Learning Discriminative and Transformation Covariant Local Feature Detectors

Xu Zhang[1], Felix X. Yu[2], Svebor Karaman[1], Shih-Fu Chang[1]
[1]Columbia University, [2] Google Research
{xu.zhang, svebor.karaman, sc250}@columbia.edu, felixyu@google.com

Abstract

Robust covariant local feature detectors are important for detecting local features that are (1) discriminative of the image content and (2) can be repeatably detected at consistent locations when the image undergoes diverse transformations. Such detectors are critical for applications such as image search and scene reconstruction. Many learning-based local feature detectors address one of these two problems while overlooking the other. In this work, we propose a novel learning-based method to simultaneously address both issues. Specifically, we extend the covariant constraint proposed by Lenc and Vedaldi [8] by defining the concepts of "standard patch" and "canonical feature" and leverage these to train a novel robust covariant detector. We show that the introduction of these concepts greatly simplifies the learning stage of the covariant detector, and also makes the detector much more robust. Extensive experiments show that our method outperforms previous hand-crafted and learning-based detectors by large margins in terms of repeatability.

1. Introduction

Local feature detectors play a critical role in computer vision applications such as image registration [29], image matching [2] and image retrieval [26]. Traditionally, local feature detectors were carefully hand-crafted to deal with scale and viewpoint changes [9, 16]. More recently, machine learning based detectors [8, 24, 25] have been proposed to deal with challenging issues such as temporal variations in webcam sequences [24]. Even though machine learning based methods have achieved great success in many computer vision problems such as object recognition, the research of learning-based local feature detectors is still quite preliminary.

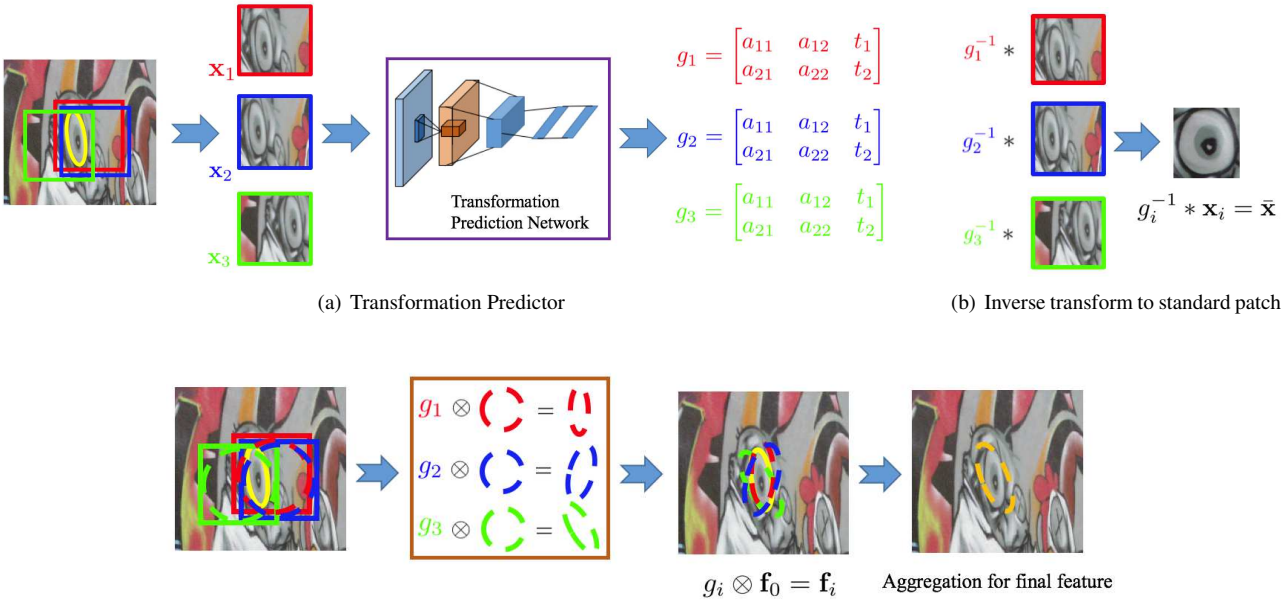
Given an image, a local feature detector outputs a set of *features*[1], *i.e.*, a set of points or a set of ellipses in the image. A good local feature detector should satisfy two important properties: (1) it can discover local discriminative information in the image, and (2) it can repeatably detect consistent patterns when the scene undergoes diverse transformations. The second property means that given an image that has undergone a geometric transformation, a good detector should output the feature same as the ones generated from the untransformed image, subject to the same geometric transformation as the one applied to the image. Such property is called the *covariant constraint* of the local feature detector and has been studied in [8].

Most of the learning-based detectors only focus on one of the two properties mentioned above. Some feature detectors are learned using manually labeled data or the output of off-the-shelf detectors as discriminative training features [6, 24, 21, 5]. However, the covariant constraint is difficult to embed in such a training pipeline, and this often leads to inability in handling transformations that are not presented in the training data. Other feature detectors are designed to primarily focus on the covariant constraint [8], but does not place equal emphasis on extraction of discriminative local content - a key property of "good" local features.

In this work, we propose a novel learning-based method to solve both issues. Specifically, we extend the covariant constraint proposed by Lenc and Vedaldi [8] by defining the concepts of "standard patch" and "canonical feature". The standard patches define discriminative patches and the *standard* position and shape of the canonical feature (e.g. a unit circle) inside the standard patch. We will show such explicit concepts make the learning process more robust and less sensitive to the initialization setting. Furthermore, we theoretically prove that the covariant constraint for all the patches sampled from images is equivalent to the covariant constraint for all standard patches. We show that this greatly simplifies the learning stage.

