Patch2Pix: Epipolar-Guided Pixel-Level Correspondences

Qunjie Zhou[1] Torsten Sattler[2] Laura Leal-Taixé[1]
[1]Technical University of Munich [2]CIIRC, Czech Technical University in Prague *

Abstract

The classical matching pipeline used for visual localization typically involves three steps: (i) local feature detection and description, (ii) feature matching, and (iii) outlier rejection. Recently emerged correspondence networks propose to perform those steps inside a single network but suffer from low matching resolution due to the memory bottleneck. In this work, we propose a new perspective to estimate correspondences in a detect-to-refine manner, where we first predict patch-level match proposals and then refine them. We present Patch2Pix, a novel refinement network that refines match proposals by regressing pixel-level matches from the local regions defined by those proposals and jointly rejecting outlier matches with confidence scores. Patch2Pix is weakly supervised to learn correspondences that are consistent with the epipolar geometry of an input image pair. We show that our refinement network significantly improves the performance of correspondence networks on image matching, homography estimation, and localization tasks. In addition, we show that our learned refinement generalizes to fully-supervised methods without re-training, which leads us to state-of-the-art localization performance. The code is available at https://github.com/GrumpyZhou/patch2pix.

1. Introduction

Finding image correspondences is a fundamental step in several computer vision tasks such as Structure-from-Motion (SfM) [36, 41] and Simultaneous Localization and Mapping (SLAM) [8, 24]. Given a pair of images, pixel-level correspondences are commonly established through a local feature matching pipeline, which involves the following three steps: i) detecting and describing local features, ii) matching the nearest neighbors using the feature descriptors, and iii) rejecting outlier matches.

Traditional hand-crafted local features such as SIFT [15] or SURF [2] are vulnerable to extreme illumination changes, motion blur and repetitive and weakly textured scenes. Therefore, recent works [5–7, 16, 17, 28, 40] propose to learn to detect and describe local features using neural networks, showing that learned features can be robustly matched under challenging conditions [6, 17, 28, 40]. Instead of focusing on improving local features, [3, 22, 38, 42] suggest to learn a filtering function from sets of correspondences to reject outlier matches. A recent method [33] further proposes to jointly learn the matching function and outlier rejection via graph neural networks and the Sinkhorn algorithm [4, 37]. Combining a learned feature [5] and learned matcher [33] has set the state-of-the-art results on several geometry tasks, showing a promising direction towards a full learnable matching pipeline.

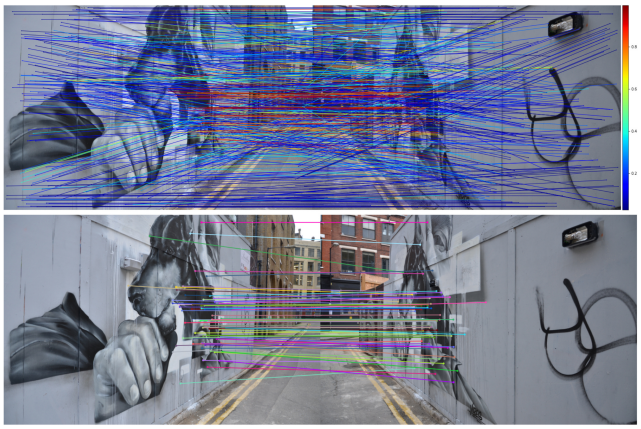

Figure 1. **An example of *Patch2Pix* correspondences.** In the top figure, the matches refined by *Patch2Pix* are coloured according to the predicted confidence scores. The less confident matches (in blue) appear mostly on the road or the blank wall. In the bottom figure, we show that the inlier matches can well handle the large viewpoint change. We show more quantitative results for handling various challenging conditions in the supp. mat (*c.f.* Sec. D).

Learning the whole matching pipeline has already been investigated in several works [13, 30, 31], where a single network directly outputs correspondences from an input image pair. The main challenge faced with those correspondence networks is how to efficiently perform matching while reaching pixel-level accuracy. In order to keep computation speed and memory footprint manageable, [29] has

*This research was funded by the Humboldt Foundation through the Sofja Kovalevskaya Award, the EU Horizon 2020 project RICAIP (grant agreeement No. 857306), and the European Regional Development Fund under project IMPACT (No. CZ.02.1.01/0.0/0.0/15 003/0000468).

to match at a rather low resolution, which is shown to be less accurate in relative pose estimation [43]. While sparse convolutions have been applied in [30] to match at higher resolution, they still do not achieve pixel-level matching. One advantage of the correspondences networks [30, 31] is that they are weakly supervised to maximize the average matching score for a matching pair and minimize it for a non-matching pair, however, they learn less effectively in pixel-level matching. This is in contrast to methods that require full supervision from ground truth (GT) correspondences [5, 6, 10, 17, 28, 33]. While the GT correspondences provide very precise signals for training, they might also add bias to the learning process. For example, using the sparse keypoints generated by an SfM pipeline with a specific detector as supervision, a keypoint detector might simply learn to replicate these detections rather than learning more general features [26]. To avoid such type of bias in the supervision, a recent work [40] proposes to use relative camera poses as weak supervision to learn local feature descriptors. Compared to the mean matching score loss used in [30, 31], they are more precise by containing the geometrical relations between the images pairs.

In this paper, we propose *Patch2Pix*, a new view for the design of correspondence networks. Inspired by the successful *detect-to-refine* practice in the object detection community [27], our network first obtains patch-level match proposals and then refines them to pixel-level matches. See an example of our matches in Fig. 1. Our novel refinement network is weakly supervised by epipolar geometry computed from relative camera poses, which are used to regress geometrically consistent pixel-wise matches within the patch proposal. Compared to [40], we optimize directly on match locations to learn matching, while they optimize through matching scores to learn feature descriptors. Our method is extensively evaluated on a set of geometry tasks, showing state-of-the-art results. We summarize our **contributions** as: i) We present a novel view for finding correspondences, where we first obtain patch-level match proposals and then refine them to pixel-level matches. ii) We develop a novel match refinement network that jointly refines the matches via regression and rejects outlier proposals. It is trained without the need for pixel-wise GT correspondences. iii) We show that our model consistently improves match accuracy of correspondence networks for image matching, homography estimation and visual localization. iv) Our model generalizes to fully supervised methods without the need for retraining, and achieves state-of-the-art results on indoor and outdoor long-term localization.

