Learning to Align Images using Weak Geometric Supervision

Jing Dong[1,2] Byron Boots[1] Frank Dellaert[1] Ranveer Chandra[2] Sudipta N. Sinha[2]

[1] Georgia Institute of Technology [2] Microsoft Research

Abstract

Image alignment tasks require accurate pixel correspondences, which are usually recovered by matching local feature descriptors. Such descriptors are often derived using supervised learning on existing datasets with ground truth correspondences. However, the cost of creating such datasets is usually prohibitive. In this paper, we propose a new approach to align two images related by an unknown 2D homography where the local descriptor is learned from scratch from the images and the homography is estimated simultaneously. Our key insight is that a siamese convolutional neural network can be trained jointly while iteratively updating the homography parameters by optimizing a single loss function. Our method is currently weakly supervised because the input images need to be roughly aligned.

We have used this method to align images of different modalities such as RGB and near-infra-red (NIR) without using any prior labeled data. Images automatically aligned by our method were then used to train descriptors that generalize to new images. We also evaluated our method on RGB images. On the HPatches benchmark [2], our method achieves comparable accuracy to deep local descriptors that were trained offline in a supervised setting.

1. Introduction

Finding pixel correspondences between multiple images is a fundamental problem in computer vision. It is a crucial ingredient in 3D reconstruction, object recognition, image analysis and editing. The correspondence problem is usually solved by extracting sparse or dense local feature descriptors in the images and matching the descriptors across images, sometimes using additional geometric constraints. Features, such as SIFT [29], SURF [6], DAISY [45] etc. provide partial invariance to scale, viewpoint and lighting change and have led to great progress in image matching.

Invariant descriptors could also be learned using data-driven approaches based on convolutional neural networks (CNN) [41, 54, 19, 3, 5, 25, 52]. These CNN-based descriptors are trained in a fully supervised setting using datasets containing ground truth image correspondence information [10]. Such datasets have been created by leveraging techniques for sparse and dense 3D reconstruction that relied on classical SIFT features [49]. However, these datasets only contain corresponding RGB image patches of the same scene. Therefore, they are unsuitable for training descriptors that can be used for aligning images of different modality [20, 40], or aligning semantically related images with different appearance[28, 43, 18, 35, 36]. The main difficulty is that recovering image correspondence and alignment in other situations requires datasets with ground truth that either do not exist or are prohibitively costly to acquire.

(a) (b) (c)

Figure 1: Aligning images of a farm taken with a UAV. (a) RGB (top) and NIR (bottom) input images. (b) Original alignment (top) and our result (bottom). (c) Zoom-in view.

In this paper, we investigate the problem of aligning two images of a rigid scene when neither handcrafted nor pre-trained features are available. Assuming that we know the images can be aligned using a specific family of geometric transforms, we propose to learn local feature descriptors from scratch while simultaneously estimating the parameters of the geometric transform specific to the image pair.

The core idea in our work is that of jointly learning the descriptor *i.e*. training a siamese convolutional neural network (CNN) and simultaneously estimating the parameters of the geometric transform. To that end, we propose to solve

a joint optimization problem where the updates to the geometric parameters are also computed using backpropagation in the same way as for the network parameters. In this work, we assume the transform to be a 2D homography and that the input images are roughly aligned, which makes our problem setting weakly supervised. We implement a multi-scale version of the optimization on image pyramids in order to handle images with greater initial misalignments.

Even though our training procedure resembles that for supervised learning with siamese networks, there are some important differences. First, the training sets for standard siamese networks are typically fixed and only the weights are updated using backpropagation and SGD. However, in our case, the set of positive pairs in our training set changes during training *i.e*. our estimate of the *true* correspondences gradually improves as the patches are resampled using the updated homography estimate computed in the previous iteration. The improvements to the homography also reduces the total training loss, as the progressively learned embedding produces a more well defined separation between the positive and negative pairs of patches from the two images.

Secondly, when an image pair is successfully aligned by our method, the trained network parameters are likely to be unreliable due to overfitting on the limited training data in the two images. However, with these alignments, we can now automatically build up a dataset with precise correspondences which could then be used to train a local descriptor with better generalization performance to new scenes. We evaluate this idea first for aligning RGB–NIR images from scratch. We also evaluate on color images in the HPatches benchmark [4] and show that our alignments computed from scratch given weak supervision are competitive to those computed by state of the art supervised methods that were trained offline on massive amounts of data.

2. Related Work

Local Descriptors. Hand-designed local features are still widely used in computer vision, *e.g*. SIFT [29], SURF [6], DAISY [45], DSP-SIFT [14] and A-SIFT [53]. While they offer partial invariance and computational efficiency, the use of learning in descriptor design promises even higher accuracy and robustness. Winder et al. [49] proposed to tune descriptor hyperparameters using a training set of ground truth correspondences to improve accuracy. Brown et al. [10] extended the framework to learn descriptors that minimized the error of a nearest neighbor classifier. Later, Simonyan et al [42] showed that further accuracy could be obtained using convex optimization to learn good strategies for pooling and dimensionality reduction.

Siamese Networks and CNNs. Bromley et al. [9] proposed siamese networks for verifying whether image pairs were related. These neural networks output feature vectors which can be compared using a suitable metric. Two identical networks (*i.e*. with parameters tied) are trained with matching and non-matching pairs and the network weights are tuned to learn invariance. Siamese networks were used to learn discriminative metrics [13], learn nonlinear dimensionality reduction [17] and has seen a resurgence in recent years for learning local descriptors, that we discuss next.

