LDAHash: Improved Matching with Smaller Descriptors

Christoph Strecha, Alexander M. Bronstein, *Member*, *IEEE*,
Michael M. Bronstein, *Member*, *IEEE*, and Pascal Fua, *Senior Member*, *IEEE*

Abstract—SIFT-like local feature descriptors are ubiquitously employed in computer vision applications such as content-based retrieval, video analysis, copy detection, object recognition, photo tourism, and 3D reconstruction. Feature descriptors can be designed to be invariant to certain classes of photometric and geometric transformations, in particular, affine and intensity scale transformations. However, real transformations that an image can undergo can only be approximately modeled in this way, and thus most descriptors are only approximately invariant in practice. Second, descriptors are usually high dimensional (e.g., SIFT is represented as a 128-dimensional vector). In large-scale retrieval and matching problems, this can pose challenges in storing and retrieving descriptor data. We map the descriptor vectors into the Hamming space in which the Hamming metric is used to compare the resulting representations. This way, we reduce the size of the descriptors by representing them as short binary strings and learn descriptor invariance from examples. We show extensive experimental validation, demonstrating the advantage of the proposed approach.

Index Terms—Local features, SIFT, DAISY, binarization, similarity-sensitive hashing, metric learning, 3D reconstruction, matching.

1 Introduction

Over the last decade, feature point descriptors such as SIFT [1] and similar methods [2], [3], [4] have become indispensable tools in the computer vision community. They are usually represented as high-dimensional vectors, such as the 128-dimensional SIFT or the 64-dimensional SURF vectors. While a descriptor's high dimensionality is not an issue when only a few hundred points need to be represented, it becomes a significant concern when millions have to be on a device with limited computational and storage resources. This happens, for example, when storing all descriptors for a large-scale urban scene on a mobile phone for image-based location purposes. Not only does this require a tremendous amount of storage, it is also slow and potentially unreliable because most recognition algorithms rely on nearest-neighbor computations and computing euclidean distances between long vectors is neither cheap nor optimal.

Consequently, there have been many recent attempts at compacting SIFT-like descriptors to allow for faster matching while retaining their outstanding recognition rates. One class of techniques relies on quantization [5], [6] and dimensionality reduction [7], [8]. While helpful, this approach is usually not sufficient to produce truly short descriptors without loss of matching performance. Another class [9], [10], [11], [12] takes advantage of training data to learn short binary codes whose distances are small for positive training pairs and large for others. This is particularly promising because not only does binarization reduce the descriptor size, but also partly increases performance, as will be shown.

Binarization is usually performed by multiplying the descriptors by a projection matrix, subtracting a threshold vector, and retaining only the sign of the result. This maps the data into a space of binary strings, greatly reducing their size on the one hand and simplifying their similarity computation (now becoming the Hamming metric, which can be computed very efficiently on modern CPUs) on the other. Another class of locality-sensitive hashing (LSH) techniques and their variants [9], [13] encode similarity of data points as the collision probability of their binary codes. While such similarity can be evaluated very efficiently, these techniques usually require a large number of hashing functions to be constructed in order to achieve competitive performance. Also, families of LSH functions have been constructed only for classes of standard metrics, such as the L_p norms, and do not allow for supervision.

In most supervised binarization techniques based on a linear projection, the matrix entries and thresholds are selected so as to preserve similarity relationships in a training set. Doing this efficiently involves solving a difficult nonlinear optimization problem and most of the existing methods offer no guarantee of finding a global optimum. By contrast, spectral hashing (SH) [14] does offer this guarantee for simple data distributions and has proved very successful. However, this approach is only weakly supervised by imposing a euclidean metric on the input data, which we will argue is not a particularly good one in our case.

- *C. Strecha is with the EPFL/IC/ISIM/CVLab, Station 14, Lausanne CH-1015, Switzerland. E-mail: christoph.strecha@epfl.ch.*
- *A.M. Bronstein is with the Department of Computer Science, Technion-Israel Institute of Technology, Room 341, Taub Building, Haifa 32000, Israel. E-mail: bron@cs.technion.ac.il.*
- *M.M. Bronstein is with the Institute of Computational Science, Faculty of Informatics, Via Giuseppe Buffi 13, Lugano 6900. E-mail: michael.bronstein@usi.ch.*
- *P. Fua is with the IC-CVLab, Station 14, EPFL, Lausanne CH-1015, Switzerland. E-mail: pascal.fua@epfl.ch.*

Manuscript received 27 Aug. 2010; revised 23 Jan. 2011; accepted 3 Mar. 2011; published online 13 May 2011.
Recommended for acceptance by F. Dellaert.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2010-08-0660.
Digital Object Identifier no. 10.1109/TPAMI.2011.103.

To better take advantage of training data composed of interest point descriptors corresponding to multiple 3D points seen under different views, we introduce a *global* optimization scheme that is inspired by an earlier *local* optimization one [10]. In [10], the entries of the projection matrix and thresholds vectors are constructed progressively using AdaBoost. Given that Adaboost is a gradient-based method [15] and that the algorithm optimizes a few matrix rows at a time, there is no guarantee the solution it finds is optimal. By contrast, we first compute a projection matrix that is designed either to solely minimize the in-class covariance of the descriptors or to jointly minimize the in-class covariance and maximize the covariance across classes, both of which can be achieved in closed form. This being done, we compute optimal thresholds that turn the projections into binary vectors so as to maximize recognition rates. In essence, we perform Linear Discriminant Analysis (LDA) on the descriptors before binarization and will therefore refer to our approach as *LDAHash*.

Our experiments show that state-of-the-art metric learning methods based, e.g., on margin maximization [16], [17] achieve exceptional performance in the low false negative rate range, which degrades significantly in the low false positive rate range. Binarization usually only deteriorates performance. In large-scale applications that involve matching keypoints against databases containing millions of them, achieving good performance in the low false positive rate range is crucial to preventing a list of potential matches from becoming unacceptably long. We use ROC curves to show that, in many different cases, the proposed method has competitive performance in the low false negative rage while significantly outperforming other methods in the low false positive range.

We also show that unlike many other techniques where binarization produces performance degradation, using our approach to binarize SIFT descriptors [1] actually improves matching performance. This is especially true in the low false positive range with 64 or 128-bit descriptors, which means that they are about 10 to 20 times shorter than the original ones. Furthermore, using competing approaches [10], [14], [18] to produce descriptors of the same size as ours results in lower matching performance over the full false positive range.

In the following section, we briefly survey existing approaches to binarization. In Section 3, we introduce our own framework. In Section 4, we describe the corresponding training methodology, training data, and analyze the impact of individual components of our approach. Finally, we present our results in Section 5.

2 Prior Work

Most approaches for compacting SIFT-like descriptors and allowing for faster matching rely on one or more of the following techniques:

2.1 Tuning

In [8], [19], [6], [20], [18], the authors use training to optimize the filtering and normalization steps that produce a SIFT-like vector. The same authors optimize in [18] over the position of the elements that make up a DAISY descriptor [4].

2.2 Quantization

The SIFT descriptor can be quantized using, for instance, only 4 bits per coordinate [5], [18], thus saving memory and speeding up matching because comparing short vectors is faster than comparing long ones. Chandrasekhar et al. [20] applied tree-coding methods for lossy compression of probability distributions to SIFT-like descriptors to obtain a compressed histogram of gradients (CHOG).

2.3 Dimensionality reduction

PCA has been extensively used to reduce the dimensionality of SIFT vectors [21], [6]. In this way, the number of bits required to describe each dimension can be reduced without loss in matching performance [6], [18]. In [22], a whitening linear transform was proposed in addition to benefit from the efficiency of fast nearest-neighbor search methods.

The three approaches above are mostly unsupervised methods and sometimes require a complex optimization scheme [20], [18]. Often, they are not specifically tuned for keypoint matching and do not usually produce descriptors as short as one would require for large-scale keypoint matching.

Our formulation relates to supervised metric learning approaches. The problem of optimizing SIFT-like descriptors can be approached from the perspective of metric learning, where many efficient approaches have been recently developed for learning similarity between data from a training set of similar and dissimilar pairs [23], [24]. In particular, *similarity-sensitive hashing* (SSH) or *locality-sensitive hashing* [9], [10], [14], [11], [12] algorithms seek to find an efficient binary representation of high-dimensional data maintaining their similarity in the new space. These methods have also been applied to *global* image descriptors and bag-of-feature representations in content-based image search [25], [26], [27], [28], video copy detection [29], and shape retrieval [30]. In [31] and [32], Hamming embedding was used to replace vector quantization in bag-of-feature construction.

There are a few appealing properties of similarity-sensitive hashing methods in large-scale descriptor matching applications. First, such methods combine the effects of dimensionality reduction and binarization, which makes the descriptors more compact and easier to store. Second, the metric between the binarized descriptors is learned from examples and renders more correctly their similarity. In particular, it is possible to take advantage of feature point redundancy and transitive closures in the training set, such as those in Fig. 3. Finally, comparison of binary descriptors is computationally very efficient and is amenable for efficient indexing.

Existing methods for similarity-sensitive hashing have a few serious drawbacks in our application. The method of Shakhnarovich [10] poses the similarity-sensitive hashing problem as boosted classification and tries to find its solution by means of a standard AdaBoost algorithm. However, given that AdaBoost is a greedy algorithm equivalent to a gradient-based method [15], there is no guarantee of global optimality of the solution. The spectral hashing algorithm [14], on the other hand, has a tacit underlying assumption of euclidean descriptor similarity, which is typically far from being correct. Moreover, it is worthwhile mentioning that spectral hashing, similarity-sensitive hashing, and similar methods have so far proven to be very efficient in *retrieval* applications for ranking the matches, in which one typically tries to achieve high recall. Thus, the operating point in these applications is at *low false negative* rates, which ensures that no relevant matches (typically, only a few) are missed. In large-scale descriptor matching, on the other hand, one has to create a list of likely candidate matches, which can be very large if the false positive rate is high. For example, given a set of 1 M descriptors, which is modest for Internet-scale applications, and 1 percent false positive rate, 10 K candidates would have to considered. Consequently, an important concern in this application is a *very low false positive* rate. As we show in the following, our approach is especially successful at this operating point, while existing algorithms show poor performance.