2. Related Work

Researchers have recently opted for leveraging deep learning to detect robust and discriminative local features [5–7, 17, 28, 40]. D2Net [6] detects keypoints by finding local maxima on CNN features at a 4-times lower resolution w.r.t. the input images, resulting in less accurate detections. Based on D2Net, ASLFeat [17] uses deformable convolutional networks and extracts feature maps at multiple levels to obtain pixel-level matches. R2D2 [28] uses dilated convolutions to preserve image resolution and predicts per-pixel keypoints and descriptors, which gains accuracy at the cost of computation and memory usage. Given the keypoints, CAPS [40] fuses features at several resolutions and obtains per-pixel descriptors by interpolation. The above methods are designed to learn local features and require a further matching step to predict the correspondences.

Matching and Outlier Rejection. Once local features are detected and described, correspondences can be obtained using Nearest Neighbor (NN) search [23] based on the Euclidean distance between the two feature representations. Outliers are normally filtered based on mutual consistency or matching scores. From a set of correspondences obtained by NN search, recent works [3, 22, 38, 42] learn networks to predict binary labels to identify outliers [22, 38, 42], or probabilities that can be used by RANSAC [9] to weight the input matches [3]. Notice, those methods do not learn the local features for matching and the matching function itself, thus they can only improve within the given set of correspondences. Recent works further propose to learn the whole matching function [10, 33]. SuperGlue [33] learns to improve SuperPoint [5] descriptors for matching using a graph neural network with attention and computes the correspondences using the Sinkhorn algorithm [4, 37]. S2DNet [10] extracts sparse features at SuperPoint keypoint locations for one image and matches them exhaustively to the dense features extracted for the other image to compute correspondences based on the peakness of similarity scores. While those methods optimize feature descriptors at keypoint locations specifically for the matching process, they do not solve the keypoint detection problem.

End-to-End Matching. Instead of solving feature detection, feature matching, and outlier rejection separately, recently correspondences networks [13, 30, 31] have emerged to accomplish all steps inside a single forward pass. NCNet uses a correlation layer [29] to perform the matching operation inside a network and further improves the matching scores by leveraging a neighborhood consistency score, which is obtained by a 4D convolution layer. Limited by the available memory, NCNet computes the correlation scores on feature maps with 16-times downscaled resolution, which has been proven not accurate enough for camera pose estimation [43]. SparseNCNet [30] uses a sparse representation of the correlation tensor by storing the top-10 similarity scores and replace dense 4D convolution with sparse convolutions. This allows SparseNCNet to obtain matches at 4-times downscaled resolution w.r.t. the origi-

nal image. DualRC-Net [13], developed concurrently with our approach, outperforms SparseNCNet by combining the matching scores obtained from coarse-resolution and fine-resolution feature maps. Instead of refining the matching scores as in [13, 30], we use regression layers to refine the match locations at image resolution.

Full versus Weak Supervision. We consider methods that require information about exact correspondences to compute their loss function as fully supervised and those that do not need GT correspondences as weakly supervised. Most local feature detectors and descriptors are trained on exact correspondences either calculated using camera poses and depth maps [6, 10, 17] or using synthetic homography transformations [5, 28], except for CAPS [40] using epipolar geometry as weak supervision. Both S2DNet [10] and SuperGlue [33] requires GT correspondences to learn feature description and matching. Outlier filtering methods [3,22,38,42] are normally weakly supervised by the geometry transformations between the pair. DualRC-Net [13] is also fully supervised on exact correspondences, while the other two correspondence networks [30, 31] are weakly-supervised to optimize the mean matching score on the level of image pairs instead of individual matches. We use epipolar geometry as weak supervision to learn geometrically consistent correspondences where the coordinates of matches are directly regressed and optimize. In contrast, CAPS [40] uses the same level of supervision to learn feature descriptors and their loss optimizes through the matching scores whose indices give the match locations. We propose our two-stage matching network, based on the concept of learned correspondences [30,31], which learns to predict geometrically consistent matches at image resolution.

3. Patch2Pix: Match Refinement Network

A benefit of correspondence networks is the potential to optimize the network directly for the feature matching objective without the need for explicitly defining keypoints. The feature detection and description are implicitly performed by the network and reflected in the found correspondences. However, there are two main issues causing the inaccuracy of the existing correspondence networks [30, 31]: i) the use of downscaled feature maps due to the memory bottleneck constrained by the size of the correlation map. This leads to every match being uncertain within two local patches. ii) Both NCNet [31] and SparseNCNet [30] have been trained with a weakly supervised loss which simply gives low scores for all matches of a non-matching pair and high scores for matches of a matching pair. This does not help identify good or bad matches, making the method unsuitable to locate pixel-accurate correspondences.

In order to fix those two sources of inaccuracies, we propose to perform matching in a two-stage *detect-to-refine* manner, which is inspired by two-step object detectors such as Faster R-CNN [27]. In the first correspondence detection stage, we adopt a correspondence network, *e.g.*, NC-Net, to predict a set of patch-level match proposals. As in Faster R-CNN, our second stage refines a match proposal in two ways: (i) using classification to identify whether a proposal is confident or not, and (ii) using regression to detect a match at pixel resolution within the local patches centered by the proposed match. Our intuition is that the correspondence network uses the high-level features to predict semantic matches at a patch-level, while our refinement network can focus on the details of the local structure to define more accurate locations for the correspondences. Finally, our network is trained with our weakly-supervised epipolar loss which enforces our matches to fulfill this geometric constraint defined by the relative camera pose. We name our network *Patch2Pix* since it predicts pixel-level matches from local patches, and the overview of the network architecture is depicted in Fig. 2. In the following, we take NC-Net as our baseline to obtain match proposals, yet we are not limited to correspondence networks to perform the match detection. We show later in our experiments that our refinement network also generalizes to other types of matching methods (*c.f.* Sec. 5.3 & 5.4). The following sections detail its architecture and training losses.

