Deep Lucas-Kanade Homography for Multimodal Image Alignment

Yiming Zhao   Xinming Huang   Ziming Zhang

Worcester Polytechnic Institute

100 Institute Rd, Worcester, MA, USA

{yzhao7, xhuang, zzhang15}@wpi.edu

Abstract

Estimating homography to align image pairs captured by different sensors or image pairs with large appearance changes is an important and general challenge for many computer vision applications. In contrast to others, we propose a generic solution to pixel-wise align multimodal image pairs by extending the traditional Lucas-Kanade algorithm with networks. The key contribution in our method is how we construct feature maps, named as deep Lucas-Kanade feature map (DLKFM). The learned DLKFM can spontaneously recognize invariant features under various appearance-changing conditions. It also has two nice properties for the Lucas-Kanade algorithm: (1) The template feature map keeps brightness consistency with the input feature map, thus the color difference is very small while they are well-aligned. (2) The Lucas-Kanade objective function built on DLKFM has a smooth landscape around ground truth homography parameters, so the iterative solution of the Lucas-Kanade can easily converge to the ground truth. With those properties, directly updating the Lucas-Kanade algorithm on our feature maps will precisely align image pairs with large appearance changes. We share the datasets, code, and demo video online [1].

1. Introduction

Pixel-wise alignment of multimodal image pair is an important problem for many computer vision applications in the field of medical imaging [12], remote sensing [45], and robotics [5]. For example, in the task of GPS denied navigation [16], the drone needs to align the RGB image from the camera with the existing map from the satellite. Aligning those two images suffers from multimodal color representations as well as appearance changes. In Fig. 1, we show a cross-season image pair with appearance changes and a Google Map and Satellite image pair with multimodal pat-
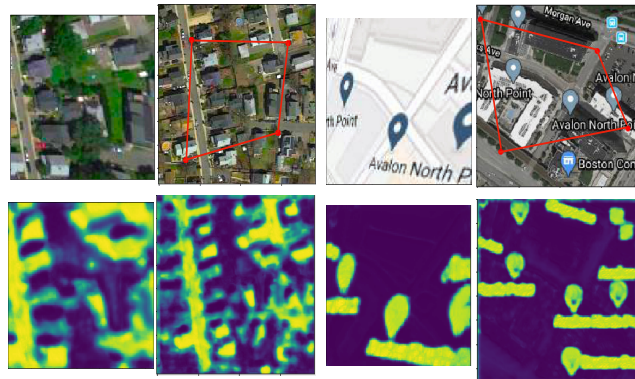

Figure 1. **(Left)** One image pair from our cross-season dataset captured in Google Earth. Plant and traffic are changing textures. Our DLKFM distinguishes invariant features which are roads and buildings. **(Right)** One image pair from our Google Map and Satellite dataset. Template comes from the static map and the input is the corresponding satellite map. Our DLKFM distinguishes invariant features which are those map labels. The red polygon is the ground truth position of the template on the input image.

tern representations. To solve this problem, we propose a generic pipeline by extending the traditional Lucas-Kanade (LK) algorithm with neural networks. The key component is how to extract the feature map, named as deep Lucas-Kanade feature map (DLKFM). DLKFM is able to spontaneously recognize invariant features in various multimodal cases, like the roads and houses in the cross-season example or map labels in the Google Map and Satellite example. DLKFM also helps the LK to converge successfully by shaping the landscape of the objective function.

Homography or perspective transform is a general 2D image transformation, which maps pixel from one image to the other, denoted as $\hat{x} = \hat{H}\hat{x}'$, where $\hat{x}$ and $\hat{x}'$ are homogeneous coordinates, $\hat{H}$ is an arbitrary 3×3 matrix. However, in 3D computer vision, parallax arises due to the change of the viewpoint, thus creates ambiguity for aligning 2D images. In this paper, we consider the same flat world assumption as [16], that the depth disparity is negligible like some UAV and remote sensing applications [32, 46]. Then, aligning the image pair is equivalent to estimating the ho-

---

[1]Codebase link: https://github.com/placeforyiming/CVPR21-Deep-Lucas-Kanade-Homography. Dr. Xinming Huang is the project manager of this paper, please contact him for further information.

mography matrix.

To estimate the homography, a straightforward solution is finding more than four matched points. This idea can be broadly summarized as the feature-based method including traditional SIFT [27], SURF [3], or recent deep learning methods D2 [13], LF-Net [34], R2D2 [37], etc. To further solve the challenge on multimodal images, some special features are also designed [45, 44]. However, those feature-based methods are facing the same generalizability problem. Even for those learning-based features, ground truth labels provided by 3D reconstruction systems are necessary [38, 39]. This prohibits the generic training of those feature detectors and descriptors on multimodal images.

Besides feature-based methods, direct method is the other way to calculate homography [2]. As the most famous one, the Lucas-Kanade family [28] has a long history and still stimulates new ideas in different aspects of computer vision [35, 43]. In this paper, we build a generic multimodal image alignment pipeline by extending Inverse Compositional Lucas-Kanade (IC-LK).

Inverse Compositional Lucas-Kanade Algorithm. Let X_I and X_T represent input image and template image, respectively, the Lucas-Kanade objective is stated as follows

$$\min_P ||X_T - W(X_I, P)||_2^2 \qquad (1)$$

where $W(\cdot|P)$ is the warping function which warps the input image based on transformation parameter set P.