The learning framework in both our work and [8] is to train a transformation predictor instead of a predictor for the existence of the local feature. Thus, the covariant constraint of the local feature detector becomes the covariant con-

[1]Note that a feature is different from a *feature descriptor*. The latter is a vector describing a local patch of an image.



(a) Transformation Predictor

(b) Inverse transform to standard patch

(c) Feature detection by predicting transformation

Figure 1. The relationship among image patch, transformation and feature. (Top) Given an image patch, we propose a transformation prediction network to predict a transformation (g_i) whose inverse (g_i^{-1}) can warp the image patch $\mathbf{x}_i$ to a standard patch $\bar{\mathbf{x}}$. (Bottom) The predicted transformation itself can also be used to map the canonical feature (dashed circle) to the feature (ellipse) observed in each image patch. Since each feature is covered by multiple overlapping image patches, the outputs from multiple image patches can be aggregated to predict the most likely location and shape of the feature.

straint of the transformation predictor. The latter is much easier to formulate, see Section 3.2 for details.

Figure 1 illustrates our approach. The whole image is divided into patches (say 32×32 pixels) using a sliding window. A deep learning-based model is applied to each patch to predict a transformation (Figure 1(a)). The transformation has two important properties: (1) the inverse of the transformation maps the observed image patch to a standard patch (Figure 1(b)). And (2) it can predict the position and the shape of the transformed feature (e.g. an ellipse) inside the image patch by applying the predicted transformation to the canonical feature (the dashed circle in Figure 1(c)). If there exists a "good" feature in the image, according to the covariant constraint, all the image patches containing the feature (patches containing the feature that have undergone different transformations) will "point" to the same feature. The final estimation of the feature can be readily determined by further analyzing the overlapped predictions (see Figure 1(c) and Section 5). In this paper, we will theoretically and experimentally show the superior performance of feature detectors based on this new formulation.

Formulating the feature detection as transformation prediction provides several unique benefits. First of all, the covariant constraint of the feature detector can be directly embedded into the optimization criterion of the training process, enabling robust feature detection under diverse transformations. Second, powerful machine learning models such as deep neural networks can be used to train the transformation predictor.

Our work makes the following contributions:

1. We define a novel formulation based on the new concepts of "standard patch" and "canonical feature" to place equal focus on discriminativeness and covariant constraint. This makes the proposed detector able to detect discriminative and repeatable features under diverse image transformations.
2. Instead of restricting our method to a specific type of local feature, *i.e.* point or blob, our approach is supported by the general theory of transformation group. It makes our approach applicable to a diverse set of features and transformations.
3. Extensive experiments over multiple standard benchmarks show that our method outperforms previous methods of both hand-crafted and learned detectors by large margins.

2. Related Work

Hand-Crafted Detectors. Previous works have shown that some image structures can be preserved under different kinds of transformations. Different kind of detectors are therefore proposed to detect different image struc-

tures to achieve covariance for different transformations. For example, corner detectors [4, 12, 28] are covariant to translation and rotation. Blob detectors [9, 1] are covariant to scale changes. Moment based detectors, such as Harris-Affine [13] and Hessian-Affine [14, 16] further extend the blob detector to be covariant to affine transformation. MSER [11], LLD [18] and ASIFT [17] are also covariant to affine transformations. The main drawback of these feature detectors are that they are hand-crafted and thus not trainable to fit different applications.

Learning-Based Detectors. As we mentioned in Section 1, learning a local feature detector needs to solve two problems: (1) how to define discriminative patterns in the image, and (2) how to detect such patterns under different conditions. Most of the learning based detectors focus on solving the first problem. Kienzle *et al*. [6] propose to learn feature detector from manually labeled data. Rosten *et al*. [21] propose to learn a fast feature detector from the FAST detector aiming at speeding up the detection. Hartmann [5] *et al*. propose to learn a keypoint detector with keypoints retained through Structure-of-Motion pipeline. The state-of-the-art method TILDE [24] learns from pre-aligned images of the same scene at different time and seasons. TILDE stacks all the aligned images and collects keypoints at the positions where the SIFT detector provides high confidence in most of the images and uses the keypoints for training. Since it also collects points missed by SIFT, it performs better than SIFT on the evaluated datasets. TaSK [22] also learns from pre-aligned images.

These approaches are good at predicting good feature for the scene that are similar to the training data. Since the covariant constraint is not embedded in training, such detectors may fail in the case when the scene is processed by unseen transformations.

Some works focus on improving the repeatability of the detector. FAST-ER [21] uses simulated annealing to optimize the parameters of the FAST detector [21] to improve the repeatability. Trujillo and Olague [23, 19] propose to use genetic programming to optimize the repeatability. Lenc and Vedaldi [8] propose to train feature detector directly from the covariant constraint. By considering the relation of the feature and the transformation, they turn the feature detection problem to a transformation regression problem. Thus, the covariant constraint can be directly learned by Siamese neural networks.

3. Preliminary

3.1. Feature and Transformation

A feature $\mathbf{f}$ is an abstract geometric structure which describes geometric properties such as position and shape. The most commonly used features are point (position), circle (position and scale) and ellipse (position, scale and shape). A class of features $\mathcal{F}$ contains different features of the same type (*e.g.* points at all possible positions).

