of pixel-level accuracy, while keypoint detectors often perform sub-pixel refinement. In parallel, several works exploit feature-metric objectives for dense bundle-adjustment [19], dense image alignment [10], or semi-dense relocalization tracking [20], exploiting the powerful representations learned by deep networks.

Figure 1: **Feature matching accuracy.** Corresponding sparse-to-dense feature matches between reference and query images. The matched detections on the query image (red) are a few pixels off from the points detected in the reference image (blue). Our pose refinement moves the points closer to their ground truth reprojections (blue), resulting in more accurate pose.

Inspired by these approaches, we propose to exploit the global alignment of features to refine the pose, and thus the position of the sparse detections. We define the feature-metric (FME) error between reference and query pixel by taking the difference in the hypercolumns corresponding to these pixels. By optimizing the feature-metric error, we obtain sub-pixel detections consistent with an absolute pose. This improves the pose accuracy in difficult conditions, can use off-the-shelf dense deep features, and is fast. We improve the selection of layers used in [5], and apply our refinement procedure to these layers. This significantly improves the performance on the RobotCar dataset, especially for night-time localization.

2. Method

We now present our feature-metric pose refinement algorithm. In Fig. 1, the feature correspondences obtained using the original S2DHM pipeline between the reference image (left) and the query image (right) are depicted. Looking closely, we observe that the inital matches (red) in the

*equal contribution

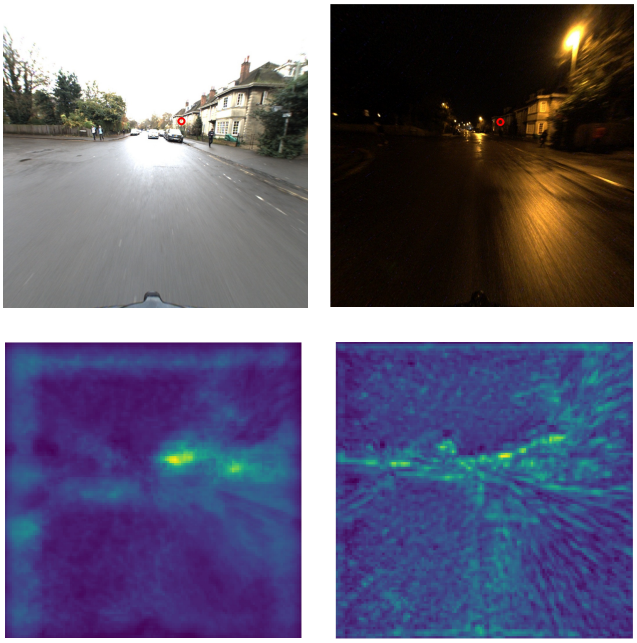

Figure 2: **Correlation map for hypercolumn matching**. First row depicts a reference (left) and a query image (right) with corresponding matches in red. Second row shows the correlation map from a reference feature to the dense query feature map. Correlation of the deeper layers of the network are shown left, while the right image shows correlations of the earlier layers of the network.

query image are a few pixels off from their corresponding reference matches (blue). This pixel-level error can result in large errors for the pose estimate [6]. In Fig. 2 we show the correlation map generated by sparse-to-dense feature matching. The bottom left image is a correlation map for the features from the deeper layers. These features provide larger radius of convergence, but have poor resolution. The bottom right image shows the feature correlation of the earlier layers of the network. It is clear that these features do not have large convergence radius, but they are useful in sub-pixel refinement of the pose due to strong gradients.

Next, we describe how we used feature gradients to do feature-metric PnP (FM-PnP) pose optimization.

2.1. Pose Refinement using Feature-Metric PnP

The proposed algorithm aims to refine the 6D camera pose of a query image with dense descriptors. The reference image can be a simple daytime image of the same scene where sparse or dense 2D-3D correspondences are available, e.g. from LIDAR or SfM.

Since the initial projection of the points in the query image needs to be within convergence radius of the feature-gradients, and in many cases those gradients are only smooth close to the optimal pose, a good initialization $\mathbf{T}_0$ is crucial. The pose obtained by running RANSAC+PnP on the S2DHM matches is often sufficiently accurate.

The feature-metric PnP algorithm can be summarized by the following steps:

1. initialize the query camera pose $\mathbf{T} = \mathbf{T}_0$;
2. project all 3D points onto the query image;
3. compute the feature-metric loss;
4. optimize the pose using Levenberg-Marquardt.

Given the query image I_q and the reference image I_r, as well as the corresponding query and reference feature maps $\mathbf{F}_q, \mathbf{F}_r \in \mathbb{R}^{W \times H \times C}$, we refine the camera pose $\mathbf{T}_q$ associated with the query image using feature-metric PnP optimization.