The Lucas-Kanade algorithm iteratively solves for the warp parameters $P_{k+1} = P_k + \Delta P$. At every iteration k, the warp increment ΔP is obtained by linearizing

$$\min_{\Delta P} ||X_T - W(X_I, P_k + \Delta P)||_2^2 \qquad (2)$$

using first order Taylor expansion

$$\min_{\Delta P} ||X_T - W(X_I, P_k) - \frac{\partial W(X_I, P_k)}{\partial P} \Delta P||_2^2 \qquad (3)$$

Here, $\partial W(X_I, P_k)/\partial P$ needs to be recomputed at every iteration as it depends on $W(X_I, P_k)$. The inverse compositional(IC) [2] avoids this recomputing by applying the warp increments ΔP to the template instead of the input

$$\min_{\Delta P} ||W(X_T, \Delta P) - W(X_I, P_k)||_2^2 \qquad (4)$$

using the warp parameter update $P_{k+1} = P_k + (\Delta P)^{-1}$ as where ΔP is the inverse increment mapping. In the corresponding linearized equation

$$\min_{\Delta P} ||X_T + \frac{\partial W(X_T, 0)}{\partial P} \Delta P - W(X_I, P_k)||_2^2 \qquad (5)$$

$\partial W(X_T, 0)/\partial P$ does not depend on P_k and can thus be pre-computed, resulting in a more efficient algorithm.

Limitation of IC-LK on Multimodal Images. IC-LK keeps accumulating ΔP toward ground truth, thus it usu-ally has an accurate performance if it can work successfully. However, Eq. 1 assumes the color difference between the input image and template image achieves the minimum value once they are aligned. This brightness consistency assumption does not hold when aligning multimodal image pairs as the input and template usually have different color pattern representations. Furthermore, minimizing Eq. 1 is a non-linear optimization task as the input image depends non-linearly on the warp parameters P, so there is no guarantee the first-order approximation solution of Eq. 5 can have a ΔP moving along the right direction to the ground truth without getting into the local minimum.

Our Solution. To overcome the limitation of IC-LK on multimodal images, we hope feature maps processed after a Siamese network can satisfy those assumptions of IC-LK. To achieve this, we design a feature constructor inspired by classical feature extractors, such as Harris Corner [9]. This feature constructor constructs one channel feature map by using eigenvalues of the local covariance matrix on output tensor. Then, we design a special loss function with two terms. The first term helps feature maps to satisfy the brightness consistency assumption. The second term shapes the landscape of the Lucas-Kanade objective function, to help the first-order linearized solution ΔP can successfully keep moving toward the ground truth homography parameters. We show examples of learned deep Lucas-Kanade feature map (DLKFM) in Fig. 1.

Contributions. In general, we extend the well-known traditional Lucas-Kanade method with neural networks. In regard to applications [16, 46, 31], we propose a generic pipeline to align multimodal images. In regard to technical contributions, we develop two new specific technologies:

- *Feature constructor on multi-channel tensors.* In order to avoid updating homography on heavy tensors, the single-channel DLKFM is constructed on the high dimensional network output. We calculate the covariance matrix of vectors in each 3×3 patch. Then, we approximate the ratio between the largest eigenvalue and the trace as an indicator to form a single channel feature map. In the ablation study section, we show this operation improves the performance by extracting meaningful features on the single-channel feature map.

- *Special loss function for the convergence of Lucas-Kanade.* We design a special loss function based on constructed feature maps. This loss function makes two feature maps satisfy brightness consistency. It further helps the convergence of the Inverse Compositional Lucas-Kanade algorithm by shaping the optimization objective with a supportive convex function. The landscape of the objective function on learned feature maps is visualized in the section of ablation study.

2. Related Work

Image alignment is one of the most fundamental computer vision tasks with a long history [2, 41, 16]. Direct methods [28], feature-based methods [25] and their combinations [1] are traditional ways to align images. Those non-learning methods often assume input and template images have the same texture pattern for the same feature, which is not the case for multimodal images. To overcome this, some researchers try to design special feature detectors [45, 44], which are usually hard to prove the generalizability.

With the coming of the deep learning epoch, people are curious if the deep neural network can learn the corresponding features on two images [26, 8]. The positive answer attracts many recent research papers investigating how to utilize the strong ability of neural networks for image matching, like semantic matching or finding dense correspondences [19, 20, 42]. Extending feature detectors and descriptors with the neural network is also a hot topic. Recent models, like D2 [13], LF-Net [34], R2D2 [37], RF-Net [40], Elf [4] are all showing strong ability to handle struggling cases for traditional feature detectors. However, many of those methods rely on ground truth feature points to train the network. Those ground truth feature points are traditional features extracted by 3D reconstruction systems, like Colmap [38, 39]. This limits the training of those methods on multimodal datasets.

A more straightforward idea is making use of the strong approximation ability of networks to directly regress homography parameters [10, 14]. Those deep homography papers usually focus on specific challenges, like how to design the unsupervised loss [32], or how to handle dynamic scenes [23, 47]. Although using networks to do regression is simple, it always can work under various conditions. However, the model accuracy is usually not guaranteed.

In contrast to above methods, we extend the traditional Inverse Compositional Lucas-Kanade method on multimodal image pairs. Two recent papers explore Lucas-Kanade with neural networks [6, 29]. Authors of one paper replace some parts of the inverse compositional solution with trainable data-driven network modules [29]. However, their method mainly focuses on solving the 3D rigid motion estimation problem. In CLKN [6], authors design a loss function to guarantee the first updating always bring relative large progress, the further iterative updating is conducted on feature maps. Since there is no special treatment for feature maps, they imply the same color pattern will become the same feature via same network parameters. This assumption does not work for multimodal image pairs as two images have different color textures for the same place.

