Co-Attention for Conditioned Image Matching

Olivia Wiles
VGG, Dept. of Eng. Science
University of Oxford
ow@robots.ox.ac.uk

Sébastien Ehrhardt
VGG, Dept. of Eng. Science
University of Oxford
hyenal@robots.ox.ac.uk

Andrew Zisserman
VGG, Dept. of Eng. Science
University of Oxford
az@robots.ox.ac.uk

Abstract

We propose a new approach to determine correspondences between image pairs in the wild under large changes in illumination, viewpoint, context, and material. While other approaches find correspondences between pairs of images by treating the images independently*, we instead condition on* both *images to implicitly take account of the differences between them. To achieve this, we introduce (i) a spatial attention mechanism (a co-attention module, CoAM) for conditioning the learned features on both images, and (ii) a distinctiveness score used to choose the best matches at test time. CoAM can be added to standard architectures and trained using self-supervision or supervised data, and achieves a significant performance improvement under hard conditions, e.g. large viewpoint changes. We demonstrate that models using CoAM achieve state of the art or competitive results on a wide range of tasks: local matching, camera localization, 3D reconstruction, and image stylization.*

1. Introduction

Determining correspondence between two images of the same scene or object is a fundamental challenge of computer vision, important for many applications ranging from optical flow and image manipulation, to 3D reconstruction and camera localization. This task is challenging due to *scene-shift*: two images of the same scene can differ dramatically due to variations in illumination (e.g. day to night), viewpoint, texture, and season (e.g. snow in winter versus flowering trees in spring).

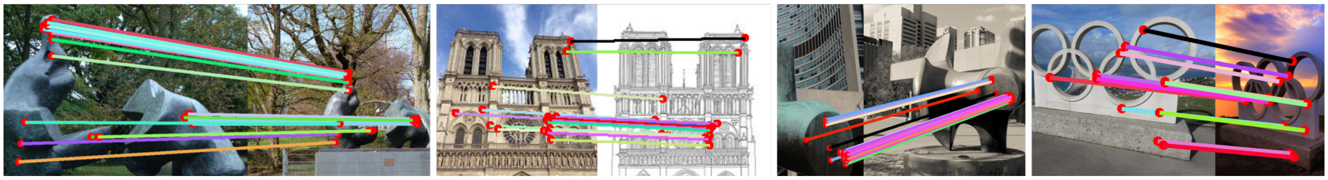
Methods that solve the correspondence task typically follow a *detect-and-describe* approach: first they *detect* distinctive regions [5, 15, 30, 34, 49] and then *describe* these regions using descriptors [5, 6, 21, 26, 30, 49] with varying degrees of invariance to scale, illumination, rotation, and affine transformations. These descriptors are then matched between images by comparing descriptors exhaustively, often using additional geometric constraints [16]. Recent approaches have sought to learn either or both of these components [3, 8, 9, 10, 13, 27, 39, 46, 47, 54, 59, 60, 73, 74]. These methods typically only find matches at textured locations, and do not find matches over smooth regions of an object. Additionally, finding these repeatable detections with invariance to scene-shift is challenging [2, 51, 56].

If prior knowledge is assumed, in terms of limited camera or temporal change (as in optical flow computation in videos), then a *dense-to-dense* approach can be used for pairs that have limited scene shift. In this case, methods typically obtain a dense feature map which is compared from one image to another by restricting the correspondence search to a small support region in the other image (based on the prior knowledge). Spatial and smoothness constraints can additionally be imposed to improve results [8, 11, 29, 53, 66, 71].

We focus on the cases where there is potentially significant scene shift (and no prior knowledge is available), and introduce a new approach for obtaining correspondences between a *pair* of images. Previous methods learn descriptors for each image *without* knowledge of the other image. Thus, their descriptors must be invariant to changes – e.g. to scale and illumination changes. However, as descriptors become increasingly invariant, they become increasingly ambiguous to match (e.g. a constant descriptor is invariant to everything but also confused for everything). We forsake this invariance and instead condition the descriptors on *both* images. This allows the descriptors to be modified based on the differences between the images (e.g. a change in global illumination). Traditionally, this was infeasible, but we can learn such a model efficiently using neural networks.

To achieve this we introduce a network (CD-UNet), which consists of two important components. First, a new spatial **C***o-Attention Module* (CoAM) that can be 'plugged into' a UNet, or similar architectures developed for single image descriptors, in order to generate descriptors conditioned on the pair of images. Second, we introduce a **D***istinctiveness* score in order to select the best matches from these descriptors.

We further investigate the utility of the CoAM under both supervised and self-supervised training. In the latter case, we augment the recent self-supervised approach of learning camera pose of [70] by using CoAMs in a plug-and-play fashion. We evaluate these trained models on a variety of



(a) (b) (c) (d)

Figure 1: Correspondences obtained with the CoAM model, which is augmented with an attention mechanism. These demonstrate the model's robustness in the face of challenging *scene shift*: changes in illumination (a,d), viewpoint (a-d), context (a,d), or style (b).

tasks: local matching, camera localization, 3D reconstruction, and style transfer. We improve over state-of-the-art (sota) models, especially under challenging conditions, and achieve sota or comparable on all tasks.

In summary, we present a key insight: that **conditioning** learned descriptors on both images should allow for improved correspondence matching under challenging conditions. As will be seen, CD-UNet is simple and scalable and eschews a number of techniques used by other methods to improve matching performance: high dimensional descriptors (we use a 64D descriptor, half the size of the current smallest descriptor), and multiple scales (we only operate at a single scale, whereas other methods use multiple scales).

2. Related Work

In this section, we review related work on finding correspondences beyond the local descriptors discussed in Sec. 1. As there is a large amount of relevant research, we focus on the most relevant work in each category.

Correspondences using an attention mechanism. Our architecture can be viewed as a generalization of the standard correlation layer used in training end-to-end models for optical flow [12, 18, 61], stereo [22] or correspondence estimation [11, 25, 53, 69, 70, 71]. This correlation layer (or attention mechanism) is used to compute a cost volume of matches from the learned descriptors.