3 Approach

Let us assume we are given a large set of keypoint descriptors. They are grouped into subsets corresponding to the same 3D points and all pairs within the subsets are therefore considered as belonging to the same class. The main idea of our method is to find a mapping from the descriptor space to the Hamming space by means of an affine map followed by a sign function such that the Hamming distance between the binarized descriptors is as close as possible to the similarity of the given data set. Our method involves two key steps:

Projection selection. We compute a projection matrix that is designed either to solely minimize the in-class covariance of the descriptors or to jointly minimize the in-class covariance and maximize the covariance across classes, both of which can be done in closed form (Sections 3.3.1 and 3.3.2).

Threshold selection. We find thresholds that can be used to binarize the projections so that the resulting binary strings maximize recognition rates. We show that this threshold selection is a separable problem that can be solved using 1D search. In the remainder of this section, we formalize these steps and describe them in more details.

3.1 Problem Formulation

Our set of keypoint descriptors is represented as n-dimensional vectors in $\mathbb{R}^n$. We attempt to find their representation in some metric space $(\mathbb{Z}, d_{\mathbb{Z}})$ by means of a map of the form $\mathbf{y} : \mathbb{R}^n \rightarrow (\mathbb{Z}, d_{\mathbb{Z}})$. The metric $d_{\mathbb{Z}} \circ (\mathbf{y} \times \mathbf{y})$ parameterizes the similarity between the feature descriptors, which may be difficult to compute in the original representation. Our goal in finding such a mapping is twofold. First, $\mathbb{Z}$ should be an efficient representation. This implies that $\mathbf{y}(\mathbf{x})$ requires significantly less storage than $\mathbf{x}$, and that $d_{\mathbb{Z}}(\mathbf{y}(\mathbf{x}), \mathbf{y}(\mathbf{x}'))$ is much easier to compute than, e.g., $\|\mathbf{x} - \mathbf{x}'\|$. Second, the metric $d_{\mathbb{Z}} \circ (\mathbf{y} \times \mathbf{y})$ should better represent some ideal descriptor similarity, in the following sense: Given a set $\mathcal{P}$ of pairs of descriptors from corresponding points in different images, e.g., the same object under a different view point (referred to as *positives*) and a set $\mathcal{N}$ of pairs of descriptors from different points (*negatives*), we would like $d_{\mathbb{Z}}(\mathbf{y}(\mathbf{x}), \mathbf{y}(\mathbf{x}')) < R$ for all $(\mathbf{x}, \mathbf{x}') \in \mathcal{P}$ and $d_{\mathbb{Z}}(\mathbf{y}(\mathbf{x}), \mathbf{y}(\mathbf{x}')) > R$ for all $(\mathbf{x}, \mathbf{x}') \in \mathcal{N}$ to hold with high probability for some range R.

Setting $\mathbb{Z}$ to be the m-dimensional Hamming space $\mathbb{H}^m = \{\pm 1\}^m$, the embedding of a descriptor $\mathbf{x}$ can be expressed as an m-dimensional binary string. Here, we limit our attention to affine embeddings of the form

$$\mathbf{y} = \mathrm{sign}(\mathbf{P}\mathbf{x} + \mathbf{t}), \tag{1}$$

where $\mathbf{P}$ is an $m \times n$ matrix and $\mathbf{t}$ is an $m \times 1$ vector; embeddings having more complicated forms can be obtained in a relatively straightforward manner by introducing kernels. Even under the optimistic assumption that real numbers can be quantized and represented by 8 bits, the size of the original descriptor is $8n$ bits, while the size of the binary representation is m bits. Thus, setting $m \ll n$ allows us to significantly alleviate the storage complexity and potentially improve descriptor indexing.

Furthermore, the descriptor dissimilarity is computed in our representation using the Hamming metric $d_{\mathbb{H}^m}(\mathbf{y}, \mathbf{y}') = \frac{m}{2} - \frac{1}{2}\sum_{i=1}^{m} \mathrm{sign}(\mathbf{y}_i \mathbf{y}'_i)$, which is done by performing an XOR operation between $\mathbf{y}$ and $\mathbf{y}'$ and counting the number of nonzero bits in the result, an operation carried out in a single instruction on modern CPU architectures (POPCNT SSE4.2).

The embedding $\mathbf{y}$ is constructed to minimize the expectation of the Hamming metric on the set positive pairs while maximizing it on the set of negative pairs. This can be expressed as minimization of the loss function:

$$L = \alpha\, \mathrm{E}\{d_{\mathbb{H}^m}(\mathbf{y}, \mathbf{y}')|\mathcal{P}\} - \mathrm{E}\{d_{\mathbb{H}^m}(\mathbf{y}, \mathbf{y}')|\mathcal{N}\}, \tag{2}$$

with respect to the projection parameters $\mathbf{P}$ and $\mathbf{t}$. Here, α is a parameter controlling the trade-off between false positive and false negative rates (higher α corresponds to lower false negative rates). In practice, the conditional expectations $\mathrm{E}\{\cdot|\mathcal{P}\}$, $\mathrm{E}\{\cdot|\mathcal{N}\}$ are replaced by averages on a training set of positive and negative pairs of descriptors, respectively.

3.2 LDAHash

Here, we note that up to constants, problem (2) is equivalent to the minimization of

$$L = \mathrm{E}\{\mathbf{y}^{\mathrm{T}}\mathbf{y}'|\mathcal{N}\} - \alpha\, \mathrm{E}\{\mathbf{y}^{\mathrm{T}}\mathbf{y}'|\mathcal{P}\} \tag{3}$$

or

$$L = \alpha\, \mathrm{E}\{\|\mathbf{y} - \mathbf{y}'\|^2|\mathcal{P}\} - \mathrm{E}\{\|\mathbf{y} - \mathbf{y}'\|^2|\mathcal{N}\}, \tag{4}$$

attempting to make the correlation of the binary codes as negative as possible for negative pairs and as positive as possible for positive pairs. Direct minimization of L is difficult since the terms $\mathbf{y}$ involve a nondifferentiable sign nonlinearity. While, in principle, smooth approximation is possible, the solution of the resulting nonconvex problem in $(m+1) \times n$ variables is challenging, typically containing thousands of unknowns.

As an alternative, we propose to relax the problem, removing the sign and minimizing a related function:

$$\tilde{L} = \alpha \mathrm{E}\{\|\mathbf{P}\mathbf{x} - \mathbf{P}\mathbf{x}'\|^2|\mathcal{P}\} - \mathrm{E}\{\|\mathbf{P}\mathbf{x} - \mathbf{P}\mathbf{x}'\|^2|\mathcal{N}\}. \tag{5}$$

The above objective is independent of the affine term $\mathbf{t}$ and optimization can be performed over the projection matrix $\mathbf{P}$ only, which we further restrict to be orthogonal. Once the optimal matrix is found, we can fix it and minimize a smooth version of (4) with respect to $\mathbf{t}$.

3.3 Projection Selection

Next, we describe two different approaches for computing $\mathbf{P}$, which we refer to as LDA and Difference of Covariances (DIF) and that we compare in Sections 4 and 5.

3.3.1 Linear Discriminant Analysis

We start by observing that

$$\mathrm{E}\{\|\mathbf{Px}-\mathbf{Px}'\|^2|\mathcal{P}\}=\mathrm{tr}\{\mathbf{P}\mathbf{\Sigma}_{\mathcal{P}}\mathbf{P}^{\mathrm{T}}\},$$

where $\mathbf{\Sigma}_{\mathcal{P}}=\mathrm{E}\{(\mathbf{x}-\mathbf{x}')(\mathbf{x}-\mathbf{x}')^{\mathrm{T}}|\mathcal{P}\}$ is the covariance matrix of the positive descriptor vector differences. This leads to

$$\tilde{L}=\alpha\,\mathrm{tr}\{\mathbf{P}\mathbf{\Sigma}_{\mathcal{P}}\mathbf{P}^{\mathrm{T}}\}-\mathrm{tr}\{\mathbf{P}\mathbf{\Sigma}_{\mathcal{N}}\mathbf{P}^{\mathrm{T}}\},$$

with $\mathbf{\Sigma}_{\mathcal{N}}=\mathrm{E}\{(\mathbf{x}-\mathbf{x}')(\mathbf{x}-\mathbf{x}')^{\mathrm{T}}|\mathcal{N}\}$ being the covariance matrix of the negative descriptor vector differences.

Transforming the coordinates by premultiplying $\mathbf{x}$ by $\mathbf{\Sigma}_{\mathcal{N}}^{-1/2}$ turns the second term of $\tilde{L}$ into a constant for any unitary $\mathbf{P}$, leaving

$$\begin{aligned}\tilde{L}&\propto\mathrm{tr}\{\mathbf{P}\mathbf{\Sigma}_{\mathcal{N}}^{-1/2}\mathbf{\Sigma}_{\mathcal{P}}\mathbf{\Sigma}_{\mathcal{N}}^{-\mathrm{T}/2}\mathbf{P}^{\mathrm{T}}\}\\&=\mathrm{tr}\{\mathbf{P}\mathbf{\Sigma}_{\mathcal{P}}\mathbf{\Sigma}_{\mathcal{N}}^{-1}\mathbf{P}^{\mathrm{T}}\}=\mathrm{tr}\{\mathbf{P}\mathbf{\Sigma}_{\mathcal{R}}\mathbf{P}^{\mathrm{T}}\},\end{aligned}\tag{6}$$

where $\mathbf{\Sigma}_{\mathcal{R}}=\mathbf{\Sigma}_{\mathcal{P}}\mathbf{\Sigma}_{\mathcal{N}}^{-1}$ is the ratio of the positive and negative covariance matrices. Since $\mathbf{\Sigma}_{\mathcal{R}}$ is a symmetric positive semidefinite matrix, it admits the eigendecomposition $\mathbf{\Sigma}_{\mathcal{R}}=\mathbf{USU}^{\mathrm{T}}$, where $\mathbf{S}$ is a nonnegative diagonal matrix. An orthogonal $m\times n$ matrix $\mathbf{P}$ minimizing the trace of $\mathbf{P}\mathbf{\Sigma}_{\mathcal{R}}\mathbf{P}^{\mathrm{T}}$ is a projection onto the space spanned by the m smallest eigenvectors of $\mathbf{\Sigma}_{\mathcal{R}}$, $\tilde{L}$ is given by

$$\mathbf{P}\mathbf{\Sigma}_{\mathcal{N}}^{-1/2}=(\mathbf{\Sigma}_{\mathcal{R}})_m^{-1/2}\mathbf{\Sigma}_{\mathcal{N}}^{-1/2}=\tilde{\mathbf{S}}_m^{-1/2}\tilde{\mathbf{U}}^{\mathrm{T}}\mathbf{\Sigma}_{\mathcal{N}}^{-1/2},\tag{7}$$

where $\tilde{\mathbf{S}}$ is the $m\times m$ matrix with the smallest eigenvalues and $\tilde{\mathbf{U}}$ is the $n\times m$ matrix with the corresponding eigenvectors (for notation brevity, we denote such a projection by $(\mathbf{\Sigma}_{\mathcal{R}})_m^{-1/2}$). This approach resembles the spirit of *linear discriminant analysis*. A similar technique has been introduced in [29] within the framework of boosted similarity learning. Note that the normalization of columns of $\mathbf{P}$ is unimportant since a sign function is applied to its output. However, we keep the normalization by the inverse square root of the variances, which makes the projected differences $\mathbf{P}(\mathbf{x}-\mathbf{x}')$ normal and white.