3. Method

3.1. Problem Settings

To overcome the limitation of Lucas-Kanade on multimodal image pairs, we extend the Inverse Compositional Lucas-Kanade on feature maps. A Siamese network, which has one template branch defined as $F(\cdot|w_T)$, and one input branch defined as $G(\cdot|w_I)$, is used to extract feature maps. Then, the objective function in Eq. 1 can be generalized as:

$$\min_P ||F(X_T, \bar{w_T}) - W(G(X_I, \bar{w_I}), P)||_2^2. \tag{6}$$

Here, we have two kinds of set of parameters, the Siamese network parameter set $\{w_T, w_I\}$, and homography parameter set P. During training, we use ground truth $\bar{P}$ to train $\{w_T, w_I\}$, then the fixed network weights $\{\bar{w_T}, \bar{w_I}\}$ are used to update P during inference to minimize Eq. 6. The iteration inverse compositional solution of Eq. 6 is

$$\Delta P = (J^T J)^{-1} J^T r, \tag{7}$$

where

$$J = \nabla F(X_T, \bar{w_T}) \cdot \frac{\partial W}{\partial P}(W(F(X_T, \bar{w_T}), 0), \tag{8}$$

$$r = \begin{bmatrix} W(G(x_I, \bar{w_I}), P)_1 - F(X_T, \bar{w_T})_1 \\ ... \\ ... \\ W(G(X_I, \bar{w_I}), P)_N - F(X_T, \bar{w_T})_N \end{bmatrix} \tag{9}$$

The derive details from Eq. 6 to Eq. 7 can be found from equation (2) to (8) here [6]. Since the updating of Eq. 7 is conducted on trained feature maps during inference, we use $\{\bar{w_T}, \bar{w_I}\}$ to represent fixed network weights during inference. In Eq. 9, N is the total number of elements on the feature maps.

3.2. Construct Single Channel Feature Map

The modern convolution layers usually output a high dimension tensor with multiple filters, like 64, 128, or more. This involves a large computational burden to update ΔP in Eq. 7 and Eq. 9. Some recent papers, which also try to involve neural networks into the pipeline of Lucas-Kanade, usually take straightforward choices like reducing the number of filters in the last layer to a small value [6], or applying channel-wise summation on the output tensor [29]. In contrast to them, we construct a single channel feature map based on eigenvalues of the local covariance matrix.

Considering a large tensor B, we apply a 3×3 sliding window to extract 9 vectors locally. Then, we calculate the covariance matrix of those 9 vectors. We use $B_{i,j}$ to denote the covariance matrix calculated by the local patch centered

at (i, j) of tensor B. Intuitively, if the information captured by this local patch is almost the same, those 9 vectors should be similar. Thus, the covariance matrix will not have a dominant eigenvalue. So we construct a one channel feature map C by using the ratio of the largest eigenvalue over the summation of all eigenvalues:

$$C_{i,j} = \frac{\lambda_{largest}}{Trace(B_{i,j})}, \qquad (10)$$

where $\lambda_{largest}$ is the largest eigenvalue of $B_{i,j}$. In traditional image processing, we as humans think corners or edges are important features for an image, like the Harris corner detector [9] or Canny edge detector [11]. In many different subjects which need matrix analysis, like statistics and signal processing, the ratio of largest eigenvalue over trace plays an important role [30]. So we think this ratio is an important feature for high dimensional tensor, just like the edge is an important feature for the image.

However, we found calculating and doing backpropagation on the largest eigenvalue for each patch is super slow with modern deep learning frameworks. Then, we design a more practical indicator by following Perron–Frobenius theorem [36, 15]. According to Perron–Frobenius theorem, the largest eigenvalue of the covariance matrix has an upper bound which is the maximum of summation of each row, and a lower bound which is the minimum of summation of each row. So the new feature map of a high dimension tensor is constructed by:

$$C_{i,j} = \frac{max_m(\sum_n B_{i,j}^{m,n}) + min_m(\sum_n B_{i,j}^{m,n})}{2 \times Trace(B_{i,j})}. \qquad (11)$$

where m, n are row and column numbers of the covariance matrix. The calculation of this feature map is just like doing regular convolution, which keeps our method practical for training and testing. We discuss more details in the section of the ablation study.

3.3. Build Loss Function

Brightness Consistency Loss Term. Once we have two feature maps from the template branch and the input branch in the Siamese network, we need to design a loss function to help those two feature maps learn meaningful features for Eq. 7. The original Lucas-Kanade objective Eq. 1 assumes that the color difference should be the minimum once two images are well aligned. We apply this assumption on feature maps to get the first term of our loss function named Brightness Consistency Loss $l_{bc} = E(w_T, w_I|\bar{P})$, where

$$E(w_T, w_I|\bar{P}) = ||F(X_T, w_T) - W(G(X_I, w_I), \bar{P})||_2^2 \qquad (12)$$

Compared with Eq. 6, l_{bc} has trainable $\{w_T, w_I\}$ with fixed ground truth $\bar{P}$ on training set. The warping function is

implemented by using STN [18] for backpropagation.

Convergence Loss Term. Original objective Eq. 1 is a highly non-linear function, whereas the iteration solution Eq. 7 relies on gradient descent approximation. To help it converge better in the right direction, we hope the objective Eq. 6 on feature maps can achieve local minimum at ground truth $\bar{P}$ and have a smooth surface around $\bar{P}$ without the other minimum. Control the convexity of a function usually needs to explicitly calculate Jacobian or Hessian, but Eq. 6 is a multivariable function of P with a Siamese network as the backbone. Instead of directly calculating Jacobian or Hessian, we consider a supportive convex function $g(Z)$ with $J(g(Z))|_{Z=\bar{P}} = 0$ to shape the local region of objective function around $\bar{P}$. The Z is a vector with eight parameters and $g(Z)$ has the minimum value at $Z = \bar{P}$. We push the optimization objective function Eq. 6 satisfy two conditions in a small region Θ around $\bar{P}$ by training:

$$Condition1 : \forall (\bar{P} + \Delta P) \in \Theta,$$

$$E(\bar{P}+\Delta P|\bar{w_T}, \bar{w_I}) - E(\bar{P}|\bar{w_T}, \bar{w_I}) \geq g(\bar{P}+\Delta P) - g(\bar{P}).$$

This condition guarantees that Eq. 6 achieves local minimum at $\bar{P}$ with a relatively larger local change than $g(Z)$.

$$Condition2 : \forall P \in \Theta, \nabla_v E(P|\bar{w_T}, \bar{w_I}) \geq \nabla_v g(P),$$

where ∇_v is the directional derivative. Since *Condition 1* is about the single point at $\bar{P}$, *Condition 2* guarantees that Eq. 6 has a steeper directional derivative than $g(Z)$ in a local region around $\bar{P}$. Compared with partial derivative, directional derivative is easier to be implemented for function without explicit expression.