We define the feature-metric error (residual) between aligned query and reference locations originating from a 3D point $\mathbf{P}_i$ as:

$$\mathbf{r}_i(\mathbf{T}_q) = \mathbf{F}_q(\pi(\mathbf{P}_i, \mathbf{T}_q)) - \mathbf{F}_r(\mathbf{p}_i). \tag{1}$$

The function $\pi(\cdot, \cdot)$ projects the point onto the 2D query image space, given a pose, and $\mathbf{p}_i = \pi(\mathbf{P}_i, \mathbf{T}_r)$ is the projection of the 3D point onto the reference image. The total feature-metric loss can be written as follows:

$$L(\mathbf{T_q}) = \sum_i \rho(\mathbf{r}_i^T \mathbf{r}_i), \tag{2}$$

where $\rho(\cdot)$ is a suitable robust loss function, eg. the Barron [2] or Huber [7] loss.

The minimization of the total feature-metric loss L is performed using the Levenberg-Marquardt (LM) algorithm [12], which iteratively linearizes and solves for an optimal update $\mathbf{\Delta T}_q$ to the solution using:

$$\mathbf{\Delta T}_q = (\mathbf{J}^T\mathbf{J} + \lambda \operatorname{diag}(\mathbf{J}^T\mathbf{J}))^{-1}\mathbf{J}^T\mathbf{r}, \tag{3}$$

where $\mathbf{J}$ is the Jacobian of the loss with respect to the pose, $\mathbf{r}$ is the residual computed in the current iteration, and λ is the regularization strength.

2.2. Integration into S2DHM

We integrate our optimization pipeline with the S2DHM feature matching [5]. We notice that the features originally used are of low resolution, while sub-pixel optimization greatly benefits from higher resolution maps. For this purpose, we add new earlier layers from the same trained VGG network. Specifically, we use the layers `conv_2_2`, `conv_3_3`, `conv_4_1`, `conv_4_3`, `conv_5_1` of the trained VGG network from [5], for sparse-to-dense matching as well as feature-metric PnP optimization. We visualize the camera-pose change and feature movement after the pose optimization.

	Robotcar-Seasons						Extended CMU - Seasons								
	Day-All			Night-All			Urban			Suburban			Park		
Threshold [m]	0.25m	0.5m	5m	0.25m	0.5m	5m	0.25m	0.5m	5m	0.25m	0.5m	5m	0.25m	0.5m	5m
Threshold [°]	2°	5°	10°	2°	5°	10°	2°	5°	10°	2°	5°	10°	2°	5°	10°
S2DHM	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
S2DHM (repr.)	46.7	77.8	95.1	28.4	69.5	94.6	-	-	-	-	-	-	-	-	-
+ FM-PnP	47.7	77.0	93.5	29.6	**70.2**	94.6	-	-	-	-	-	-	-	-	-
S2DHM NL	50.7	79.5	95.2	33.0	68.6	94.6	63.3	82.2	95.9	55.7	78.0	94.7	50.6	**72.1**	88.1
+ FM-PnP	**52.8**	**80.0**	**95.2**	**34.4**	69.7	**94.6**	62.9	81.9	**95.9**	55.1	77.6	**94.7**	50.1	72.0	**88.1**

Table 1: **Localization results.** We report the percent of estimated poses below three thresholds (fine, medium, coarse) on different sequences of the Robotcar-Seasons [11] and Extended-CMU-Seasons [1] dataset. Blue highlights the (equally) best performance in each threshold. The first row is the data of S2DHM as reported by Germain et. al. [5], while the second row is a rerun of their code with our parameters explained in section 2.3. **S2DHM NL** refers to our run of S2DHM with features as described in section 2.2. Rows with **+ FM-PnP** show the localization accuracy after optimizing for the FME.

2.3. Implementation Details

We evaluate our approach on the RobotCar-Seasons [11] and Extended CMU-Seasons [1] dataset. We use the 3D pointclouds triangulated by Germain *et al.* [5] for both datasets. As a baseline implementation for image retrieval and feature extraction, we used the publicly available source code of Germain [5] and integrated our FM-PnP in there. For both datasets, we retrieve $N = 30$ images and we compute the initial pose with RANSAC+PnP, using a reprojection threshold of 12.0 pixels and pre-filtering the S2DHM matches with a ratio factor $f = 0.006$ [5]. The database image with the highest number of inliers and the corresponding estimated pose are selected for our FM-PnP refinement. If all image have fewer than 12 inlier matches, the estimated poses are deemed unstable and we instead select the first retrieved image and its own pose.