A geometric transformation g is a function whose domain and range are sets of points. Some examples are translation, rotation and scaling. A transformation group G contains a set of transformations. We define three transformation operators:

- $\otimes$ applies a transformation on a feature. For example, $g \otimes \mathbf{f}_1 = \mathbf{f}_2$ means moving and warping $\mathbf{f}_1$ by g to generate a new feature $\mathbf{f}_2$.
- $*$ applies a transformation on an image or image patch, such as translating, rotating and shearing the image (patch).
- $\circ$ is the composition of transformations, which is also the group operation in G. For example, $g \circ h$ is the transformation of applying h and then g: $(g \circ h) \otimes \mathbf{f} = g \otimes (h \otimes \mathbf{f})$.

Transformations in G have the following properties:

- For any g and h in G, $g \circ h$ is also in G.
- There is an identity element e in G. For any g in G, g^{-1} is also in G and $g^{-1} \circ g = e$.
- The $\circ$ operation in G is associative.

Typical transformation groups include translation group, rotation group and affine group. Considering the theory of transformation group provides us a tool to build a general theory framework for learning feature detector on any transformation group.

Feature and transformation are closely related. There is one important case, when there exists a bijection between the feature and the transformation. Let $\mathbf{f}_0 \in \mathcal{F}$ be a fixed *canonical feature*, we say a class of features $\mathcal{F}$ *resolves* a group of transformation G, when

Definition 1 *$\mathcal{F}$ resolves G if*

1. *For any $\mathbf{f} \in \mathcal{F}$, there exists $g \in G$, such that $g \otimes \mathbf{f}_0 = \mathbf{f}$.*
2. *For any $g \in G$, $g \otimes \mathbf{f}_0 \in \mathcal{F}$.*
3. *For any $g_1, g_2 \in G$, $g_1 \otimes \mathbf{f}_0 = g_2 \otimes \mathbf{f}_0 \Rightarrow g_1 = g_2$.*

For example, the class of point resolves the translation group and the class of orientated ellipse resolves the affine group without reflection. $\mathcal{F}$ resolving G gives us several good properties. By detecting the feature $\mathbf{f} = g \otimes \mathbf{f}_0$, we can estimate the transformation performed on the image. The image patch containing $\mathbf{f}$ then can be normalized to a "standard patch" (a patch containing the canonical feature $\mathbf{f}_0$) by applying g^{-1} to the original image patch. Thus, the descriptor extracted from the standard patch would be invariant to the group of transformation G.

3.2. Learning Covariant Detector

Following previous works [24, 25], we train a detector from small image patches, one patch $\mathbf{x} \in \mathcal{X}$ is supposed to have at most one local feature. $\mathcal{X}$ is the set of all the image patches. Given an image patch $\mathbf{x}$, one intuitive way to train a feature detector is to train a classifier or regressor to predict a confidence score (real number) for the presence of a local feature in $\mathbf{x}$ [24, 25]. Since a transformation cannot be applied to a real number, it is challenging to factor the covariant constraint into this approach.

Recently, Lenc and Vedaldi [8] proposed a novel framework for training a covariant local feature detector from scratch by explicitly embedding covariant constraint into the loss function. In the rest of this section, we will briefly review this work and discuss limitations and ways for improvement. We define the feature detector as:

Definition 2 *A feature detector* $\psi : \mathbf{x} \mapsto \mathbf{f}$ *maps an image patch* $\mathbf{x}$ *to a feature* $\mathbf{f}$ *in or overlapped with* $\mathbf{x}$.

Thus, the covariant constraint of the feature detector can be defined as:

Definition 3 *A feature detector* $\psi : \mathbf{x} \mapsto \mathbf{f}$ *is said to be covariant, if*

$$\forall \mathbf{x} \in \mathcal{X}, g \in G : \psi(g * \mathbf{x}) = g \otimes \psi(\mathbf{x}). \tag{1}$$

Directly dealing with the feature is intuitive but cumbersome. However, when $\mathcal{F}$ resolves G, it is possible to deal with transformation instead of feature, since there exists a bijective mapping between $\mathcal{F}$ and G. Thus, the transformation regressor $\phi : \mathbf{x} \mapsto g$ can substitute the role of feature detector ψ. (1) can be rewritten as:

$$\forall \mathbf{x} \in \mathcal{X}, g \in G : \phi(g * \mathbf{x}) = g \circ \phi(\mathbf{x}). \tag{2}$$

Lenc and Vedaldi [8] proposed to directly use the covariant constraint to train the feature detector. In order to optimize the covariant constraint, they randomly sample n patches $\{\mathbf{x}_i\}$, $i = 1, \ldots, n$, from training images. For each training patch $\mathbf{x}_i$, a transformation g_i is randomly generated and applied to $\mathbf{x}_i$. The transformed image patch is further cropped to make it the same size as $\mathbf{x}_i$. The cropped patch is denoted as $g_i * \mathbf{x}_i$. The image patch, the transformation and the transformed image patch form a training triplet. Given n training triplets $\{(\mathbf{x}_i, g_i * \mathbf{x}_i, g_i)\}$, $i = 1, \ldots, n$, they defined the learning problem as:

$$\phi = \arg\min_{\phi} \sum_{i=1}^{n} d(\phi(g_i * \mathbf{x}_i), g_i \circ \phi(\mathbf{x}_i))^2, \tag{3}$$

where $d(\cdot, \cdot)$ is the distance between two transformations [27]. In practice, they parameterized the transformation as a matrix, and used the matrix Frobenius norm of the transformations' difference as the distance of two transformations. Thus, the loss of the covariant constraint can be realized as:

$$\ell_{covariant} = \sum_{i=1}^{n} \parallel \phi(g_i * \mathbf{x}_i) - g_i \circ \phi(\mathbf{x}_i) \parallel_F^2 . \tag{4}$$

The learning objective can be rewritten as:

$$\phi = \arg\min_{\phi} \sum_{i=1}^{n} \parallel \phi(g_i * \mathbf{x}_i) - g_i \circ \phi(\mathbf{x}_i) \parallel_F^2 . \tag{5}$$

The problem can be solved by Siamese neural networks. Detecting local features in the whole image can be done by applying the local feature detector to all the image patches in the image, see Section 5.3 for the details.