3.3.2 Difference of Covariances

An alternative approach can be derived by observing that

$$\tilde{L}=\mathrm{tr}\{\mathbf{P}\mathbf{\Sigma}_{\mathcal{D}}\mathbf{P}^{\mathrm{T}}\},$$

where $\mathbf{\Sigma}_{\mathcal{D}}=\alpha\mathbf{\Sigma}_{\mathcal{P}}-\mathbf{\Sigma}_{\mathcal{N}}$. This yields

$$\mathbf{P}=(\mathbf{\Sigma}_{\mathcal{D}})_m^{-1/2},\tag{8}$$

where at most m smallest *negative* eigenvectors are selected. This selection of the projection matrix will be referred to as *covariance difference* and denoted by DIF. Note that it allows controlling the trade-off between false positive and negative rates through the parameter α, which is impossible in the LDA approach.

The limit $\alpha\to\infty$ is of particular interest as it yields $\mathbf{\Sigma}_{\mathcal{D}}\propto\mathbf{\Sigma}_{\mathcal{P}}$. In this case, the negative covariance does not play any role in the training, which is equivalent to assuming that the differences of negative descriptor vectors are white Gaussian, $\mathbf{\Sigma}_{\mathcal{N}}=\mathbf{I}$. The corresponding projection matrix is given by

$$\mathbf{P}=(\mathbf{\Sigma}_{\mathcal{P}})_m^{-1/2}.\tag{9}$$

The main advantage of this approach is that it allows learning the projection in a semi-supervised setting when only positive pairs are available.

In general, a fully supervised approach is advantageous over its semi-supervised counterpart, which assumes a sometimes unrealistic unit covariance of the negative class differences. However, unlike the positive training set containing only pairs of knowingly matching descriptors, the negative set might be contaminated by positive pairs (a situation usually referred to as *label noise*). If such a contamination is significant, the semi-supervised setting is likely to perform better.

3.4 Threshold Selection

Given the projection matrix $\mathbf{P}$ selected as described in the previous section, our next step is to minimize a smooth version of the loss function (3),

$$\begin{aligned}L&=\mathrm{E}\{\mathrm{sign}(\mathbf{Px}+\mathbf{t})^{\mathrm{T}}\mathrm{sign}(\mathbf{Px}'+\mathbf{t})|\mathcal{N}\}\\&\quad-\alpha\mathrm{E}\{\mathrm{sign}(\mathbf{Px}+\mathbf{t})^{\mathrm{T}}\mathrm{sign}(\mathbf{Px}'+\mathbf{t})|\mathcal{P}\}\\&=\sum_{i=1}^{m}\mathrm{E}\{\mathrm{sign}(\mathbf{p}_i^{\mathrm{T}}\mathbf{x}+t_i)\mathrm{sign}(\mathbf{p}_i^{\mathrm{T}}\mathbf{x}'+t_i)|\mathcal{N}\}\\&\quad-\alpha\mathrm{E}\{\mathrm{sign}(\mathbf{p}_i^{\mathrm{T}}\mathbf{x}+t_i)\mathrm{sign}(\mathbf{p}_i^{\mathrm{T}}\mathbf{x}'+t_i)|\mathcal{P}\},\end{aligned}\tag{10}$$

with respect to the thresholds $\mathbf{t}$, where $\mathbf{p}_i^{\mathrm{T}}$ denotes the ith row of $\mathbf{P}$ and t_i denotes the ith element of $\mathbf{t}$. Observe that due to its separable form, the problem can be split into independent subproblems:

$$\begin{aligned}\min_{t_i}\quad&\mathrm{E}\{\mathrm{sign}((\mathbf{p}_i^{\mathrm{T}}\mathbf{x}+t_i)(\mathbf{p}_i^{\mathrm{T}}\mathbf{x}'+t_i))|\mathcal{N}\}\\&-\alpha\mathrm{E}\{\mathrm{sign}((\mathbf{p}_i^{\mathrm{T}}\mathbf{x}+t_i)(\mathbf{p}_i^{\mathrm{T}}\mathbf{x}'+t_i))|\mathcal{P}\},\end{aligned}\tag{11}$$

which in turn can be solved using simple 1D search over each threshold t_i.

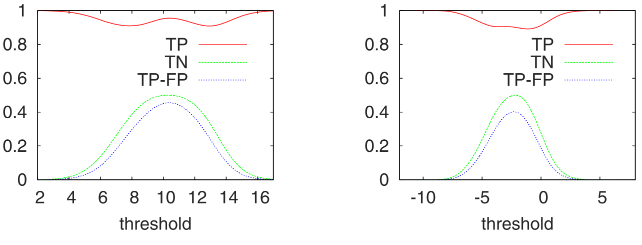
Let $y=\mathbf{p}_i^{\mathrm{T}}\mathbf{x}$ and $y'=\mathbf{p}_i^{\mathrm{T}}\mathbf{x}'$ be the ith element of the projected training vectors $\mathbf{x}$ and $\mathbf{x}'$. The ith bits of $\mathbf{y}$ and $\mathbf{y}'$ coincide if $t_i<\min\{y,y'\}$ or $t_i>\max\{y,y'\}$, and differ if $\min\{y,y'\}\le t_i\le\max\{y,y'\}$. For a given value of the threshold, we express the false negative rate as

$$\begin{aligned}\mathrm{FN}(t)&=\Pr(\min\{y,y'\}\ge t\text{ or }\max\{y,y'\}<t|\mathcal{P})\\&=1-\Pr(\min\{y,y'\}<t|\mathcal{P})+\Pr(\max\{y,y'\}<t|\mathcal{P})\\&=1-\mathrm{cdf}(\min\{y,y'\}|\mathcal{P})+\mathrm{cdf}(\max\{y,y'\}|\mathcal{P})\end{aligned}\tag{12}$$

with cdf standing for cumulative distribution function. Similarly, the false positive rate can be expressed as

$$\begin{aligned}\mathrm{FP}(t)&=\Pr(\min\{y,y'\}<t\le\max\{y,y'\}|\mathcal{N})\\&=1-\Pr(\min\{y,y'\}\ge t\text{ or }\max\{y,y'\}<t|\mathcal{N})\\&=\mathrm{cdf}(\min\{y,y'\}|\mathcal{N})-\mathrm{cdf}(\max\{y,y'\}|\mathcal{N}).\end{aligned}\tag{13}$$

We compute histograms of minimal and maximal values of projected positive and negative pairs, from which the cumulative densities are estimated. The optimal threshold t_i



(a) (b)

Fig. 1. The probability density functions for the classification performance for positive and negative training examples (a) for the first two dimensions and (b) for DIF.

is selected to minimize FP + FN (or, alternatively, maximize TN + TP, where TP = 1 − FN and TN = 1 − FP are the true positive and true negative rates, respectively). Fig. 1 visualizes TP, TN, and TP − FP for the first two components $i = 1, 2$ of the projections LDA and DIF.

4 Training Methodology

In this section, we first describe our ground truth training and evaluation data. We then evaluate different aspects of our binary descriptors.

4.1 Ground Truth Data

To build our ground truth database, we used sets of calibrated images for which we show the 3D point model and a member image in Figs. 3, 4, 14, 15, and 16. These data sets contain images we acquired ourself, such as those in Figs. 14 and 15, and sometimes over extended periods of time (Fig. 3). Those of Figs. 3, 4, and 15 contain images downloaded from the Internet or are fully acquired from this source, as in the case of Fig. 16.

We used our own calibration pipeline [33] to register them and to compute internal and external camera parameters as well as a sparse set of 3D points, each corresponding to a single keypoint track. First, pairwise keypoint correspondences are established using Vedaldi's [34] SIFT [1] descriptors that we compared using the standard L_2 norm. These are transformed into keypoint tracks which are used to grow initial reconstructions that have been obtained by a robust fit of pairwise essential matrices. This standard procedure is similar to [35] and we refer to this and our work [33] for more information.

Because our data set contains multiple views of the same scene, we have many conjunctive closure matches [36] such as the one depicted by the blue line in Fig. 3 (bottom): A keypoint that is matched in two other images, as depicted by the green lines, gives rise to an additional match in these other two images. Since they may be quite different from each other, the L_2 distance between the corresponding descriptors may be large. Yet, the descriptors in all three images will be treated as belonging to the same class, which is key to learning a metric that can achieve better matching performance than the original L_2 norm. In our data sets, these conjunctive closures partially build long chains for which individual pairs can have quite large L_2 norm as one can see in Fig. 2. In practice, we consider only chains with five or more keypoints, i.e., 3D points that are visible in at least five images.

For the negative examples, we randomly sampled the same number of keypoint pairs and checked that none of them belonged to the positive set.