Simo et al. [41] use siamese networks to train CNN descriptors, proposing an efficient training procedure that searches for difficult positive and negative pairs and includes them while training their model iteratively. DeepCompare [54], MatchNet [19] both use CNNs to compute descriptors but also to compute the similarity score. In principle they are related to MC-CNN [47] proposed for dense stereo matching. These method tend to be more accurate but much more computationally expensive. DeepCompare [25], PN-Net [3] and TFeat [5] replace siamese networks with triplets networks. The triplet loss encourages positives pairs to have higher similarity compared to negative pairs such that the pairs share a data point which provides better context. LIFT [52] is yet another CNN model trained to predict 2D keypoints with orientations in addition to descriptors. All these methods use supervised learning and need ground truth correspondence. In contrast, we do not need ground truth but assume that the input images are related via an unknown 2D homography and are approximately aligned.

CNNs have been used in other ways for image correspondence problems. Deep Matching [48] is a multiscale semi-dense matching method that, inspired by CNNs, interleaves convolutions and max-pooling on image pyramids. Later, it was extended to a fully-trainable variant [44]. Rocco et al [35, 36] propose methods for semantical alignment of arbitary image pairs going from a supervised to a weakly supervised setting. Although, our work shares a similar motivation, our network is designed to learn local descriptors in the siamese setting whereas their CNN architecture are trained to learn global representations.

Datasets. The recent siamese networks have mostly been trained on the Multi-View Stereo Correspondence (MVS-Corr) dataset [10] built using multi-view stereo and structure from motion on Internet photo collections. Schmidt et al. [38] used dense RGB-D SLAM instead of multi-view stereo in a similar spirit and Bergamo et al. [7] leveraged correspondences obtained from structure from motion to train discriminative codebooks for place recognition. The HPatches dataset [4] was created to benchmark local descriptor performance and provides consistent evaluation metrics for different tasks like patch retrieval and image matching. Another recent benchmark [39] measures the impact of different descriptors by checking whether they lead to more accurate image-based 3d reconstructions.

Multi-modal image alignment. Mutual information is a widely used method for image alignment [46], popular for medical image registration [31]. However, the procedure

is iterative and relies on good initialization and more difficult to use on natural color images or when matching local patches [2]. While variational methods for multi-modal image matching were explored earlier on [20], recent focus has been on learning keypoints and descriptors suitable for multispectral images [16, 11] or developing robust matching costs [40] for dense multi-modal optical flow estimation. For multi-modal images, computing an automatic initialization robust to arbitrary differences in image scale, position, rotation etc. is still challenging although robust bootstrapping techniques have shown promise [50].

Other learning-based methods. Techniques for learning to align images have been studied before. Miller et al. [26] proposed *congealing*, a procedure to jointly align multiple images by minimizing entropy across stacks of aligned pixels. Huang et al. extended this line of work to the unsupervised setting using handcrafted features [22] and later by learning the features from scratch [21]. These methods work for well defined categories such as faces, and can be used to automate dataset creation [51]. Our work has a similar motivation but we focus on matching images of arbitrary scenes. Recently, the inverse compositional Lucas-Kanade algorithm was used with convolutional feature maps [12] and was also used to improve spatial transformer networks [27]. However, these models must be trained in a supervised manner. Our focus in this work is instead on weakly supervised learning for local descriptors.

3. Preliminaries

Let us briefly review how local descriptors are learned using siamese networks and how these networks are trained. The neural network takes square patches as input and outputs a feature descriptor vector. We denote $n \times n$ pixel patches taken from a c channel image as $\mathbf{x} \in \mathbb{R}^{n \times n \times c}$. The network outputs a d–dimensional vector that is denoted as $\boldsymbol{f}(\mathbf{x};\theta) \in \mathbb{R}^d$. Here, θ is a vector denoting the network parameters (or weights). We sometimes use $\boldsymbol{f}(\mathbf{x})$ instead of $\boldsymbol{f}(\mathbf{x};\theta)$ for notational brevity.

The training data consists of two sets of pairs of patches. Each patch pair is denoted by $\{\mathbf{x}, \mathbf{x}'\}$. The first set denoted by $\mathcal{P}$ contains true correspondences *i.e.* pairs of patches from different images that depict the same scene point or object. The negative set denoted by $\mathcal{N}$ contains pairs of patches that are not corresponding and depict different scene points and objects. Training the network involves learning an embedding, where $\|\boldsymbol{f}(\mathbf{x}) - \boldsymbol{f}(\mathbf{x}')\|_2$ is small for positive pairs $\{\mathbf{x}, \mathbf{x}'\} \in \mathcal{P}$, and the distance is large for negative pairs $\{\mathbf{x}, \mathbf{x}'\} \in \mathcal{N}$. The loss function used during training is called contrastive loss [41, 3, 5, 17]. It is defined for input pairs and has the following form.

$$\begin{aligned} \boldsymbol{L}_0(\mathbf{x}, \mathbf{x}'; \theta) &= \|\boldsymbol{f}(\mathbf{x};\theta) - \boldsymbol{f}(\mathbf{x}';\theta)\|_2 \\ \boldsymbol{L}_1(\mathbf{x}, \mathbf{x}'; \theta) &= \max(0,\, \mu - \|\boldsymbol{f}(\mathbf{x};\theta) - \boldsymbol{f}(\mathbf{x}';\theta)\|_2) \end{aligned} \tag{1}$$

The hyperparameter μ denotes a margin and is often set to the value 1. The loss function $\boldsymbol{L}_0$ encourages pairs of vectors to have smaller pairwise distances. In contrast, the hinge loss $\boldsymbol{L}_1$ encourages the pairwise distances to increase and imposes a penalty when those distances become smaller than the margin μ. The network parameters θ are computed by solving the following optimization problem.

$$\underset{\theta}{\operatorname{argmin}} \Big(\sum_{i=1}^{|\mathcal{P}|} \boldsymbol{L}_0(\mathbf{x}_i, \mathbf{x}'_i; \theta) + \sum_{j=1}^{|\mathcal{N}|} \boldsymbol{L}_1(\mathbf{x}_j, \mathbf{x}'_j; \theta) \Big) \tag{2}$$

Stochastic gradient descent (SGD) is often used for this problem and cross-validation is used to pick the best model.