The main drawback of this work is that the solution of (5) may be not unique. Let's assume that $\phi^*(\cdot)$ is a solution of (5), or $\phi^*(\cdot)$ minimizes (4). Firstly, we assume the group operation $\circ$ is addition, which is the case for many transformation groups, such as translation group. For any $g' \in G$, let $\phi'(\cdot) = \phi^*(\cdot) + g'$, then

$$\begin{aligned} &\parallel \phi'(g_i * \mathbf{x}_i) - (g_i + \phi'(\mathbf{x}_i)) \parallel_F^2 \\ =& \parallel \phi^*(g_i * \mathbf{x}_i) + g' - (g_i + \phi^*(\mathbf{x}_i) + g') \parallel_F^2 \\ =& \parallel \phi^*(g_i * \mathbf{x}_i) - (g_i + \phi^*(\mathbf{x}_i)) \parallel_F^2, \end{aligned} \tag{6}$$

which means $\phi'(\cdot)$ is also a solution to (5). When the group operation $\circ$ is multiplication, which is the case for many transformation groups, such as similarity group, for any $g' \in G$, such that $\parallel g' \parallel_F^2 \leq 1$, let $\phi'(\cdot) = \phi^*(\cdot)g'$, we have,

$$\begin{aligned} &\parallel \phi'(g_i * \mathbf{x}_i) - g_i \phi'(\mathbf{x}_i) \parallel_F^2 \\ =& \parallel \phi^*(g_i * \mathbf{x}_i)g' - g_i \phi^*(\mathbf{x}_i)g' \parallel_F^2 \\ \leq& \parallel \phi^*(g_i * \mathbf{x}_i) - g_i \phi^*(\mathbf{x}_i) \parallel_F^2 \parallel g' \parallel_F^2 . \end{aligned} \tag{7}$$

Since $0 \leq \parallel g' \parallel_F^2 \leq 1$, $\phi'(\cdot)$ is also a solution or even a better solution to (5). This drawback may result in two issues: (1) the training is sensitive to the initialization of the neural network, different initializations may lead to different ϕ, and (2) since the solution may differ from the "correct" solution by a fixed transformation, the feature detected by the detector may also differ from the canonical feature by the transformation. This may move a good local feature to a useless one.

In this paper, we prove this problem can be solved by embedding the *standard patches* into the training pipeline. We further show that the concept of standard patches can greatly simplify the covariant constraint.

4. Learning From Standard Patches

Generally speaking, *standard patches* are patches that are discriminative and diverse enough for learning to regress the target transformation group. However, to find out the exact set of standard patches is not an easy problem. Following previous works [24, 5] that used the results of an

existing off-the-shelf detector as anchors, we adopt a similar strategy to choose the potential standard patches. In this section, we denote a standard patch as $\bar{\mathbf{x}}$, and the set of standard patches as $\bar{\mathcal{X}}$.

As mentioned in Definition 1, *canonical feature* is the reference feature mapped to the identity in the transformation group. Theoretically, the canonical feature can be chosen arbitrarily. For better clarify, in this paper, the canonical feature is defined as the central point of the standard patch (for point detection) or the inscribed circle in the standard patch (for blob detection). For each transformation group, the canonical feature is uniquely defined.

If any of the standard patch is sent to the feature detector, the output of the feature detector is the canonical feature. Formally, for all $\bar{\mathbf{x}} \in \bar{\mathcal{X}}$, $\psi(\bar{\mathbf{x}}) = \mathbf{f}_0$. Since the canonical feature $\mathbf{f}_0 \in \mathcal{F}$ is mapped to the identity e in the transformation group G, we have, for all $\bar{\mathbf{x}} \in \bar{\mathcal{X}}$, $\phi(\bar{\mathbf{x}}) = e$. Given m standard patches, it is natural to design a loss to realize this identity constraint:

$$\ell_{identity} = \sum_{j=1}^{m} \parallel \phi(\bar{\mathbf{x}}_j) - e \parallel_F^2 . \tag{8}$$

By considering (4) and (8), given n training patches $\{\mathbf{x}_i\}$, $i = 1, \ldots, n$, we can generate n triplets of patches $\{(\mathbf{x}_i, g_i * \mathbf{x}_i, g_i)\}$, $i = 1, \ldots, n$, as mentioned in Section 3.2, together with m standard patches $\{\bar{\mathbf{x}}_j\}$, $j = 1, \ldots, m$, the learning problem is given by:

$$\begin{aligned} \phi = \arg\min_{\phi} & \sum_{i=1}^{n} \parallel \phi(g_i * \mathbf{x}_i) - g_i \circ \phi(\mathbf{x}_i) \parallel_F^2 \\ & + \alpha \sum_{j=1}^{m} \parallel \phi(\bar{\mathbf{x}}_j) - e \parallel_F^2, \end{aligned} \tag{9}$$

where α is a trade-off parameter between the covariant constraint (4) and the identity constraint (8). The identity constraint performs as a regularization term. By anchoring the transformation of the standard patches to identity, the non-uniqueness issue mentioned in Section 3.2 is resolved.