This training database is more specific than the one used in [8] and [19], where the authors use a calibrated database of images and their dense multiview stereo correspondences. However, calibration and dense stereo information is used to extract the image patches which are centered around 3D point projections and use these to build a training database of positive matches. In our framework, we use the calibration only to geometrically *verify* SIFT matches as being consistent with the camera parameters and with the 3D structure. The 2D position, scale, and orientation of the original interest points is kept such that we can perform

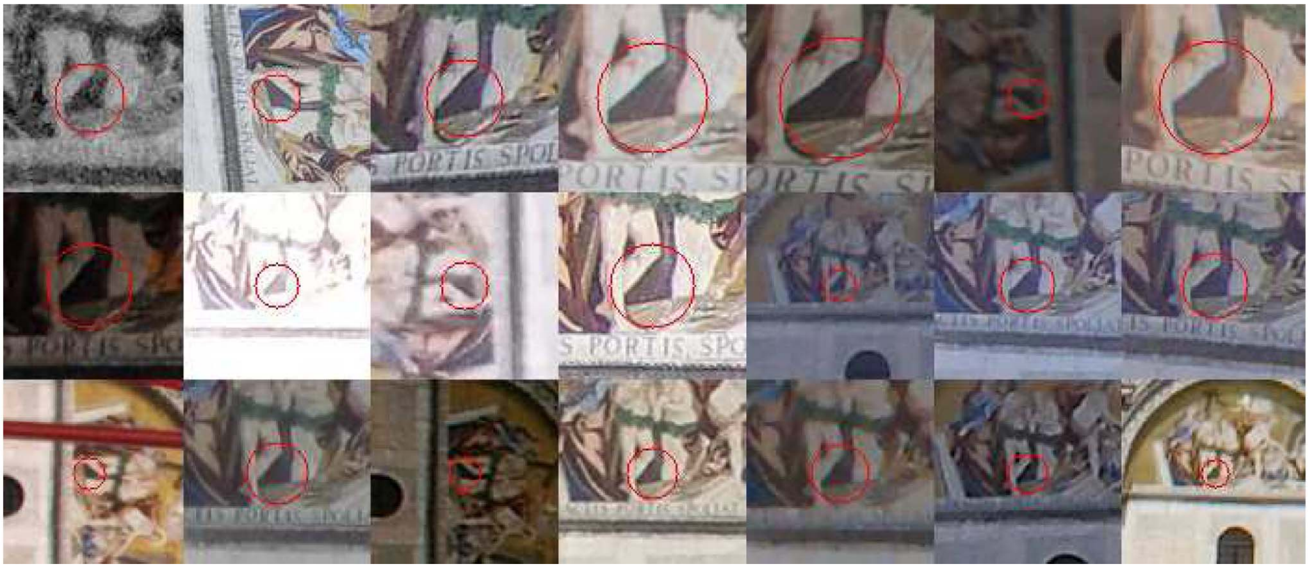


Fig. 2. Some of the keypoints from the same 3D point for the Venice data set in Fig. 16 are shown as an example. The red circle shows the keypoint (DoG) position and its scale. The track was extracted by consecutive SIFT L_2 matching, which makes it possible to include keypoint pairs (conjunctive closures) that are quite different into the training and evaluation set.

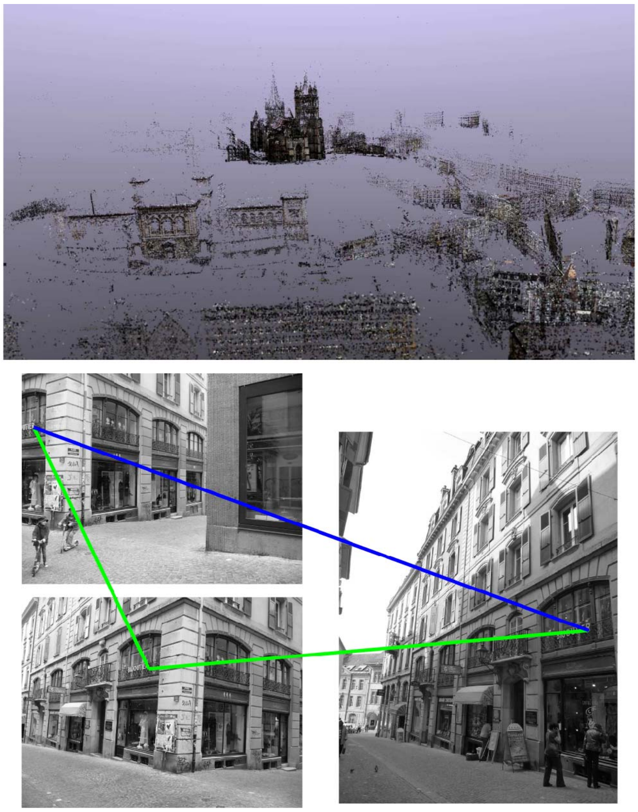


Fig. 3. Top row: Calibrated model of Lausanne with 4,485 images and 1.264 M 3D points that are computed from 9.9 M feature points. Bottom row: Three sample images from the data set with a transitive closure indicated.

learning on the data that are actually extracted by the combination of SIFT keypoint detection (Difference of Gaussians (DoG)) and description.

In [6] and [18], stereo correspondences are used to build a training database of positive keypoint pairs, similar to ours. This approach has advantages if the computed stereo correspondences are reliable even for image pairs with strong appearance changes. However, it is likely that ground truth correspondences for which SIFT already give good results are overrepresented by this strategy [18]. Here, we put more effort into build long chains of subsequent matches that end up describing the huge variability of features represented by the same 3D point.

To train our descriptors, we use the Lausanne data set of Fig. 3. Approximately, 9.9 M feature points are extracted and their triangulation produced about 1.3 M 3D points, such as those depicted in the top of Fig. 3. The urban area represented here covers nearly 2 square kilometers and encompasses the appearance statistics of man-made scenes. Vegetation also appears but is not extensively represented in this database. This training database finally consists of about 72 M positive and negative matching pairs from nearly 8 M keypoints. For testing, we used the data sets in Figs. 4, 14, 15, and 16 as well as Lidar ground truth data and planar image pairs as described in Section 5.1.

4.2 Parameter Evaluation

In the following, we evaluate the two steps in our optimization: 1) the computation of $\mathbf{P}$, which results in a dimensionality reduced floating-point feature vector and 2) the estimation of the thresholds that perform the binarization. For this evaluation, we use a set of images from different cities of Figs. 4, 14, 15, and 16. These provide positive and negative matching examples, which we use to compute the ROC statistics for different descriptor distances, i.e., L_2 ball or Hamming cube. We use the same negative samples in all cases.

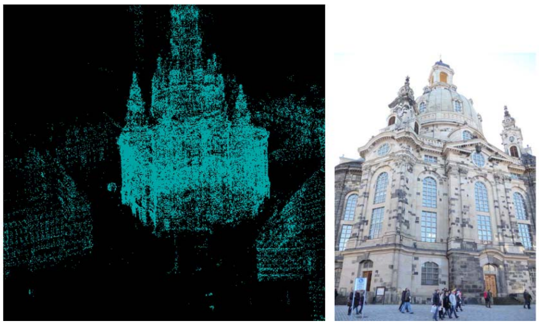


Fig. 4. The Dresden data set used for the evaluation in Figs. 6 and 7 contains 4,551,124 positive and negative matches, which are obtained by geometric verification using the full calibration.

All ROC curves are plotted in log scale for the FP rate, since the operating point for large-scale image retrieval systems requires very low FP rates. For example, a value of $\mathrm{FP} = 0.01$ (1 percent) for the Dresden data set with 4.5 M positive and negative matching examples will result in 45 K false positives, which is far more than retrieval systems could possibly handle. We are thus interested in performance at $\mathrm{FP} \ll 1\%$.

Throughout the paper, we use the following convention to the algorithms we compare: *Metric-Projection-Size*. The *metric* can either be L_2 (euclidean) or H (Hamming on the binarized vectors). *Projection* denotes the way in which the projection matrix $\mathbf{P}$ is computed: LDA (linear discriminant according to (7)) or DIF (difference of covariances according to (8)). *Size* denotes the descriptor length in bits.

4.3 The Choice of α in DIF Projections

Fig. 5 shows the performance of the DIF formulation when the relative influence of positive and negative training data is varied. This is achieved by α in (8). $\alpha = 10$ leads to the best

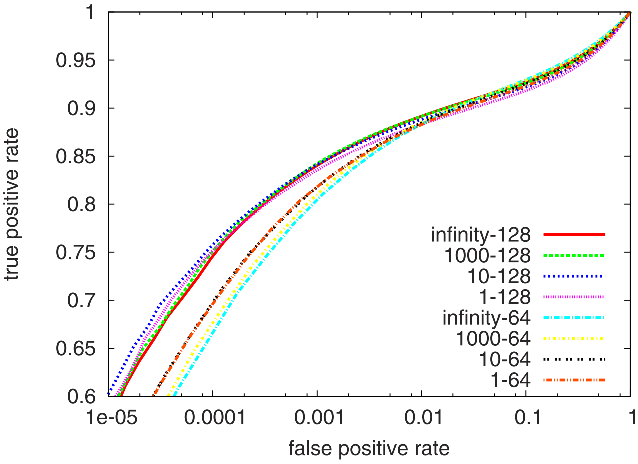


Fig. 5. Performance evaluation for the DIF binarization as a function of α for 128 and 64 bits on the Dresden data set shown in Fig. 4. The label on each curve indicates α—number of bits.

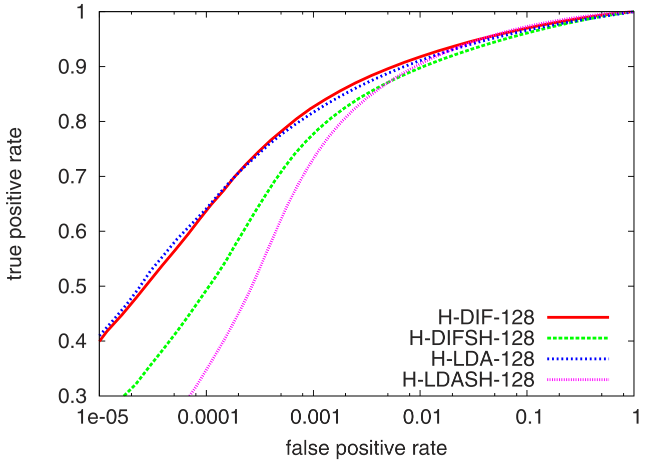


Fig. 6. Performance evaluation for the binarization used in spectral hashing [14] (denoted by the ending SH for each projection) with our proposed threshold optimization in Section 3.4 for the Venice data set shown in Fig. 16. Note that our threshold selection outperforms the corresponding SH formulation over the full false positive range.

results for both 128 and 64-bit descriptors. Note that this experiment also includes the case where only positive matches are taken into account, i.e., the approach with $\alpha = \infty$. All remaining results in this paper will therefore use $\alpha = 10$ and we denote the corresponding binarization by DIF.

4.4 Linear Projection

We estimated the parameters $\mathbf{P}$ of our projection matrix of (1) to produce descriptors of size $m = 64$ and 128 for DIF and LDA. The projection by $\mathbf{P}$ results in floating point descriptors $\mathbf{y} = \mathbf{P}\mathbf{x}$ which we compare in Fig. 7(left) to SIFT [1], [34] and to DAISY [6], [18]. For DAISY, we used software provided by Simon Winder, who also suggested the optimal parameters.[1]

As shown in Fig. 7(left), LDA projections improve the results when compared to SIFT. By contrast, DIF projections perform worse than the original SIFT descriptors. This effect is stronger when we reduce the dimensions to 64. However, after binarization, these results change as will be shown next.