4. Algorithm

Given an image pair $\{I, I'\}$, we assume that we know the family of 2D warping functions $\mathbf{w}(\cdot\,;\psi) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ with parameter vector ψ that transforms a pixel location or a patch in I to the corresponding location or patch in I'. We will now present our method to estimate the parameters ψ while simultaneously computing the network parameters θ. The resulting embedding $\boldsymbol{f}(\mathbf{x};\theta)$ will be specific to I and I'.

Before describing our weakly supervised method is full detail, we first discuss a simpler learning problem, that of learning θ when the parameter ψ is known. This problem can be solved using supervised learning. We then motivate how this method can be extended to the weakly supervised setting and finally present the proposed approach to learn ψ and θ jointly using a joint optimization framework. Finally, we also present a hybrid variant of siamese and pseudo siamese networks which is useful in cases where the input image modalities differ a lot *e.g.* RGB and NIR.

4.1. Descriptor learning from an aligned image pair

When the alignment parameters ψ for an image pair are known, we can use the warping function to extract corresponding patches from the images to train $\boldsymbol{f}(\mathbf{x};\theta)$. We first select a set of image patches in the first image I. For each such patch $\mathbf{x}$ in I, we find the warped patch $\mathbf{w}(\mathbf{x};\psi)$ in I' and insert them into the set of positive pairs $\mathcal{P}$. Similarly, we construct the set of negative pairs $\mathcal{N}$ by randomly sampling patches in I' whose location in the image are at least τ pixels away from the location of the true corresponding patch. The model can now be trained by minimizing a pairwise loss function, similar to the one in Equation 1.

$$\begin{aligned} \boldsymbol{L}_0(\mathbf{x}; \psi, \theta) &= \|\boldsymbol{f}(\mathbf{x};\theta) - \boldsymbol{f}(\mathbf{w}(\mathbf{x};\psi);\theta)\|_2 \\ \boldsymbol{L}_1(\mathbf{x}, \mathbf{x}'; \theta) &= \max(0,\, \mu - \|\boldsymbol{f}(\mathbf{x};\theta) - \boldsymbol{f}(\mathbf{x}';\theta)\|_2) \end{aligned} \tag{3}$$

The network parameters are learned by minimizing the following objective over the pairs in $\mathcal{P}$ and $\mathcal{N}$.

$$\underset{\theta}{\operatorname{argmin}} \Big(\sum_{i=1}^{|\mathcal{P}|} \boldsymbol{L}_0(\mathbf{x}_i; \psi, \theta) + \sum_{j=1}^{|\mathcal{N}|} \boldsymbol{L}_1(\mathbf{x}_j, \mathbf{x}'_j; \theta) \Big) \tag{4}$$

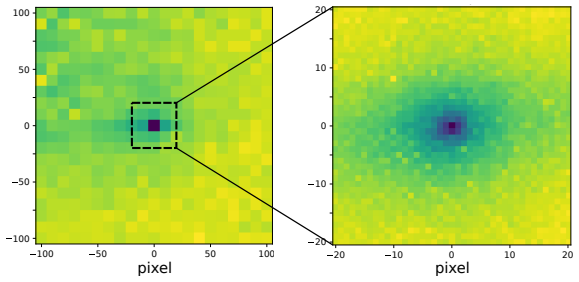

Figure 2: Visualization of final training losses (see Eq. 3) for supervised method on an image pair, where 2D translations (up to $\pm$ 100 pixels) were added to the true locations in the second image. Darker color indicates lower loss.

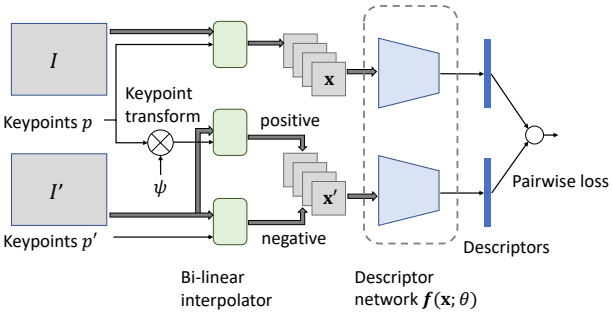

Figure 3: Overview of the proposed joint model.

This method still uses supervised learning but does so in an uncommon setting. The embedding $\boldsymbol{f}(\mathbf{x};\theta)$ is being learned from only a single image pair. Hence, the training set is small and the model is very likely to overfit to the data. When the alignment parameter ψ is accurate, the embedding is still meaningful for this image. We will now analyze what happens when the alignment is inaccurate and the correspondences it induces are imprecise.

4.2. From supervised to weakly supervised

To simulate the effect of imprecise alignment on an image pair, we added x and y translational offsets to the true warping function to generate several imprecise candidates. For each alignment candidate, we applied the supervised method just described previously to learn a different embedding $\boldsymbol{f}(\mathbf{x};\theta)$ from scratch and recorded the loss obtained at the end of training. Figure 2 shows a visualization of the training loss for all x and y offsets up to $\pm$ 100 pixels.

The visualization shows the effect of misalignment on the training loss. There is a well defined minimum at the center, *i.e.* when there is no misalignment. More over, the cost surface is smooth and has a stable gradient near the center and does not have any significant spurious local minima. This observation motivated the question. *Can we minimize the same pairwise loss to also iteratively estimate the alignment ψ as we train the neural network?*

This leads to our *self-supervised* method to jointly learn the descriptor and the warping parameters. We still use the pairwise loss function defined in Eq. 3, but the warping parameter ψ is no longer assumed to be known. Instead it is a variable in the loss function optimization.

$$\theta^*, \psi^* = \underset{\theta,\psi}{\operatorname{argmin}} \left(\sum_{i=1}^{|\mathcal{P}|} \boldsymbol{L}_0(\mathbf{x}_i;\psi,\theta) + \sum_{j=1}^{|\mathcal{N}|} \boldsymbol{L}_1(\mathbf{x}_j,\mathbf{x}'_j;\theta) \right) \tag{5}$$

Our self-supervised method does not need any transformation supervision. However, the training loss as shown will in general be nonconvex and local minima can exist. With iterative optimization techniques based on gradient descent, there is no guarantee of convergence to the correct estimate of ψ starting from an arbitrary initial value. Thus, our method is *weakly supervised*. It needs a reasonable initialization of ψ or requires the input images to be roughly aligned. While this sounds like a limitation, coarse alignment is available in many cases and our empirical evidence shows that our method can handle a fairly large amount of misalignment.

