Discriminative Learning of Local Image Descriptors

Matthew Brown, *Member, IEEE,* Gang Hua, *Member, IEEE* and Simon Winder, *Member, IEEE*

***Abstract*— In this paper we explore methods for learning local image descriptors from training data. We describe a set of building blocks for constructing descriptors which can be combined together and jointly optimized so as to minimize the error of a nearest-neighbour classifier. We consider both linear and non-linear transforms with dimensionality reduction, and make use of discriminant learning techniques such as Linear Discriminant Analysis (LDA) and Powell minimization to solve for the parameters. Using these techniques we obtain descriptors that exceed state-of-the-art performance with low dimensionality. In addition to new experiments and recommendations for descriptor learning, we are also making available a new and realistic ground truth dataset based on multi-view stereo data.**

***Index Terms*— image descriptors, local features, discriminative learning, SIFT**

I. Introduction

LOCAL feature matching has rapidly emerged to become the dominant paradigm for recognition and registration in computer vision. In traditional vision tasks such as panoramic stitching [1], [2] and structure from motion [3], [4], it has largely replaced direct methods due to its speed, robustness, and the ability to work without initialization.

It is also used in many recognition problems. Vector quantizing feature descriptors to finite vocabularies and using the analogue of “visual words” has enabled visual recognition to scale into the millions of images [5], [6]. Also the statistical properties of local features and visual words have been exploited by many researchers for object class recognition problems [7], [8], [9].

However, despite the proliferation of learning techniques that are being employed for higher level visual tasks, the majority of researchers still rely upon a small selection of hand coded feature transforms for the lower level processing. A good survey of some of the more common techniques can be found in [10], [11]. Some exceptions to this rule and good examples of low-level feature learning include the work of Lepetit and Fua [12], Shotton et al [13] and Babenko [14]. Lepetit and Fua [12] showed that randomized trees based on simple pixel differences could be an effective low level operation. This idea was extended by Shotton et al [13], who demonstrated a compelling scheme for object class recognition. Babenko et al. [14] showed that boosting could be applied to learn point based feature matching representations from a large training dataset. Another example of learning low level image operations is the Berkeley edge detector [15], which, rather than being optimized for recognition performance per se, is designed to mimic human edge labellings.

Progress in image feature matching improved rapidly following Schmid and Mohr’s work on indexing using grey-value invariants [16]. This represented a step forward over previous approaches to invariant recognition that had largely been based on geometrical entities such as edges and contours [17]. Another landmark paper in the area was the work of Lowe [18], [19] who demonstrated the importance of scale invariance and a non-linear, edge-based descriptor transformation inspired by the ideas of Hubel and Wiesel [20]. Since then small improvements have resulted, mainly due to improved spatial pooling arrangements that are more closely linked to the errors present in the interest point detection process [11], [21], [22].

One criticism of the local image descriptor designs described above has been the high dimensionality of descriptors (e.g., 128 dimensions for SIFT). Dimensionality reduction techniques can help here, and have also been used to design features as well. A first attempt was PCA-SIFT [23], which used the principal components of gradient patches to form local descriptors. Whilst this provides some benefits in reducing noise in the descriptors, a better approach is to find projections that actively discriminate between classes [24], instead of just modelling the total data variance. Such techniques have been extensively studied in the face recognition literature [25], [26], [27].

Our work attempts to improve on the state of the art in local descriptor matching by learning optimal low-level image operations using a large and realistic training dataset. In contrast to previous approaches that have used only planar transformations [11] or jittered patches [12] we use actual 3D correspondences obtained via a stereo depth map. This allows us to design descriptors that are optimized for the non-planar transformations and illumination changes that result from viewing a truly 3D scene. We note that Moreels and Perona have also proposed a technique for evaluating 3D feature matches based on trifocal constraints [28]. Our work extends this approach by giving us the ability to generate new correspondences at arbitrary locations and also to reason about visibility.

To generate correspondences, we leverage recent improvements in multi-view stereo matching [29], [30]. In contrast to previous approaches [31], this allows us to generate correspondences for arbitrary interest points and to model true interest point noise. We explore two methodologies for feature learning. The first uses parametric models inspired by previous successful feature designs, and Powell minimization [32] to solve for the parameters. The second uses non-parametric dimensionality reduction techniques common in the face recognition literature. Our training and test datasets containing approximately 2.5×10^6 labelled image patches are being made available online at http://www.cs.ubc.ca/∼mbrown/patchdata/patchdata.html.

A. Contributions

The main contributions of this work are as follows:

Matthew Brown is with the Computer Vision Laboratory, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland. Email: matthew.brown@epfl.ch. Gang Hua is with Nokia Research Center Hollywood, 2400 Broadway, D-500, Santa Monica, CA 90404. Email: ganghua@gmail.com. Simon Winder is with the Interactive Visual Media group at Microsoft Research, One Microsoft way, Redmond, WA 98052. Email: swinder@microsoft.com

1) We present a new ground-truth dataset for descriptor learning, making use of multi-view stereo from large 3D reconstructions. This allows us to optimize descriptors for real interest point detections. We will be making this dataset available to the community.
2) We extend previous work in parametric and non-parametric descriptor learning, and provide recommendations for future designs.
3) We conduct several new experiments, including reducing dynamic range to minimize the number of bits used by our feature descriptors (important for scalability) and optimizing descriptors for different types of interest point (e.g., Harris and DOG).

II. Ground Truth Dataset

To generate ground truth data for our descriptor matching problems, we make use of recent advances in multi-view image recognition and correspondence. Recent improvements in wide-baseline matching and structure from motion have made it possible to find matches and compute cameras for datasets containing thousands of images, with greatly varying pose and illumination conditions [33], [34]. Furthermore, advances in multi-view stereo have made it possible to reconstruct dense surface models for such images despite the greatly varying imaging conditions [29], [30].

We view these 3D reconstructions as a possible source of training data for object recognition problems. Previous work [31] used re-projections of 3D point clouds to establish correspondences between images, adding synthetic jitter to emulate the noise introduced in the interest point detection process. This approach, whilst being straightforward to implement, has the disadvantage of allowing training data to be collected only at discrete locations, and fails to model true interest point noise.