3.1. Refinement: Pixel-level Matching

Feature Extraction. Given a pair of images (I_A, I_B), a CNN backbone with L layers extracts the feature maps from each image. We consider $\{f_1^A\}_{l=0}^L$ and $\{f_l^B\}_{l=0}^L$ to be the activation maps at layer l for images I_A and I_B, respectively. At the layer index $l = 0$, the feature map is the input image itself, *i.e.*, $f_0^A = I_A$ and $f_0^B = I_B$. For an image with spatial resolution $H \times W$, the spatial dimension of feature map f_l is $H/2^l \times W/2^l$ for $l \in [0, L-1]$. For the last layer, we set the convolution stride as 1 to prevent losing too much resolution. The feature maps are extracted once and used in both the correspondence detection and refinement stages. The detection stage uses only the last layer features which contain more high-level information, while the refinement stage uses the features before the last layer, which contain more low-level details.

From match proposals to patches. Given a match proposal $m_i = (p_i^A, p_i^B) = (x_i^A, y_i^A, x_i^B, y_i^B)$, the goal of our refinement stage is to find accurate matches on the pixel level by searching for a pixel-wise match inside local regions. As the proposals were matched on a downscaled feature map, an error by one pixel in the feature map leads to inaccuracy of 2^{L-1} pixels in the images. Therefore, we define the search region as the $S \times S$ local patches centered at p_i^A and p_i^B, where we consider $S > 2^{L-1}$ to cover a larger region than the original $2^{L-1} \times 2^{L-1}$ local patches. Once

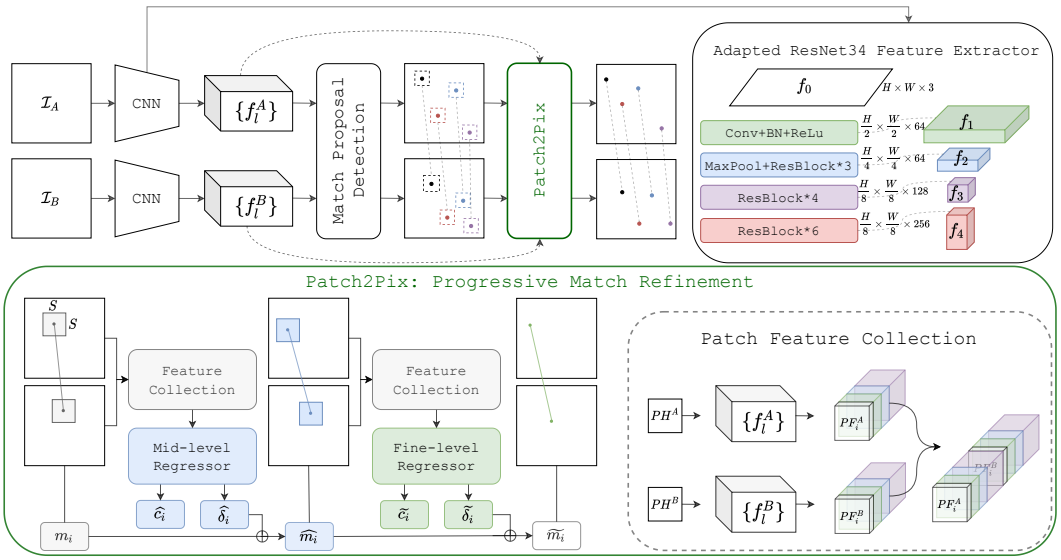

Figure 2. **Correspondence Refinement with *Patch2Pix*.** *Top:* For a pair of images, features are first extracted using our adapted ResNet34 backbone and fed into a correspondence network, *e.g.*, NC matching layer [31], to detect match proposals. Those proposals are then refined by *Patch2Pix*, which re-uses the extracted feature maps. *Bottom:* We design two levels of regressors with the same architecture to progressively refine the match proposals at image resolution. For a pair of $S \times S$ local patches centered at a match proposal m_i, the features of the patches are collected as the input to our mid-level regressor to output (i) a confidence score $\widehat{c}_i$ which indicates the quality of the match proposal and (ii) a pixel-level local match $\widehat{\delta}_i$ found within the local patches. The updated match proposal $\widehat{m_i}$ updates the search space accordingly through a new pair of local patches. The fine-level regressor outputs the final confidence score $\widetilde{c}_i$ and $\widetilde{\delta}_i$ to obtained the final pixel-accurate match $\widetilde{m_i}$. The whole network is trained under weak supervision without the need for explicit GT correspondences.

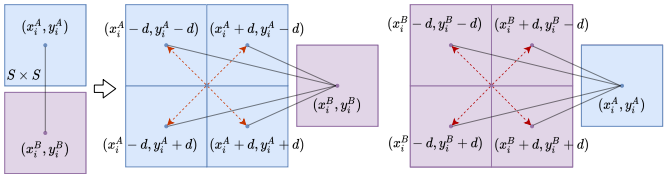

Figure 3. **Patch Expansion.** Given a match proposal $p_i^A = (x_i^A, y_i^A)$ and $p_i^B = (x_i^B, y_i^B)$, we move p_i^A towards its four corners by moving along the x- and y-axes by d pixels, which are matched to p_i^B to compose 4 new match proposals. Repeating it also from p_i^B to p_i^A, leads to 8 match proposals in total, which allows us to search in two $2S \times 2S$ local regions, compared to the original $S \times S$ patches.

we obtain a set of local patch pairs for all match proposals, the pixel-level matches are regressed by our network from the feature maps of the local patch pairs. We describe each component in detail below.