4.3. Joint alignment and weakly supervised learning

In this paper, we assume that the warping model $\mathbf{w}$ is based on a 2D homography. Thus, we can handle image pairs from a purely zooming or rotating camera or overlapping images of a planar scene from one or more cameras.

We now describe the joint optimization for the homography case in more details. To generate patches from an image, we extract multiple randomly sampled 2D keypoints from the image. Each keypoint $\mathbf{p}$ has a position (x, y), orientation ϕ and scale s, from which we can resample a $n \times n$ square patch using bilinear interpolation. Mathematically, we write this as $\mathbf{x} = \mathrm{B}(\mathbf{p}; I)$. To obtain corresponding patches, we transform the keypoint $\mathbf{p}$ in the image I using the homography to obtain the transformed keypoint $\mathbf{p}'$ in I' and then resample the patch associated with $\mathbf{p}'$. Here $\mathbf{w}_k$ is the same warping function as $\mathbf{w}$, but $\mathbf{w}_k$ transforms keypoints instead of patches, which will be detailed in Section 5.2. Mathematically, we have,

$$\mathbf{x} = \mathrm{B}(\mathbf{p}; I),\ \ \mathbf{w}(\mathbf{x};\psi) = \mathrm{B}(\mathbf{p}'; I') = \mathrm{B}(\mathbf{w}_k(\mathbf{p};\psi); I'). \tag{6}$$

Bilinear interpolation is differentiable with respect to the homography parameters. When we substitute $\mathbf{w}(\mathbf{x};\psi)$ from Eq. 6 into Eqs. 3 and 5, the new training loss remains differentiable with respect to θ and ψ. Thus, we can use backpropagation to compute the derivatives. Figure 3 illustrates our model. Since the parameter vector ψ gets iteratively updated, we compute the positive pairs using the updated homography from scratch and regenerate the positive training

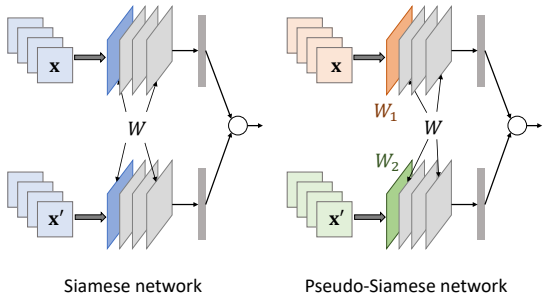

Figure 4: We train standard siamese networks and a special form of pseudo-siamese networks with unshared weights in the first layer to adapt to different image modalities.

set on-the-fly in each iteration. However, the set of negative pairs remains fixed throughout. The neural network architecture, implementation details of keypoint warping and the homography parametrization are discussed in Section 5.

4.4. Partially shared pseudo siamese network

Since siamese networks uses shared weights for both input pairs, they may perform poorly when the input image pairs have different modalities, *e.g.* RGB and NIR, since one set of parameters may not capture the statistics of both modalities. Similar ideas for unsharing network weights have been investigated in domain adaptation and transfer learning [37, 30]. In *pseudo-siamese* networks, both networks have the same structure but have different copies of the weights. They have been used for matching different modalities [1, 34]. but do not provide a significant gain when both inputs have the same modality [54]. With twice the parameters, pseudo-siamese networks tend to perform worse than siamese networks when training data is limited.

We explore a special case of pseudo-siameses networks where the weights of only the first layer of both networks are different *i.e.* the first layer is unshared, but the remaining parameters of the network are shared (see Figure 4). This allows the first layer to adapt to the different modalities but only adds a small number of parameters to the model. This is important for us since the training sets are quite small.

5. Implementation details

In this section, we present important details for implementing our approach and describe our CNN architecture.

5.1. Homography parameterization for SGD

Unlike second-order methods *e.g.* Gauss-Newton, first-order methods like SGD are not curvature-aware and have trouble optimizing ill-conditioned problems when different dimensions of the parameter space differ a lot in scale. The usual 8-DOF parameterization for the homography H has

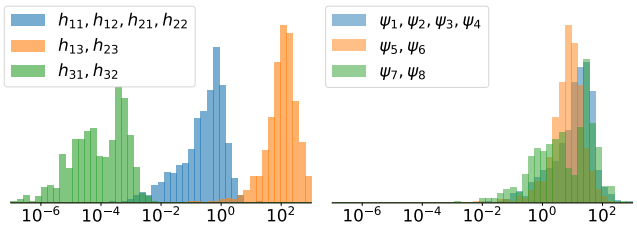

Figure 5: Homography parameter's distributions of dataset [4] in log-scale, left is before normalization in H, and right is normalization in ψ with $\alpha = 64$.

eight parameters often have very different scales.

$$H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix}. \tag{7}$$

Figure 5 shows the scale variation of the eight parameters for some homographies in the HPatches dataset [4]. Therefore, we choose a well-conditioned homography parametrization that is more suitable for use with SGD. Based on *dimensional analysis* [8] in physics, we check the *unit* of H the homography mapping although, there are no physical units.