In this work, we use dense surface models obtained via stereo matching to establish correspondences between images. Note that because of the epipolar and multi-view constraints, stereo matching is a much easier problem than unconstrained 2D feature matching. We can thus generate correspondences via local stereo matching and multi-view consistency constraints that will be very challenging for wide baseline feature matching methods to match. We can also learn descriptors that are optimized for actual (and arbitrary) interest point detections, finding corresponding points by transferring their positions via the depth maps.

We make use of camera calibration information and dense multi-view stereo data for three datasets containing over 1000 images provided by [34] and [30]. In a similar spirit to [31], we extract patches around each interest point and store them in a large dataset on disk for efficient processing and learning. We detect Difference of Gaussian (DOG) interest points with associated position, scale and orientation in the manner of [19] (we also experiment with multi-scale Harris corners in Section VI-E). This results in around 1000 interest points per image.

For each interest point detected, we compute the position, scale and orientation of the local region when mapped into each neighbouring image. These parameters are solved for by a least-squares procedure. We do this by creating a uniform, dense point sampling (once per pixel) within the feature footprint in the first image. These points are then transferred via the depth map into the second image. In general the sampled points will not undergo an exact similarity transform, due to depth variations and perspective effects, so we estimate the best translation, rotation and scale between the corresponding image regions by least squares.

First, we check to see if the interest point is visible in the neighbouring image using the visibility maps supplied by [30] (a visibility map is defined over each neighbouring image, and each pixel has the label 1 if the corresponding point in the reference image is visible, and 0 otherwise). We then declare interest points that are detected within 5 pixels of position, 0.25 octaves of scale and $\pi/8$ radians in angle to be "matches". Those falling outside $2\times$ these ranges are defined to be "non-matches". Interest point detections that are in between these ranges are deemed to be ambiguous and not used in learning or testing. We chose fairly small ranges for position, orientation and scale tolerance to suit our intended applications in automatic stitching and structure from motion. However, for category recognition problems one might choose larger ranges that should result in more position invariance but less discriminative representations. See Figures 1 and 2 for examples of correspondences and image patches generated by this process.

III. Descriptor Algorithm

In previous work [31] we have noted that many existing descriptors described in the literature, while appearing quite different, can be constructed using a common modular framework consisting of processing stages similar to Figure 3. At each stage, different candidate block algorithms (described below) may be swapped in and out to produce a new overall descriptor. In addition, some candidates have free parameters that we can adjust in order to maximize the performance of the descriptor as a whole. Certain of these algorithmic combinations give rise to published descriptors but many are untested. Using this structure allows us to examine the contribution of each building block in detail and obtain a better covering of the space of possible algorithms.

Our approach to learning descriptors is therefore to put together a combination of building blocks and then optimize the parameters of these blocks using learning to obtain the best match/no-match classification performance. This contrasts with prior attempts to hand tune descriptor parameters and helps to put each algorithm on the same footing so that we can obtain and compare best performances.

Figure 3 shows the overall learning framework for building robust local image descriptors. The input is a set of image patches, which may be extracted from the neighbourhood of any interest point detector. The processing stages consist of the following:

G-block Gaussian smoothing is applied to the input patch.

T-blocks We perform a range of non-linear transformations to the smoothed patch. These include operations such as angle-quantized gradients and rectified steerable filters, and typically resemble the "simple-cell" stage in human visual processing.

S-blocks/E-blocks We perform spatial pooling of the above filter responses. S-blocks use parametrized pooling regions, E-blocks are non-parametric. This stage resembles the "complex-cell" operations in visual processing.

N-blocks We normalize the output patch to account for photometric variations. This stage may optionally be followed by another E-block, to reduce the number of dimensions at the output.

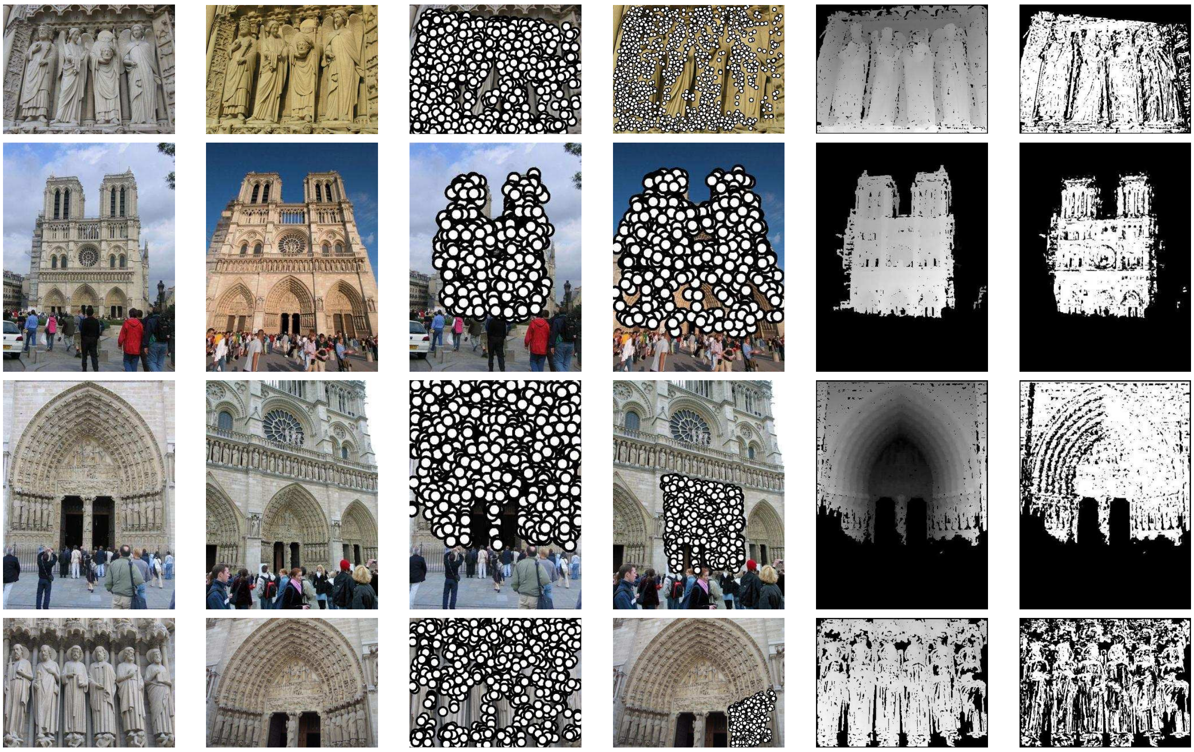


Fig. 1. Generating ground truth correspondences. To generate the ground truth image correspondences needed as input to our algorithms, we use multi-view stereo data provided by Goesele et al [30]. Interest points are detected in the reference image, and transferred to each neighbouring image via the depth map. If the projected point is visible, we look for interest points within a specified range of position, orientation and scale, and declare these to be matches. Points lying outside of twice this range are declared to be non-matches. This is the basic input to our learning algorithms. Left to right: reference image, neighbour image, reference matches, neighbour matches, depth map, visibility map.