In correspondence estimation, the learned descriptors are limited in spatial resolution [25, 69, 71] so that the entire volume can be computed. This is too coarse for geometric matching, so other methods use a hierarchical approach [11, 53, 70]. In optical flow [12, 18, 42, 61] and stereo [22], the cost volume is only applied within a limited support region for a single descriptor (e.g. a square region or a raster line) and typically at a lower resolution. Moreover, these methods implicitly assume photometric consistency between frames: their quality degrades the more the frames differ in time, as the pose and viewpoint progressively change.

Unlike these methods, we apply attention at multiple stages in our network, so that the final descriptors *themselves* are conditioned on both images. This should be beneficial for challenging image pairs where one, final comparison is unable to encompass all possible scene-shifts between two images. To find matches at the image level without performing an exhaustive comparison, we use a modified hinge loss to enforce that true descriptors are nearby and false ones further away.

Dense correspondence matching with prior knowledge. Given a static scene and initial camera estimation, a second algorithm, e.g. PatchMatch [4, 57], can be used to find dense correspondences between the images and obtain a full 3D reconstruction. If the images have been rectified using multiple-view geometry [16] and have limited *scene shift*, stereo algorithms such as [14, 41, 62, 72] (and reviewed by [63]) can be used to obtain a full 3D reconstruction.

While not directly related, [23, 58] condition on a second image by iterative warping. This requires multiple passes through a network for each image pair and uses pre-trained descriptors as opposed to training end-to-end.

Also related are approaches that seek to learn correspondence between similar scenes [29] or instances of the same semantic class [23, 38, 44, 45].

Local descriptors for image retrieval. Another form of correspondence is to find relevant images in a database using a query image. Related works use an aggregation of local descriptors from a CNN [7, 67]. Again, these methods generate descriptors for the dataset images independently of the query image, whereas the descriptors we extract for the input image are conditioned on both images.

Stylization for robust correspondence matching. Our idea of conditioning the output of one image on another has interesting connections to stylization and associated generative models [20, 40, 68]. Additionally, a recent line of work studies how training on stylized images can improve robustness in correspondence matching [32]. As opposed to enforcing invariance to style, CD-UNet and the other architectures considered, learn how to leverage differing styles (as the precise style may be useful) via our CoAM.

3. Method

Our task is to find dense correspondences between a pair of images of the same scene. This proceeds in two stages. The first stage obtains dense descriptor vectors for each image and a distinctiveness score. The descriptors are conditioned on *both* images so they only have to be invariant to the changes particular to that pair of images. The second stage compares these descriptor vectors to obtain a set of high quality matches. We first describe in Sec. 3.1 our full architecture CD-UNet, and how it is trained in Sec. 3.2. CD-

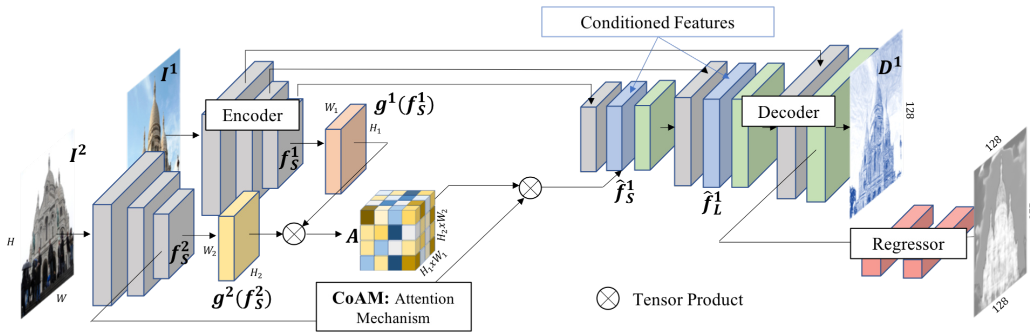


Figure 2: Overview of CD-UNet for obtaining co-attended descriptors. Descriptor vectors D^1 for one input image I^1 are conditioned on another I^2 using our CoAM. This module can be applied at multiple layers in the model hierarchy (we show one for clarity). The conditioned features are then *decoded* to obtain D^1. We also *regress* a distinctiveness mask which is used at test time to ignore unmatchable regions (e.g. the sky or regions visible in only one image). The descriptor vectors D^2 for I^2 are obtained by swapping the input images.

UNet consists of a set of Co-Attention Modules (CoAMs) and a distinctiveness score regressor, which are incorporated into a standard UNet architecture. Then, in Sec. 3.3, we describe how CoAM is incorporated into the recent CAPSNet architecture [70] and trained in a self-supervised manner.

3.1. A UNet encoder-decoder with CoAM

The architecture for obtaining descriptor vectors and a distinctiveness score for one image I^1 (Fig. 2), is composed of four components. The first component, the **encoder**, projects both images I^1, I^2 to obtain feature maps at two resolutions: f_L^i, f_S^i. The second component, the **attention mechanism (CoAM)**, is used to determine spatial correspondence between the feature maps of the different images and obtain conditioned feature maps. The third component, the **decoder**, concatenates the conditioned feature maps with the original feature maps. These are decoded to obtain a grid of spatial descriptor vectors D^1 (which are conditioned on both images). The final component, the **regressor**, learns a distinctiveness score for each grid position, which encodes how likely the match is to be accurate. To obtain descriptor vectors D^2 for the other image, we operate precisely as described above, except that the order of the input images is flipped. This gives a grid of descriptor vectors D^1, D^2 for images I^1, I^2 respectively.

Encoder. Given two images of the same scene, $I^1 \in \mathbb{R}^{H\times W\times 3}$ and $I^2 \in \mathbb{R}^{H\times W\times 3}$, we obtain spatial feature maps: f_L^i and f_S^i at a $l$arger and $s$maller resolution. These will be concatenated within a UNet framework [48] and injected into the decoder. A CNN with shared parameters is used to encode the images and obtain these spatial feature maps. In practice, we use the feature maps after the last two blocks in a ResNet50 [17] architecture.