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + 1}, y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + 1}. \tag{8}$$

Here, x, y, x' and y' are in pixels. So to reach dimensional homogeneity, $h_{11}, h_{12}, h_{21}, h_{22}$ must have *unit* 1, h_{13}, h_{23} must have pixel *unit*, and h_{31}, h_{32} must have pixel^{-1} *unit*. Thus, we normalize H using the image dimension $w \times h$ and this gives the following 8-dimensional vector for ψ.

$$\alpha\left[h_{11} - 1, h_{12}, h_{21}, h_{22} - 1, \frac{h_{13}}{w}, \frac{h_{23}}{h}, h_{31}w, h_{32}h\right] \tag{9}$$

Here, α is a scale hyperparameter that we set to 64. This parametrization is scale normalized (see Figure 5) and conveniently maps $\psi = 0$ to the identity homography.

5.2. Keypoint transformation under homography

The position (x', y') of a warped keypoint $\mathbf{p}'$ is given by Eq. 8. To find the orientation ϕ' and scale s' of the point $\mathbf{p}'$, we treat $\mathbf{p}$ as a 2D vector $[v_0, v_1]^T$ with orientation ϕ and length s with origin (x, y) in image I. Then, we have

$$v_0 \propto s\cos\phi, \qquad v_1 \propto s\sin\phi \tag{10}$$

Warping a keypoint is the same as finding the warped vector $(v_0', v_1')^T$ in I'. Assuming the vectors are short, we can use finite differences on Eq. 8 to compute the values of

$$v_0' \approx \frac{h_{11}v_0 + h_{12}v_1}{h_{31}x + h_{32}y + 1}, v_1' \approx \frac{h_{21}v_0 + h_{22}v_1}{h_{31}x + h_{32}y + 1} \tag{11}$$

The orientation and scale of $\mathbf{p}'$ can be computed by substituting values of v'_0 and v'_1 from Eq. 11 into these equations.

$$\phi' = \arctan\left(\frac{v'_1}{v'_0}\right), s' = \sqrt{(v'_0)^2 + (v'_1)^2} \quad (12)$$

5.3. CNN training

To obtain image patches to train our model, we select keypoint sets $\{\mathbf{p}\}$ and $\{\mathbf{p}'\}$ with sufficient randomization to reduce the effect of overfitting. We sample approximately 4000 keypoints in each image. The positions of these keypoints are sampled from areas with sufficient contrast by selecting areas were the local gradient magnitude exceeds 0.05[1] and selecting pixels randomly from a uniform distribution. The orientation is sampled uniformly from the range $[0, 2\pi]$ and the scale values are sampled such that $\log_2 s$ is distributed uniformly in the interval $[0, 4]$. We also check that the warped keypoint $\mathbf{p}'$ lies inside the boundary of image I'. Although the cropped patches extracted from $\{\mathbf{p}\}$ and $\{\mathbf{p}'\}$ spatially overlap, selecting the orientation and scales randomly provides effective data augmentation.

We use a shallow network architecture: `Conv(16,32,5,1)-Tanh-MaxPool(12,-,2,2)-Conv(6,64,3,1)-Tanh-FC(256)`. The parameters of each layer are denoted by (INPUT,CHANNEL,KERNEL,STRIDE). The patch size is 16×16 pixels and the descriptor dimension is $d = 256$. Our design is based on prior work [5, 41] but other architectures could be used as well. Since we have relatively fewer and small patches to train our model, we favor architectures with lower capacity to reduce the risk of overfitting. The optimization is done using SGD with batch size of 64 and momentum of 0.9, with an initial learning rate of 10^{-4} which is temporally annealed.

To address the issue of potentially large initial misalignments, the training is done in a coarse-to-fine fashion on an image pyramid, where the coarsest level has about 80 pixels on the longer side. The joint alignment and network training starts at coarsest pyramid level after which the estimated parameter ψ^* is used to initialize ψ at the next level. The network parameters θ^* are discarded as descriptors learned at coarser scales are not suitable for patches at finer scales. The parameters ψ^* and θ^* from the finest level are saved.

6. Experiments

We first evaluate our method on matching color images from HPatches [4] to compare with existing methods. We then test our method for aligning aerial RGB and NIR images that were captured using an off the shelf drone. Our implementation is based on Tensorflow [15]. All experiments were done using a NVIDIA GTX 1080Ti GPU, and aligning each image pair takes about 90 seconds.

[1] image intensities have mean 0 and standard deviation 1

On both datasets, we quantitatively evaluate ours descriptor to SIFT [29] and DAISY [45]. We also compare to four learned approaches – DeepCompare [54] with one (-s) and two stream (-s2s) siamese networks, DeepDesc [41] and DeepPatchMatch [25] which are currently amongst the top ranked methods on HPatches and were trained on the MVSCorr dataset [10]. We use the *mean average precision (mAP)* metric to evaluate descriptor performance. To evaluate an image pair (I,I'), we first extract many pairs of corresponding patches from them. We then extract descriptors from the patches. Then, for each descriptor extracted from I, we independently compute the exact nearest neighbor within the set of descriptors extracted from I', in the $L2$ distance sense. The average precision (AP) is the fraction of times the correct nearest neighbor was found. The mean AP (mAP) is the average AP across multiple image pairs.

6.1. Evaluation on the HPatches dataset

HPatches [4] contains image pairs rated at EASY, HARD and TOUGH difficulty levels for 116 scenes where 59 scenes have viewpoint variations and the rest vary in illumination. The evaluation protocol in HPatches involves calculating mAP on pre-extracted pairs of grayscale patches. We implement this protocol but using our own patches so that descriptors computed from RGB patches can also be evaluated. To generate the evaluation patches, we detect up to 2000 SIFT keypoints [29] in the first image, warp the underlying patches by the ground truth homographies and sampled the corresponding patches in the second image.

6.1.1 Descriptor performance evaluation

We evaluate four variants of our method for grayscale and RGB patches combined with the siamese network (**Ours-s**) and the pseudo siamese network (**Ours-ps**) respectively. The descriptor mAP evaluation is summarized in Table 1. All six baselines were based on grayscale patches. The coarse alignment needed by our method for initialization was simulated by adding random 2D translational perturbations to the true homography, where the shift is equal to 5% of the image size (see later for results with larger shifts).