In order to detect all possible features $\mathbf{f} \in \mathcal{F}$, the feature detector has to be covariant to all image patches of all $\mathbf{f} \in \mathcal{F}$. It requires to collect a huge number of training triplets to minimize the covariant loss in (9). The following proposition greatly simplifies this problem:

Proposition 1 *A transformation regressor ϕ is covariant for all the patches, if and only if it is covariant for all the standard patches. Formally,*

$$\begin{aligned} & \forall \mathbf{x} \in \mathcal{X}, g \in G, \phi(g * \mathbf{x}) = g \circ \phi(\mathbf{x}) \iff \\ & \forall \bar{\mathbf{x}} \in \bar{\mathcal{X}}, g \in G, \phi(g * \bar{\mathbf{x}}) = g \circ \phi(\bar{\mathbf{x}}). \end{aligned} \tag{10}$$

Since $\bar{\mathcal{X}}$ is a subset of $\mathcal{X}$, from left to right is trivial. Let's prove Proposition 1 from right to left. For all $\mathbf{x} \in \mathcal{X}$, there exists one transformation $g_1 \in G$ that maps one standard patch $\bar{\mathbf{x}}$ to $\mathbf{x}$, $\mathbf{x} = g_1 * \bar{\mathbf{x}}$. Or the inverse of the transformation g_1^{-1} maps the image patch $\mathbf{x}$ to its corresponding standard patch $\bar{\mathbf{x}}$. Thus, for all $g \in G$,

$$\begin{aligned} \phi(g * \mathbf{x}) &= \phi(g * (g_1 * \bar{\mathbf{x}})) = \phi((g \circ g_1) * \bar{\mathbf{x}}) \\ &= (g \circ g_1) \circ \phi(\bar{\mathbf{x}}) = g \circ (g_1 \circ \phi(\bar{\mathbf{x}})) \\ &= g \circ \phi(g_1 * \bar{\mathbf{x}}) = g \circ \phi(\mathbf{x}). \end{aligned} \tag{11}$$

The second equality is due to the composition in the transformation group. The third and the fifth equalities are due to the covariant constraint of the standard patches. The fourth equality is based on the associative property of the transformation group.

Proposition 1 gives us a strong result that in order to train a covariant feature detector for all the image patches, we only need to train a covariant feature detector for all the standard patches. Thus, the previous training problem (9) can be simplified as, given n standard patches $\{\bar{\mathbf{x}}_i\}$, $i = 1, \ldots, n$, generating n triplets of training data $\{(\bar{\mathbf{x}}_i, g_i * \bar{\mathbf{x}}_i, g_i)\}$, $i = 1, \ldots, n$, getting ϕ via,

$$\begin{aligned} \phi = \arg\min_{\phi} & \sum_{i=1}^{n} (\parallel \phi(g_i * \bar{\mathbf{x}}_i) - g_i \circ \phi(\bar{\mathbf{x}}_i) \parallel_F^2 \\ & + \alpha \parallel \phi(\bar{\mathbf{x}}_i) - e \parallel_F^2), \end{aligned} \tag{12}$$

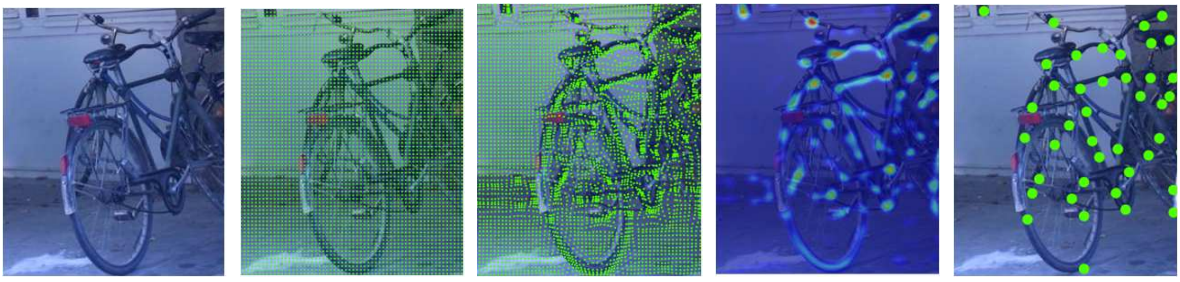
which means we only need to train based on standard patches instead of all possible patches.

5. Implementation

5.1. Collecting Training Data

In practice, we choose the output of TILDE [24] detector as candidates of the standard patches. An empirical study of choosing the results of different detectors as candidate standard patches is shown in Section 6.3. Among all, the state-of-the-art local feature detector TILDE-P24 [24] gives us the best performance. We also notice that the model trained with output of TILDE-P24 has the lowest identity loss, compared with the models trained with the output of other detectors, which shows the output of TILDE-P24 may be more consistent to the identity constraint. For each output feature of TILDE detector, a 32×32 image patch around the feature is treated as one standard patch. In practice, we keep a 51×51 image patch since generating the transformed patch may use the neighboring pixels near the standard patch.