4.5 Binarization

In Fig. 6, we compare our supervised threshold optimization with the spectral hashing approach [14], which has been shown to outperform many other hashing approaches such as restricted Boltzmann machines and locality-sensitive hashing [14]. Spectral hashing first applies a PCA projection of the feature space. Then, the bounding box of all feature vectors is computed and the binarization is realized by looking at the sign of the analytical eigenfunctions in that box for each dimension. The SH approach selects the m smallest of those eigenfunctions. Instead of applying PCA projections, we show the performance of this particular binarization scheme for DIF and LDA projections, denoted as H-DIFSH-128 and H-LDASH-128. This is compared to our supervised threshold optimization (H-DIF-128 and H-LDA-128) in Fig. 6. One can see that our supervised binarization scheme, as described in Section 3.4 does increase performance substantially over the corresponding unsupervised spectral hashing formulation. Note also that SH binarization is related to feature discretization, which tries to approximate floating-point feature vectors by fewer bits in each dimension. Without sorting the m smallest eigenfunctions or equally scaling each dimension of the feature space to the same range, SH corresponds to a discretization of each feature dimension into several bits.[2] Unsupervised feature discretization, as used by Brown et al. [18], will therefore show a similar behavior as SH binarization does.

4.6 Combined Comparison

In Fig. 7(right), we show the final result of our binarized descriptors in comparison to other approaches. One can see that if the data are transformed according to the covariance structure of the feature space (by LDA or DIF), we get a significant performance boost by using the Hamming metric on binarized descriptors. This can be seen even for H-DIF-128, for which the unbinarized version L2-DIF-4096 performs worse than SIFT. If, on the other hand, the feature space is not aligned with the covariance structure, binarization does not improve, e.g., for random orthogonal projections H-RANORTH-128. Fig. 7 also shows the results of similarity-sensitive hashing proposed in [10] and used in [40], the results of DAISY [6], [18], and spectral hashing [14]. Our approach shows significantly better performance in the interesting area of low false positive rates and reaches the performance of the other descriptors for high true positive rates with a much smaller descriptor size. In the next sections (Sections 5.1 and 5.2), we show similar or better behavior on more difficult data sets of our approach on many other test sequences.

Note also the improvement of the binarization with respect to the unbinarized projection by comparing Figs. 7(left) and 7(right) for LDA and DIF. An improvement by quantization was also reported by Brown et al. [18], where the range of each descriptor coordinate has been binarized to fit various bit sizes.

In Fig. 8, we show the performance with varying number of bits for DIF binarization and we compare it to the SIFT baseline performance.

5 Experimental Evaluation

In this section, we compare the performance of our approach to metric learning against state-of-the-art methods [10], [14], [18] and use SIFT [1] as a baseline. We first do this using image pairs for which LIDAR data, and therefore ground truth correspondences, are available. We then move on to the large-scale data sets presented in Section 5.2 to validate our approach in a more challenging context.

5.1 LIDAR Ground Truth Evaluation

We evaluated the performance of our binarized descriptor on publicly available data sets [41], [38], for which camera parameters and the ground truth 3D model are available. The dense ground truth cloud of 3D points was obtained by using LIDAR and was registered to the images, making it

[1] The DAISY parameters used: 1) the keypoint scale, which transforms the SIFT scale parameter to DAISY scale, was set to 1.6 and 2) the descriptor **T2 4 2r6s** making up a 52-dimensional feature representation of unsigned char values was used in all experiments. For additional details, see [6] and [18].

[2] The number of bits depends on the frequency of the harmonic eigenfunctions and can be chosen (see [14] for more details).

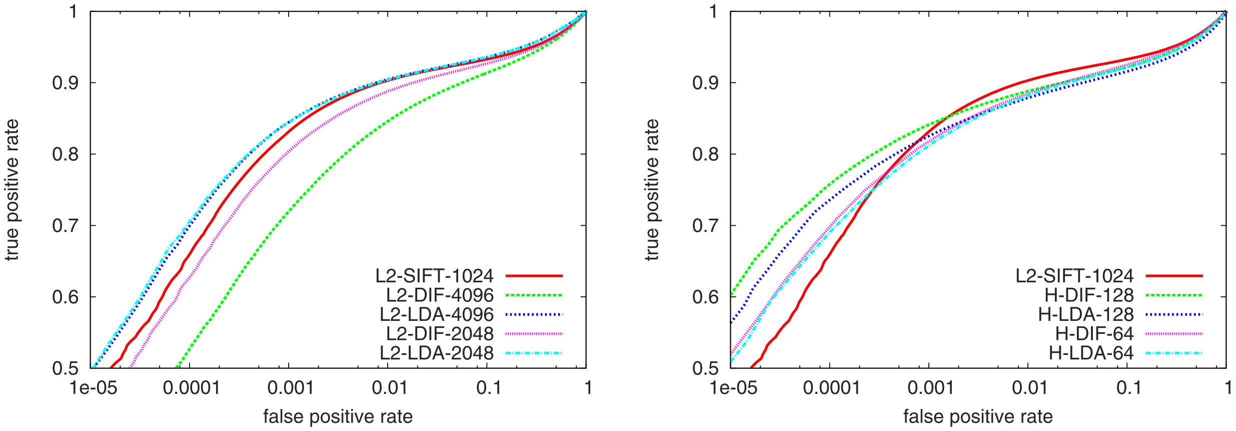


Fig. 7. Left: Performance evaluation for the projection $\mathbf{P}$ for our methods (DIF and LDA) in comparison to the original SIFT and to the DAISY descriptor on the Dresden data set shown in Fig. 4. Right: Performance evaluation for various descriptors for the same data set after binarization. We compare our binary descriptors with Locality-Sensitive Hashing in [10] (H-SSH-128), DAISY [6] (L2-DAISY-416), SIFT [1] (L2-SIFT-1024), and random orthogonal projections (H-RANORT-128). Note that binarization improves the performance for the interesting area of the ROC curves at a low false positive rate.

easy to find the corresponding pixel in any image to a pixel in any other. Occluded areas can by identified, and have been excluded from the evaluation, by geometric visibility reasoning. These high precision evaluation data do contain real 3D distortions which are different from the well-known data set of Mikolajczyk et al. [2], where the images are related by a single homography. It does therefore allow us to evaluate more realistic scenarios.

We focus on two pairs of the Fountain-P11 and the Herz-Jesu-P8 data sets depicted in Fig. 9. For both data sets, we present the results for a small baseline and a wide-baseline setting. These data sets and the evaluation procedure will be publically available [39]. In addition, we show results on the standard graffiti and wall data sets of Mikolajczyk et al. [2], which consist of planar scenes, making it easy to establish dense correspondence by a homography.

In Figs. 10, 11, and 12, we plot ROC and precision-recall curves that summarize the corresponding matching performance using the various descriptors. These curves were obtained as follows: First, SIFT keypoints were detected in all images. From these, we filtered out all keypoints for which there were no ground truth matches, either due to missing LIDAR data or occlusions. For each of the remaining keypoints in one image, we search for the corresponding keypoint in the other image and check whether it is less than 2 pixels[3] away from the ground truth LIDAR match. To enforce consistency, we switched the roles of the images and performed the same operation. This provided us with ground truth keypoint correspondences and we further did the evaluation only on those keypoints. By varying the matching threshold on either the L_2 norm or Hamming distance, we counted the number of true and false positives to obtain the ROC curves. By using the same set of keypoints, the recall is defined by the relative amount of true positives and precision by the number of true positive relative to the total number of retrieved keypoints.

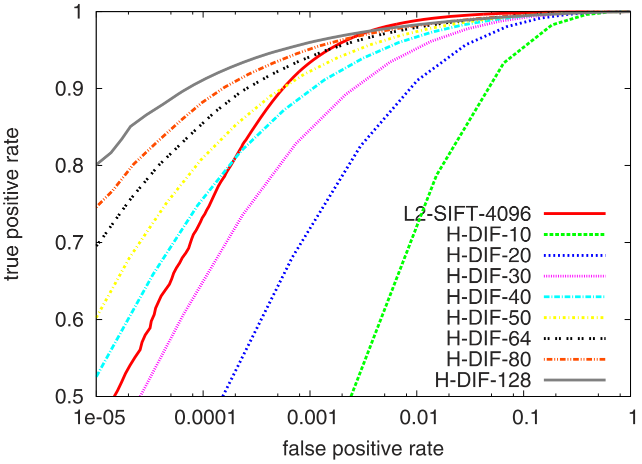


Fig. 8. Performance of DIF with varying number of bits on the Karls bridge data set of Prague [37]. As a reference, we include the original SIFT performance.

In the fountain-P11 and Herz-Jesu-P8 data sets (Figs. 10 and 11), the 128-bit binary descriptors significantly outperform SIFT. This performance boost is achieved with a descriptor size which is eight times less than the number of bits original SIFT requires (1,024). Even if we halve the size of our descriptors to 64 bits, we get results that are similar and in some cases superior to those of SIFT in accuracy, while being 16 times more compact. This dependence of the descriptor size is depicted in Fig. 13. These experiments show a significant improvement of DAISY when compared to SIFT, which was also reported by their authors in [6] and [18]. When compared to current state-of-the-art hashing approaches [14], spectral hashing, and similarity-sensitive hashing, using the same descriptor size (128 bits), we can appreciate a performance boost over the full precision/FP range. Our DIF projections are slightly better than LDA projections and still perform very well with only 64 bits. In the Mikolajczyk data set 12, the results do not show a clear direction. This is grounded in the small number of ground truth matches (680 and 375) that make matching confusions less likely and on the fact that the image pairs are relatively easy.

[3] We used this value since we are primarily interested in high precision matches which are needed for calibration purposes. We also checked different values and obtained very similar results.