In this evaluation, **Ours-s-Grayscale** is ranked lower than pretrained CNN but the gap in mAP is small – 0.57 (ours) *vs*. 0.59–0.62 [54, 41, 25]. This is expected, since those CNNs were trained offline on a massive patch dataset whereas in our case, the network is optimized from scratch on each image pair. SIFT and DAISY have much lower mAP of 0.39 and 0.47 respectively. However, our RGB networks have the highest overall accuracy. The pseudo siamese (mAP = 0.633) and the siamese network (mAP = 0.631) both had similar accuracies. The siamese variant **Ours-s-RGB** has the highest mAP in four out of the six subgroups. This evaluation shows that the descriptors learnt

Colorspace	Method	Viewpoint			Illumination			Average
		EASY	HARD	TOUGH	EASY	HARD	TOUGH	
Grayscale	SIFT [29]	0.485	0.297	0.192	0.549	0.431	0.372	0.388
	DAISY [45]	0.683	0.491	0.336	0.579	0.416	0.327	0.472
	DeepCompare-s [54]	0.781	0.542	0.361	0.779	0.632	0.532	0.605
	DeepCompare-s2s [54]	**0.803**	**0.581**	**0.396**	0.776	0.631	0.542	**0.622**
	DeepDesc [41]	0.781	0.554	0.360	**0.792**	**0.655**	0.547	0.615
	DeepPatchMatch [25]	0.770	0.541	0.359	0.760	0.610	0.509	0.592
	Ours-s-Grayscale	0.700	0.426	0.309	0.790	0.644	**0.549**	0.570
	Ours-ps-Grayscale	0.729	0.446	0.290	0.760	0.624	0.520	0.562
RGB	**Ours-s-RGB**	0.763	0.525	**0.397**	**0.813**	**0.679**	**0.607**	0.631
	Ours-ps-RGB	0.781	0.563	0.388	0.808	0.664	0.592	**0.633**

Table 1: HPatches evaluation: mAP for six baselines and variants of our method. For the six groups (columns), the best grayscale method is in bold. When the RGB method has a higher mAP than the best grayscale method, it is also in bold.

from scratch by our method are representative even though we do not expect them to generalize to new images.

6.1.2 Analyzing robustness to initial alignment error

We also analyze the robustness of our method to increasing amounts of error in the initial alignment. Figure 6 shows the results of these experiments. The plots show both the homography estimation error and the mAP scores for models trained on the six subgroups starting with a different amount of translational perturbation. The homography error is equal to the average warping error of the true feature matches under the estimated homography, after normalizing the error by the image size. The plots show that the homography error is very low for small perturbations (up to 7.5%) which indicates accurate alignment. The alignment error does increase with higher perturbation. However, it is worth noting that here, we adhere to the standard descriptor evaluation protocol and avoid RANSAC and geometric constraints to *robustly* estimate the homography which would have easily yielded more accurate alignments.

For perturbations between 0.0–0.1, the mAP curves are mostly flat *i.e*. the mAP drop is fairly small. The mAP drops by an amount between 0.01 and 0.04 for the six subgroups. In these plots, a perturbation of 0.1 is equal to a 2D shift of 10% of the image size. As the perturbation increases, the accuracy of our method drops gradually. At 0.2, the mAP is still reasonably high for the EASY and HARD groups whereas performance drops more for the TOUGH group.

6.2. Evaluation on RGB and NIR images

Next, we report our evaluation of RGB–NIR image alignment which has useful applications in precision agriculture (see example in Figure 1). The main difficulty here is due to frequent gradient reversal and the lack of correlation in the visible spectrum and the NIR band (770–810nm). We collected 50 RGB-NIR image pairs and manually annotated sparse correspondences in them and then recovered ground truth homographies. We split these into two sets – a training set of 40 pairs and a test set of 10 pairs. We evaluate the descriptor performance and homography estimation accuracy on the training set, which we present next.

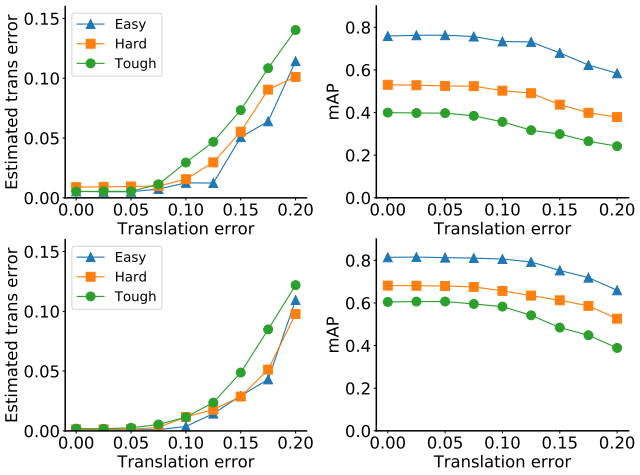

Figure 6: The average error of the estimated homography and mAP score of our method on EASY, HARD and TOUGH pairs for a range of initial translation errors (expressed as a fraction of the image size). Upper and lower rows are for "v" and "i" groups respectively in HPatches [4] (see text).

6.2.1 Descriptor performance evaluation

We calculate the mAP of our descriptors on the 40 training pairs using the same protocol used for HPatches. In each case, an identity transformation was used for initialization. We compared our method with the six baselines where the color patches were converted to grayscale. The first column of Table. 2 summarizes the descriptor performance evalua-

Method	RGB vs. NIR	
	Train Set(40)	Test Set(10)
SIFT	0.098	0.085
DAISY	0.030	0.040
DeepCompare-s	0.068	0.071
DeepCompare-s2s	0.101	0.098
DeepDesc	0.070	0.066
DeepPatchMatch	0.089	0.080
Ours-s	0.404	–
Ours-ps	**0.603**	–
Automatic-s (supervised)	–	0.475
Automatic-ps (supervised)	–	**0.556**

Table 2: Evaluation of RGB–NIR image alignment. The first column shows the mAP on the 40 training pairs where our method clearly outperforms all baselines. The second column shows the mAP on the 10 test pairs. Those descriptors were trained in supervised fashion on a training set generated using our automatic weakly supervised method.

tion for this case. Our method performs significantly outperforms all existing methods. The existing learned methods were not trained for these modalities and are not applicable when datasets with ground truth correspondences are unavailable. In this case, the pseudo-siamese network (mAP = 0.603) is much more accurate than the siamese network (mAP = 0.404) for the reason discussed earlier.