Local Patch Expansion. We further propose a patch expansion mechanism to expand the search region by including the neighboring regions, as illustrated in Fig. 3. We first move p_i^A towards its four corners along the x- and y-axes, each by d pixels. This gives us four anchor points for p_i^A that we match to p_i^B to compose four new match proposals. Similarly, we also expand p_i^B to get its four corner anchors and match them to p_i^A, giving us another four new match proposals. In the end, the expanded eight proposals identify eight pairs of $S \times S$ local patches. We set $d = S/2$ pixels so that the expanded search region defined by the expanded patches has size $2S \times 2S$ and still covers the original $S \times S$ searching space. The patch expansion to the patch proposals M_{patch} is especially useful during training since the network is forced to identify the correct proposal among spatially close and similar features.We show in the supp. mat (Sec. B) that our expansion mechanism can speed up the learning process and also improves the model performance. While one can also apply it during the inference to increase the search region, it will lead to a higher computation overhead. We thus refrain from using it during testing.

Progressive Match Regression. In order to locate pixel-level matches, we define the refinement task as finding a good match inside the pair of local patches. We achieve this using two regressors with the same architecture, *i.e.*, the mid-level and the fine-level regressor, to progressively identify the final match, which is shown in the lower part of Fig. 2. Given a pair of $S \times S$ patches, we first collect the corresponding feature information from previously extracted activation maps, *i.e.*, $\{f_l^A\}, \{f_l^B\}$. For every point location (x, y) on the patch, its corresponding location on the l-layer feature map is $(x/2^l, y/2^l)$. We select all features from the layers $\{0, \ldots, L-1\}$ and concatenate them into a single feature vector. The two gathered feature patches PF_i^A and PF_i^B are concatenated along the feature dimension and fed into our mid-level regressor. The regressor first aggregates the input features with two convolutional layers into a compact feature vector, which is then pro-

cessed by two fully connected (fc) layers, and finally outputs our network predictions from two heads implemented as two fc layers. The first head is a regression head, which outputs a set of local matches $\widehat{M}_{\Delta} := \{\widehat{\delta}_i\}_{i=1}^{N} \subset R^4$ inside the $S \times S$ local patches w.r.t. their center pixels, where $\widehat{\delta}_i = (\widehat{\delta x_i^A}, \widehat{\delta y_i^A}, \widehat{\delta x_i^B}, \widehat{\delta y_i^B})$. In the second head, *i.e.*, the classification head, we apply a sigmoid function to the outputs of the fc layer to obtain the confidence scores $\widehat{\mathcal{C}}_{pixel} = (\widehat{c_1}, \ldots, \widehat{c_N}) \in R^N$, which express the validity of the detected matches. This allows us to detect and discard bad match proposals that cannot deliver a good pixel-wise match. We obtain the mid-level matches $\widehat{M}_{pixel} := \{\widehat{m_i}\}_{i=1}^{N}$ by adding the local matches to patch matches, *i.e.*, $\widehat{m_i} = m_i + \widehat{\delta}_i$. Features are collected again for the new set of local $S \times S$ patch pairs centered by the mid-level matches and fed into the fine-level regressor, which follows the same procedure as the mid-level regression to output the final pixel-level matches $\widetilde{M}_{pixel} := \{\widetilde{m_i}\}_{i=1}^{N}$ and the confidence scores $\widetilde{\mathcal{C}}_{pixel} = (\widetilde{c_1}, \ldots, \widetilde{c_N}) \in R^N$.

3.2. Losses

Our pixel-level matching loss $\mathcal{L}_{pixel}$ involves two terms: (i) a classification loss $\mathcal{L}_{cls}$ for the confidence scores, trained to predict whether a match proposal contains a true match or not, and (ii) a geometric loss $\mathcal{L}_{geo}$ to judge the accuracy of the regressed matches. The final loss is defined as $\mathcal{L}_{pixel} = \alpha\mathcal{L}_{cls} + \mathcal{L}_{geo}$, where α is a weighting parameter to balance the two losses. We empirically set $\alpha = 10$ based on the magnitude of the two losses during training.

Sampson distance. To identify pixel-level matches, we supervise the network to find correspondences that agree with the epipolar geometry between an image pair. It defines that the two correctly matched points should lie on their corresponding epipolar lines when being projected to the other image using the relative camera pose transformation. How much a match prediction fulfills the epipolar geometry can be precisely measured by the Sampson distance. Given a match m_i and the fundamental matrix $F \in R^{3\times3}$ computed by the relative camera pose of the image pair, its Sampson distance ϕ_i measures the geometric error of the match w.r.t. the fundamental matrix [11], which is defined as:

$$\phi_i = \frac{((P_i^B)^T F P_i^A)^2}{(FP_i^A)_1^2 + (FP_i^A)_2^2 + (F^T P_i^B)_1^2 + (F^T P_i^B)_2^2}, \quad (1)$$

where $P_i^A = (x_i^A, y_i^A, 1)^T, P_i^B = (x_i^B, y_i^B, 1)^T$ and $(FP_i^A)_k^2, (FP_i^B)_k^2$ represent the square of the k-th entry of the vector FP_i^A, FP_i^B.

Classification loss. Given a pair of patches obtained from a match proposal $m_i = (x_i^A, y_i^A, x_i^B, y_i^B)$, we label the pair as positive, hence define its classification label as $c_i^* = 1$, if $\phi_i < \theta_{cls}$. Here, θ_{cls} is our geometric distance threshold for classification. All the others pairs are labeled as negative. Given the set of predicted confidence scores $\mathcal{C}$ and the binary labels $\mathcal{C}^*$, we use the weighted binary cross entropy to measure the classification loss as

$$\mathcal{B}(\mathcal{C}, \mathcal{C}^*) = -\frac{1}{N}\sum_{i=1}^{N} w c_i^* \, log\, c_i + (1 - c_i^*)\, log\,(1 - c_i)\,, \quad (2)$$

where the weight $w = |\{c_i^* | c_i^* = 0\}| / |\{c_i^* | c_i^* = 1\}|$ is the factor to balance the amount of positive and negative patch pairs. We have separate thresholds $\widehat{\theta}_{cls}$ and $\widetilde{\theta}_{cls}$ used in the mid-level and the fine-level classification loss, which are summed to get the total classification loss $\mathcal{L}_{cls}$.