In general, the T-block stage extracts useful features from the data like edge or local frequency information, and the S-block stage pools these features locally to make the representation insensitive to positional shift. These stages are similar to the simple/complex cells in the human visual cortex[36]. It's important that the T-block stage introduces some non-linearity, otherwise the smoothing step amounts to simply blurring the image. Also, the N-block normalization is critical as many factors such as lighting, reflectance and camera response have a large effect on the actual pixel values.

These processing stages have been combined into 3 different pipelines, as shown in the figure. Each stage has trainable parameters, which are learnt using our ground truth dataset of match/non-match pairs. In the remainder of this section, we will take a more detailed look at the parametrization of each of these building blocks.

A. Pre-smoothing (G-block)

We smooth the image pixels using a Gaussian kernel of standard deviation σ_s as a pre-processing stage to allow the descriptor to adapt to an appropriate scale relative to the interest point scale. This stage is optional and can be included in the T-block processing (below) if desired.

B. Transformation (T-block)

The transformation block maps the smoothed input patch onto a grid with one length k vector with positive elements per output sample. In this paper, the output grid was given the same resolution as the input patch, i.e., 64×64. Various forms of linear or non-linear transformations or classifiers are possible and have been described previously [31]. In this paper we restrict our choice to the following T-blocks which were found to perform well:

[T1] We evaluate the gradient vector at each sample and recover its magnitude m and orientation θ. We then quantize the orientation to k directions and construct a vector of length k such that m is linearly allocated to the two circularly adjacent vector elements i and $i+1$ representing $\theta_i < \theta < \theta_{i+1}$ according to the proximity to these quantization centres. All other elements are zero. This process is equivalent to the orientation binning used in SIFT and GLOH[11]. For the T1a-variant we use $k = 4$ directions and for the T1b-variant we use $k = 8$ directions.

[T2] We evaluate the gradient vector at each sample and rectify its x and y components to produce a vector of length 4 for the T2a-variant: $\{|\nabla_x| - \nabla_x; |\nabla_x| + \nabla_x; |\nabla_y| - \nabla_y; |\nabla_y| + \nabla_y\}$. This provides a natural sine-weighted quantization of orientation into 4 directions. Alternatively for T2b, we extend this to 8 directions by concatenating an additional length 4 vector using ∇_{45} which is the gradient vector rotated through $45°$.

[T3] We apply steerable filters at each sample location using n orientations and compute the responses from quadrature pairs [37] with rectification to give a length $k = 4n$ vector in a similar way to the gradient computation described above so that the positive and negative parts of the quadrature filter responses are placed in different vector elements. We tried two kinds of steerable filters: those based on a second derivatives provide broader scale and orientation tuning while fourth order filters give narrow scale and orientation tuning that can discriminate multiple orientations at each location in the input patch. These filters were implemented using the example coefficients given in [37]. The variants were

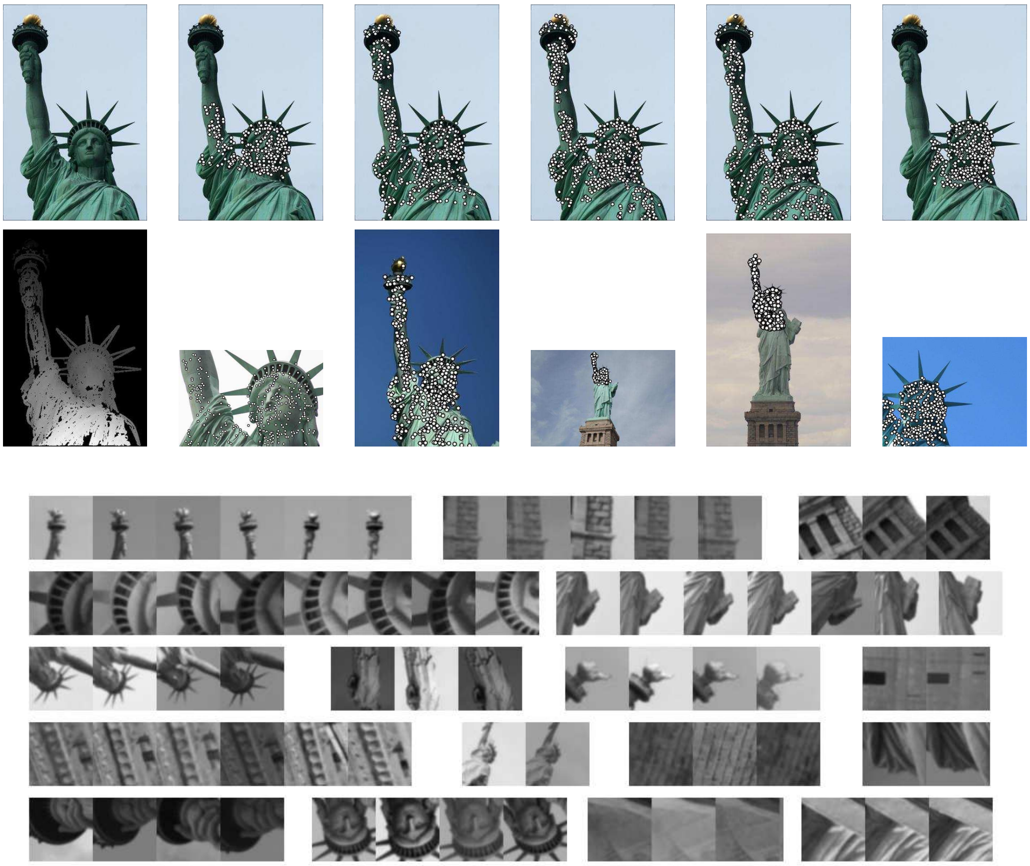


Fig. 2. Patch correspondences from the Liberty dataset. Top rows: reference image and depth map (left column), generated point correspondences (other columns). Note the wide variation in viewpoints and scales. Bottom rows: patches extracted from this dataset. Patches are considered to be “matching” if the detected interest points are within 5 pixels in position, 0.25 octaves of scale and $\pi/8$ radians in angle.