6.2.2 Analyzing robustness to initial alignment error

Once again, we evaluate our method by adding translational shifts to the ground truth alignment and simulate initializations of increasing difficulty. Since mutual information is sometimes used to align images of different modalities [46], we compare our pseudo-siamese network to an advanced mutual information based image registration method – `Elastix` [24] which uses Mattes' mutual information metric [32], with the same coarse to fine pyramid resolution as our method. The results are shown in Figure 7. With increasing translational perturbation, the mAP decreases gradually and the homography error increases as expected. However, our method is consistently more reliable than `Elastix`. In particular, our method has near-zero homography error (*i.e.* all pairs in training set are accurately aligned) up to a perturbation of 0.05 whereas `Elastix` is accurate only up to a much smaller perturbation of 0.025.

6.2.3 Automatic descriptor learning

We now evaluate the effectiveness of our method for automatically building a dataset without human annotation. We train a supervised descriptor on such a dataset and test the descriptor performance. We run our method on 40

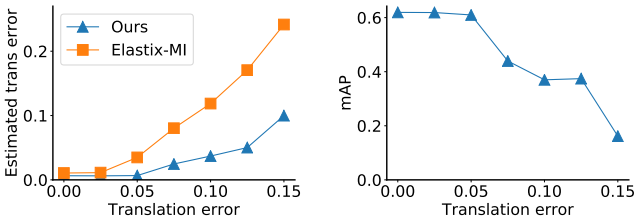

Figure 7: Homography errors (alignment accuracy) and mAP (descriptor performance) for our method and `Elastix` [24] which uses mutual information for image registration when the error in the initial alignment is varied. Our method can handle much more initial misalignment.

RGB/NIR image pairs. Using the homography obtained from each pair we extract correspondences. Specifically, SIFT keypoints extracted from the RGB images are warped by the estimated homography. We construct a set of 62K positive pairs in the training set. The supervised descriptor is a siamese CNN architecture (see Section 5.3), and trained using the same hyperparameters described earlier except that fewer training epochs were used.

This supervised descriptor network is evaluated using the same protocol as before but on the 10 test image pairs. The last two rows of Table. 2 shows the mAP for this method. The mAP of 0.556 is comparable to that of our weakly supervised method and much higher than all the existing methods. These results were obtained without tuning the network's architecture or its hyperparameters and show that our weakly supervised method is feasible for automatically learning local descriptors from coarsely aligned images.

7. Conclusion

We proposed a new weakly supervised method to align two images related by an unknown 2D homography when a coarse alignment is known. The key idea here is to train a local descriptor siamese network from scratch while jointly estimating the homography. We show how the descriptor parameters and the geometric parameters can be updated jointly within a single optimization. There are several avenues for future work. We will explore the idea of learning one network from several image pairs while estimating a unique alignment for each pair and also investigate extensions for making the method fully automatic. Finally, being able to handle general image pairs which have significant parallax would make the approach more widely applicable.

Acknowledgements. Part of the work was done while author Jing Dong was an intern at Microsoft Research. This work was also supported in part by National Institute of Food and Agriculture, USDA, under 2014-67021-22556.

References

[1] C. A. Aguilera, F. J. Aguilera, A. D. Sappa, C. Aguilera, and R. Toledo. Learning cross-spectral similarity measures with deep convolutional neural networks. In *CVPR Workshops*, pages 267–275, 2016. 5

[2] A. Andronache, M. von Siebenthal, G. Székely, and P. C. Cattin. Non-rigid registration of multi-modal images using both mutual information and cross-correlation. *Medical image analysis*, 12 1:3–15, 2008. 3

[3] V. Balntas, E. Johns, L. Tang, and K. Mikolajczyk. PN-Net: Conjoined triple deep network for learning local image descriptors. *arXiv preprint arXiv:1601.05030*, 2016. 1, 2, 3

[4] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 4, page 6, 2017. 2, 5, 6, 7

[5] V. Balntas, E. Riba, D. Ponsa, and K. Mikolajczyk. Learning local feature descriptors with triplets and shallow convolutional neural networks. In *BMVC*, volume 1, page 3, 2016. 1, 2, 3, 6

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008. 1, 2

[7] A. Bergamo, S. N. Sinha, and L. Torresani. Leveraging structure from motion to learn discriminative codebooks for scalable landmark classification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 763–770. IEEE, 2013. 2

[8] P. W. Bridgman. *Dimensional analysis*. Yale University Press, 1922. 5

[9] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a" siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, pages 737–744, 1994. 2

[10] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):43–57, 2011. 1, 2, 6

[11] M. Brown and S. Süsstrunk. Multi-spectral sift for scene category recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 177–184. IEEE, 2011. 3

[12] C.-H. Chang, C.-N. Chou, and E. Y. Chang. Clkn: Cascaded lucas-kanade networks for image alignment. In *CVPR*, 2017. 3

[13] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005. 2

[14] J. Dong and S. Soatto. Domain-size pooling in local descriptors: Dsp-sift. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 5097–5106. IEEE, 2015. 2

[15] M. A. et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 6

[16] D. Firmenichy, M. Brown, and S. Süsstrunk. Multispectral interest points for rgb-nir image registration. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 181–184. IEEE, 2011. 3

[17] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 1735–1742. IEEE, 2006. 2, 3

[18] B. Ham, M. Cho, C. Schmid, and J. Ponce. Proposal flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3475–3484, 2016. 1

[19] X. Han, T. Leung, Y. Jia, R. Sukthankar, and A. C. Berg. MatchNet: Unifying feature and metric learning for patch-based matching. In *CVPR*, pages 3279–3286, 2015. 1, 2

[20] G. Hermosillo, C. Chefd'Hotel, and O. Faugeras. Variational methods for multimodal image matching. *International Journal of Computer Vision*, 50(3):329–343, 2002. 1, 3

[21] G. Huang, M. Mattar, H. Lee, and E. G. Learned-Miller. Learning to align from scratch. In *Advances in neural information processing systems*, pages 764–772, 2012. 3

[22] G. B. Huang, V. Jain, and E. Learned-Miller. Unsupervised joint alignment of complex images. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 3

[23] J. Kim, V. Kolmogorov, and R. Zabih. Visual correspondence using energy minimization and mutual information. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1033–1040. IEEE, 2003.