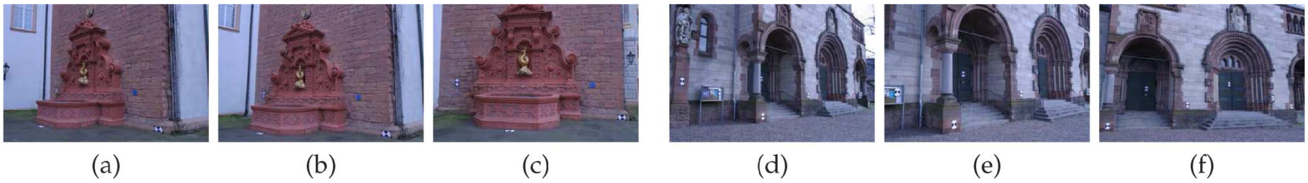
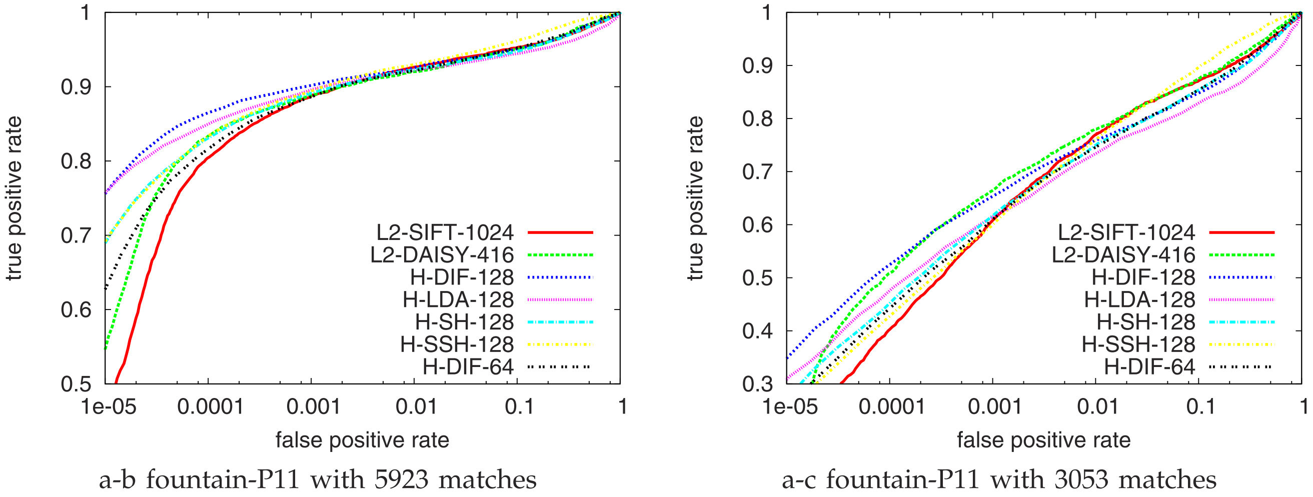


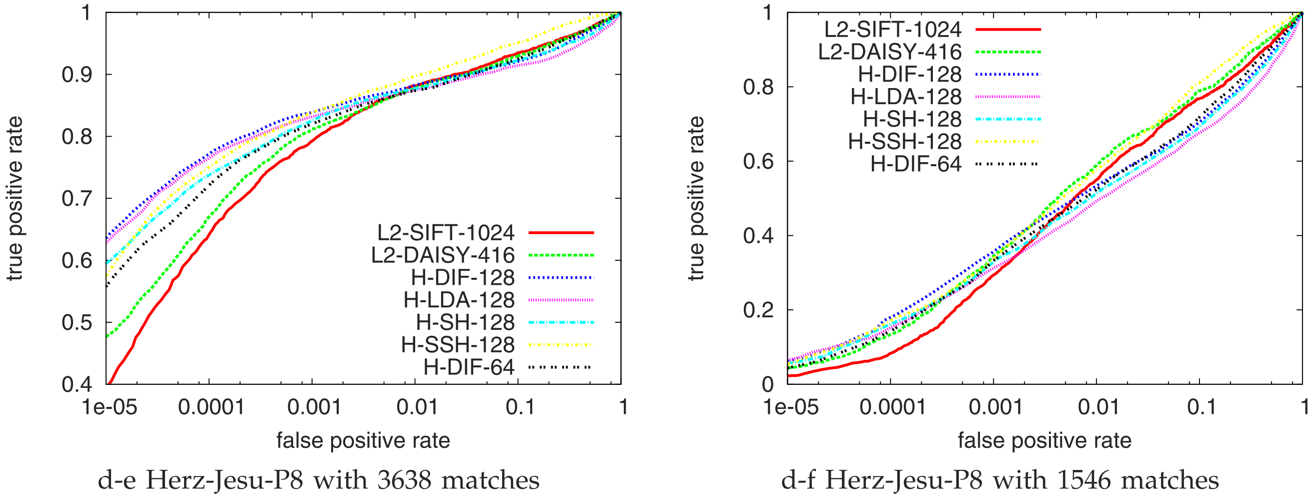
Fig. 9. Images used for quantitative evaluation. Dense ground truth correspondences are available from LIDAR measurements for fountain-P11 (top) and Herz-Jesu-P8 (bottom) [38]. The matching performance of the image pairs a-b and a-c as well as d-e and d-f is shown in Figs. 10 and 11. The data are publically available [39].



a-b fountain-P11 with 5923 matches

a-c fountain-P11 with 3053 matches

Fig. 10. ROC curves for binarized and original SIFT as well as DAISY, SH, and SSH on the fountain image pairs shown in Fig. 9. When using 128-bit descriptors, we systematically outperform all other methods and perform at least similarly when using 64-bit descriptors. Precision versus recall curves are shown in [37].



d-e Herz-Jesu-P8 with 3638 matches

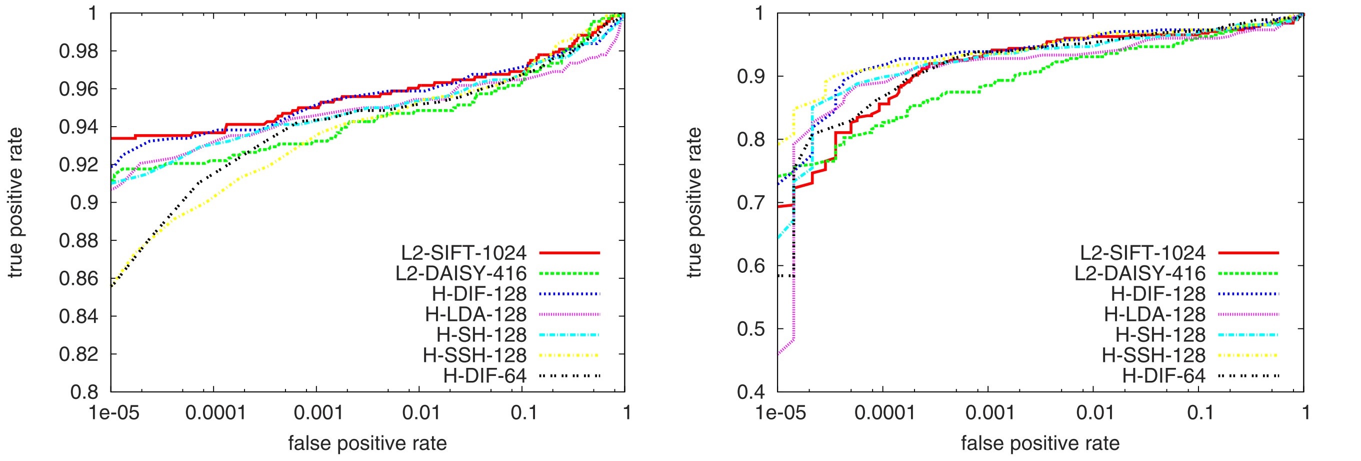
d-f Herz-Jesu-P8 with 1546 matches

Fig. 11. ROC curves for binarized and original SIFT as well as DAISY, SH, and SSH on the Herz-Jesu image pairs shown in Fig. 9. When using 128-bit descriptors, we systematically outperform all other methods and perform at least similarly when using 64-bit descriptors. Precision versus recall curves are shown in [37].

5.2 Large-Scale Ground Truth Evaluation

To test our hashing scheme for large-scale keypoint retrieval on substantially different images, we calibrated four other data sets depicted in Figs. 14, 15, and 16 using SIFT L_2 norm matching as described in Section 4.1. The first data set consists of 71 aerial images (41 M pixels) and the other three of 192, 107, and 310 urban images. All data sets contain millions of matching examples and especially the Venice data set, with about 13 million data points, also covers interesting situations with strong light and scale changes. The ROC curves are shown in Figs. 14, 15, and 16. Overall, we get an improvement in performance for these large-scale data sets, which indicates that our learning scheme generalizes properly and scales well.

The first three data sets are relatively easy. Baselines in these data sets are small and many of the images are taken under similar light conditions, which is especially true for the aerial data set in Fig. 14. As a result, the improvement of our metric learning is less pronounced than in the last example of Venice (Fig. 16). This data set contains images from photo community collections taken by many different users at different times. One can notice here a significant improvement for 128-bit LDA and DIF projections as well as for 64-bit DIF projections for low false positive rates. More particularly,



(a) wall img1-img2 with 680 matches

(b) graffity img1-img2 with 375 matches

Fig. 12. ROC curves for binarized and original SIFT as well as DAISY, SH, and SSH on the image pairs of wall (a) and graffiti (b) (top) from [2]. Precision versus recall curves are shown in [37].

as can be conducted from the graphs, we retrieve the correct keypoint in 83 percent (78 percent) of the cases with 128 (64) bits at $\mathrm{FP} = 0.001$ (corresponding to 12,796 false positives in total), which is substantially better than SIFT and DAISY-416 with 56 and 69 percent, respectively. At the same time, we need only 12.5 percent (6.25 percent) of the space and bandwidth to store and transfer the descriptors for processing. The difference is much more outspoken if we go to more realistic, lower values of the false positive rate.

If we compare the performance of the descriptors with 128 bits and less, we outperform the other approaches SSH, SH, and DAISY-128 over the full false positive range.

The improvement of our metric learning scheme can be explained by the large amount of conjunctive closure matches in our training set. They are true matches, in that they correspond to the projection of the same physical 3D point but may be relatively far apart when compared by the SIFT L_2 norm. Our hashing scheme accounts for that and brings those keypoints closer in the Hamming space. This results in an even greater performance boost over SIFT when wide baseline and small baseline are compared, as seen in Figs. 10 and 11, and when the images contain strong appearance changes, as in the Venice data set shown in Fig. 16. We note that the use of a single global projection of the data is potentially limiting full exploitation of the wide-baseline data. Training a sequence of projections where a subsequent projection is trained on the errors of the previous ones could allow circumventing this limitation.