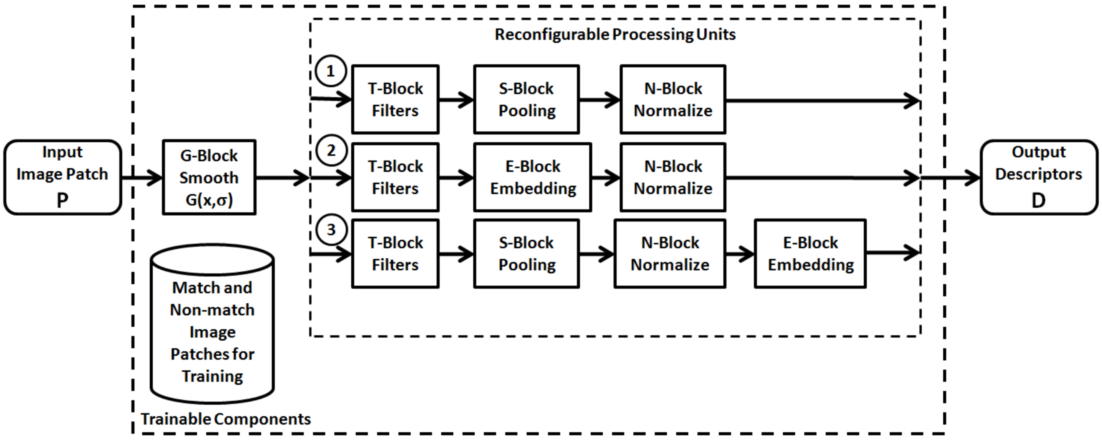
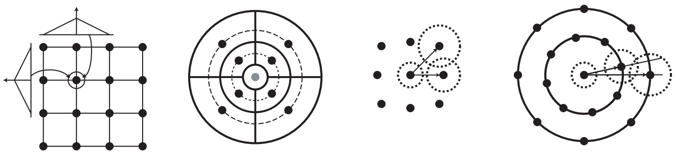


Fig. 3. Schematic showing the learning algorithms explored for building local image descriptors. Three overall pipelines have been explored: (1) uses parametric parameter optimization, (‘S’ blocks) using Powell Minimization as in [31]; (2) uses optimal linear projections (‘E’ blocks), found via LDA as in [35]; and a third approach (3) combines a stage of (1) followed by the linear projection step in (2).

T3g: 2nd order, 4 orientations; T3h: 4th order 4 orientations; T3i: 2nd order, 8 orientations; and T3j: 4th order, 8 orientations.

[T4] We compute two isotropic Difference of Gaussians (DOG) responses with different centre scales at each location by convolving the already smoothed patch with three new Gaussians (one additional larger centre and two surrounds). The two linear DOG filter outputs are then used to generate a length 4 vector by rectifying their responses into positive and negative parts as described above for gradient vectors. We set the ratio between the centre and surround space constants to 1.4. The pre-smoothing stage sets the size of the first DOG centre and so we use one additional parameter to set the relative size of the second DOG centre.



S1: SIFT grid with bilinear weights

S2: GLOH polar grid with bilinear radial and angular weights

S3: 3x3 grid with Gaussian weights

S4: 17 polar samples with Gaussian weights

Fig. 4. Examples of the different spatial summation blocks. For S3 and S4, the positions of the samples and the sizes of the Gaussian summation zones were parametrized in a symmetric manner.

C. Spatial Pooling (S-block)

Many descriptor algorithms incorporate some form of histogramming. In our pooling stage we spatially accumulate weighted vectors from the previous stage to give N linearly summed vectors of length k and these are concatenated to form a descriptor of kN dimensions where $N \in \{3, 9, 16, 17, 25\}$. We now describe the different spatial arrangements of pooling and the different forms of weighting:

[S1] We used a square grid of pooling centres (see Figure 4), with the overall footprint size of this grid being a parameter. The vectors from the previous stage were summed together spatially by bilinearly weighting them according to their distance from the pooling centres as in the SIFT descriptor [19] so that the width of the bilinear function is dictated by the output sample spacing. We use sub-pixel interpolation throughout as this allows continuous control over the size of the descriptor grid. Note that all these summation operations are performed independently for each of the k vector elements.

[S2] We used the spatial histogramming scheme of the GLOH descriptor introduced by Mikolajczyk and Schmid [11]. This uses a polar arrangement of summing regions as shown in Figure 4. We used three variants of this arrangement with 3, 9 and 17 regions, depending on the number of angular segments in the outer two rings (zero, 4, or 8). The radii of the centres of the middle and outer regions and the outer edge of the outer region were parameters that were available for learning. Input vectors are bilinearly weighted in polar coordinates so that each vector contributes to multiple regions. As a last step, each of the final vectors from the N pooling regions is normalized by the area of its summation region.

[S3] We used normalized Gaussian weighting functions to sum input vectors over local pooling regions arranged on a 3×3, 4×4 or 5×5 grid. The sizes of each Gaussian and the positions of the grid samples were parameters that could be learned. Figure 4 displays the symmetric 3×3 arrangement with two position parameters and three Gaussian widths.

[S4] We tried the same approach as S3 but instead used a polar arrangement of Gaussian pooling regions with 17 or 25 sample centres. Parameters were used to specify the ring radii and the size of the Gaussian kernel associated with all samples in each ring (Figure 4). The rotational phase angle of the spatial positioning of middle ring samples was also a parameter that could be learned. This configuration was introduced in [31] and named the DAISY descriptor by [38].

D. Embedding (E-block)

Embedding methods are prevalent in the face recognition literature [24], [25], and have been used by some authors for building local image descriptors [23], [35], [39]. Discriminative linear embedding can identify more robust image descriptors, whilst simultaneously reducing the number of dimensions. We summarize the different embedding methods we have used for E-blocks below (see also the objective functions in Section V).