Geometric loss. To avoid training our regressors to refine matches within match proposals which are going to be classified as non-valid, for every refined match, we optimize its geometric loss only if the Sampson distance of its parent match proposal is within a certain threshold θ_{geo}. Our geometric loss is the average Sampson distance of the set of refined matches that we want to optimize. We use thresholds $\widehat{\theta}_{geo}$ and $\widetilde{\theta}_{geo}$ for the mid-level and the fine-level geometric loss accordingly and the sum of the two losses gives the total geometric loss $\mathcal{L}_{geo}$.

4. Implementation Details

We train *Patch2Pix* with match proposals detected by our adapted NCNet, *i.e.*, the pre-trained NC matching layer from [31], to match features extracted from our backbone. Our refinement network is trained on the large-scale outdoor dataset MegaDepth [14], where we construct 60661 matching pairs. We set the distance thresholds to compute the training losses (*c.f.* Sec. 3.2) as $\widehat{\theta}_{cls} = \widehat{\theta}_{geo} = 50$ for the mid-level regression and $\widetilde{\theta}_{cls} = \widetilde{\theta}_{geo} = 5$ for the fine-level regression. We constantly set the local patch size to $S = 16$ pixels at image resolution. The pixel-level matching is optimized using Adam [12] with an initial learning rate of $5e^{-4}$ for 5 epochs and then $1e^{-4}$ until it converges. A mini-batch input contains 4 pairs of images with resolution 480×320. We present architecture details about our regressor and our adapted NCNet [31], training data processing, hyper-parameter ablation, and qualitatively results of our matches in the supp. mat. (*c.f.* Sec. A & B).

5. Evaluation on Geometrical Tasks

5.1. Image Matching

As our first experiment, we evaluate *Patch2Pix* on the HPatches [1] sequences under the image matching task, where a method is supposed to detect correspondences between an input image pair. We follow the setup proposed in D2Net [6] and report the mean matching accuracy (MMA) [19] under thresholds varying from 1 to 10 pixels, together with the numbers of matches and features.

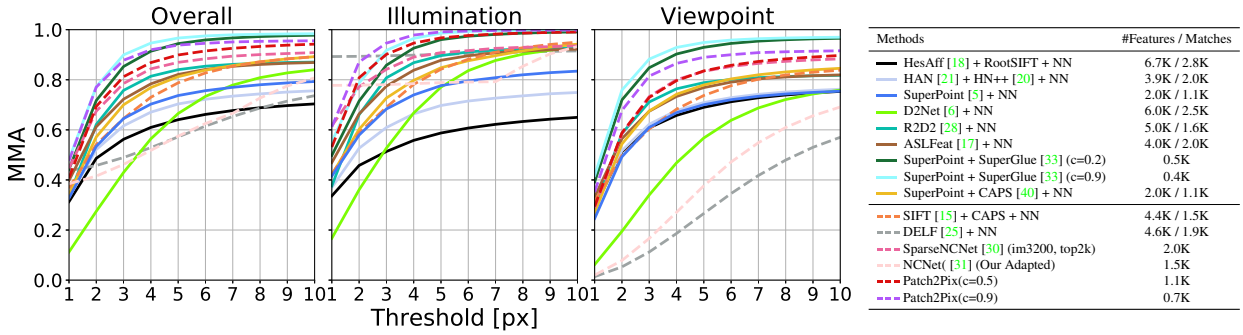

Figure 4. **Image Matching on HPatches [1].** We denote weakly-supervised methods with dashed lines and methods based on full supervision with solid lines.

Experimental setup. We use the confidence scores produced by the fine-level regressor to filter out outliers and study its performance under two settings, *i.e.*, $c = 0.5/0.9$, which present a trade-off between quantity and quality of the matches. To show the effectiveness of our refinement concept, we compare to our NCNet baseline, which provides our match proposals. For NCNet and *Patch2Pix*, we resize images to have a larger side of 1024 to reduce runtime. We also compare to SparseNCNet [30], which is the most similar one to ours among related works, since it also builds upon NCNet and aims to improve the accuracy of its matches through a re-localization mechanism. Besides comparing to several local feature methods that use NN Search for matching, we further consider SuperPoint [5] features matched with SuperGlue [33] and study its performance under their default threshold $c = 0.2$ and a higher threshold $c = 0.9$ for outlier rejection.

Results. As shown in Fig. 4, NCNet performs competitively for illumination sequences with constant viewpoints, which is a special case for NCNet since it uses fixed upsampling to bring patch matches to pixel correspondences. While its performance under illumination changes reveals its efficiency in patch-level matching, its accuracy under viewpoint changes reveals its insufficient pixel-level matching performance. Our refinement network brings patch-level matches predicted by NCNet to pixel-level correspondences, which drastically improves the matching accuracy under viewpoint changes and further improves under illumination changes. When comparing *Patch2Pix* to all weakly supervised methods, our model is the best at both thresholds under illumination changes. For viewpoint changes, our model with threshold $c = 0.9$ is the best and SparseNCNet performs similar to our model under threshold $c = 0.5$. Compared to the methods trained with full supervision, our model with threshold $c = 0.9$ outperforms all of them under illumination variations. For viewpoint changes, we are less accurate than SuperPoint + SuperGlue but still, we outperform all the other fully-supervised methods. Looking at the curves and the table in Fig. 4 together, both SuperPoint + SuperGlue and our method improve performance when using a higher threshold to remove less confident predictions.

5.2. Homography Estimation

Having accurate matches does not necessarily mean accurate geometry relations can be estimated from them since the distribution and number of matches are also important when estimating geometric relations. Therefore, we next evaluate *Patch2Pix* on the same HPatches [1] sequences for homography estimation.

