Selecting and robustifying local image descriptors
Everts, I.

Citation for published version (APA):
Everts, I. (2014). Selecting and robustifying local image descriptors

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: http://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.
Per-Patch Metric Learning for Robust Image Matching *

3.1 Introduction

Viewing and lighting condition changes in real-world scenes cause substantial variations in image feature representations. Significant progress has been made in developing image representations that are invariant to visual changes due to photometric [30, 32] or geometric [5, 51, 58] transformations. These invariant image representations are beneficial for applications such as object recognition, image retrieval and scene recognition.

A full invariant representation, unfortunately, leads to a decrease in discriminative power [93]. This trade-off is due to discriminant variations that an invariant representation cannot capture. For example, under rotational invariance a ‘6’ is identical to a ‘9’. Another disadvantage of invariant image representations is that they negatively effect stability [33, 90]. This is due to their sensitivity to noise when the image signal is low or ambiguous. For example, a rotational invariant that chooses the dominant orientation [58] is unstable when multiple equally dominant orientations are present.

Current invariant methods are always ‘on’. One can either choose to use the invariance, or choose not to use it; there is no middle-ground. It is not possible to have invariance for only some shading or only slight rotations. These rigid properties of current invariants play a central role in the trade-off between in-

*Submitted to the Asian Conference on Computer Vision in extended form [40].
variance and discriminative power. Here, we propose to replace these binary on/off invariants, by steering the invariance to a limited range of disturbances. Such a limited degree of invariance is called robustness. For example, in the case of rotation, the proposed method can learn that a ‘6’ is only invariant up to $\pm 45^\circ$ of (and thus robust to) rotation, therefore eliminating the confusion with ‘9’.

By allowing a degree of invariance, a single global image representation cannot be used as it essentially depends on the specific image content how the limited transformation range will take effect, which is illustrated by the ‘6’ and ‘9’ example. Robustness must be achieved on a per-patch basis. Figure 3.1 (top) illustrates that the feature distributions after a transformation depend on the patch content, since even instances within the same class behave differently (red versus yellow). The bottom row of Figure 3.1 illustrates the effect of steerable invariance applied to each patch.

In order to achieve robustness, a Mahalanobis metric is computed for each individual patch. In effect, the metric weights the subset of feature dimensions that require robustness. For this, a relevant subset of transformations is generated and a metric that is specific for only those transformations is learned. Two approaches are presented for learning the metric: (i) full and (ii) direct. In the full method, synthetic image patches are generated, descriptors are extracted for each patch, and the robustness is obtained through a metric that is learned on these descriptors. In the direct approach, we only once generate a transfor-
3.2 Related Work

Figure 3.2: The flow of the proposed method. A range of transformations are applied on each query patch. With the transformations we learn a robust patch-specific metric, and use this map to estimate the metric from the patch directly without explicitly generating any synthetic images. The approach is illustrated in Figure 3.2. For learning the metric we here focus only on geometric image transformations.

3.2 Related Work

Figure 3.3 relates the proposed approach of this chapter to existing literature. Approaches that aim to achieve (full) invariance either use a transformation model based on the laws of physics [30, 32] or a model of the observed variations [5, 51, 56, 58]. The two disadvantages of invariants — stability and discrim-
In contrast to employing pre-determined models, (deep) learning methods learn invariant features from unsupervised training examples [41, 70, 77]. Such methods do not explicitly model invariance as they attain robustness from training examples. Therefore, learning methods require large amounts of training data which is hard to obtain. Moreover, pure learning approaches do not directly incorporate known physical laws of the world. In this work we use a hybrid approach of modelling robustness by learning from generated geometric training data.