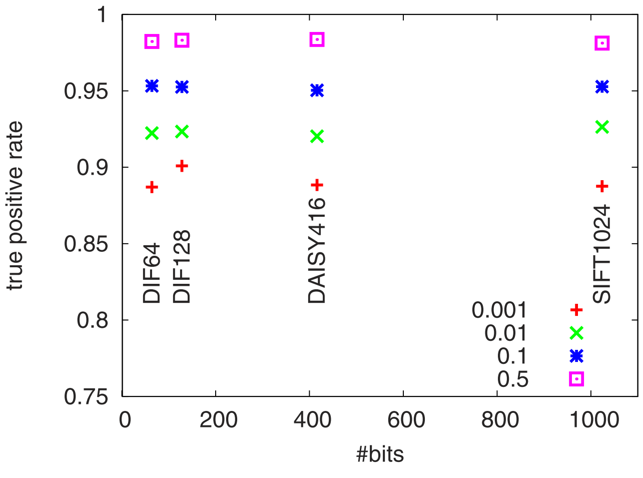


Fig. 13. Descriptor performance as a function of their size for the fountain data set in Fig. 10(top left) for various false positive rates.

Our evaluation confirms earlier results on the performance of the (52-dimensional) DAISY descriptor [6], [18]

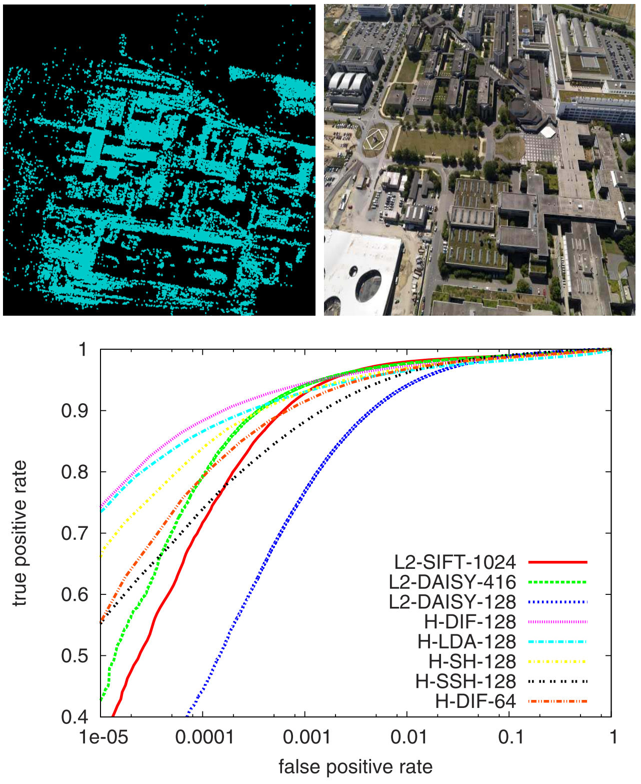


Fig. 14. ROC curves for our binary descriptors together with original SIFT, DAISY [6], spectral hashing [14], and boosted learning in [10] on an aerial image set with 6,375,139 positive and negative matching examples. Note that this test image set is also very different from our terrestrial image training set in that more vegetation is present. The performances H-DIF-16 and H-LDA-16 indicate a good generalization of our learning procedure.

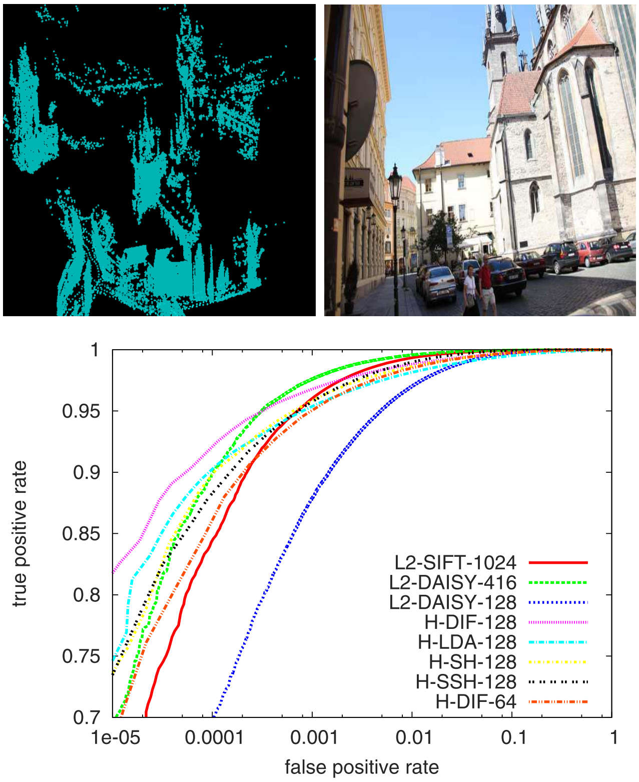


Fig. 15. ROC curves similar to Fig. 14 on the urban data set of Prague with 2,027,389 positive and negative matching examples.

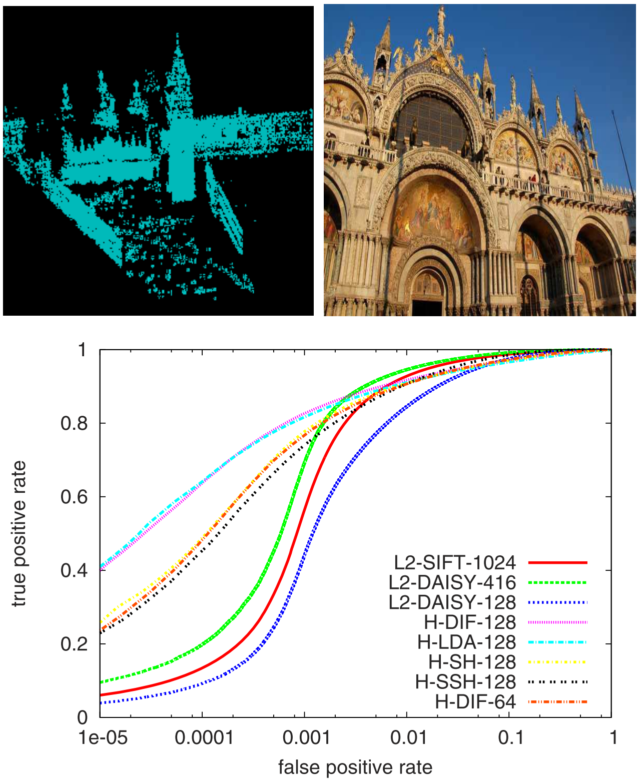


Fig. 16. ROC curves for our learned binary descriptors together with original SIFT, DAISY [6], [18], spectral hashing [14], and boosted learning in [10] on the flickr data set of Venice, with 12,796,971 positive and negative matching examples. This data set contains images taken by different cameras and with different light, weather, and seasonal conditions. For this reason and for its size, it is the most challenging data set.

when compared to SIFT, which is visible especially in the large-scale data sets. To build the DAISY descriptor, an extensive optimization of the filter locations that are used to fill up the descriptor bins has been performed. This was not done here. Surprisingly, the good low false positive performance of our descriptors when compared to DAISY-416 is consistent and could be explained by the difference in generating the training data (as discussed in Section 4.1) and by the fact that DAISY does not use supervision for its last, quantization step. We think that this is important and show here, as seen in Fig. 6, that it leads to a larger performance boost than the unsupervised quantization strategy used by DAISY.

Our experiments show that DIF projections perform slightly better than LDA projections.

5.3 Dependence on Keypoint Detector

Local keypoint descriptors are often highly coupled to keypoint detectors since computation time can be saved by this strategy. For all evaluation so far, we used the SIFT-related keypoint detector which is based on Difference of Gaussians [1]. DAISY [6] and SURF [3] use other keypoint detectors, which are based on Laplacians and Hessians, respectively. An evaluation on the matching performance for SIFT, DAISY, and SURF *with their own* keypoint detectors is shown in Fig. 17. For a fair comparison, we sampled for each keypoint detector a constant number of 5,000 matches for the fountain data set in Figs. 9a and b. The results show that the DoG keypoint detector performs best and that DAISY gives better results on those keypoints when compared to its own keypoint detections.

6 Conclusions

We presented a novel and simple approach to produce a binary string from a SIFT descriptor. Our approach first aligns the SIFT descriptors according to the problem specific covariance structure. In the resulting vector space, all SIFT descriptors have diagonal covariance. We can then

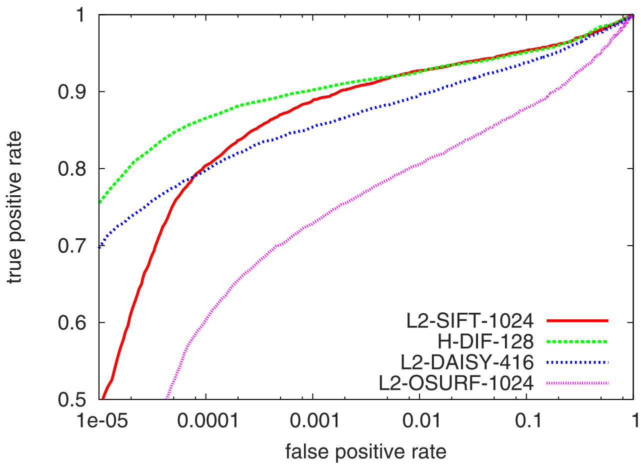


Fig. 17. ROC curves for the performance of the descriptors on their own keypoint detector with L2-SIFT-1024 and H-DIF-128 using DoG keypoints, L2-DAISY-416 using Laplacian Keypoints [6], and L2-OSURF [3] using Hessian keypoints. We use 5,000 ground truth keypoints on the fountain data set depicted in Fig. 9a-b.

estimate reliable thresholds that perform the binarization according to an appropriate cost function. This approach is very fast and can be used for many other applications for which similar training data are available.

We showed in this paper that this very simple and general approach leads to outstanding matching results with a very compact descriptor. Our resulting binary descriptor performs better than original SIFT [1], [34] and DAISY [6], [18] in the low false positive range, which is the interesting range for large-scale keypoint retrieval applications. Thereby, our 128-bit version requires only $\approx$10% of the size SIFT uses to ($\approx$25% of the DAISY size, respectively) describe keypoints. When compared to locality-sensitive hashing [10] and spectral hashing proposed by Weiss et al. [14], which use the same number of bits to encode keypoints, our descriptors perform better in the whole false positive range. This is also true if we compare to a reduced size DAISY with 128 bits.