[E1] We perform principal component analysis (PCA) on the input vectors. This is a non-discriminative technique and is used mostly for comparison purposes.

[E2] We find projections that minimize the ratio of in-class variance for match pairs to the variance of all match pairs. This is similar to Locality Preserving Projections (LPP) [25].

[E4] We find projections that minimize the ratio of variance between matched and non-matched pairs. This is similar to Local Discriminative Embedding [26].

[E6] We find projections that minimize the ratio of in-class variance for match pairs to the total data variance. We call this generalized local discriminative embedding (GLDE). If the number of classes is large, this objective function will be similar to [E2] and [E4] [35].

[E3], [E5] and **[E7]** are the same as [E2], [E4] and [E6] with the addition of orthogonality constraints which ensure that each of the projection directions are mutually orthogonal [40], [27], [41].

E. Post Normalization (N-block)

We use normalization to remove the descriptor dependency on image contrast and to introduce robustness.

For parametric descriptors, we employ the SIFT style normalization approach which involves range clipping descriptor elements. Our slightly modified algorithm consists of four steps: (1) Normalize to a unit vector, (2) clip all the elements of the vector that are above a threshold κ by computing $v'_i = \min(v_i, \kappa)$, (3) re-normalize to a unit vector, and (4) repeat from step 2 until convergence or a maximum number of iterations has been reached. This procedure has the effect of reducing the dynamic range of the descriptor and creating a robust function for matching. The threshold κ was available for learning.

In the case of the non-parametric descriptors of Figure 3(2), we normalize the descriptor to a unit vector.

IV. Learning Parametric Descriptors

This section corresponds to Pipeline 1 in figure 3. The input to the modular descriptor is a 64×64 image patch and the final output is a descriptor vector of $D = kN$ numbers where k is the

T-block dimension and N is the number of S-block summation regions.

We evaluate descriptor performance and carry out learning using our ground-truth data sets consisting of match and non-match pairs. For each pair we compute the Euclidean distance between descriptor vectors and form two histograms of this value for all true matching and non-matching cases in the data set. A good descriptor minimizes the amount of overlap of these histograms. We integrate the two histograms to obtain an ROC curve which plots correctly detected matches as a fraction of all true matches against incorrectly detected matches as a fraction of all true non-matches. We compute the area under the ROC curve as a final score for descriptor performance and aim to maximize this value. Other choices for quality measures are possible depending on the application but we choose ROC area as a robust and fairly generic measure. In terms of reporting our results on the test set, however, we choose to indicate performance in terms of the percentage of false matches present when 95% of all correct matches are detected.

We jointly optimized parameter values of G, T, S, and N-blocks by using Powell's multidimensional direction set method [32] to maximize the ROC area. We initialized the optimization with reasonable choices of parameters.

Each ROC area measure was evaluated using one run over the training data set. After each run we updated the parameters and repeated the evaluation until the change in ROC area was small. In order to avoid over-fitting we used a careful parametrization of the descriptors using as few parameters as possible (typically 5–11 depending on descriptor type). Once we had determined optimal parameters, we re-ran the evaluation over our testing data set to obtain the final ROC curves and error rates.

V. Learning Non-Parametric Descriptors

This section corresponds to Pipeline 2 in figure 3. In this section, we attempt to learn the spatial pooling component of the descriptor pipeline without committing to any particular parametrization. To do this, we make use of linear embedding techniques as described in Section III-D. Instead of using numerical gradient descent methods such as Powell minimization to optimize parametrized descriptors, the embedding methods solve directly for a set of optimal linear projections. The projected output vector in this embedding space becomes the final image descriptor. Although Pipeline 2 also involves parameters for T and N-blocks, these are learned independently using Powell Minimization as described above. We leave the joint optimization of these parameters for future work.

The input to the embedding learning algorithms is a set of match/non-match labelled image pairs that have been processed by different processing units (T-blocks), i.e.,

$$\mathcal{S} = \{\mathbf{x}_i = \mathcal{T}(\mathbf{p}_i), \mathbf{x}_j = \mathcal{T}(\mathbf{p}_j), l_{ij}\}. \tag{1}$$

In Equation 1, $\mathbf{p}_k$ is an input image patch, $\mathcal{T}(\cdot)$ represents a composite set of different image processing units presented in Section III, $\mathbf{x}_k$ is the output vector of $\mathcal{T}(\cdot)$, and l_{ij} takes binary value to indicate if patch $\mathbf{p}_i$ and $\mathbf{p}_j$ are match ($l_{ij} = 1$) or non-match ($l_{ij} = 0$). We now present the mathematical formulation of the different embedding learning algorithms.

A. *Objective functions of different embedding methods.*

Our E2 block attempts to maximize the ratio of the projected variance of all $\mathbf{x}_i$ in the match patch pair set to that of the difference vectors $\mathbf{x}_i - \mathbf{x}_j$. Letting $\mathbf{w}$ be the projection vector, we can write this mathematically as follows:

$$J_1(\mathbf{w}) = \frac{\sum_{l_{ij}=1} \left(\mathbf{w}^T \mathbf{x}_i\right)^2}{\sum_{l_{ij}=1} \left(\mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j)\right)^2}. \tag{2}$$

The intuition for this objective function is that in projection space, we try to minimize the distance between the match pairs while at the same time keeping the overall projected variance of all vectors in the match pair set as big as possible. This is similar to the Laplacian eigen-map adopted in previous works such as the locality preserving projections [25].

Alternatively, motivated by local discriminative embedding [26], the E4 block optimizes the following objective function:

$$J_2(\mathbf{w}) = \frac{\sum_{l_{ij}=0} \left(\mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j)\right)^2}{\sum_{l_{ij}=1} \left(\mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j)\right)^2}. \tag{3}$$

By maximizing $J_2(\mathbf{w})$, we are seeking the embedding space under which the distances between match pairs are minimized and the distances between non-match pairs are maximized.