Synthetically generated data can be used to directly create variation in the train and test samples [10, 26, 50]. Other brute-force methods like ASIFT [60] generate a full range of affine transformations for both training and testing images which are used in an exhaustive matching scheme. Similar to our work, Simard et al. [82] avoid brute-force approaches and use synthetically generated images to learn a robust distance metric which is tangent to the manifold that is spanned by the generated transformations. We also learn a robust metric, however, where Simard et al. [82] require pixel values to estimate a manifold, our method estimates a Mahalanobis distance, which is applicable to any feature representation such as SIFT or LBP. To improve discriminability of a local descriptor, Cai et al [10] also propose to learn a projection matrix for a limited range of affine transformations through generated data. However, it is important to note that the authors learn a global projection whereas it is proposed...
in this chapter to learn patch specific projections. Figure 3.1 illustrates that the same transformations applied to even the same class instances has different effects for different patches. These variations are thus patch-dependent and might not reflect the appropriate effect on other patches from the same class.

In a supervised scenario, Mahalanobis metrics can be learned to allow class-specific weighting of feature dimensions [3, 43, 57, 100]. Such local approaches learn multiple distance metrics for various parts of the feature space [103, 100] or for every single image [25], as opposed to learning a single global metric. We derive inspiration from these local methods to learn a distance metric for each patch, however we do not use class labels. The proposed method learns either from generated instances (full metric), or directly from a patch, without explicitly generating any instances (direct metric). This chapter thus contributes by the development and evaluation of novel methods for per-patch metric learning.

### 3.3 A Single-patch Metric

We first develop a metric that is learned from synthetically generated geometric distortions.

**Geometric Transformations** Images are subject to geometric distortions introduced by perspective effects caused by view point changes. For small patches, the perspective transformation $(x', y')^T$ can be approximated by an affine transformation for a given point $(x, y)^T$ as

$$
\begin{pmatrix}
    x' \\
    y' \\
    1
\end{pmatrix} =
\begin{pmatrix}
    s_x \cos \alpha & -t_x \sin \alpha & t_x \\
    t_y \sin \alpha & s_y \cos \alpha & t_y \\
    0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
    x \\
    y \\
    1
\end{pmatrix},
$$

(3.1)

where $s$ denotes scale, $\tau$ represents shearing, $\alpha$ is the rotation angle, and $t$ denotes translation. Examples of the transformations are depicted in Figure 3.4. Sample generation involves repetitive random selection from appropriate parameter ranges.
3.3.1 Metric Learning

A Mahalanobis distance metric between image features $x_i$ and $x_j$ is parameterized by the matrix $M$

$$d_M(x_i, x_j) = \sqrt{(x_i - x_j)^T M (x_i - x_j)},$$  \hspace{1cm} (3.2)

where $M$ is a positive semi-definite matrix. Eq. (3.2) simplifies to the Euclidean distance when $M$ is the identity matrix. Otherwise, $M$ considers different weights for different feature dimensions.

A straightforward approach to compute $M$ is to use $M = C^{-1}$ where $C$ is the empirical feature covariance of the training data [3, 43]. The rationale behind using the inverse covariance as a metric is that for objects of the same class, a high variance in a feature value means that this feature is not very stable. The most informative feature dimensions are those that have low variance on training data.

For large-scale matching it is convenient to use fast indexing techniques such as trees [57]. Such techniques typically work with the Euclidean distance. To this end, the metric can be rewritten to $(x_i - x_j)^T W^T W (x_i - x_j) = ||W x_i - W x_j||^2$, where $W = M^{-\frac{1}{2}}$. This effectively scales the feature space with an affine transformation $W$ to allow the Euclidean distance to be used for metric $M$.

Robustness is obtained by generating a limited range of transformations for
3.3 A Single-patch Metric

a single patch. From these generated samples the covariance $C$ is estimated, which is then used to compute $M$.

We coin this simple method for metric learning the full approach, since it needs to fully generate a large number of transformations for a patch in order to extract the image features and estimate the covariance matrix.

3.3.2 Direct Metric Estimation

Instead of using the brute-force approach of computing a covariance matrix by explicitly generating all patches and extracting features for each of them, we propose a direct approach to estimate the metric per patch. In the direct approach, the covariance is estimated from a single patch.