CoAM Attention Module. We wish to concatenate features from both images in order to condition the model on both input images. However, for a given spatial location, the relevant (corresponding) feature in the other image may not be at the same spatial location. As a result, we use an attention mechanism to model long range dependencies.

In detail, the attention mechanism is used to determine where a location i in one set of features g from one image should attend to in another set of features h from another image [69]. For each location i in g, it obtains a feature $\hat{g}_i$ that is a weighted sum over all spatial features in h where A is the similarity matrix comparing g and h using the inner product followed by the softmax normalization step.

$$\hat{g}_i = \sum_j A_{ij} h_j \qquad A_{ij} = \frac{\exp(g_i^T h_j)}{\sum_k \exp(g_i^T h_k)} \qquad (1)$$

To apply this attention mechanism, we operate as follows for f_L^1 (and similarly for f_S^1). First, to perform dimensionality reduction (as is standard), the features are projected with two MLPs $g^1(\cdot), g^2(\cdot)$: $g = g^1(f_L^1)$, $h = g^2(f_L^2)$. The attended features $\hat{f}_L^1$ are then computed using the projected features as in (1). This gives a new feature map of the features in I^2 at the corresponding position in I^1.

Decoder: Conditioned Features. The attended features are incorporated into a UNet [48] architecture to obtain a grid of spatial descriptors $D^1 \in \mathbb{R}^{H\times W\times D}$ (Fig. 2). The attended features $\hat{f}_L^1$ and $\hat{f}_S^1$ are concatenated with the original features and passed through the decoder portion of the UNet. The resulting feature map is $L2$ normalized over the channel dimension to obtain the final descriptors. This step ensures that the final descriptors are conditioned on both images.

Regressor: Distinctiveness Score. We regress a distinctiveness score $r(\cdot)_{ij} \in [0, 1]$, for each pixel (i, j), which approximates its matchability and is used at test time to select the best matches. $r(\cdot)_{ij}$ approximates how often the descriptor at (i, j) is confused with negatives in the other image. If it is near 1, the descriptor is uniquely matched; if it is near 0, the descriptor is often confused. To regress these values, we use an MLP, $r(\cdot)$, on top of the unnormalized descriptor maps.

Determining Matches at Test Time. We want matches at locations k and l in images I^1 and I^2 respectively that are accurate and distinctive (e.g. no matches in the sky). We use the scalar product to compare the normalized descriptor vectors to find the best matches and the distinctiveness score to determine the most distinctive matches. The following similarity score c_{kl} combines these properties such that a value near 1 indicates a distinct and accurate match:

$$c_{kl} = r(D_k^1) r(D_l^2) \left[\left(D_k^1\right)^T D_l^2 \right]. \tag{2}$$

Finally, we select the best K matches. First, we exhaustively compare all descriptors in both images. Then, we only select those matches that are mutual nearest neighbours: e.g. if the best match for location m in one image is location n in another, and the best match for location n is m, then (n, m) is a good match. So if the following holds:

$$m = \texttt{argmax}_j c_{nj} \quad \text{and} \quad n = \texttt{argmax}_i c_{im}. \tag{3}$$

These matches are ranked according to their similarity score and the top K selected.

3.2. Supervised Training and Loss Functions

Selecting Correspondences at Train Time. Given a ground-truth correspondence map, we randomly select L positive correspondences. For each positive correspondence, we randomly select a large number ($N = 512$) of negative correspondences. These randomly chosen positive and negative correspondences are used to compute both the distinctiveness and correspondence losses.

Correspondence Loss. The correspondence loss is used to enforce that the normalized descriptor maps D^1 and D^2 can be compared using the scalar product to obtain the best matches. At a location i in D^1 and j in D^2 then the standard Euclidean distance metric $d(D_i^1, D_j^2)$ should be near 0 if the corresponding normalized descriptor vectors are a match.

To train these descriptors, we use a standard contrastive hinge loss to separate true and false correspondences (we consider other contrastive losses in the appendix). For the set $\mathcal{P}$ of L true pairs, the loss $\mathcal{L}_p$ enforces that the distance between descriptors is near 0. For the set $\mathcal{N}$ of LN negative pairs, the loss $\mathcal{L}_n$ enforces that the distance between descriptors should be above a margin M.

$$\mathcal{L}_p = \frac{1}{L} \sum_{(x,y) \in \mathcal{P}} d(D_x^1, D_y^2) \tag{4}$$

$$\mathcal{L}_n = \frac{1}{LN} \sum_{(x,\hat{y}) \in \mathcal{N}} \max(0, M + c_x - d(D_x^1, D_{\hat{y}}^2)). \tag{5}$$

$c_x = d(D_x^1, D_y^2), (x, y) \in \mathcal{P}$ re-weights the distance of the false correspondence according to that of the positive one: the less confident the true match, the further the negative one must be from M [10].

Distinctiveness Loss. To learn the $r(\cdot)$ MLP, we need an estimate of how often a descriptor in one image is confused with the wrong descriptors in the other image. Given a set $\mathcal{N}$ of N negative matches in the other image and the margin M, the number of times a descriptor at location x is confused is $m_x = \sum_{\hat{y} \in \mathcal{N}} \mathbb{1}(d(D_x^1, D_{\hat{y}}^2) < M)$. This value is used to regress $r(\cdot)$, which is near 1 if the feature has a unique match (the true match), near 0 otherwise (τ is a hyper-parameter set to $\frac{1}{4}$):

$$\mathcal{L}_r = \frac{1}{L} \sum_{(x,\cdot) \in \mathcal{P}} |r(D_x^1), \frac{1}{(1 + m_x)^\tau}|_1. \tag{6}$$

Training Setup. CD-UNet is trained on MegaDepth [28], which consists of a variety of landmarks, registered using SfM [55]. As each landmark consists of many images taken under differing conditions, we can obtain matches between images that are unmatchable when considered independently.