We perform our FM-PnP on the selected database image, using the 3D points labeled by RANSAC as inliers. We optimize for 50 iterations using the vanilla L2 loss and an adaptive damping factor $\lambda = 0.1$. The features of the 2D projections in the (1024×1024) images are bilinearly interpolated from the (256×256) dense feature maps. Feature gradients are obtained by applying Sobel filters, and are similarly bilinearly interpolated. Correspondences with query image pixel coordinates outside of the feature map are excluded at each iteration, but are reconsidered if they are back in the supported domain at the next iteration.

3. Experimental Results

Localization results. We benchmark our method against S2DHM [5], which achieved state-of-the-art results in localization for the night-query on the Robotcar-Seasons dataset. We use the evaluation procedure described in [18] on all sequences of both the datasets. In Table 1, the percentage of images below three pose thresholds are reported as a metric for pose estimation accuracy on the Robotcar-Seasons [11] and Extended CMU-Seasons [1] datasets. To the best of our knowledge, we achieve state-of-the-art results on the challenging nighttime scenario in the Robotcar-Seasons dataset. We also improve S2DHM on daytime query images in Robotcar. Our method contributes the biggest improvements in the lowest threshold, where the initial pose is close enough that our FME optimizer can converge and improve the pose.

Our method however decreases the accuracy in the CMU dataset on the lower thresholds. We observed that the majority of the drop comes from using different layers and a different ratio factor than Germain et. al. [5], which we optimized for the Robotcar-Seasons dataset. Still, FM-PnP decreases the accuracy in the lowest threshold by around 0.5%, although reducing the FME. The exact reasoning for that needs further investigation, but it suggests that a reduction of the FME does not always correlate with a better pose. Also see Fig. 1, where a few correspondences are further away from the reference pixel location after the optimization.

Pose change. In Table 2, we report the absolute pose- and FME change observed in both datasets. Despite the decrease in FME being rather small (relative change is around 2.5%), and correspondences only moving by a few pixels, the pose still changes by about $0.5m$, suggesting that we will mostly improve in the lower thresholds and supporting the reported improvements with FM-PnP in Table 1. In the Extended CMU-Seasons dataset, the FME reduction through FM-PnP is even smaller, and we observe worse results in terms of localization accuracy there. This hints that we converged into a local minima, which is potentially a worse estimation than the pose obtained from RANSAC+PnP.

	Pose change [m]		FME change [-]	
	median	95%	median	95%
Robotcar	0.449	2.216	6.401E-3	1.593E-2
Extended CMU	0.226	5.421	7.435E-4	1.00E-1

Table 2: **Observed pose and FME changes.** Median and 95%-Quantile of the absolute pose- and FME-change through FM-PnP in S2DHM with the new layers.

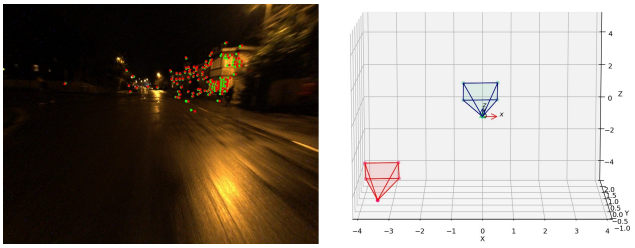

Figure 3: **Pose update and feature points.** A small movement of the query points, in the left image, results in a translation of more than 5 meters between the initial (blue) and the final (red) camera poses, as shown in the right image.

As we can see in Fig. 1, after the optimization most of the keypoints are moving towards the ground truth reference points (from red to green) resulting in a more accurate pose estimate. In Fig. 3 we also observe that even with a small movement of the matched points (left), camera movements are very large. Still, some points in Fig. 1 are moving away from the ground truth reference points. This can be the case where the 3D structure associated to the reference image is not very accurate.

Generalization. To show the applicability of our method on other features, we used the dense descriptors from D2-Net [4] within the S2DHM framework and optimized the poses for the feature-metric error there. Table 3 shows that our proposed optimization also improves the localization accuracy despite not being tuned for these features.

Feature matches. Fig. 4 shows the inlier correspondences between a reference (day) and a query (night) image from the RobotCar-Seasons dataset with a 2px reprojection error threshold. The matches in the top image are obtained from RANSAC+PnP after S2DHM. The improved final pose estimate through our FME-optimization increases the number of inlier correspondences.