In the following, let $x$ be a column vector, with dimensionality $N$. For clarity, we start with pixel values to explain the direct metric estimation, i.e. $x$ is a vector of image pixels.

**Pixel Values** The direct metric is a combination of two terms, a transformation probability $D_T$ that a pixel moves to a different position after transformation $T$, and the patch-specific covariance term $V$ of the feature values.

Let $D_T$ be a symmetric matrix of size $N \times N$, containing the probabilities $P_T(i|j)$ of a pixel at position $j$ affecting the position of pixel $i$ under a transformation $T$. Note that this transformation is independent of the actual feature values. Matrix $D_T$ represents the per-pixel transformation probability which is determined by simulating a large set of transformations and comparing the transformed patch with the ground truth location obtained through the homography in Eq. (3.1).

The matrix $V$ of size $N \times N$ is the variance matrix with elements $\sigma(i,j)$ representing the covariance of pixel values at position $i$ with respect to position $j$. To compute the covariance matrix, we need the expected average weighted pixel value of the pixel values in $x$ at position $i$ after the transformation $T$, which is given by

$$E[x_T(i)] = \sum_{j} P_T(i|j)x(j) = D_Tx^T, \quad (3.3)$$

where $x_T$ represents the pixel vector $x$ after the transformation $T$. The expected value of the transformed image $x_T$ is denoted by $E[x_T(i)]$ and represents the average image of all transformations that are present in $D_T$. See the top row of Figure 3.5 for some example images of $E[x_T(i)]$. 
The covariance $\sigma(i, j)$ after transformation $T$ is then
\[
\sigma(i, j) = E[(E[x^T(i)] - x_T(i))(E[x^T(j)] - x_T(j))].
\]

Note that in the transformation of $x_T(i)$ and $x_T(j)$ it is allowed for pixels to move independently to other pixels. To illustrate this, in the middle row of Figure 3.5 several images $x_T$ are created by sampling the distribution of motion probabilities $D_T$. None of these images are actually used in the direct metric estimation approach. Eq. (3.4) can be rewritten in matrix form to obtain $V$ directly
\[
W_V = ([[D_T > 0]] \bullet (D_T \cdot x^T \cdot 1 - (x^T \cdot 1)^T),
\]
\[
V = \frac{1}{N^2} D_T \cdot W_v W_V^T,
\]
where $1$ is a column vector of all ones, $\bullet$ denotes element-wise multiplication and $[[::]]$ indicate Iverson brackets which resolves a (matrix) element to $1$ when the argument is true, and $0$ otherwise. The metric is computed by $M = V^{-1}$, and the transformation by $W = V^{-\frac{1}{2}}$.

SIFT For other descriptors, the $D_T$ matrix of transformation probabilities can be reused and is not required to be recomputed. The $V$ matrix, however, has to be adapted to the specific form of the descriptor. In the case of SIFT, $D_T$ is converted to $D_T^{sift}$ of size $128 \times 128$. In contrast to generating all possible SIFT values, this transformation only has to be computed once.

The 128 dimensions of SIFT comprise of a $4 \times 4$ spatial grid and $8$ angular bins ($4 \times 4 \times 8$). We use the pixel-based transformation probabilities $D_T$ to directly calculate the probability $P_T^{sift}(i|j)$ for spatial SIFT bins $i$ and $j$ with
\[
P_T^{sift}(i|j) = \sum_{y=0}^{N^2} [[f(x) = i]][[f(y) = j]]P_T(x|y),
\]
where the function $f(x)$ maps the pixel at location $x$ to the correct SIFT-bin $i$. For the angular transformation probabilities, we assume independence with the spatial bins. The unit circle is sampled with 360 vectors and transformations are applied to these vectors. Since the original orientation is known, this results in counting how often a vector switches bins after the transformations, as illustrated in Figure 3.6. The joint 128x128 matrix $D_{\text{sift}}^T$ is calculated by multiplying the angular probabilities with the spatial probabilities.