In our implementation, we use the simplest $g(Z) = \sum_{i=1}^{8}(Z_i - \bar{P}_i)^2$ as the supportive function. Then, the loss term for *Condition 1* can be defined as:

$$l_{con1} = -\frac{1}{M} \sum_{m=1}^{M} minimum(0, E(\bar{P} + \Delta P_m|w_T, w_I)$$

$$- E(\bar{P}|w_T, w_I) - \sum_{i=1}^{8}(\Delta P_i^m)^2), \qquad (13)$$

where ΔP_i^m is a sampled random noise of the i-th variable in vector P, we sample M times for each training batch.

For *Condition 2*, we define $\forall (\bar{P} + \Delta P) \in \Theta$, $P = \bar{P} + \lambda \Delta P$, where $\lambda \in (0, 1)$. Then, we have:

$$\nabla_{\Delta P} E(P|\bar{w_T}, \bar{w_I}) = \lim_{(1-\lambda) \to 0}$$
$$\frac{E(\bar{P} + \Delta P|\bar{w_T}, \bar{w_I}) - E(\bar{P} + \lambda \Delta P|\bar{w_T}, \bar{w_I})}{(1-\lambda)||\Delta P||}, \qquad (14)$$

$$\nabla_{\Delta P} g(P) = \lim_{(1-\lambda) \to 0} \frac{\Delta P^2 - (\lambda \Delta P)^2}{(1-\lambda)||\Delta P||}, \qquad (15)$$
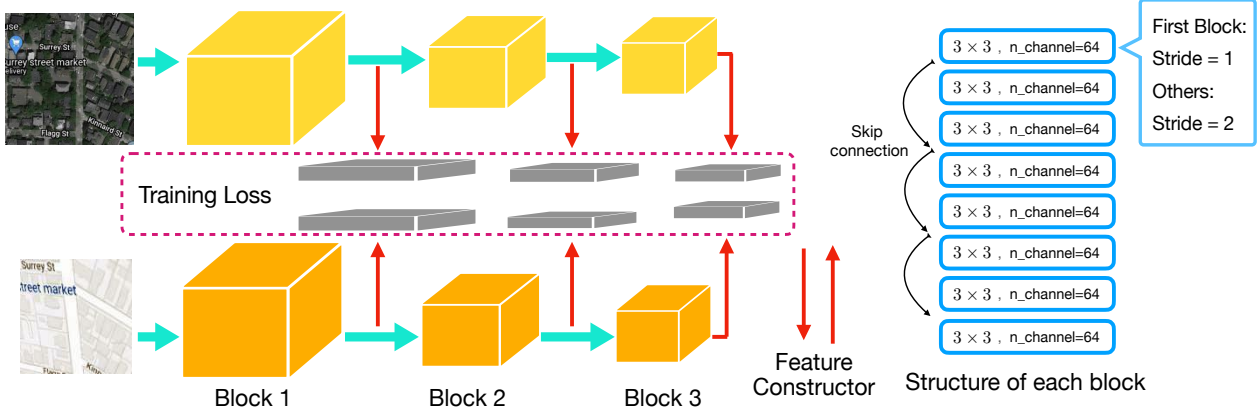
Figure 2. The Siamese network has three identical blocks except for the stride in the first layer. The first layers in the second and third block have stride 2 to downsample the input tensor. We construct one channel feature maps on the output tensors of each block. The red arrow represents the feature constructor developed in this paper. Each block contains a few layers with residual skip connection. The loss is built on constructed feature maps during training.

Combining Eq. 14 and Eq. 15, we have the mathematical expression of *Condition 2* as

$$\forall (\bar{P} + \Delta P) \in \Theta, E(\bar{P} + \Delta P | \bar{w}_T, \bar{w}_I)$$
$$- E(\bar{P} + \lambda \Delta P | \bar{w}_T, \bar{w}_I) \geq (1 - \lambda^2) \sum_{i=1}^{8} (\Delta P_i^m)^2. \quad (16)$$

The loss term for *Condition 2* can be defined as :

$$l_{con2} = -\frac{1}{M} \sum_{m=1}^{M} minimum(0, E(\bar{P} + \Delta P_m | w_T, w_I)$$
$$- E(\bar{P} + \lambda \Delta P_m | w_T, w_I) - (1 - \lambda^2) \sum_{i=1}^{8} (\Delta P_i^m)^2). \quad (17)$$

Since we have $\lambda \in (0, 1)$ and $(1 - \lambda) \to 0$, we choose $\lambda = 0.8$ for the implementation.

Finally, the Convergence Loss is defined as $l_{converge} = l_{con1} + l_{con2}$. The total loss function is the summation of the Brightness Consistency Loss and the Convergence Loss

$$l_{total} = l_{bc} + \gamma l_{converge} \quad (18)$$

where γ is a hyperparameter to control the balance. We choose $\gamma = 0.1$ in this paper.

3.4. Network Structure

We show the network structure in Fig. 2. The Siamese network has three identical blocks except for the first layer. In each block, there are 8 convolution layers with 64 filters connected with the residual. The first layers in the second and third blocks have stride 2 to downsample tensors. All the other layers have 3×3 kernels with stride 1.

During training, the input image and template image go through two branches. Then, we extract output tensors after each block, and apply Eq. 11 to construct feature maps for both input and template. The loss function Eq. 18 is defined based on constructed feature maps in three different scales.

During inference, we have feature maps in three different scales. The Eq. 7 is updated bottom-up from coarse to fine. We give an example of the updating during inference in Fig. 3. Note that the iterative solution needs an initial guess to start with and a stopping criterion to end, we discuss those two in the section of experiments.