	Robotcar-Seasons					
	Day-All			Night-All		
Threshold [m]	0.25m	0.5m	5m	0.25m	0.5m	5m
Threshold [°]	2°	5°	10°	2°	5°	10°
D2-Net	39.8	73.5	94.9	14.5	47.0	89.4
+ FM-PnP	**41.1**	**74.1**	**94.9**	**14.8**	**47.0**	**89.4**

Table 3: **Optimization on D2-Net descriptors.** Reported percent of estimated poses below three localization thresholds (fine, medium, coarse) before and after using FM-PnP on the D2-Net [4] features. Blue highlights the (equally) best performance in each threshold.

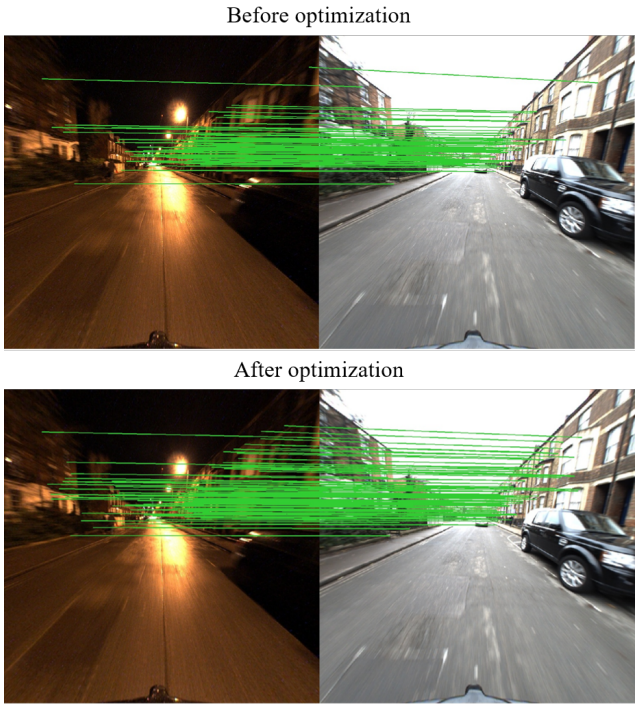

Figure 4: **Feature matches.** Inlier correspondences before and after FM-PnP. We obtain more correspondences after the optimization.

4. Conclusion

We show that using feature-metric error optimization, we can improve the pose estimates. In the future, integrating this optimization into a robust pose-estimation pipeline should result in even more accurate poses. Also, if we could train a network in a way that the obtained dense descriptors have a more prominent gradient towards the FME minimas, we could increase the radius of convergence for our method, so that FM-PnP is less impacted by wrong initial poses.

References

[1] H Badino, D Huber, and T Kanade. The CMU visual localization data set. *Computer Vision Group*, 2011. 1, 3

[2] Jonathan T Barron. A general and adaptive robust loss function. In *CVPR*, 2019. 2

[3] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *CVPR Workshop on Deep Learning for Visual SLAM*, 2018. 1

[4] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable CNN for joint detection and description of local features. In *CVPR*, 2019. 1, 4

[5] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. Sparse-to-dense hypercolumn matching for long-term visual localization. In *3DV*, 2019. 1, 2, 3

[6] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2DNet: Learning accurate correspondences for sparse-to-dense feature matching. *arXiv:2004.01673*, 2020. 1, 2

[7] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992. 2

[8] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1

[9] Zixin Luo, Lei Zhou, Xuyang Bai, Hongkai Chen, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. ASLFeat: Learning local features of accurate shape and localization. *CVPR*, 2020. 1

[10] Zhaoyang Lv, Frank Dellaert, James Rehg, and Andreas Geiger. Taking a deeper look at the inverse compositional algorithm. In *CVPR*, 2019. 1

[11] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017. 1, 3

[12] Jorge Nocedal and Stephen Wright. *Numerical optimization*. Springer Science & Business Media, 2006. 2

[13] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: Learning local features from images. In *NeurIPS*, 2018. 1

[14] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2D2: Repeatable and reliable detector and descriptor. In *NeurIPS*, 2019. 1

[15] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 1

[16] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020. 1

[17] Torsten Sattler, Bastian Leibe, and Leif Kobbelt. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE PAMI*, 39(9):1744–1756, 2017. 1

[18] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, et al. Benchmarking 6DOF outdoor visual localization in changing conditions. In *CVPR*, 2018. 1, 3

[19] Chengzhou Tang and Ping Tan. BA-Net: Dense bundle adjustment network. In *ICLR*, 2019. 1

[20] Lukas von Stumberg, Patrick Wenzel, Qadeer Khan, and Daniel Cremers. GN-Net: The Gauss-Newton loss for multi-weather relocalization. *RA-L and ICRA*, 5(2):890–897, 2020. 1