A third objective function (E6 blocks) unifies the above two objective functions under certain conditions [35]:

$$J_3(\mathbf{w}) = \frac{\sum_{\mathbf{x}_i \in \mathcal{S}} \left(\mathbf{w}^T \mathbf{x}_i\right)^2}{\sum_{l_{ij}=1} \left(\mathbf{w}^T(\mathbf{x}_i - \mathbf{x}_j)\right)^2}. \tag{4}$$

All three objective functions J_1, J_2, and J_3 can be written in matrix form as

$$J_i(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{A}_i \mathbf{w}}{\mathbf{w}^T \mathbf{B} \mathbf{w}}. \tag{5}$$

where

$$\begin{aligned}
\mathbf{A}_1 &= \sum_{\mathcal{S}} (\sum_j l_{ij}) \mathbf{x}_i \mathbf{x}_i^T & (6)\\
\mathbf{A}_2 &= \sum_{l_{ij}=0} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T & (7)\\
\mathbf{A}_3 &= \sum_{\mathbf{x}_i \in \mathcal{S}} \mathbf{x}_i \mathbf{x}_i^T & (8)\\
\mathbf{B} &= \sum_{l_{ij}=1} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T. & (9)
\end{aligned}$$

In the following, for ease of presentation, we use $\mathbf{A}$ to represent any of $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$. Setting the derivative of our objective function (Equation 5) to zero gives

$$\frac{\partial J}{\partial \mathbf{w}} = \frac{2\mathbf{A}\mathbf{w}(\mathbf{w}^T\mathbf{B}\mathbf{w}) - 2(\mathbf{w}^T\mathbf{A}\mathbf{w})\mathbf{B}\mathbf{w}}{(\mathbf{w}^T\mathbf{B}\mathbf{w})^2} = 0 \tag{10}$$

which implies that the optimal $\mathbf{w}$ is given by the solution to a generalized eigenvalue problem

$$\mathbf{A}\mathbf{w} = \lambda \mathbf{B}\mathbf{w} \tag{11}$$

where $\lambda = \mathbf{w}^T\mathbf{A}\mathbf{w}/\mathbf{w}^T\mathbf{B}\mathbf{w}$. Equation 11 is solved using standard techniques, and the first K generalized eigenvectors are chosen to form the embedding space.

E3, E5 and E7 blocks place orthogonality constraints on the corresponding E2, E4 and E6 blocks, respectively. The mathematical formulation is quite straightforward: Suppose we have already obtained $k-1$ orthogonal projections for the embedding, i.e.,

$$\mathbf{W}_k = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{k-1}], \tag{12}$$

to pursue the k^{th} vector, we solve the following optimization problem:

$$\arg\max_{\mathbf{w}} \quad \frac{\mathbf{w}^T\mathbf{A}\mathbf{w}}{\mathbf{w}^T\mathbf{B}\mathbf{w}} \tag{13}$$

$$s.t. \quad \mathbf{w}^T\mathbf{w}_1 = 0 \tag{14}$$

$$\mathbf{w}^T\mathbf{w}_2 = 0 \tag{15}$$

$$\ldots \tag{16}$$

$$\mathbf{w}^T\mathbf{w}_{k-1} = 0. \tag{17}$$

By formulating the Lagrangian, it can be shown that the solution to this problem can be found by solving the following eigenvalue problem [27], [41]:

$$\hat{\mathbf{M}}\mathbf{w} = ((\mathbf{I} - \mathbf{B}^{-1}\mathbf{W}_k\mathbf{Q}_k^{-1}\mathbf{W}_k^T)\mathbf{B}^{-1}\mathbf{A})\mathbf{w} = \lambda\mathbf{w}, \tag{18}$$

where

$$\mathbf{Q}_k = \mathbf{W}_k^T\mathbf{B}^{-1}\mathbf{W}_k. \tag{19}$$

The optimal $\mathbf{w}_k$ is then the eigenvector associated with the largest eigenvalue in Equation 18. We omit the details of the derivation of the solution here but refer readers to [27], [41].

B. Power regularization

A common problem with the linear discriminative formulation in Equation 5 is the issue of over-fitting. This occurs because projections $\mathbf{w}$ which are essentially noise can appear discriminative in the absence of sufficient data. This issue is exacerbated by the high dimensional input vectors used in our experiments (typically several hundred to several thousands of dimensions). To mitigate the problem, we adopt a power regularization cost function to force the discriminative projections to lie in the signal subspace. To do this, we first perform eigenvalue decomposition for the $\mathbf{B}$ matrix in Equation 5, i.e., $\mathbf{B} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$. Here $\mathbf{\Lambda}$ is a diagonal matrix with $\Lambda_{ii} = \lambda_i$ being the i^{th} eigenvalue of $\mathbf{B}$ and $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$. We then regularize $\mathbf{\Lambda}$ by clipping its diagonal elements against a minimal value λ_r, where

$$\lambda_i' = \max(\lambda_i, \lambda_r). \tag{20}$$

We choose r such that $\sum_{i \geq r} \lambda_i$ accounts for a portion α of the total power, i.e.,

$$r = \min_k \;\; s.t. \; \frac{\sum_{i=k}^{n} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \leq \alpha. \tag{21}$$

Figure 5 shows the top 10 projections learnt from a set of match/non-match image patches with different power regularization rate α. The only pre-processing applied to these patches was bias-gain normalization. As we can clearly observe, as α decreases from 0.2 to 0 (top to bottom), the projections become increasingly noisy.

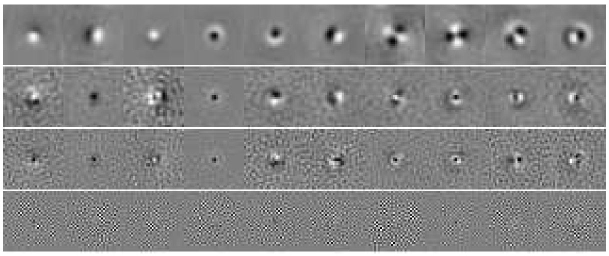


Fig. 5. The first 10 projections learned from normalized image patches in a match/non-match image patch set using $J_2(\mathbf{w})$ with different power regularization rate [35]. From top to bottom, α takes the value of 0.2, 0.1, 0.02 and 0, respectively. Notice that the projections become progressively noisier as the power regularization is reduced.