We train the features end-to-end, but train the distinctiveness score separately by not allowing gradients to flow. In practice we backpropagate on all randomly chosen positive pairs $\mathcal{L}_p$, negative pairs $\mathcal{L}_n$, and additionally the hardest $H = 3$ negative pairs for each positive pair.

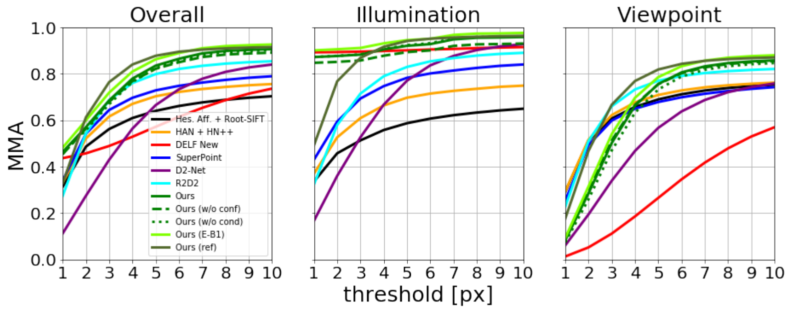
The model is trained with a learning rate of 0.0001, the ADAM optimizer [24], a batch size of 16, M=1, L=512, and N=512. At train time we use an image size of 256, at test time an image size of 512. We use K=2000 for HPatches and Aachen, and K=8192 when performing SfM. For SfM, we find it is important to use more, rougher correspondences to obtain more coverage in the 3D reconstruction.

3.3. Self-supervised training – the CAPSNet [70] with CoAM

In this section we describe how CoAM can be added to the CAPSNet architecture of [70] and trained using the self-supervised framework of [70].

CAPSNet consists of a UNet style architecture, which predicts features at a coarse and fine level. The matches at a coarse level are used to guide feature matching at the finer level. These features are trained using two losses. First, an epipolar loss enforces that matches should satisfy epipolar constraints. Second, a cycle consistency loss enforces that, for a match between two images, the best match for the local descriptor in one image should also be the best match for the local descriptor in the other. Using this approach, the authors achieve high quality results at pose estimation on the challenging MegaDepth test set.

As the descriptor model is a UNet style architecture, and it is trained in an end-to-end fashion, we operate in a very similar manner to the UNet architecture with CoAM of Sec. 3.1, by again adding CoAMs to condition descriptors on both images. We use the CoAM to inject attended features from the other image at either a coarse level or at a fine and coarse



Method	# Matches
Hes. det. + RootSift [1, 30]	2.8K
HAN + HN++ [36, 35]	2.0K
LF-Net [39]	0.2K
SuperPoint [9]	0.9K
DELF [37]	1.9K
D2 Trained (SS) [10]	2.5K
R2D2 [43]	1.8K
Ours	2.0K

Figure 3: HPatches [2]. Comparison with sota using the mean matching accuracy for different pixel thresholds on the HPatches dataset. We also report the mean matches extracted per image pair. For this dataset, one desires more matches with high accuracy. Our method achieves superior performance when images vary by illumination for all thresholds, and by viewpoint for thresholds > 6px. By a simple refinement strategy (**ours (ref)**), we achieve sota for all thresholds on both viewpoint and illumination.

level (precise details are given in the appendix). In both cases, this leads to an addition of less than 15% of the total weights of the original network.

The loss functions used to train the conditioned local descriptors are unchanged from the original CAPSNet work.

Training Setup. We train the model as done in [70]: for 200K iterations, using a batch size of 6 images, and an image size of 480×640.

3.4. Discussion

Here we discuss some of the benefits of conditioning using CoAM as opposed to operating directly on local descriptors and keypoints as done in SuperGLUE [50]. First, our module is trained end-to-end and does not introduce an extra step in the matching pipeline of comparing pretrained descriptors. Second, our descriptors are learned, so our method is not dependent on the quality of the extracted descriptors. Finally, SuperGLUE scales with the number of extracted keypoints, hampering its performance and utility on tasks that require finding a large number of correspondences (e.g. SFM). As the CoAM is plugged in as a component of our network, our method scales with image size. For reference, on a single GPU, to extract 2k keypoints on a 256×256 image, our method runs in 97ms while SuperGLUE would add an overhead of $\approx$270ms as reported in the original paper. Further, our method would scale with little overhead to more keypoints at the given image size.

Our method requires an exhaustive match of all image pairs. While we find that we can run the full, exhaustive pipeline on reasonably large datasets ($\approx$ 1500 images) in Sec. 4.2.2, we envision two stages when using our method in truly large scale settings. First, a coarser, faster method can be used as a preprocessing step to remove spurious pairs and our method subsequently used in a second stage to find high quality correspondences.

4. Experiments I: Supervised Co-AM

In this section we evaluate the CD-UNet architecture (UNet encoder-decoder with CoAM and distinctiveness score as in Fig. 2) on four challenging downstream tasks under full supervision. In Sec. 5 the benefits of the co-attention module are evaluated under self-supervised training [70].

The first task directly assesses how well CD-UNet can estimate correspondences between images pairs. The second task uses the correspondences to perform camera localization. In these tasks we ablate the utility of the CoAM and distinctiveness score components of the architecture. The third task obtains high quality 3D reconstructions in challenging situations, with a large amount of *scene shift*. The final task is stylization, and assesses CD-UNet's matches, when extracted in a dense manner, on a downstream task.

In general we find that CD-UNet achieves state of the art or comparable results and that the CoAM is useful especially in challenging conditions (e.g. when there is a large viewpoint change).