Figure 3.6: Angular transformations. Original angular SIFT binning is depicted by the white lines, whereas the red colored segments show how orientation vectors switch bins.

Finally, we note that SIFT is $L_2$ normalized, which obscures the true probabilities of the angular bins. Hence, $L_2$ normalization is performed after applying $W$ (Section 3.3.1). The computation of $M_{\text{V}}^\text{sift}$ is identical to the pixel value case in eq. Eq. (3.6). The input feature $I$ is now the unnormalized 128d SIFT vector.

3.4 Experiments

The full and direct metric estimation methods are evaluated on two different publicly available datasets for image matching: MNIST [49] and ALOI [29]. For both datasets we consider metric learning for achieving geometric robustness. Photometric robustness is assessed on ALOI only.

3.4.1 Geometric Robustness

Here, we evaluate per-patch metric learning under geometric transformations. The performance of the proposed methods is compared against other existing methods, and we evaluate how different combinations of affine changes affect matching performance.

The optimal affine parameter ranges are obtained on separate validation sets for both MNIST and ALOI. Parameters are repetitively sampled (1500 times)
from these ranges to either generate and apply geometric transformations in the *full* approach or to estimate the transformation probabilities $P_{T(i|j)}$ in the *direct* approach.

Classification is performed by feature matching in a 1-NN classification scheme.

### MNIST Matching

The MNIST dataset consists of 60,000 handwritten digits for training and another 10,000 for testing, see e.g. Figure 3.4 and Figure 3.5 for examples. Both raw pixels and SIFT are considered as image features.

Parameter selection is performed on the first 200 images of the training set. The best transformation combination for MNIST dataset comprises moderate amounts of translation (in [-2:2]), rotation (in [-.1:.1]) and shearing (in [-.125:.125]) (see Eq. (3.1)) for SIFT features. Scale has no influence, because the digit size is almost the same throughout the dataset. Shearing and rotation help to learn variations such as *italic* writing while translation helps to learn small shifts of the digits from the center.

<table>
<thead>
<tr>
<th>Method</th>
<th>Cl. Rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>96.57</td>
</tr>
<tr>
<td>SIFT + Rotation invariance</td>
<td>95.61</td>
</tr>
<tr>
<td>SIFT + Affine invariance</td>
<td>96.55</td>
</tr>
<tr>
<td>SIFT + RCA [3]</td>
<td>96.16</td>
</tr>
<tr>
<td><strong>SIFT + Proposed-full</strong></td>
<td><strong>96.87</strong></td>
</tr>
<tr>
<td><strong>SIFT + Proposed-direct</strong></td>
<td><strong>96.49</strong></td>
</tr>
<tr>
<td>Raw Pixel</td>
<td>96.91</td>
</tr>
<tr>
<td>Raw Pixel + one sided TD [82]</td>
<td>98.12</td>
</tr>
<tr>
<td>Raw Pixel + two sided TD [82]</td>
<td><strong>98.61</strong></td>
</tr>
<tr>
<td>Raw Pixel + Proposed-direct</td>
<td>96.88</td>
</tr>
<tr>
<td>SIFT + SVM (linear)</td>
<td>97.40</td>
</tr>
<tr>
<td>SIFT + LDA + 1-NN</td>
<td>97.39</td>
</tr>
<tr>
<td>SIFT + LMNN 1-NN Mahalanobis [100]</td>
<td><strong>97.50</strong></td>
</tr>
</tbody>
</table>

*Table 3.1:* Classification rates for different methods on MNIST dataset.

Classification performance is compared against unsupervised and supervised
3.4 Experiments

state-of-the-art methods in Table 3.1. Best performance for this dataset is obtained with tangent distance (TD) [82]. As discussed in Section 3.2, Simard et al. invoke prior information by generating small global transformations. This supports our idea of exploiting prior information for steering the invariance. TD however requires pixel values as image features, which makes applicability of this approach harder for some other feature representation such as SIFT.