Very good performance for low false positive rates can be obtained by using as few as 64 bits (H-DIF-64), which makes it possible to search efficiently in a large database. Matching is very fast for binary descriptors even for exhaustive search, since only an XOR followed by a bit count is needed to compute the Hamming distance (in some modern CPUs, bit counting is implemented as a single instruction). Moreover, binary descriptors with the Hamming metric can be indexed efficiently on existing database management systems, a direction we intend to explore in future research. We believe that matching of our binary representations can be performed very fast even on mobile devices and release our binarizations for SIFT into the public domain [42].

Philosophically, our approach addresses the gap between *modeling* and *learning* in feature descriptor design. The recent trend in computer vision literature has been to construct feature descriptors that would theoretically be invariant to certain transformations such as rotations or affine transformations. However, such transformations are only approximations of the real image formation model, and thus the descriptor is never truly invariant. Augmenting it with a metric learning approach, it is possible to learn invariance to typical transformations that may appear in a natural scene. It would be interesting to explore the tradeoff between how much effort should be invested in modeling invariance versus learning it from examples.

Interesting further research could look at other descriptors such as DAISY [6], SURF [3], or BRIEF [43], which are faster to compute and to learn a similar binarization. We also plan to investigate the performance of an additional network layer to reduce the size of our current binary descriptors even further and without loss in performance.

Acknowledgments

The authors would like to thank Matthew Brown for useful discussion and Simon Winder for providing his DAISY binary and for testing DAISY on the fountain sequence. They further acknowledge Flickr users for providing their images for the Venice and for parts of the Prague and Lausanne data sets as well as SenseFly (www.sensefly.com) for capturing the aerial image of the EPFL. Michael M. Bronstein is partially supported by the Swiss High-Performance and High-Productivity Computing (HP2C).

References

[1] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[2] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A Comparison of Affine Region Detectors," *Int'l J. Computer Vision* vol. 65, nos. 1/2, pp. 43-72, 2005.

[3] H. Bay, A. Ess, T. Tuytelaars, and L.V. Gool, "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding,* vol. 10, no. 3, pp. 346-359, 2008.

[4] E. Tola, V. Lepetit, and P. Fua, "Daisy: An Efficient Dense Descriptor Applied to Wide Baseline Stereo," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 5, pp. 815-830, May 2010.

[5] T. Tuytelaars and C. Schmid, "Vector Quantizing Feature Space with a Regular Lattice," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[6] S. Winder, G. Hua, and M. Brown, "Picking the Best DAISY," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2009.

[7] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 10, pp. 1615-1630, Oct. 2005.

[8] G. Hua, M. Brown, and S. Winder, "Discriminant Embedding for Local Image Descriptors," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[9] A. Gionis, P. Indik, and R. Motwani, "Similarity Search in High Dimensions via Hashing," *Proc. Int'l Conf. Very Large Databases,* 2004.

[10] G. Shakhnarovich, "Learning Task-Specific Similarity," PhD dissertation, Massachusetts Inst. of Technology, 2005.

[11] B. Kulis and T. Darrell, "Learning to Hash with Binary Reconstructive Embeddings," *Proc. Neural Information Processing Systems,* pp. 1042-1050, 2009.

[12] M. Raginsky and S. Lazebnik, "Locality-Sensitive Binary Codes from Shift-Invariant Kernels," *Proc. Advances in Neural Information Processing Systems,* 2009.

[13] M. Bawa, T. Condie, and P. Ganesan, "LSH Forest: Self-Tuning Indexes for Similarity Search," *Proc. 14th Int'l Conf. World Wide Web,* pp. 651-660, 2005.

[14] Y. Weiss, A. Torralba, and R. Fergus, "Spectral Hashing," *Proc. Advances in Neural Information Processing Systems,* vol. 21, pp. 1753-1760, 2009.

[15] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting Algorithms as Gradient Descent," *Proc. Neural Information Processing Systems,* pp. 512-518, 2000.

[16] K. Weinberger and L. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," *The J. Machine Learning Research,* vol. 10, pp. 207-244, 2009.

[17] C. Shen, J. Kim, L. Wang, and A. van den Hengel, "Positive Semidefinite Metric Learning with Boosting," *Proc. Computing Research Repository,* 2009.

[18] M. Brown, G. Hua, and S. Winder, "Discriminative Learning of Local Image Descriptors," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 1, pp. 43-57, Jan. 2011.

[19] S. Winder and M. Brown, "Learning Local Image Descriptors," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2007.

[20] V. Chandrasekhar, G. Takacs, D.M. Chen, S.S. Tsai, R. Grzeszczuk, and B. Girod, "Chog: Compressed Histogram of Gradients a Low Bit-Rate Feature Descriptor," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 2504-2511, 2009.

[21] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *Proc. IEEE Computer Vision and Pattern Recognition,* pp. 257-263, June 2003.

[22] K. Mikolajczyk and J. Matas, "Improving Descriptors for Fast Tree Matching by Optimal Linear Projection," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[23] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios, "Boostmap: A Method for Efficient Approximate Similarity Ranking," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* 2004.

[24] J. Wang, S. Kumar, and S.F. Chang, "Sequential Projection Learning for Hashing with Compact Codes," *Proc. Int'l Conf. Machine Learning,* 2010.

[25] P. Jain, B. Kulis, and K. Grauman, "Fast Image Search for Learned Metrics," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[26] A. Torralba, R. Fergus, and W.T. Freeman, "80 Million Tiny Images: A Large Dataset for Non-Parametric Object and Scene Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 11, pp. 1958-1970, Nov. 2008.

[27] H. Jégou, M. Douze, and C. Schmid, "Packing Bag-of-Features," *Proc. IEEE Int'l Conf. Computer Vision,* 2009.

[28] J. Wang, S. Kumar, and S.F. Chang, "Semi-Supervised Hashing for Scalable Image Retrieval," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

[29] A.M. Bronstein, M.M. Bronstein, and R. Kimmel, "Video Genome," technical report, Cornell Univ. Library, 2010.

[30] A. Bronstein, M. Bronstein, M. Ovsjanikov, and L. Guibas, "Shape Google: Geometric Words and Expressions for Invariant Shape Retrieval," *ACM Trans. Graphics,* vol. 30, 2011.

[31] H. Jegou, M. Douze, and C. Schmid, "Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search," *Proc. European Conf. Computer Vision,* pp. 304-317, 2008.

[32] H. Jégou, M. Douze, and C. Schmid, "Product Quantization for Nearest Neighbor Search," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 1, pp. 117-128, Jan. 2011.

[33] C. Strecha, T. Pylvanainen, and P. Fua, "Dynamic and Scalable Large Scale Image Reconstruction," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2010.

[34] A. Vedaldi, "An Open Implementation of the SIFT Detector and Descriptor," Technical Report 070012, Computer Science Dept., Univ. of California Los Angeles, 2007.

[35] N. Snavely, S. Seitz, and R. Szeliski, "Photo Tourism: Exploring Photo Collections in 3D," *Proc. ACM SIGGRAPH,* pp. 835-846, 2006.

[36] K. Heath, N. Gelfand, M. Ovsjanikov, M. Aanjaneya, and L.J. Guibas, "Image Webs: Computing and Exploiting Connectivity in Image Collections," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.

[37] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua, "LDAHash: Improved Matching with Smaller Descriptors," Technical Report EPFL-REPORT-152487, 2010.

[38] C. Strecha, W. von Hansen, L.V. Gool, P. Fua, and U. Thoennessen, "On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[39] C. Strecha and P. Fua, "Local Keypoint Evaluation," http://cvlab.epfl.ch/data/, 2010.

[40] A. Torralba, R. Fergus, and Y. Weiss, "Small Codes and Large Databases for Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2008.

[41] C. Strecha, "Multi-View Evaluation," http://cvlab.epfl.ch/data, 2008.

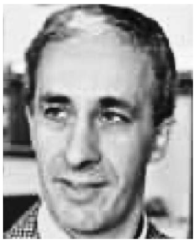
[42] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua, "LDAHash," http://cvlab.epfl.ch/software, 2010.

[43] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," *Proc. European Conf. Computer Vision,* 2010.

Christoph Strecha received the degree in physics from the University of Leipzig, Germany, and the PhD degree from the Catholic University of Leuven, Belgium, in 2008. He did his PhD thesis in computer vision in the field of multiview stereo. He joined EPFL (Swiss Federal Institute of Technology) in 2008 where he works as a postdoctoral researcher in the Computer Vision Group. His research interests include photogrammetry, structure from motion techniques, city modeling, multiview stereo, and optimization-based techniques for image analysis and synthesis. He is the cochair of Commission III/1 of the International Society for Photogrammetry and Remote Sensing and the founder of Pix4D.

Alexander M. Bronstein received the MSc degree summa cum laude in 2005 from the Department of Electrical Engineering and the PhD degree in 2007, respectively, from the Department of Computer Science of the Technion-Israel Institute of Technology, and in 2010 joined the School of Electrical Engineering at Tel Aviv University. Prior to that, he served as a scientist and the vice president of video technology at a Silicon Valley startup company Novafora, Inc., and held visiting appointments at Stanford University, Politecnico di Milano, and the University of Verona. His main research interests are computational shape analysis, computer vision, and machine learning. He is a member of the IEEE.

Michael M. Bronstein received the BSc degree summa cum laude in 2002 from the Department of Electrical Engineering and the PhD degree with distinction in 2007, respectively, from the Department of Computer Science of the Technion-Israel Institute of Technology. He is an assistant professor in the Institute of Computational Science of the Faculty of Informatics, Università della Svizzera Italiana (USI), Lugano, Switzerland. Prior to joining USI, he held a visiting appointment at Stanford university. His main research interests are theoretical and computational methods in metric geometry and their application to problems in computer vision, pattern recognition, shape analysis, computer graphics, image processing, and machine learning. He has authored a book, more than 60 publications in leading journals and conferences, and more than a dozen patents. He is a member of the IEEE.



Pascal Fua received the engineering degree from the Ecole Polytechnique, Paris, in 1984 and the PhD degree in computer science from the University of Orsay in 1989. He joined EPFL (Swiss Federal Institute of Technology) in 1996, where he is now a professor in the School of Computer and Communication Science. Before that, he worked at SRI International and at INRIA Sophia-Antipolis as a computer scientist. His research interests include shape modeling and motion recovery from images, analysis of microscopy images, and augmented reality. He has (co)authored more than 150 publications in refereed journals and conferences. He has been an associate editor of the journal *IEEE Transactions for Pattern Analysis and Machine Intelligence* and has often been a program committee member, area chair, and program chair of major vision conferences. He is a senior member of the IEEE.