[24] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. Pluim. Elastix: a toolbox for intensity-based medical image registration. *IEEE transactions on medical imaging*, 29(1):196–205, 2010. 8

[25] B. Kumar, G. Carneiro, I. Reid, et al. Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In *CVPR*, pages 5385–5394, 2016. 1, 2, 6, 7

[26] E. G. Learned-Miller. Data driven image models through continuous joint alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(2):236–250, 2006. 3

[27] C.-H. Lin and S. Lucey. Inverse compositional spatial transformer networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 2252–2260. IEEE, 2017. 3

[28] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. In *Dense Image Correspondences for Computer Vision*, pages 15–49. Springer, 2016. 1

[29] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 1, 2, 6, 7

[30] Z. Ma, Y. Yang, Y. Cai, N. Sebe, and A. G. Hauptmann. Knowledge adaptation for ad hoc multimedia event detection

with few exemplars. In *Proceedings of the 20th ACM international conference on Multimedia*, pages 469–478. ACM, 2012. 5

[31] F. Maes, A. Collignon, D. Vandermeulen, G. Marchal, and P. Suetens. Multimodality image registration by maximization of mutual information. *IEEE transactions on Medical Imaging*, 16(2):187–198, 1997. 2

[32] D. Mattes, D. R. Haynor, H. Vesselle, T. K. Lewellen, and W. Eubank. PET-CT image registration in the chest using free-form deformations. *IEEE transactions on medical imaging*, 22(1):120–128, 2003. 8

[33] D. Mishkin, J. Matas, and M. Perdoch. Mods: Fast and robust method for two-view matching. *Computer Vision and Image Understanding*, 141:81–93, 2015.

[34] L. Mou, M. Schmitt, Y. Wang, and X. X. Zhu. A CNN for the identification of corresponding patches in SAR and optical imagery of urban scenes. In *Urban Remote Sensing Event (JURSE), 2017 Joint*, pages 1–4. IEEE, 2017. 5

[35] I. Rocco, R. Arandjelovic, and J. Sivic. Convolutional neural network architecture for geometric matching. In *Proc. CVPR*, volume 2, 2017. 1, 2

[36] I. Rocco, R. Arandjelović, and J. Sivic. End-to-end weakly-supervised semantic alignment. *arXiv preprint arXiv:1712.06861*, 2017. 1, 2

[37] A. Rozantsev, M. Salzmann, and P. Fua. Beyond sharing weights for deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 5

[38] T. Schmidt, R. Newcombe, and D. Fox. Self-supervised visual descriptor learning for dense correspondence. *IEEE Robotics and Automation Letters*, 2(2):420–427, 2017. 2

[39] J. L. Schönberger, H. Hardmeier, T. Sattler, and M. Pollefeys. Comparative evaluation of hand-crafted and learned local features. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 6959–6968. IEEE, 2017. 2

[40] X. Shen, L. Xu, Q. Zhang, and J. Jia. Multi-modal and multispectral registration for natural images. In *European Conference on Computer Vision*, pages 309–324. Springer, 2014. 1, 3

[41] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, pages 118–126, 2015. 1, 2, 3, 6, 7

[42] K. Simonyan, A. Vedaldi, and A. Zisserman. Learning local feature descriptors using convex optimisation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1573–1585, 2014. 2

[43] T. Taniai, S. N. Sinha, and Y. Sato. Joint recovery of dense correspondence and cosegmentation in two images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4246–4255, 2016. 1

[44] J. Thewlis, S. Zheng, P. H. Torr, and A. Vedaldi. Fully-trainable deep matching. *arXiv preprint arXiv:1609.03532*, 2016. 2

[45] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):815–830, 2010. 1, 2, 6, 7

[46] P. Viola and W. M. Wells III. Alignment by maximization of mutual information. *International journal of computer vision*, 24(2):137–154, 1997. 2, 8

[47] J. Žbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *JMLR*, 17(65):1–32, 2016. 2

[48] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. DeepFlow: Large displacement optical flow with deep matching. In *IEEE Intenational Conference on Computer Vision (ICCV)*, Sydney, Australia, Dec. 2013. 2

[49] S. A. Winder and M. Brown. Learning local image descriptors. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 1, 2

[50] G. Yang, C. V. Stewart, M. Sofka, and C.-L. Tsai. Registration of challenging image pairs: Initialization, estimation, and decision. *IEEE transactions on pattern analysis and machine intelligence*, 29(11), 2007. 3

[51] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 3

[52] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned invariant feature transform. In *ECCV*, pages 467–483. Springer, 2016. 1, 2

[53] G. Yu and J.-M. Morel. Asift: An algorithm for fully affine invariant comparison. *Image Processing On Line*, 1:11–38, 2011. 2

[54] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, pages 4353–4361. IEEE, 2015. 1, 2, 5, 6, 7

[55] Y. Zhong, Y. Dai, and H. Li. Self-supervised learning for stereo matching with self-improving ability. *arXiv preprint arXiv:1709.00930*, 2017.

[56] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. *arXiv preprint arXiv:1704.07813*, 2017.