Experimental setup. We follow the corner correctness metric used in [5, 33, 40] and report the percentage of correctly estimated homographies whose average corner error distance is below 1/3/5 pixels. In the following experiments, where geometrics relations are estimated using RANSAC-based solvers, we use $c = 0.25$ as our default confidence threshold, which overall gives us good performance across tasks. The intuition of setting a lower threshold is to filter out some very bad matches but leave as much information as possible for RANSAC to do its own outlier rejection. We compare to methods that are more competitive in the matching task which are categorized based on their supervision types: fully supervised (Full), weakly supervised (Weak), and mixed (Mix) if both types are used. We run all methods under our environment and measure the matching time from the input images to the output matches. We provide more experimental setup details in our supp. mat (*c.f.* Sec. C).

Results. From the results shown in Tab. 1, we observe again that NCNet performs extremely well under illumination changes due to their fixed upsampling (*c.f.* Sec. 5.2). Here, we verify that the improvement of matches by *Patch2Pix* under viewpoint changes is also reflected in the quality of the estimated homographies. Both SparseNCNet and our method are based on the concept of improving match accuracy by searching inside the matched local patches to progressively re-locate a more accurate match in higher resolution feature maps. While our method predicts matches at the original resolution and is fully learnable, their non-learning approach produces matches at a 4-times downscaled resolution. As we show in Tab. 1, our refinement network is more powerful than their re-localization

Method	Overall	Illumination	Viewpoint	Supervision	#Matches	Time (s)
	Accuracy (%, $\epsilon < 1/3/5$ px)					
SuperPoint [5] + NN	0.46 / 0.78 / 0.85	0.57 / 0.92 / 0.97	0.35 / 0.65 / 0.74	Full	1.1K	0.12
D2Net [6] + NN	0.38 / 0.72 / 0.81	0.65 / 0.95 / **0.98**	0.13 / 0.51 / 0.65	Full	2.5K	1.61
R2D2 [28] + NN	0.47 / 0.78 / 0.83	0.63 / 0.93 / **0.98**	0.33 / 0.64 / 0.70	Full	1.6K	2.34
ASLFeat [17] + NN	0.48 / 0.81 / 0.88	0.63 / 0.94 / **0.98**	0.34 / 0.69 / 0.78	Full	2.0K	0.66
SuperPoint + SuperGlue [33]	**0.51 / 0.83 / 0.89**	0.62 / 0.93 / **0.98**	**0.41/ 0.73/ 0.81**	Full	0.5K	0.14
SuperPoint + CAPS [40] + NN	0.49 / 0.79 / 0.86	0.62 / 0.93 / 0.98	0.36 / 0.65 / 0.75	Mix	1.1K	0.36
SIFT + CAPS [40] + NN	0.36 / 0.76 / 0.85	0.48 / 0.89 / 0.95	0.26 / 0.65 / 0.76	Weak	1.5K	0.73
SparseNCNet [30] (im3200, top2k)	0.36 / 0.66 / 0.76	0.62 / 0.92 / 0.97	0.13 / 0.41 / 0.57	Weak	2.0K	5.83
NCNet [31] (Our Adapted)	0.48 / 0.61 / 0.71	**0.98 / 0.98 / 0.98**	0.02 / 0.28 / 0.46	Weak	1.5K	0.83
Patch2Pix	**0.51** / 0.79 / 0.86	0.72 / 0.95 / **0.98**	0.32 / 0.64 / 0.75	Weak	1.3K	1.24
Oracle	0.00 / 0.15 / 0.54	0.00 / 0.23 / 0.7	0.00 / 0.07 / 0.39	-	2.5K	0.04
Patch2Pix (w.Oracle)	0.55 / 0.85 / 0.92	0.68 / 0.95 / 0.99	0.43 / 0.76 / 0.82	Weak	2.5K	0.76

Table 1. **Homography Estimation on Hpatches [1].** We report the percentage of correctly estimated homographies whose average corner error distance is below 1/3/5 pixels. We denote the supervision type with 'Full' for fully-supervised methods, 'Weak' for weakly-supervised ones, and 'Mix' for those used both types. We mark the best accuracy in **bold**.

mechanism, improving the overall accuracy within 1 pixel by 15 percent. For illumination changes, we are the second-best after NCNet, but we are better than all fully supervised methods. Under viewpoint variations, we are the best at 1-pixel error among weakly-supervised methods and we achieve very close overall accuracy to the best fully supervised method SuperPoint + SuperGlue.

Oracle Investigation. Since our method can filter out bad proposals but not generate new ones, our performance will suffer if NCNet fails to produce enough valid proposals, which might be the reason for our relatively lower performance on viewpoint changes. In order to test our hypothesis, we replace NCNet with an Oracle matcher to predict match proposals. Given a pair of images, our Oracle first random selects 2.5K matches from the GT correspondences computed using the GT homography and then randomly moves each point involved in a match within the 12×12 local patch centered at the GT location. In this way, we obtain our synthetic match proposals where we know there exists at least one GT correspondence inside the 16×16 local patches centered by those match proposals, which allows us to measure the performance of our true contribution, the refinement network. As shown in Tab. 1, the low accuracy of matches produced by our Oracle evidently verifies that the matching task left for our refinement network is still challenging. Our results are largely improved by using the Oracle proposals, which means our current refinement network is heavily limited by the performance of NCNet. Therefore, in the following localization experiments, to see the potential of our refinement network, we will also investigate the performance when using SuperPoint + SuperGlue to generate match proposals.

5.3. Outdoor Localization on Aachen Day-Night

We further show the potential of our approach by evaluating *Patch2Pix* on the Aachen Day-Night benchmark (v1.0) [34, 35] for outdoor localization under day-night illumination changes.