The appendix includes further ablations to validate our choices (e.g. the loss function and grid size) and datasets (e.g. (1) YFCC100M [65] which shows our superior results and the utility of both the distinctiveness score and the CoAM, and (2) a new, challenging SFM dataset). Finally, it includes qualitative samples for each of the experiments discussed in the following, including HPatches, Aachen, SFM, and stylization.

Ablations. The full model uses the ResNet50 [17] backbone, the CoAMs and the distinctivness score to reweight matches. We ablate multiple variants. The first (**ours**) is our full model. The second (**ours w/o conf**) is our model without the distinctiveness score but only the scalar product. The third (**ours w/o cond**) is our model without conditioning (i.e. the CoAMs). The final variant (**ours-E-B1**) is our full model but using an EfficientNet-B1 backbone [64]. This ablation uses a smaller (7M params vs 23M params) and faster (0.7GFlops vs 4.1GFlops) backbone architecture; it is more suitable for

Table 1: Aachen Day-Night [51]. Higher is better. Ours does comparably or better than other sota setups. * indicates the method was trained on the Aachen dataset.

Method	Threshold Accuracy		
	0.25m (2°)	0.5m (5°)	5m (10°)
Upright RootSIFT [30]	36.7	54.1	72.5
DenseSFM [51]	39.8	60.2	84.7
Han+, HN++ [36, 35]	39.8	61.2	77.6
Superpoint [9]	42.8	57.1	75.5
DELF [37]	39.8	61.2	85.7
D2-Net [10]	**44.9**	66.3	**88.8**
R2D2* [43]	**45.9**	66.3	**88.8**
Ours w/o cond	42.9	62.2	87.8
Ours w/o conf	43.9	64.3	86.7
Ours	**44.9**	**70.4**	**88.8**
Ours (E-B1)	**44.9**	68.4	**88.8**

Table 2: SfM. We compare our approach to using SIFT features on 3D reconstruction. ↑: higher is better. ↓: lower is better.

		Large SfM		
	Landmark:	Madrid Met.	Gen.	Tow. of Lon.
# Reg. Ims ↑	**SIFT [30]:**	500	1035	804
	Ours:	**702**	**1072**	**967**
# Sparse Pts ↑	**SIFT [30]:**	116K	338K	239K
	Ours:	**256K**	**570K**	**452K**
Track Len ↑	**SIFT [30]:**	**6.32**	5.52	**7.76**
	Ours:	6.09	**6.60**	5.82
Reproj Err (px) ↓	**SIFT [30]:**	**0.60**	**0.69**	**0.61**
	Ours:	1.30	1.34	1.32
# Dense Pts ↑	**SIFT [30]:**	**1.8M**	**4.2M**	**3.1M**
	Ours:	1.1M	2.1M	1.8M

practical applications.

4.1. Correspondence Evaluation

We test our model on local matching by evaluating on the HPatches [2] benchmark. We compare to a number of baselines and achieve state-of-the-art results.

HPatches Benchmark. The HPatches benchmark evaluates the ability of a model to find accurate correspondences between pairs of images, related by a homography, that vary in terms of illumination or viewpoint. We follow the standard setup used by D2Net [10] by selecting 108 of the 116 sequences which show 6 images of larger and larger illumination and viewpoint changes. The first image is matched against the other 5, giving 540 pairs.

Evaluation Setup. We follow the evaluation setup of D2Net [10]. For each image pair, we compute the number of correct matches (using the known homography) and report the average number of correct matches as a function of the pixel threshold error in Fig. 3. We then compare to a number of detect-then-describe baselines used in D2Net using their software: RootSIFT [1, 30] with the Affine keypoint detector [33], HesAffNet [36] with HardNet++ [35], LF-Net [39], SuperPoint [9], DELF [37]; as well as to D2Net [10] and R2D2 [43]. These methods vary in terms of whether the detector and descriptors are hand crafted or learned.

Results. As shown in Fig. 3, all variants of our model outperform previous methods for larger pixel thresholds, demonstrating the practicality and robustness of our approach. In comparison to other methods, CD-UNet performs extremely well when the images vary in illumination: it outperforms all other methods. CD-UNet is superior under viewpoint changes for larger pixel thresholds ($>$ 6px). Using the smaller, more efficient (**ours-E-B1**) actually improves performance over the larger ResNet model (**ours**). A simple refinement strategy (described in the appendix) boosts our model's performance under viewpoint changes, giving results superior or comparable to sota methods for all thresholds for viewpoint and illumination changes. Compared to the other evaluation datasets, e.g. [51] below, the components of our model have a limited impact on performance on this benchmark, presumably because this dataset has less *scene shift* than the others.

4.2. Using Correspondences for 3D Reconstruction

In this section, we evaluate the robustness of our approach on images that vary significantly in terms of illumination and viewpoint, and our model's ability to scale to larger datasets. CD-UNet achieves sota or comparable results on all datasets.

4.2.1 Camera Localization

Aachen Benchmark. In order to evaluate our approach under large illumination changes, we use the Aachen Day-Night dataset [51, 52]. For each of the 98 query night-time images, the goal is to localize the image against a set of day-time images using predicted correspondences.

Evaluation Setup. The evaluation measure is the percentage of night time cameras that are localized within a given error threshold [51]. We use the pipeline and evaluation server of [51] with the matches automatically obtained with our method (Sec. 3.1). We compare against RootSIFT descriptors from DoG keypoints [30], HardNet++ with HesAffNet features [35, 36], DELF [37], SuperPoint [9], D2Net [10] and DenseSFM [51].

Results. Tab. 1 shows that our method does comparably or better than other sota approaches. They also show the utility of the distinctiveness score and CoAM. These results imply that the traditional approach of first finding reliably detectable regions may be unnecessary; using a grid to exhaustively find matches is, perhaps surprisingly, superior in this case. These results also show that our architectural improvements (i.e. using an attention mechanism and distinctiveness score) boost performance and that an efficient architecture (**Ours-E-B1**) has a small impact on performance.