4. Experiments

4.1. Datasets

To test the performance of our method, we prepare three datasets. We utilize Adam [21] as our optimizer with a learning rate 0.0001 to train 10 epochs for all three datasets. As discussed in the section of method, we set $\gamma = 0.1$, $\lambda = 0.8$ and $M = 4$. All the experiments are conducted on a single RTX 2080 Ti.

MSCOCO [24] is a widely used large scale dataset. Recent deep homography papers also choose this dataset as a test bench. We exactly follow the settings of those papers [10, 6, 23]. The whole training set of MSCOCO is used for our training, and we sample 6K images from the validation set as our test. Each sample is resized to 196×196 as an input image. Then, we randomly choose four points in four 64×64 boxes at the corner and warp the chosen region to 128×128 as the template. An example is showed in Fig. 4. The use of this regular dataset is serving for two purposes. First, since we implement several baseline methods, including DHN [10], MHN [23], SIFT with Ransac, etc, observing the performance drop from the same model on multimodal datasets will illustrate the challenge of estimating homography on multimodal images. Second, we want to show our method can also work on the regular image alignment task.
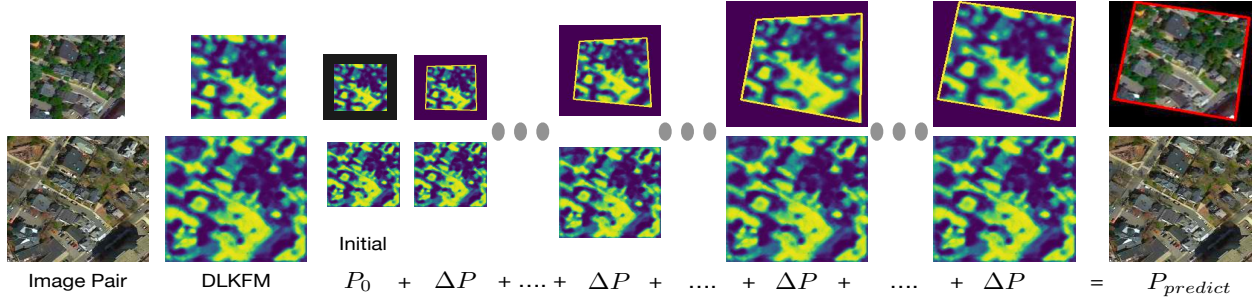
Figure 3. For a given initial guess, the Eq. 7 is used to calculate ΔP on input and template DLKFM for updating P. The updating starts from the smallest scale with a certain stop criterion on each scale. More examples of iterative updating are shown in the demo video.

Figure 4. From left to right: MSCOCO, Google Earth cross season dataset, Google static map and satellite dataset. We show how to prepare the input and template images for those three datasets.

Google Earth provides high-resolution satellite images captured on different dates. So, we can choose images for the same place captured in different seasons. Here we are using satellite images saved on 04/2018 and 06/2019 about the Great Boston area. One sample is provided in the second row of Fig. 4 to show that input and template images have appearance changes like different traffic and plant conditions. We randomly sample around 8K images as training, and around 1K as testing. Since randomly sampling will choose some completely textureless regions, like the surface of the river, we manually remove some images to finally get 850 test samples.

Google Maps and Satellite are multimodal images provided by Google Static Map API. Two corresponding images belong to static google maps and satellite maps respectively. One example is shown in the third row of Fig. 4, we can see they have completely different color pattern representations of the same place. We randomly sample around 8K images in the same place as training. For testing, we follow the same way as cross-season Google Earth dataset, to prepare 888 sample pairs finally.

4.2. Evaluation Metric

The same as recent papers [6, 23], we are using average corner error in pixel as the evaluation metric:

$$e_c(P, \bar{P}) = \frac{1}{4} \sum_{j=1}^{4} ||W(e_j, P) - W(e_j, \bar{P})||_2, \quad (19)$$

where $e_1, ..., e_4$ are four corners in template image. The corner error measures the L_2 distance of the warped corners and then takes the average over the four corners.

4.3. Initial Homography

Updating Eq. 7 on DLKFM needs an initial homography P_0. The initial used in Fig. 3 is a pre-assigned homography matrix that we simply set (31,31), (31,159), (159,159), (159,31) as four corners of a 128×128 template image on 192×192 input. Note, the closer the initial homography to the ground truth, the larger chance Lucas-Kanade will converge to it. In some applications, like tracking [35] or navigation [7], the previous frame can serve as the initial candidate since the relative pose is small. Some traditional methods, like fast affine template matching [22], can also work to provide an initial guess with the brightness consistency property of our feature maps.

In this paper, we choose two recent network-based deep homography models, DHN [10] and MHN [23], to respectively provide the initial homography. This helps our solution achieve state-of-the-art performance on recent benchmark settings [6, 23]. It is also able to show the strong ability of our method to work as a post-process refiner that keeps improving the alignment accuracy based on existing or future solutions.