Method	Supervision	Localized Queries (%, 0.25m,2°/0.5m,5°/1.0m, 10°) Day	Night
Local Feature Evaluation on Night-time Queries			
SuperPoint [5] + NN	Full	-	73.5 / 79.6 / 88.8
D2Net [6] + NN	Full	-	74.5 / 86.7 / **100.0**
R2D2 [28] + NN	Full	-	76.5 / **90.8** / **100.0**
SuperPoint + S2DNet [10]	Full	-	74.5 / 84.7 / **100.0**
ASLFeat [17] + NN	Full	-	77.6 / 89.8 / **100.0**
SuperPoint + CAPS [40] + NN	Mix	-	**82.7** / 87.8 / **100.0**
DualRC-Net [13]	Full	-	79.6 / 88.8 / 100.0
SIFT + CAPS [40] + NN	Weak	-	77.6 / 86.7 / 99.0
SparseNCNet [30]	Weak	-	76.5 / 84.7 / 98.0
Patch2Pix	Weak	-	79.6 / 87.8 / **100.0**
Full Localization with HLOC [32]			
SuperPoint [5] + NN	Full	85.4 / 93.3 / 97.2	75.5 / 86.7 / 92.9
SuperPoint + CAPS [40] + NN	Mix	86.3 / 93.0 / 95.9	83.7 / 90.8 / 96.9
SuperPoint + SuperGlue [33]	Full	**89.6** / 95.4 / **98.8**	86.7 / 93.9 / **100.0**
Patch2Pix	Weak	84.6 / 92.1 / 96.5	82.7 / 92.9 / 99.0
Patch2Pix (w.CAPS)	Mix	86.7 / 93.7 / 96.7	85.7 / 92.9 / 99.0
Patch2Pix (w.SuperGlue)	Mix	89.2 / **95.5** / 98.5	**87.8 / 94.9 / 100.0**

Table 2. **Evaluation on Aachen Day-Night Benchmark (v1.0) [34, 35].** We report the percentage of correctly localized queries under specific error thresholds. We follow the supervision notations described in Tab. 1 and mark the best results in **bold**.

Experimental Setup. To localize Aachen night-time queries, we follow the evaluation setup from the website[1]. For evaluation on day-time and night-time images together, we adopt the hierarchical localization pipeline (HLOC[2]) proposed in [32]. Matching methods are then plugged into the pipeline to estimate 2D correspondences. We report the percentage of correctly localized queries under specific error thresholds. We test our *Patch2Pix* model with NCNet proposals and SuperPoint [5] + SuperGlue [33] proposals. Note, the model has been only trained on NCNet proposals. Due to the triangulation stage inside the localization pipeline, we quantize our matches by representing keypoints that are closer than 4 pixels to each other with their mean location. We provide a more detailed discussion of the quantization inside our supp. mat (*c.f.* Sec. C).

Results. As shown in Tab. 2, for local feature evalua-

[1] https://github.com/tsattler/visuallocalizationbenchmark
[2] https://github.com/cvg/Hierarchical-Localization

Method	Supervision	Localized Queries (%, 0.25m/0.5m/1.0m, 10°) DUC1	DUC2
SuperPoint [5] + NN	Full	40.4 / 58.1 / 69.7	42.0 / 58.8 / 69.5
D2Net [6] + NN	Full	38.4 / 56.1 / 71.2	37.4 / 55.0 / 64.9
R2D2 [28] + NN	Full	36.4 / 57.6 / 74.2	45.0 / 60.3 / 67.9
SuperPoint + SuperGlue [33]	Full	49.0 / **68.7** / 80.8	53.4 / 77.1 / **82.4**
SuperPoint + CAPS [40] + NN	Mix	40.9 / 60.6 / 72.7	43.5 / 58.8 / 68.7
SIFT + CAPS [40] + NN	Weak	38.4 / 56.6 / 70.7	35.1 / 48.9 / 58.8
SparseNCNet [30]	Weak	41.9 / 62.1 / 72.7	35.1 / 48.1 / 55.0
Patch2Pix	Weak	44.4 / 66.7 / 78.3	49.6 / 64.9 / 72.5
Patch2Pix (w.SuperPoint+CAPS)	Mix	42.4 / 62.6 / 76.3	43.5 / 61.1 / 71.0
Patch2Pix (w.SuperGlue)	Mix	**50.0** / 68.2 / **81.8**	**57.3** / **77.9** / 80.2

Table 3. **InLoc [39] Benchmark Results.** We report the percentage of correctly localized queries under specific error thresholds. Methods are evaluated inside the HLOC [32] pipeline to share the same retrieval pairs, RANSAC threshold, *etc*. We use the supervision notation from Tab. 1 and mark the best results in **bold**.

tion on night-time queries, we outperform the other two weakly-supervised methods. While being worse than SuperPoint [5] + CAPS [40], which involves both full and weak supervision, we are on-par or better than all the other fully-supervised methods. For full localization on all queries using HLOC, we show we are better than SuperPoint + NN on night queries and competitively on day-time images. By further substituting NCNet match proposals with SuperGlue proposals, we are competitive to SuperGlue on day-time images and outperform them slightly on night queries. Our intuition is that we benefit from our epipolar geometry supervision which learns potentially more general features without having any bias from the training data, which is further supported by our next experiment.

5.4. Indoor Localization on InLoc

Finally, we evaluate *Patch2Pix* on the InLoc benchmark [39] for large-scale indoor localization. The large textureless areas and repetitive structures present in its scenes makes this dataset very challenging.

Experimental Setup. Following SuperGlue [33], we evaluate a matching method by using their predicted correspondences inside HLOC for localization. We report the percentage of correctly localized queries under specific error thresholds. It is worth noting that compared to the evaluation on Aachen Day-Night, where our method looses accuracy up to 4 pixels due to the quantization, we have a fairer comparison on InLoc (where no triangulation is needed) to other methods. The results directly reflect the effect of our refinement when combined with other methods. Except for SuperPoint+SuperGlue, we evaluate several configurations of the other methods and compare to their best results. Please see the supp. mat. for more details (*c.f*. Sec. C).

Results. As shown in Tab. 3, *Patch2Pix* is the best among weakly supervised methods and outperforms all other methods except for SuperPoint + SuperGlue. Notice, we are 14.5 % better than SparseNCNet on DUC2 at the finest error, which further highlights that our learned refinement network is more effective than their hand-crafted relocalization mechanism. Further looking at the last rows of Tab. 3, our refinement network achieves the overall best performance among all methods when we replace NCNet proposals with more accurate proposals predicted by SuperPoint + SuperGlue. By searching inside the local regions of SuperPoint keypoints that are matched by SuperGlue, our network is able to detect more accurate and robust matches to outperform SuperPoint + SuperGlue. This implies that epipolar geometry is a promising type of supervision for the matching task. While CAPS is also trained with epipolar loss, its performance still largely relies on the keypoint detection stage. In contrast, we bypass the keypoint detection errors by working directly on the potential matches.