To generate a training triplet $(\bar{\mathbf{x}}_i, g_i * \bar{\mathbf{x}}_i, g_i)$, given a 51×51 image patch, a 32 × 32 standard patch $\bar{\mathbf{x}}_i$ is extracted from the center of the large patch. Depending on different transformation groups, g_i can be generated by a combination of (1) scaling in both x and y axes with two factors uniformly sampled from $[0.85, 1.15]$; (2) shearing in both x and y axes with two factors uniformly sampled from $[-0.15, 0.15]$; (3) translation in both x and y axes with two factors uniformly sampled from $[-8, 8]$; (4) rotation uniformly sampled from $[0, 360°]$. Then g_i is applied to the



(a) Input image (b) Grid points (c) Transformed points (d) Density of the point (e) Output keypoints

Figure 2. Detecting local feature in the image using the transformation predictor.

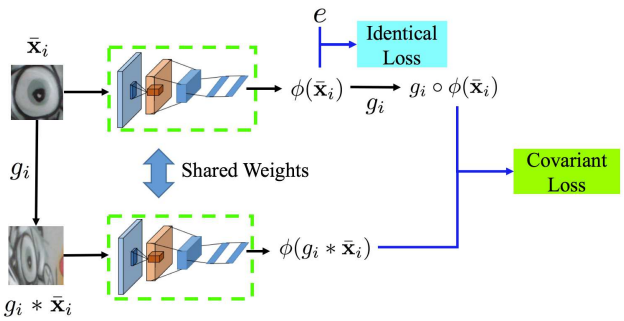


Figure 3. Training pipeline of the proposed transformation prediction network.

51×51 image patch to get a large transformed patch. We get the final transformed patch $g_i * \bar{\mathbf{x}}_i$ by also cropping a 32×32 patch from the center of the large transformed patch. In order to get enough variation, for each standard patch, we generate 24 transformed triplets with different transformations. It should be noted that some patches may still lack enough information to regress the required group of transformation. For example, a corner at any scale may look the same, and may not be able to discern the scale change. Such ambiguity can be eliminated by pre-filtering the candidate standard patches that lack cues to regress the required transformation.

5.2. The Neural Network-Based Model

One benefit of formulating the feature detector as the transformation predictor is that it provides the possibility to use a powerful regressor such as deep neural networks to predict the transformation. Considering the problem defined in (12), both the standard patch and the transformed patch will simultaneously go through the same regressor to calculate the covariant loss. It is natural to use a modified Siamese networks to solve this problem. Figure 3 shows the training pipeline of the proposed method. Different from the conventional Siamese structure, the "identity loss" defined in (8) is calculated on the branch which processes the standard patch.

Since the input of network is simply a 32×32 color image patch and the detection time is also very important for local feature detector, we only use a compact architecture in our experiments. The first layer is a convolutional layer of kernel size 5×5 and 32 output channels followed by a 2×2 max pooling layer. The second convolutional layer has a kernel size of 5×5 and 128 output channels, it is also followed by a 2×2 max pooling layer. The third convolutional layer has a kernel size of 3×3 and 128 output channels. The fourth convolutional layer has a kernel size of 3×3 and 256 output channels. The final layer is an 1×1 convolutional layer with the number of output channels that equal to the number of parameters in the regressed transformation. We use the ReLU activation in all the convolutional layers.

5.3. Global Feature Detector

So far, we have only discussed the local feature detector (or transformation predictor) for an image patch. To detect all the features in the whole image, the transformation predictor is applied at all image locations. Since our neural network has the "fully-convolutional" structure, it can be applied to images with any size and aspect ratio. Figure 2 shows the case of point detector. The canonical feature in all the image patches define a dense grid in the image (Figure 2(b)). Each green point is the center of an image patch. The predicted transformation "moves" the canonical feature to the closest local feature. Due to the covariant constraint, all the image patches overlapped with a "good" feature will transform their canonical feature to this feature (Figure 2(c)). The density of the points reflects the stability of the local feature (Figure 2(d)). We use a voting method to estimate the density of the points. Specifically, each image patch votes to the grid points in Figure 2(b) using bilinear interpolation. Non-maxima suppression is applied to choose the grid point which is the local maximum in the vote map (Figure 2(e)). For detectors that are covariant to more than translation, to avoid voting in high-dimensional space, we only vote for the position of the local feature, the final shape of the local feature is predicted by

the patch associated with the chosen point.

Since the neural network can only process image patches with a fixed size, our model can only resolve a small range of scale changes. In order to improve the performance over a larger scale range, features are extracted on a 5-level image pyramid. The bottom level is the original image, and image in an upper level is the image in current level down-sampled by a factor of $\sqrt{2}$ with Gaussian smoothing.

Like most CNNs, the proposed network architecture contains pooling layer which down-samples the output size for efficiency. In order to perform dense feature detection, one solution is to remove the pooling layers. Another solution is to reapply the CNN to slightly shifted images. The *à trous* algorithm provides an efficient way to do that by reusing precomputed results of previous denser layers [10, 20, 8]. In this work, we only consider the down-sampled output.

6. Experiments

6.1. Datasets and Settings

We use three datasets which are viewed as standard in evaluating feature detectors in previous works.

Webcam [24] contains 6 sequences, each sequence has 140 images taken from the same scene. Among them, 100 images are for training, 20 images for validation and 20 images for test. It contains drastic time and season changes of one scene, including day and night, rain and snow, winter and summer which are challenging for local feature detector.

EF [28] has 5 sequences of 38 images. Each sequence is a scene observed under different conditions. The dataset contains drastic illumination and background clutter changes.

VGG-Affine [15] is a traditional dataset for local feature detector evaluation. It contains 8 sequences of 48 images with varying viewpoints, lighting conditions and compression rates.