SIFT does not benefit from rotation invariance. This is due to the associated decrease in discriminative power. However, affine invariance does not affect the performance. The per-patch full metric preserves discriminative power while moderately benefiting from improved robustness, as for example illustrated by the ‘6’ vs ‘9’ example. The direct metric assumes independent pixel movements which makes it difficult to preserve shape information, as illustrated by the performance of the ‘Proposed-direct’ methods in Table 3.1. Note that ‘Raw Pixel + Proposed-full’ is not considered as the computational load associated to generating enough examples and computing covariance is prohibitively large.

Supervised methods generally outperform unsupervised methods on MNIST. To show the strength of the proposed metric learning approaches, we next consider a matching task in which no class labels are available and pixel-based methods such as TD fail due to increased visual complexity.

ALOI Matching

The ALOI dataset contains 1000 objects under varying imaging conditions. The variations are due to illumination direction, illumination color and camera viewpoint. Here, we focus on geometric transformations and thus only consider patches recorded from different viewpoints under canonical illumination conditions, i.e. with all light sources switched on. The recording setup consists of three cameras positioned next to each other. For the following matching experiments, patches are sampled from the two outer cameras (that are furthest apart). To find the matching pairs from two outer cameras, the same procedure is followed as in [21]. In total 8300 matching pairs are extracted of which 200 are used for validation. The patch size is $64 \times 64$, whereas SIFT is extracted from a $56 \times 56$ window around the center to avoid the artificial border edges due to the geometric transformations for the full metric method. Parameter selection is performed as described in Section 3.4.1 on the validation set.

The influence of sweeping each individual transformation range on the validation set is illustrated in Figure 3.8. The figures show that the direct and full metric agree more or less on their performance changes for different parameter
settings. A joint optimization of parameters on the ALOI dataset yields translation in \([-2:2]\) and shearing in \([-.1:.1]\) for SIFT descriptors. Scale and orientation do not affect performance as the positional difference between cameras do not yield large scale and orientation variations.

<table>
<thead>
<tr>
<th></th>
<th>SIFT</th>
<th>+Rot. Inv.</th>
<th>Full Metric</th>
<th>Direct Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Classification rates on the geometric ALOI dataset based on SIFT features.</strong></td>
<td>54.85%</td>
<td>13.86%</td>
<td>61.42%</td>
<td>61.07%</td>
</tr>
</tbody>
</table>

The results in Table 3.2 show that the proposed full and direct SIFT-metric methods have significant improvement of \(6.57\%\) and \(6.22\%\) over regular SIFT performance respectively. The substantial performance increase for the ALOI dataset is due to the fact that viewpoint variations degrade SIFT performance for matching. Considering rotation invariance for SIFT leads to a dramatic performance loss as most discriminative information is ignored and due to the visual complexity it is harder to estimate a dominant gradient orientation.

Considering raw pixels and tangent distance on this set, we obtain \(16.54\%\) and \(23.47\%\) respectively. As expected, the raw pixel matching performance drastically drops for this dataset. Furthermore, as texture rather than shape is prominent in ALOI patches, the direct metric does not suffer substantially from its assumption with respect to independent pixel movement.

As the dataset contains no class labels, the supervised methods used for comparison in Section 3.4.1 cannot be included here.
3.4 Experiments

3.4.2 Photometric Robustness

We next evaluate the proposed direct method for photometric robustness. Matching patch pairs are selected from the same viewpoint but with different illumination. The same evaluation procedure is followed as in Section 3.4.1. With SIFT matching, we obtain a baseline result of 73.40% matching accuracy. Using the direct metric from the geometric ALOI dataset increases the performance to 76.88% accuracy.