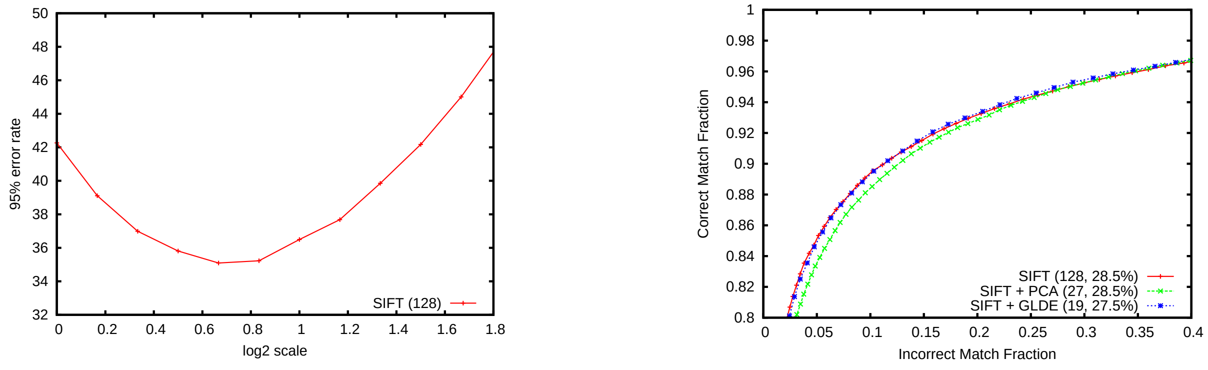
VI. Experiments

We performed experiments using the parametric and non-parametric descriptor formulations described above, using our new test dataset. The following results all apply to Difference of Gaussian (DOG) interest points. For experiments using Harris corners, see Section VI-E. In each case we have compared to Lowe's original implementation of SIFT. Since SIFT performs descriptor sampling at a certain scale relative to the Difference of Gaussian peak, we have optimized over this scaling parameter to ensure that a fair comparison is made (see Figure 6).

For the results presented in this paper, we used three test sets (Yosemite, Notre Dame, and Liberty) which were obtained by extracting scale and orientation normalized 64×64 patches around DOG interest points as described in Section II. Typically four training and test set combinations were used: Yosemite–Notre Dame, Yosemite–Liberty, Notre Dame–Yosemite, and Notre Dame–Liberty, where the first of the pair is the training set. In addition a "synthetic" training set was obtained which incorporated artificial geometric jitter as described in [31]. Training sets typically contained from 10,000 to 500,000 patch pairs depending on the application while test sets always contained 100,000 pairs. The training and test sets contained 50% match pairs, and 50% non-match pairs. During training and testing, we recomputed all match/non-match descriptor distances as the descriptor transformation varied, sweeping a threshold on the descriptor distance to generate an ROC curve. Note that using predefined match/non-match pairs eliminates the need to recompute nearest neighbours in the 100,000 element test set, which would be computationally very demanding. In addition to presenting ROC curves, we give many results in terms of the 95% error rate which is the percent of incorrect matches obtained when 95% of the true matches are found (Section IV).

A. Parametric Descriptors

We obtained very good results using combinations of the parametric descriptor blocks of Section III, exceeding the performance of SIFT by around 1/3 in terms of 95% error rates. We chose to focus specifically on four combinations that were shown to have merit in [31]. These included a combination of angle quantized gradients (T1) or steerable filters (T3) with log-polar (S2) or Gaussian (S4) summation regions. Other combinations with T2, T4, S1, S3 performed less well. Example ROC curves are shown in Figure 7 and 8, and all error rates are given in Table I (all tables show the 95% error rate with the optimal number of dimensions given in parentheses).



(a)

(b)

Fig. 6. Results for Lowe-SIFT descriptors: (a) shows the solution for the optimal SIFT descriptor footprint using the Liberty dataset. Note that the performance is quite sensitive to this parameter, so it must be set carefully. (b) shows ROC curves when using this optimal patch scaling and the Yosemite dataset for testing. We also tried using PCA and GLDE on the SIFT descriptors (shown in the other curves). GLDE gave only small improvement in performance (1% error at 95% true positives) to Lowe's algorithm, but substantially reduced the number of dimensions from 128 to 19. PCA also gives a large dimensionality reduction for only a small drop in performance.

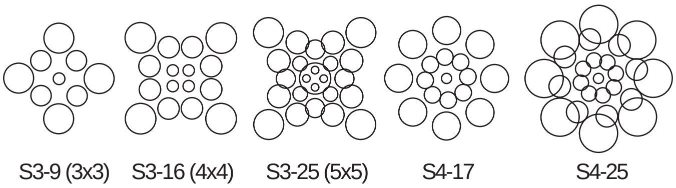


Fig. 9. Optimal summation regions are foveated and this is despite initialization with a rectangular arrangement in the case of S3.

On three of the four datasets, the best performance was achieved by the T3h-S4-25 combination, which is a combination of steerable filters with 25 Gaussian summation regions arranged in concentric rings. We found that when optimized over our training dataset, these summation regions tended to converge to a foveated shape, with larger and more widely space summation regions further from the centre (see Figure 9). This structure is reminiscent of the geometric blur work of [22], and similar arrangements were independently suggested and named DAISY descriptors by [38]. Rectangular arrays of summation regions were found to have lower performance and their results are not included here.

Note that the performance of these parametric descriptors is uniformly strong in comparison to SIFT, but the downside of this method is that the number of dimensions is very large (typically several hundred).

B. Non-Parametric Descriptors

The ROC curves for training on Yosemite and testing on Notre Dame using Non-Parametric descriptors are shown in Figure 10. To summarize the remaining results, we have created tables showing the 95% error rates only.

Table II shows the best results for each T-block using the scheme of Figure 3(2) over all subspace methods that we tried (PCA, LDE, LPP, GLDE and orthogonal variants). Also shown are results for applying subspace methods to raw bias-gain normalized pixel patches and gain normalized gradients. We see that the T3 (steerable filter) block performs the best, followed by T1 (angle-quantized gradients) and T2 (rectified gradients). In half of the cases the combination of T3 and E-block learning beat SIFT. Table III shows the best results for each E-block over all T-block filters. LPP is the clear winner when trained on Yosemite. For Notre Dame the case is not so clear, and no one method performs consistently well. The best results for each subspace method are almost always using T3.