4.4. Stopping Criterion

The iterative solution Eq. 7 needs a stopping criterion during the inference. For each ΔP on P_k, the average four corner error can still be used to evaluate relative improvement of ΔP as $e_c(P_k + \Delta P, P_k)$. In this paper, we set the stopping criterion as $e_c(P_k + \Delta P, P_k) < 1$ pixel for smallest scale, $e_c(P_k + \Delta P, P_k) < 0.1$ pixel for middle scale, and
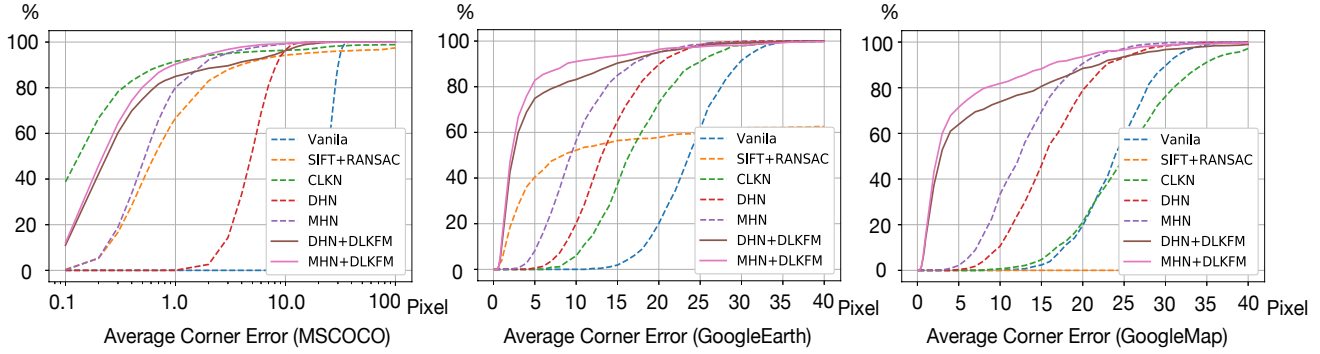
Figure 5. The x-axis is the mean average pixel error, the y-axis is the cumulative percentage of test images which have lower average pixel error than x. Since almost all methods work well on MSCOCO, the x-axis is in a log scale for this dataset. We use two regression methods DHN [10] and MHN [23] to provide the initial P_0 as they can work on all three datasets. It is clear to see that updating Eq. 7 on our deep Lucas-Kanade feature map (DLKFM) significantly improves the alignment accuracy.

$e_c(P_k + \Delta P, P_k) < 0.01$ pixel for largest scale. To further control the inference time, the maximum iteration number on each scale is 30.

4.5. Model Performance

We show the performance of our method and all baseline methods in Fig. 5. In general, our solution outperforms the others on multimodal datasets with a large margin. We give further discussions about each method below.

Vanilla. Based on how we generate templates, the vanilla means centers of four corner boxes are used as the prediction. This is a baseline for guessing without knowledge.

SIFT+RANSAC. This is a typical traditional method to calculate homography. We use OpenCV implementation of SIFT, BF Matcher, and RANSAC. From Fig. 5, we can see it achieves pretty accurate performance on the MSCOCO dataset which has the exact same template and input images. However, on cross-season Google Earth dataset, it performs better than vanilla for only 60% of images, and it completely fails to calculate homography on Google Maps.
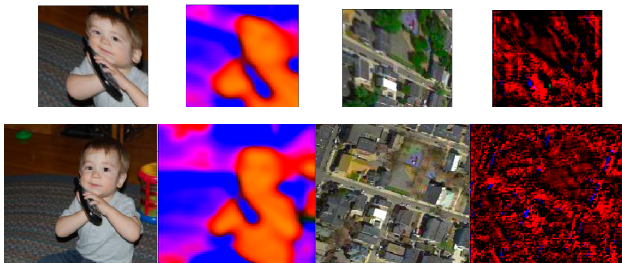

Figure 6. This figure illustrates why CLKN [6] can not work on multimodality case. (**Left**): the images and learned feature maps of CLKN [6] on MSCOCO, we can see that feature maps extract meaningful semantic information. (**Righ**): the images and learned feature maps on cross season Google Earth, we can see the learned feature maps are no longer meaningful.

CLKN [6]. This method extends inverse compositional Lucas-Kanade on feature maps. Based on vanilla as initial, it relies on training to help the first updating get relatively large progress. However, since CLKN [6] does not set any constrain on feature maps, it fails to align multimodal images. We show the feature maps learned by CLKN on the MSCOCO dataset and the Google Earth cross-season dataset respectively in Fig. 6. It is clear to see learned feature maps on multimodal images are not meaningful.

DHN [10] and MHN [23]. Those two methods are using the network as a regression function to directly output homography parameters. A recent technical report gives a comprehensive comparison between those two methods and many traditional methods under various conditions [33]. MHN [23] is a more advanced version as it combines three DHN [10] in a multi-scale cascaded way. Though the idea is straightforward, we find both two methods always can give meaningful results for different cases. However, compared with the regular MSCOCO dataset, we observe the performance drop of those two methods on multimodal datasets showed in Table 1. For example, MHN [23] performs much better than DHN [10], and aligns 95% of total test images on MSCOCO within 3-pixel error, but it fails to align almost all test multimodal image pairs within 3 average pixels.

DHN + DLKFM and MHN + DLKFM. As our major contribution, DLKFM guarantees the local minimum and local smoothing of the Lucas-Kanade objective function. The iterative updating solution of Eq. 7 needs the initial homography to start with. As discussed before, we choose results from regression models DHN [10] and MHN [23] as the initial P_0 respectively. This makes our pipeline generic, which is not limited to the experimental settings in this paper. According to Fig. 5 and one recent technical report [33], those regression models are always able to give relatively good performance under different conditions. In Table. 1, we show the improvement by updating Eq. 7 on DLKFM. On multimodal datasets, our method largely improves the accuracy by helping most of the image pairs get smaller average corner errors than 3 pixels. On MSCOCO, even though

Dataset	Method	<3 pixel	<10 pixel
MSCOCO	DHN[10]	14%	97%
	DHN+DLKFM	90%	97%
	MHN[23]	95%	99%
	MHN+DLKFM	97%	99%
GoogleEarth	DHN[10]	0%	20%
	DHN+DLKFM	60%	83%
	MHN[23]	1%	56%
	MHN+DLKFM	66%	91%
GoogleMap	DHN[10]	0%	10%
	DHN+DLKFM	52%	74%
	MHN[23]	0%	33%
	MHN+DLKFM	60%	81%

Table 1. The percentage of test images with a smaller average corner error than 3 and 10 pixels. Our method largely improves the performance of existing methods by using them as initials.
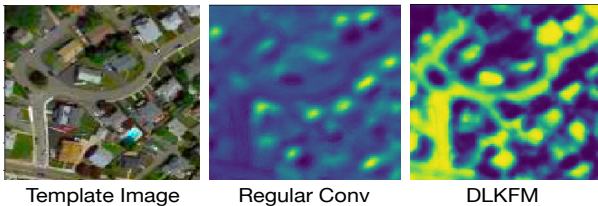

Template Image   Regular Conv   DLKFM

Figure 7. The demonstration of the meaningful feature map extracted by our feature constructor, DLKFM.