The direct method does not rely on synthetic photometric data generation, but nonetheless substantially outperforms SIFT when confronted with photometric variations in the data. Thus, the direct approach, which is developed for achieving robustness against geometric transformations, is also applicable in the context of photometric variations. This positive side effect is due to the fact that the proposed method is robust against signal changes in general, which may also be caused by variations in illumination.
3.4.3 Complexity Analysis

We perform an analysis of the computational complexity of the per-patch metric learning approaches that are considered, i.e. by either explicitly applying geometric transformations or directly per patch. The analysis is based on the steps involved in matching a query patch to a single database patch. The load associated to the simulation of geometric transformations is denoted by $T$. This involves applying the actual affine transformation to the query patch. Its contribution to the complexity depends on the number of repetitions, i.e. the size of the dataset from which to estimate the covariance, and the patch size. The computational load associated to estimating the full covariance $C_{full}$ is determined by the actual computation of covariance from the generated set of descriptors. Direct computation $C_{direct}$ involves one matrix multiplication of the descriptor with the displacement matrix, subtracting the result from the original and squaring the result. The displacement matrix is optimized offline on a holdout set and is therefore not relevant to the analysis of runtime complexity. Computing the metric $M$ involves inverting the covariance and decomposing the result. Applying the metric to a patch descriptor involves a matrix multiplication and is denoted by $A$. This is performed on both the query and the database patch. Finally, the computational load associated to distance computation is denoted by $D$. The computational complexity of metric learning based on full covariance is then $O(T + C_{full} + M + 2A + D)$, whereas the proposed direct approach is associated to a complexity of $O(C_{direct} + M + 2A + D)$. Thus, doing away with the necessity of actually applying the transformations and computing the covariance grants a big advantage to the direct approach. The graph in Figure 3.9 compares the full and direct approaches for increasing patch size.

![Figure 3.9: Full and direct metric efficiency comparison. 10 repeats, 100 random transformations.](image-url)
3.4 Experiments

3.4.4 Extensive Analysis of the Proposed Method

To illustrate the effect of per-patch metric learning, we consider the examples in Figure 3.10. Figure 3.10 shows matching pairs of image patches, and their difference image. These patches are extracted from two different viewpoints, by which a geometric distortion is introduced. We observe from Figure 3.10.a that the regions in the patch that remain stable across viewpoints (i.e. have zero difference) are those that lack structure. For example, the homogeneously red colored regions of the patches in Figure 3.10.a are unaffected by the transformation. By repetitively simulating such (affine) transformations and recording the resultant descriptors, a dataset is generated which exhibits low variance for those descriptor dimensions that remain stable under these image transformations. In mahalanobis metric learning setups, these variations are inverted and consequently translate to high weights in the distance metric. Considering again the example in figure Figure 3.10a, the metric will thus emphasize structureless regions such as the red colored part at the bottom. More uncertainty is associated to regions near prominent structure such as edges, which is reflected by strong energy in the difference image. Consequently, a metric learnt from highly textured patches such as the ones in figure Figure 3.10.b will grant low weight to a large amount of descriptor dimensions (and thereby affecting the distance measure only marginally). At the other side of the ‘texturedness spectrum’, a metric learnt from a completely homogeneous patch such as in Figure 3.10.c will grant high weight to all descriptor dimensions, which leaves the distance measure basically unaltered. It is thus the structure-based localization of structureless regions that is performed by per-patch metric learning, where most effect is expected for patches that are neither homogeneous nor highly textured. Note that one such metric could be derived directly from image derivatives. That is, image derivatives can be thought of as the expected variance per pixel when the image is subject to small translations. Also note that it is essential to perform per-patch processing in order to obtain these kind of metrics.
3.5 Discussion

In this chapter, we propose a generic patch-specific robust metric learning method to improve matching performance of local descriptors. We show that a full invariant representation leads to a decrease in discriminative power of descriptors. Therefore, we propose a per-patch metric learning method that allows invariance to only an optimizable transformation range. Two approaches for learning the metric are presented: full and direct. The proposed approaches are validated on two publicly available datasets exhibiting geometric and photometric image transformations. It is shown that the proposed approaches outperform the original SIFT descriptor in terms of matching performance.