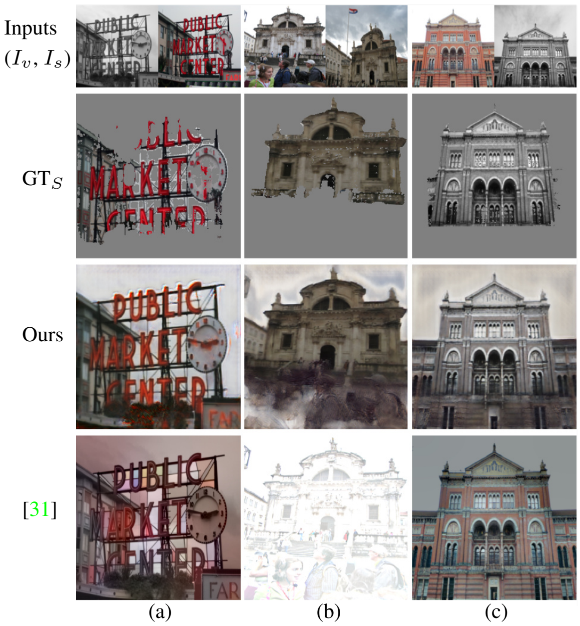


Figure 4: Stylization. Given I_s and I_v, the task is to generate an image with the pose and viewpoint of I_v and style of I_s. We show results for CoAM and a baseline that uses semantics [31]. We also show the resampled image (GT$_S$) which is computed using the true correspondences from the MegaDepth dataset [28]. While [31] works well for easy cases, it sometimes copies style from I_v (as shown by the red background in (a) and red hued building in (c)). [31] also fails if the semantic prediction is incorrect (e.g. (b)).

4.2.2 Structure from Motion (SfM)

The objective here is to evaluate the correspondences obtained with our model for the task of 3D reconstruction.

SfM Dataset. The assessment is on the standard SfM Local Feature Evaluation Benchmark [56] that contains many ($\approx$ 1500) internet images of three landmarks: Madrid Metropolis, Gendarmenmarkt, and Tower of London.

Baselines. We compare to SIFT [30]. This method first finds repeatably detectable regions for which features are extracted and compared between images. This method works well when there are distinctive textured regions that can be found and matched. Our method, however, conditions on both images, so our approach should be more robust when there are fewer textured regions or where there are significant variations between the images as it can make use of auxiliary information from the other image. Additional, less challenging baselines are given in the appendix.

Results. Tab. 2 shows that, using CD-UNet, we consistently register more images and obtain more sparsely reconstructed 3D points (visualizations are in the appendix). However, the pixel error is higher and there are fewer dense points. These differing results are somewhat explained by the implicit trade off between number of points and reprojection error [70].

Figure 5: CoAM Attention. We visualize the predicted attention (A) for sample image pairs. The red dot in I^1 denotes the point for which we compute the attention. It is not clear apriori what the attention module should do, but it does attend to relevant, similar regions in the other image and is dependent on the query location.

However, clearly our results are competitive with SIFT.

4.3. Using Correspondences for Stylization

Previously, we focused on using our matching pipeline for extracting a set of correspondences to be used for localization and 3D reconstruction. Here we evaluate how well our features can be used for a task that requires dense matching: stylization. The goal is, given two images I_s, I_v of the same scene, to generate an image with the style of I_s but the pose and viewpoint of I_v.

Setup. To achieve this, we first use CD-UNet to transform I_s into the position of I_v. The approach is simple: instead of only choosing the mutual nearest neighbours as in (2), we consider the best match for *every* pixel location. Then, the color of the best match in I_s is used to color the corresponding location in I_v. This gives the *sampled image*. The next step is to remove artefacts. We do this by training a refinement model on top of the sampled image in order to obtain an image I_g in the pose of I_v and style of I_s. Full details of the architecture and training are given in the supp.

Relation to Pix2Pix [19]. In a standard image to image translation task (e.g. Pix2Pix [19]), the two images (e.g. image and semantic map) are aligned. In our case, the images are not aligned. We effectively use our correspondences to align the images and then run a variant of Pix2Pix.

Experimental Setup. To evaluate our results, we use the test set of the MegaDepth dataset (these are landmarks unseen at training time). We randomly select 400 pairs of images and designate one the viewpoint I_v image and the other the style image I_s. We task the models to generate a new image I_g with the style of I_s in the viewpoint of I_v. From the MegaDepth dataset, we can obtain ground truth correspondence for regions in both images and so the true values of I_g for this region. The reported error metric is the mean $L1$ distance between the generated image and true value within this region.

Method	Accuracy on MegaDepth		
	easy	*medium*	*hard*
CAPS [70] w/ SIFT Kp.	91.3 / **52.7**	82.5 / 57.9	65.8 / 61.3
Ours (C CoAM)	91.7 / 52.1	**82.9** / **58.6**	**69.3** / 62.4
Ours (C+F CoAMs)	**91.9** / 52.3	82.8 / 58.4	68.8 / **63.4**

Table 3: Results on the MegaDepth dataset on three increasingly challenging subsets (*easy*, *medium*, and *hard*) for both angular / translational errors: $(\cdot)$ / $(\cdot)$. The results show that augmenting the baseline model with our CoAMs improves performance, especially on the challenging viewpoint images, demonstrating the utility of conditioning the descriptors on *both* images under these conditions.

Results. We compare against a stylization approach that uses semantics to perform style transfer [31] in Fig. 4. We also determine the $L1$ error for both setups and obtain 0.22 for [31] and 0.14 for our method, demonstrating that our method is more accurate for regions that can be put in correspondence. The qualitative results demonstrate that our method is more robust, as [31] produces poor results if the semantic prediction is wrong and sometimes copies style from I_v as opposed to I_s (e.g. it creates a colored I_g image when I_s is grey-scale). As we sample from I_s in the first step and then refine the sampled image, our model rarely copies style from I_v. Finally, our full method runs in seconds at test time whereas [31] takes minutes due to a computationally intensive iterative refinement strategy.