MHN [23] is already very accurate, we can still improve it.

5. Ablation Study

We proposes two specific technologies in this paper. The feature constructor constructs one channel feature map based on a high dimension tensor. The special designed loss function guarantees the success of Inverse Compositional Lucas-Kanade on learned feature maps. We verify those two technical contributions here.

5.1. Effectiveness of Feature Constructor

Initial Method	Feature type	<3 pixel	<10 pixel
DHN [10]	Regular Conv	47%	72%
	DLKFM	60%	83%
MHN [23]	Regular Conv	55%	83%
	DLKFM	66%	91%

Table 2. Performance comparison on cross season Google Earth.

Since Eq. 8 and Eq. 9 need to be calculated with all values from the output tensor, we reduce the number of channels to one by using Eq. 11. The other common way is using a regular convolutional layer with fewer filters [6, 29]. We conduct an extra experiment on cross-season Google Earth dataset by replacing our feature constructor with a regular one filter convolutional layer. We list the comparison performance in Table 2. To further demonstrate the effective-
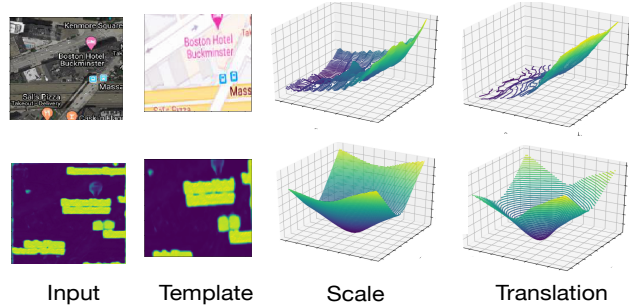

Input   Template   Scale   Translation

Figure 8. **(Top)** The LK objective function on the original multimodal image pair from Google Map and Satellite. **(Bottom)** The objective function on our DLKFM. We respectively perturb the scale and translation in two directions around the ground truth whereas keeping the other six parameters fixed.

ness of our feature constructor, we visualize the learned feature map in Fig. 7. It is clear that our feature constructor helps the feature map extract meaningful features, which are invariant roads and houses. In recent papers [46, 31], those features are learned with semantic labels, like the building label provided by OpenStreetMap [17], whereas our method spontaneously recognizes those patterns.

5.2. Effectiveness of Designed Loss Function

Our special designed loss function has two terms. The first term helps two feature maps keep the brightness consistency, which has been presented in Fig. 1. The second term helps the convergence of Eq. 6 by shaping the landscape of optimization function around the ground truth value $\bar{P}$, that is $\bar{P} = [\bar{P}_{11}, \bar{P}_{12}, \bar{P}_{13}, \bar{P}_{21}, \bar{P}_{22}, \bar{P}_{23}, \bar{P}_{31}, \bar{P}_{32}]$. To verify it, we keep six parameters fixed, and perturb the other two around the ground truth. On the image coordinate, we perturb scale $(\bar{P}_{11}, \bar{P}_{22})$ in two directions and translation $(\bar{P}_{13}, \bar{P}_{23})$ in two directions, shown in Fig. 8. The sample is from our multimodal Google Map and Satellite dataset. For original image pair, we can see the ΔP has no chance to converge as the $\bar{P}$ is even not the minimum. On the contrary, the landscape of the objective function on learned DLKFM shows a smooth surface with the minimum at $\bar{P}$. Therefore, by minimizing the objective, ΔP will keep moving toward the ground truth $\bar{P}$.

6. Conclusion

This paper proposes a generic method to align multimodal images by extending Lucas-Kanade on feature maps. Solving this fundamental challenge has the potential to provide solutions for many computer vision applications in the filed like UAV or remote sensing. Two specific techniques have been developed as part of our method. Since how to extract meaningful features and how to shape the objective function are both widely existing challenges, we believe those two technical contributions can give a hint to those related computer vision or machine learning tasks.

References

[1] Epameinondas Antonakos, Joan Alabort-i Medina, Georgios Tzimiropoulos, and Stefanos P Zafeiriou. Feature-based lucas–kanade and active appearance models. *IEEE Transactions on Image Processing*, 24(9):2617–2632, 2015. 3

[2] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International journal of computer vision*, 56(3):221–255, 2004. 2, 3

[3] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006. 2

[4] Assia Benbihi, Matthieu Geist, and Cédric Pradalier. Elf: Embedded localisation of features in pre-trained cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7940–7949, 2019. 3

[5] Cesar Cadena, Anthony R Dick, and Ian D Reid. Multimodal auto-encoders as joint estimators for robotics scene understanding. In *Robotics: Science and Systems*, volume 5, page 1, 2016. 1

[6] Che-Han Chang, Chun-Nan Chou, and Edward Y Chang. Clkn: Cascaded lucas-kanade networks for image alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2213–2221, 2017. 3, 5, 6, 7, 8

[7] Haiyang Chao, Yu Gu, and Marcello Napolitano. A survey of optical flow techniques for robotics navigation applications. *Journal of Intelligent & Robotic Systems*, 73(1-4):361–372, 2014. 6

[8] Christopher B Choy, JunYoung Gwak, Silvio Savarese, and Manmohan Chandraker. Universal correspondence network. In *Advances in Neural Information Processing Systems*, pages 2414–2422, 2016. 3

[9] Konstantinos G Derpanis. The harris corner detector. *York University*, pages 1–2, 2004. 2, 4

[10] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016. 3, 5, 6, 7, 8

[11] Lijun Ding and Ardeshir Goshtasby. On the canny edge detector. *Pattern Recognition*, 34(3):721–725, 2001. 4