Our evaluation metric is the repeatability defined in [14] which considers the position, scale and the shape of the feature. Given an image pair and the transformation between two images, two regions in different images are deemed to correspond if the overlap error of one region and the projected region (the region of the other image projected by the ground-truth transformation) is less than 0.4. The repeatability is defined as the ratio of the number of correspondences and the smaller number of regions in the pair of images. Only the regions located in the part that are shared by both images are considered.

The main drawback of the repeatability is that when the number of the feature is high, the feature may be randomly "matched" due to the high density of the features. Hence, we calculate the repeatability on extracting both 1000 features and 200 features per image.

We compare our approach to scale-covariant handcrafted feature detectors (SIFT [9], SURF [1], MSER [11], Harris Laplace (HarLap) [14] and Hessian Laplace (HesLap) [14]), and affine-covariant feature detectors (Harris Affine (HarAff) [14] and Hessian Affine (HesAff) [14]). We also compare our approach with learned feature detectors such as FAST [21], TILDE [24] and Covariant Point Detector (CovDet) [8]. Note that TILDE is currently the state-of-the-art method. Three versions of TILDE [24] detectors based on different regressors are used for evaluation: T-CNN (based on convolutional neural network), T-P (based on piece-wise linear regressor), and T-P24 (an approximation of T-P). For CovDet, we use the model provided by [8]. For TILDE [24] and Covdet [8] detectors, we also extract keypoint from multi-scale image pyramid as described in Section 5.3. Since larger feature always leads to higher repeatability, in order to make fair comparison, for TILDE, CovDet and our detector, as in [24], a scale of 10 is used as the scale of the keypoint in each pyramid level.

Both TILDE and our method share the same training images from the Mexico subset of the Webcam dataset. For TILDE, we use the model provided by the authors. For our method, we run the T-P24 detector trained on the Mexico subset on the training images (also in the Mexico subset), then randomly extract about 5k patches from the detection results and generate 120k training triplets as detailed in Section 5.1. All the methods are evaluated on all the subsets except Mexico subset in the Webcam dataset.

For network training, the learning rate is set to 0.01 and the training batch size is 128. We found empirically that the network usually converges after 5 epochs. With a TITAN X, the training takes about 30 minutes based on our implementation with TensorFlow [3]. For detection, our implementation achieves 10 FPS for running our model on a 1000×700 pixels of image with a TITAN X GPU.

6.2. Repeatability

Table 1 shows the average repeatability on different datasets. Our method is a clear winner on almost all the settings. In particular it outperforms the state-of-the-art learning based method T-P24 [24]. Although trained using the output of T-P24, our detector further incorporates synthesized patches in training and uses regressed transformation to predict the location of the feature. These steps make our detector surpass T-P24. Our method also outperforms the baseline method CovDet [8]. The main differences between our method and CovDet are the introduction of the identity loss (8) and the using of discriminative standard patches instead of randomly sampled patches. It shows that defining which feature is discriminative by defining the standard patches is very important for training a good local feature detector. In summary, the superior performance of our detector confirms the importance of (1) defining discriminative patches; (2) the covariant constraint.

Method	Webcam		EF		VGG	
	#Feature		#Feature		#Feature	
	1000	200	1000	200	1000	200
SIFT	29.5	19.1	20.8	10.9	47.1	41.7
SURF	46.0	33.4	39.7	23.4	61.2	58.3
MSER	45.1	29.4	37.1	18.9	54.1	38.4
SFOP	43.8	25.6	36.1	21.7	51.2	44.9
HesLap	51.1	37.2	38.8	28.0	66.7	60.0
HarLap	48.2	44.5	35.7	33.4	60.5	55.5
HesAff	42.5	34.5	26.6	21.8	66.4	59.6
HarAff	38.4	33.6	22.7	20.2	57.3	55.7
FAST	56.3	41.1	32.0	28.9	53.8	44.1
T-P	35.4	29.0	26.3	16.3	54.6	46.1
T-P24	61.7	45.1	45.4	32.3	64.4	57.6
T-CNN	51.4	36.7	38.0	21.8	50.7	40.6
CovDet	49.9	32.2	42.7	23.8	62.0	48.0
Ours	**68.4**	**52.6**	**46.6**	**36.3**	**70.2**	**61.2**

Table 1. Repeatability (%) of different methods on all datasets.

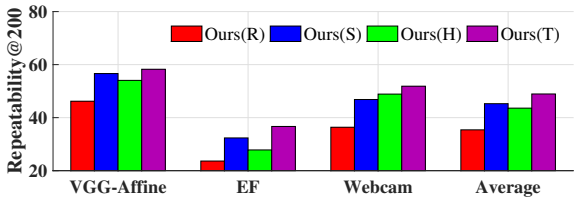


Figure 4. Repeatability (%) of our feature detector trained with different standard patches.

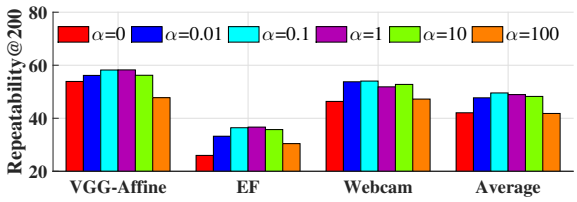


Figure 5. Repeatability (%) of our feature detector trained with different α values.

6.3. Effects of Standard Patches and Identity Loss

Figure 4 shows the performance of our detector trained with different standard patches. We collect standard patches in 4 different ways: randomly sampled patches (Ours(R)), detection result of SIFT detector (Ours(S)), detection result of Hessian Affine detector (Ours(H)) and detection result of T-P24 detector (Ours(T)). Randomly sampled patches perform the worst and T-P24 gives the best result.