Generalization By evaluating *Patch2Pix* on image matching (*c.f*. Sec. 5.1) and homography estimation (*c.f*. Sec. 5.2), we validate our refinement concept by showing dramatic improvements over NCNet matches. While our network has been trained only on NCNet-type of proposals, we show that our refinement network provides distinct improvements, on both indoor and outdoor localization, by switching from the match proposals produced by NCNet to SuperPoint + SuperGlue proposals without the need for retraining. This highlights that our refinement network learns the general task of predicting matches from a pair of local patches, which works across different scene types and is independent of how the local patch pair has been obtained. Such general matching capability can be used to further improve the existing methods. As shown in Tab. 2 and Tab. 3, both SuperPoint + SuperGlue and SuperPoint + CAPS get improved by using our refinement network.

6. Conclusion

In this paper, we proposed a new paradigm to predict correspondences in a two-stage *detect-to-refine* manner, where the first stage focuses on capturing the semantic high-level information and the second stage focuses on the detailed structures inside local patches. To investigate the potential of this concept, we developed a novel refinement network, which leverages regression to directly output the locations of matches from CNN features and jointly predict confidence scores for outlier rejection. Our network was weakly supervised by epipolar geometry to detect geometrically consistent correspondences.We showed that our refinement network consistently improved our correspondence network baseline on a variety of geometry tasks. We further showed that our model trained with proposals predicted by a correspondence network generalizes well to other types of proposals during testing. By applying our refinement to the best fully-supervised method without retraining, we achieved state-of-the-art results on challenging long-term localization tasks.

References

[1] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, pages 5173–5182, 2017. 5, 6, 7

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *CVIU*, 110(3):346–359, 2008. 1

[3] Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *ICCV*, pages 4322–4331, 2019. 1, 2, 3

[4] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, pages 2292–2300, 2013. 1, 2

[5] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *CVPR Workshops*, pages 224–236, 2018. 1, 2, 3, 6, 7, 8

[6] 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, 2, 3, 5, 6, 7, 8

[7] Patrick Ebel, Anastasiia Mishchuk, Kwang Moo Yi, Pascal Fua, and Eduard Trulls. Beyond cartesian representations for local descriptors. In *CVPR*, pages 253–262, 2019. 1, 2

[8] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *PAMI*, 2018. 1

[9] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *CACM*, 24(6):381–395, 1981. 2

[10] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2dnet: Learning accurate correspondences for sparse-to-dense feature matching. *ECCV*, 2020. 2, 3, 7

[11] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 5

[12] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 5

[13] Xinghui Li, Kai Han, Shuda Li, and Victor Adrian Prisacariu. Dual-resolution correspondence networks. In *NeurIPS*, 2020. 1, 2, 3, 7

[14] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018. 5

[15] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 6

[16] Zixin Luo, Tianwei Shen, Lei Zhou, Jiahui Zhang, Yao Yao, Shiwei Li, Tian Fang, and Long Quan. Contextdesc: Local descriptor augmentation with cross-modality context. In *CVPR*, pages 2527–2536, 2019. 1

[17] 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. In *CVPR*, pages 6589–6598, 2020. 1, 2, 3, 6, 7

[18] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004. 6

[19] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *TPAMI*, 27(10):1615–1630, 2005. 5

[20] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NeurIPS*, pages 4826–4837, 2017. 6

[21] Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Repeatability is not enough: Learning affine regions via discriminability. In *ECCV*, pages 284–300, 2018. 6

[22] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *CVPR*, pages 2666–2674, 2018. 1, 2, 3

[23] Marius Muja and David G Lowe. Scalable nearest neighbor algorithms for high dimensional data. *PAMI*, 36(11):2227–2240, 2014. 2

[24] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *TRO*, 31(5):1147–1163, 2015. 1

[25] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *ICCV*, pages 3456–3465, 2017. 6

[26] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: learning local features from images. In *NeurIPS*, pages 6234–6244, 2018. 2

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 2, 3

[28] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *NeurIPS*, pages 12405–12415, 2019. 1, 2, 3, 6, 7, 8

[29] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *CVPR*, pages 6148–6157, 2017. 1, 2

[30] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. *ECCV*, 2020. 1, 2, 3, 6, 7, 8

[31] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *NeurIPS*, 2018. 1, 2, 3, 4, 5, 6, 7

[32] Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *CVPR*, 2019. 7, 8

[33] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, pages 4938–4947, 2020. 1, 2, 3, 6, 7, 8

[34] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8601–8610, 2018. 7

[35] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image Retrieval for Image-Based Localization Revisited. In *British Machine Vision Conference (BMCV)*, 2012. 7

[36] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 1

[37] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *PJM*, 21(2):343–348, 1967. 1, 2

[38] Weiwei Sun, Wei Jiang, Eduard Trulls, Andrea Tagliasacchi, and Kwang Moo Yi. Acne: Attentive context normalization for robust permutation-equivariant learning. In *CVPR*, pages 11286–11295, 2020. 1, 2, 3

[39] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii. InLoc: Indoor Visual Localization with Dense Matching and View Synthesis. In *CVPR*, 2018. 8

[40] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *ECCV*, 2020. 1, 2, 3, 6, 7, 8

[41] Changchang Wu. Towards linear-time incremental structure from motion. In *3DV*, pages 127–134. IEEE, 2013. 1

[42] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. *ICCV*, 2019. 1, 2, 3

[43] Qunjie Zhou, Torsten Sattler, Marc Pollefeys, and Laura Leal-Taixe. To learn or not to learn: Visual localization from essential matrices. In *ICRA*, pages 3319–3326. IEEE, 2020. 2