5. Experiments II: CoAM with CAPSNet

Next, we evaluate CoAM when injected into the CAPSNet architecture [70] and trained in a self-supervised manner. We again validate our hypothesis that conditioning on two images is preferable in this setting, as it improves results on the downstream task of pose prediction. Finally, we visualize and investigate the learned attention to obtain an intuition into how CoAM is being used by the network.

5.1. Camera Pose Prediction

This experiment follows that of [70]. The aim is to estimate the relative camera pose between pairs of images extracted at random from the MegaDepth test set. The pairs of images are divided into three subsets depending on the relative angular change: **easy** ($[0°, 15°]$), **medium** ($[15°, 30°]$), and **hard** ($[30°, 60°]$). Each subset has at least 1000 pairs.

In order to determine the relative camera pose, we follow the approach of [70]. The essential matrix is extracted by using the mutual nearest neighbour correspondences and known camera intrinsic parameters. The essential matrix is decomposed into the rotation and translation matrices. The estimated angular change in rotation and translation is then compared to the ground truth. If the difference between the predicted and ground truth is less than a threshold of $10°$, the prediction is considered correct.

We consider two variants of injecting CoAM into the CAPSNet architecture. First, (C CoAM) only injects one CoAM at a coarse resolution. Second, (C+F CoAM) injects two CoAMs at a coarse and a fine resolution. We report the percentage of correct images for rotational and translational errors separately in Tab. 3. These results demonstrate that using a CoAM does indeed improve over the baseline model, especially on the harder angle pairs. Injecting further CoAMs does not substantially increase performance but it consistently performs better than the original CAPSNet model. This demonstrates the value of using our CoAM to condition descriptors on both images.

5.2. Visualisation of CoAM's Attention

Finally, we visualize CoAM's predicted attention in Fig. 5 to obtain an intuition of how the additional image is used to improve the learned descriptors. We note that there is no clear a priori knowledge of what the model *should* attend to. The attention module could find regions of similar texture but varying style in order to be invariant to the style. Or the module could attend to the right location in the other image. However, the qualitative results imply that the model is making use of the CoAM to attend to relevant regions.

Additionally, we quantify how invariant the descriptors are with the CoAM and without. We use the sets of images in the HPatches benchmark that vary in illumination. One image is kept fixed (the target) and the other varied (the query). We then evaluate how much the query image's descriptors vary from those of the target by computing the L1 error. Our descriptors differ on average by 0.30 ± 0.14, whereas [70]'s descriptors differ more, by 0.41 ± 0.17. This validates that the CoAM increases the invariance of corresponding descriptors under large amounts of *scene shift*.

6. Conclusion

We investigated a new approach for obtaining correspondences for image pairs using a co-attention module and distinctiveness score. The central insight was that, using neural networks, descriptors can be conditioned on *both* images. This allows greater flexibility, as the descriptors only need to be invariant to changes between the pair of images. Using this insight, our simple model improved the quality of the learned descriptors over those of a baseline model on multiple tasks and in both a supervised and self-supervised setting. We would expect further improvements with larger, more complex models.

Acknowledgements. The authors thank Noah Snavely and David Murray for feedback and discussions. This work was funded by an EPSRC studentship, EPSRC Programme Grant Seebibyte EP/M013774/1, a Royal Society Research Professorship, and ERC 638009-IDIU.

References

[1] Relja Arandjelović and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *Proc. CVPR*, 2012. 5, 6

[2] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. Hpatch: A benchmark and evaluation of handcrafted and learned local descriptors. In *Proc. CVPR*, 2017. 1, 5, 6

[3] Vassileios Balntas, Edgar Riba, Daniel Ponsa, and Krystian Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *Proc. BMVC*, 2016. 1

[4] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. In *ACM Transactions on Graphics (ToG)*, volume 28, page 24. ACM, 2009. 2

[5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *Proc. ECCV*, May 2006. 1

[6] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary robust independent elementary features. In *Proc. ECCV*, 2010. 1

[7] Bingyi Cao, Andre Araujo, and Jack Sim. Unifying deep local and global features for efficient image search. In *Proc. ECCV*, 2020. 2

[8] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *NeurIPS*, 2016. 1

[9] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-supervised interest point detection and description. In *CVPR workshops*, 2018. 1, 5, 6

[10] Mihai Dusmanu, Ignacio Rocco, Tomas Pajdla, Marc Pollefeys, Josef Sivic, Akihiko Torii, and Torsten Sattler. D2-Net: A trainable cnn for joint description and detection of local features. In *Proc. CVPR*, 2019. 1, 4, 5, 6

[11] Mohammed E Fathy, Quoc-Huy Tran, M Zeeshan Zia, Paul Vernaza, and Manmohan Chandraker. Hierarchical metric learning and matching for 2D and 3D geometric correspondences. In *Proc. ECCV*, 2018. 1, 2

[12] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proc. ICCV*, 2015. 2

[13] Hugo Germain, Guillaume Bourmaud, and Vincent Lepetit. S2DNet: Learning accurate correspondences for sparse-to-dense feature matching. *Proc. ECCV*, 2020. 1

[14] Marsha Jo Hannah. *Computer matching of areas in stereo images*. PhD thesis, 1974. 2

[15] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Proc. Alvey Vision Conf.*, pages 147–151, 1988. 1

[16] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000. 1, 2

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proc. CVPR*, 2016. 3, 5

[18] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *Proc. CVPR*, 2017. 2

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*, 2017. 7

[20] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proc. CVPR*, 2019. 2

[21] Yan Ke and Rahul Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proc. CVPR*, Jun 2004. 1

[22] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. End-to-end learning of geometry and context for deep stereo regression. In *Proc. CVPR*, 2017. 2

[23] Seungryong Kim, Stephen Lin, Sang Ryul Jeon, Dongbo Min, and Kwanghoon Sohn. Recurrent transformer networks for semantic correspondence. In *NeurIPS*, 2018. 2