In (12), the α parameter trades off the identity (8) and covariant (4) constraints. The identity constraint is one of the key contributions of our work. In Figure 5, we see that an adequate choice of α leads to much better performance than setting α to 0 or 100, *i.e.*, using a single constraint.

Dataset	Detector				
	SIFT	HesAff	T-P24	CovDet	Ours
Webcam	12.9	13.8	13.4	12.0	**19.4**
VGG	42.8	35.3	44.5	43.1	**50.7**
EF	**10.2**	5.4	5.2	4.8	6.2
Average	22.0	18.2	21.0	20.0	**25.4**

Table 2. Matching score (%) of different detectors.

6.4. Matching Score

To show the matching performance, we evaluate matching score on the above three benchmark datasets via VLBenchmarks [7]. Matching score is the ratio between the number of the correct matches and the smaller number of the detected regions in the pair of images [16]. A match is correct if the two corresponding regions (defined in Section 6.2) are nearest neighbors in the descriptor space. Since the goal of this experiment is to compare the detectors, not the descriptor, SIFT descriptor [9] is used for all the detectors. Table 2 summarizes the matching scores. Although our detector achieves higher repeatability than SIFT detector, the matching score of our detector is lower than that of SIFT detector in EF dataset. One possible reason is that EF dataset contains drastic background clutter changes. The feature detected by our detector may contain background changes that can't be matched via the descriptor.

7. Conclusion

Good local feature detectors should have two properties: (1) detect discriminative image features, and (2) detect the same feature under diverse transformations. Most of the previous works only focus on one of the constraints and ignore the other. In this work, we propose a new method to simultaneously address the two properties by extending the covariant constraint of [8] with the concept of "standard patches". We further prove that the covariant constraint can be greatly simplified by the "standard patches". Our prototype implementation chooses the output of the TILDE detector as the candidate of standard patches and resulting in significantly improved result in terms of repeatability. However, how to choose the "standard patches" is still an open issue deserving more studies in the future. Our prototype implementation can be downloaded from https://github.com/ColumbiaDVMM/Transform_Covariant_Detector.

Acknowledgement This material is based upon work supported by the United States Air Force Research Laboratory (AFRL) and the Defense Advanced Research Projects Agency (DARPA) under Contract No. FA8750-16-C-0166. Any opinions, findings and conclusions or recommendations expressed in this material are solely the responsibility of the authors and does not necessarily represent the official views of AFRL, DARPA, or the U.S. Government.

References

[1] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded Up Robust Features. In *ECCV*. 2006.

[2] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *CVPR*, 2005.

[3] M. A. *et al*. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015.

[4] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, 1988.

[5] W. Hartmann, M. Havlena, and K. Schindler. Predicting matchability. In *CVPR*, 2014.

[6] W. Kienzle, F. A. Wichmann, B. Scholkopf, and M. O. Franz. Learning an interest operator from human eye movements. In *CVPRW*, 2006.

[7] K. Lenc, V. Gulshan, and A. Vedaldi. Vlbenchmarks. http://www.vlfeat.org/benchmarks/, 2011.

[8] K. Lenc and A. Vedaldi. Learning covariant feature detectors. In *ECCV Workshop on Geometry Meets Deep Learning*, 2016.

[9] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *IJCV*, 60(2):91–110.

[10] S. Mallat. *A wavelet tour of signal processing*. Academic press, 1999.

[11] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 2004.

[12] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *ICCV*, 2001.

[13] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *ECCV*, 2002.

[14] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 2004.

[15] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *TPAMI*, 2005.

[16] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A Comparison of Affine Region Detectors. *IJCV*, 2005.

[17] J. Morel and G. Yu. ASIFT: A New Framework for Fully Affine Invariant Image Comparison. *SIAM Journal on Imaging Sciences*, 2009.

[18] P. Musé, F. Sur, F. Cao, Y. Gousseau, and J.-M. Morel. An a contrario decision method for shape element recognition. *IJCV*, 2006.

[19] G. Olague and L. Trujillo. Evolutionary-computer-assisted design of image operators that detect interest points using genetic programming. *Image and Vision Computing*, 2011.

[20] G. Papandreou, I. Kokkinos, and P.-A. Savalle. Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection. In *CVPR*, 2015.

[21] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. In *ECCV*, 2006.

[22] C. Strecha, A. Lindner, K. Ali, and P. Fua. Training for task specific keypoint detection. In *Joint Pattern Recognition Symposium*, 2009.

[23] L. Trujillo and G. Olague. Synthesis of interest point detectors through genetic programming. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM, 2006.

[24] Y. Verdie, K. M. Yi, P. Fua, and V. Lepetit. Tilde: A temporally invariant learned detector. In *CVPR*, pages 5279–5288. IEEE, 2015.

[25] K. M. Yi, E. Trulls, V. Lepetit, and P. Fua. LIFT: Learned Invariant Feature Transform. In *ECCV*, 2016.

[26] H. Zhang, Z.-J. Zha, Y. Yang, S. Yan, Y. Gao, and T.-S. Chua. Attribute-augmented semantic hierarchy: towards bridging semantic gap and intention gap in image retrieval. In *MM*. ACM, 2013.

[27] L. Zheng, Q. Liu, X. Wang, and A. Maleki. lp-based complex approximate message passing with application to sparse stepped frequency radar. *Signal Processing*, 2017.

[28] C. L. Zitnick and K. Ramnath. Edge foci interest points. In *ICCV*. IEEE, 2011.

[29] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and vision computing*, 2003.