[12] Jose Dolz, Karthik Gopinath, Jing Yuan, Herve Lombaert, Christian Desrosiers, and Ismail Ben Ayed. Hyperdensenet: a hyper-densely connected cnn for multi-modal image segmentation. *IEEE transactions on medical imaging*, 38(5):1116–1126, 2018. 1

[13] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8092–8101, 2019. 2, 3

[14] Farzan Erlik Nowruzi, Robert Laganiere, and Nathalie Japkowicz. Homography estimation from image pairs with hierarchical convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 913–920, 2017. 3

[15] Georg Frobenius, Ferdinand Georg Frobenius, Ferdinand Georg Frobenius, Ferdinand Georg Frobenius, and Ger-

[16] many Mathematician. Über matrizen aus nicht negativen elementen. 1912. 4

[16] Hunter Goforth and Simon Lucey. Gps-denied uav localization using pre-existing satellite imagery. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 2974–2980. IEEE, 2019. 1, 2, 3

[17] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008. 8

[18] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 4

[19] Seungryong Kim, Stephen Lin, Sang Ryul Jeon, Dongbo Min, and Kwanghoon Sohn. Recurrent transformer networks for semantic correspondence. In *Advances in Neural Information Processing Systems*, pages 6126–6136, 2018. 3

[20] Seungryong Kim, Dongbo Min, Somi Jeong, Sunok Kim, Sangryul Jeon, and Kwanghoon Sohn. Semantic attribute matching networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12339–12348, 2019. 3

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[22] Simon Korman, Daniel Reichman, Gilad Tsur, and Shai Avidan. Fast-match: Fast affine template matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2331–2338, 2013. 6

[23] Hoang Le, Feng Liu, Shu Zhang, and Aseem Agarwala. Deep homography estimation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7652–7661, 2020. 3, 5, 6, 7, 8

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[25] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2010. 3

[26] Jonathan L Long, Ning Zhang, and Trevor Darrell. Do convnets learn correspondence? In *Advances in neural information processing systems*, pages 1601–1609, 2014. 3

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

[28] Bruce D Lucas, Takeo Kanade, et al. An iterative image registration technique with an application to stereo vision. 1981. 2, 3

[29] Zhaoyang Lv, Frank Dellaert, James M Rehg, and Andreas Geiger. Taking a deeper look at the inverse compositional algorithm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4581–4590, 2019. 3, 8

[30] Boaz Nadler. On the distribution of the ratio of the largest eigenvalue to the trace of a wishart matrix. *Journal of Multivariate Analysis*, 102(2):363–371, 2011. 4

[31] Ahmed Nassar, Karim Amer, Reda ElHakim, and Mohamed ElHelw. A deep cnn-based framework for enhanced aerial imagery registration with applications to uav geolocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1513–1523, 2018. 2, 8

[32] Ty Nguyen, Steven W Chen, Shreyas S Shivakumar, Camillo Jose Taylor, and Vijay Kumar. Unsupervised deep homography: A fast and robust homography estimation model. *IEEE Robotics and Automation Letters*, 3(3):2346–2353, 2018. 1, 3

[33] David Niblick and Avinash Kak. Homography estimation with convolutional neural networks under conditions of variance. *arXiv preprint arXiv:2010.01041*, 2020. 7

[34] Yuki Ono, Eduard Trulls, Pascal Fua, and Kwang Moo Yi. Lf-net: learning local features from images. In *Advances in neural information processing systems*, pages 6234–6244, 2018. 2, 3

[35] Shaul Oron, Aharon Bar-Hille, and Shai Avidan. Extended lucas-kanade tracking. In *European Conference on Computer Vision*, pages 142–156. Springer, 2014. 2, 6

[36] Oskar Perron. Zur theorie der matrices. *Mathematische Annalen*, 64(2):248–263, 1907. 4

[37] Jerome Revaud, Cesar De Souza, Martin Humenberger, and Philippe Weinzaepfel. R2d2: Reliable and repeatable detector and descriptor. In *Advances in Neural Information Processing Systems*, pages 12405–12415, 2019. 2, 3

[38] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4104–4113, 2016. 2, 3

[39] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision*, pages 501–518. Springer, 2016. 2, 3

[40] Xuelun Shen, Cheng Wang, Xin Li, Zenglei Yu, Jonathan Li, Chenglu Wen, Ming Cheng, and Zijian He. Rf-net: An end-to-end image matching network based on receptive field. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8132–8140, 2019. 3

[41] Richard Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 2(1):1–104, 2006. 3

[42] Prune Truong, Martin Danelljan, and Radu Timofte. Glunet: Global-local universal network for dense flow and correspondences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6258–6268, 2020. 3

[43] Chaoyang Wang, Hamed Kiani Galoogahi, Chen-Hsuan Lin, and Simon Lucey. Deep-lk for efficient adaptive object tracking. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 627–634. IEEE, 2018. 2

[44] Yuanxin Ye, Lorenzo Bruzzone, Jie Shan, Francesca Bovolo, and Qing Zhu. Fast and robust matching for multimodal remote sensing image registration. *IEEE Transactions on Geoscience and Remote Sensing*, 57(11):9059–9070, 2019. 2, 3

[45] Yuanxin Ye, Jie Shan, Lorenzo Bruzzone, and Li Shen. Robust registration of multimodal remote sensing images based on structural similarity. *IEEE Transactions on Geoscience and Remote Sensing*, 55(5):2941–2958, 2017. 1, 2, 3

[46] Armand Zampieri, Guillaume Charpiat, Nicolas Girard, and Yuliya Tarabalka. Multimodal image alignment through a multiscale chain of neural networks with application to remote sensing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 657–673, 2018. 1, 2, 8

[47] Jirong Zhang, Chuan Wang, Shuaicheng Liu, Lanpeng Jia, Nianjin Ye, Jue Wang, Ji Zhou, and Jian Sun. Content-aware unsupervised deep homography estimation. In *European Conference on Computer Vision*, pages 653–669. Springer, 2020. 3