[24] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *Proc. ICLR*, 2014. 4

[25] Zihang Lai, Erika Lu, and Weidi Xie. MAST: A memory-augmented self-supervised tracker. In *Proc. CVPR*, 2020. 2

[26] Stefan Leutenegger, Margarita Chli, and Roland Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Proc. ICCV*, pages 2548–2555, 2011. 1

[27] Shuda Li, Kai Han, Theo W. Costain, Henry Howard-Jenkins, and Victor Prisacariu. Correspondence networks with adaptive neighbourhood consensus. In *Proc. CVPR*, 2020. 1

[28] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proc. CVPR*, 2018. 4, 7

[29] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift Flow: Dense correspondence across scenes and its applications. *IEEE PAMI*, 33(5):978–994, 2010. 1, 2

[30] David Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 1, 5, 6, 7

[31] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. *Proc. CVPR*, 2017. 7, 8

[32] Iaroslav Melekhov, Gabriel J Brostow, Juho Kannala, and Daniyar Turmukhambetov. Image stylization for robust features. *arXiv preprint arXiv:2008.06959*, 2020. 2

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

[34] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, Jiri Matas, Frederik Schaffalitzky, Timor Kadir, and Luc Van Gool. A comparison of affine region detectors. *IJCV*, 65(1/2):43–72, 2005. 1

[35] Anastasiya Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor's margins: Local descriptor learning loss. In *NeurIPS*, 2017. 5, 6

[36] Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Repeatability is not enough: Learning affine regions via discriminability. In *Proc. ECCV*, 2018. 5, 6

- [37] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proc. ICCV*, 2017. 5, 6
- [38] David Novotny, Diane Larlus, and Andrea Vedaldi. AnchorNet: A weakly supervised network to learn geometry-sensitive features for semantic matching. In *Proc. CVPR*, 2017. 2
- [39] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. LF-Net: learning local features from images. In *NeurIPS*, 2018. 1, 5, 6
- [40] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proc. CVPR*, 2019. 2
- [41] Marc Pollefeys. *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*. PhD thesis, ESAT-PSI, K.U.Leuven, 1999. 2
- [42] Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. In *Proc. CVPR*, 2017. 2
- [43] 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. 5, 6
- [44] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proc. CVPR*, 2017. 2
- [45] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. End-to-end weakly-supervised semantic alignment. In *Proc. CVPR*, 2017. 2
- [46] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. Efficient neighbourhood consensus networks via submanifold sparse convolutions. In *Proc. ECCV*, 2020. 1
- [47] Ignacio Rocco, Mircea Cimpoi, Relja Arandjelović, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Neighbourhood consensus networks. In *NeurIPS*, 2017. 1
- [48] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. MICCAI*, pages 234–241. Springer, 2015. 3
- [49] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proc. ICCV*, pages 2564–2571, 2011. 1
- [50] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *Proc. CVPR*, 2020. 5
- [51] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl Kahl, and Tomas Pajdla. Benchmarking 6dof outdoor visual localization in changing conditions. In *Proc. CVPR*, 2018. 1, 6
- [52] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image retrieval for image-based localization revisited. In *Proc. BMVC*, 2012. 6
- [53] Nikolay Savinov, Lubor Ladicky, and Marc Pollefeys. Matching neural paths: transfer from recognition to correspondence search. In *NeurIPS*, 2017. 1, 2
- [54] Nikolay Savinov, Akihito Seki, Lubor Ladicky, Torsten Sattler, and Marc Pollefeys. Quad-Networks: unsupervised learning to rank for interest point detection. In *Proc. CVPR*, 2017. 1
- [55] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proc. CVPR*, 2016. 4
- [56] Johannes L Schönberger, Hans Hardmeier, Torsten Sattler, and Marc Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Proc. CVPR*, 2017. 1, 7
- [57] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *Proc. ECCV*, 2016. 2
- [58] Xi Shen, François Darmon, Alexei A Efros, and Mathieu Aubry. RANSAC-Flow: generic two-stage image alignment. *Proc. ECCV*, 2020. 2
- [59] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *Proc. ICCV*, 2015. 1
- [60] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Learning local feature descriptors using convex optimisation. *IEEE PAMI*, 2014. 1
- [61] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proc. CVPR*, 2018. 2
- [62] Jian Sun, Yin Li, Sing Bing Kang, and Heung-Yeung Shum. Symmetric stereo matching for occlusion handling. In *Proc. CVPR*, 2005. 2
- [63] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 2
- [64] Mingxing Tan and Quoc V Le. EfficientNet: Rethinking model scaling for convolutional neural networks. *Proc. ICML*, 2019. 5
- [65] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: the new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 5
- [66] Engin Tola, Vincent Lepetit, and Pascal Fua. A fast local descriptor for dense matching. In *Proc. CVPR*, 2008. 1
- [67] Giorgos Tolias, Tomas Jenicek, and Ondřej Chum. Learning and aggregating deep local descriptors for instance-level recognition. In *Proc. ECCV*, 2020. 2
- [68] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proc. ICML*, 2016. 2
- [69] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *Proc. ECCV*, 2018. 2, 3
- [70] Qianqian Wang, Xiaowei Zhou, Bharath Hariharan, and Noah Snavely. Learning feature descriptors using camera pose supervision. In *Proc. ECCV*, 2020. 1, 2, 3, 4, 5, 7, 8
- [71] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *Proc. CVPR*, 2019. 1, 2
- [72] Oliver Woodford, Philip Torr, Ian Reid, and Andrew Fitzgibbon. Global stereo reconstruction under second order smoothness priors. *IEEE PAMI*, 31(12):2115–2128, 2009. 2
- [73] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. LIFT: Learned invariant feature transform. In *Proc. ECCV*, 2016. 1
- [74] Linguang Zhang and Szymon Rusinkiewicz. Learning to detect features in texture images. In *Proc. CVPR*, 2018. 1