To investigate sensitivity to training data, we tested on the Liberty set using training on both Notre Dame and Yosemite. For the non-parametric descriptor learning it seems that the Yosemite dataset was best for training, whereas for the parametric descriptors the performance was comparable (within 1-2%) for both datasets. In general the results from the E-block learning are less strong and more variable than the parametric S-block techniques. Certain combinations, such as T3/LPP were able to generate SIFT beating performance (e.g. 19.29% vs 26.10% on the Yosemite/Notre Dame test case), but many other combinations did not. The principal advantage of these techniques is that dimensionality reduction is simultaneously achieved, so the number of dimensions is typically low (e.g. 32 dimensions in the case of T3/LPP).

C. Dimension reduced parametric descriptors

Parametric descriptor learning yielded excellent performance with high dimensionality, whereas the non-parametric learning gave us a very small number of dimensions but with a slightly inferior performance. Thus it seems natural to combine these approaches. We did this by running a stage of non-parametric dimensionality reduction after a stage of parametric learning. This corresponds to Pipeline 3 in Figure 3. Note that we did not attempt to jointly optimize for the embedding and parametric descriptors, although this could be a good direction for future work. The results are shown in Figure 11 and Table IV. This approach gave us the overall best results, with typically 1-2% less error than parametric S-blocks alone, and far fewer dimensions (∼30-40). Although LDA gave much better results than PCA when applied to raw pixel data [35], running PCA on the outputs of S-block learning gave equal or better results to LDA. It may be that LDA is slightly overfitting in cases where a discriminative representation has already been found. For half the datasets, the best results were

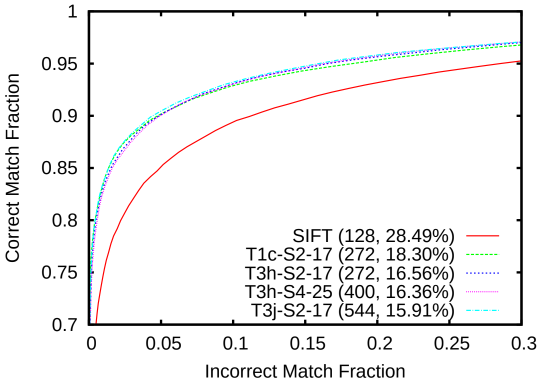


Fig. 7. ROC curves for parametrized descriptors. Training on Notre Dame and testing on Yosemite.

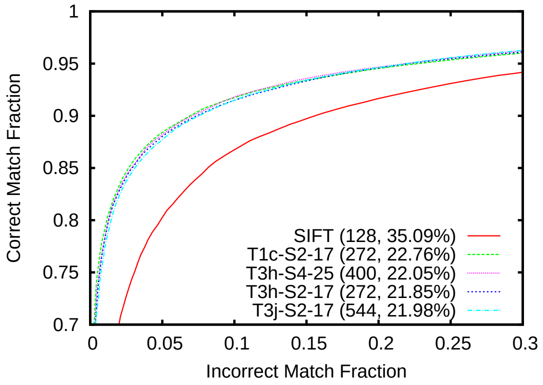


Fig. 8. ROC curves for parametrized descriptors. Training on Notre Dame and testing on Liberty.

Train	Test	T1c-S2-17	T3h-S4-25	T3h-S2-17	T3j-S2-17	SIFT
Yosemite	Notre Dame	$17.90_{(272)}$	$\mathbf{14.43}_{(400)}$	$15.44_{(272)}$	$15.87_{(544)}$	$26.10_{(128)}$
Yosemite	Liberty	$23.00_{(272)}$	$\mathbf{20.48}_{(400)}$	$22.00_{(272)}$	$22.28_{(544)}$	$35.09_{(128)}$
Notre Dame	Yosemite	$18.30_{(272)}$	$16.35_{(400)}$	$16.56_{(272)}$	$\mathbf{15.91}_{(544)}$	$28.50_{(128)}$
Notre Dame	Liberty	$22.76_{(272)}$	$\mathbf{21.85}_{(400)}$	$22.05_{(272)}$	$21.98_{(544)}$	$35.09_{(128)}$
Synthetic	Liberty	$29.50_{(272)}$	$\mathbf{24.25}_{(400)}$	$25.74_{(272)}$	$32.36_{(544)}$	$35.09_{(128)}$

TABLE I

Parametric descriptor results. 95% error rates are shown, with the number of dimensions in parenthesis.

		normalized	normalized					
Training Set	Test Set	pixels	gradients	T1	T2	T3	T4	SIFT
Yosemite	Notre Dame	$37.17_{(14)}$	$32.09_{(15)}$	$25.68_{(24)}$	$27.78_{(33)}$	$\mathbf{19.29}_{(32)}$	$35.37_{(28)}$	$26.10_{(128)}$
Yosemite	Liberty	$56.33_{(14)}$	$51.63_{(15)}$	$38.55_{(24)}$	$41.10_{(20)}$	$\mathbf{31.10}_{(32)}$	$47.74_{(28)}$	$35.09_{(128)}$
Notre Dame	Yosemite	$43.37_{(27)}$	$38.36_{(19)}$	$33.59_{(21)}$	$33.99_{(40)}$	$\mathbf{31.27}_{(19)}$	$42.39_{(27)}$	$28.50_{(128)}$
Notre Dame	Liberty	$55.70_{(27)}$	$52.62_{(17)}$	$41.37_{(24)}$	$43.80_{(15)}$	$\mathbf{36.54}_{(19)}$	$50.63_{(27)}$	$35.09_{(128)}$
Synthetic	Notre Dame	$37.85_{(15)}$	$39.15_{(24)}$	$24.47_{(32)}$	$24.47_{(32)}$	$\mathbf{22.94}_{(30)}$	$34.41_{(28)}$	$26.10_{(128)}$

TABLE II

Best T-block results over all subspace methods.

Training	Test	PCA	GLDE	GOLDE	LDE	OLDE	LPP	OLPP	SIFT
Yosemite	Notre D.	$40.36_{(29)}$	$24.20_{(28)}$	$26.24_{(31)}$	$24.65_{(31)}$	$25.01_{(27)}$	$\mathbf{19.29}_{(32)}$	$23.71_{(31)}$	$26.10_{(128)}$
Yosemite	Liberty	$53.20_{(29)}$	$35.76_{(28)}$	$43.35_{(31)}$	$34.97_{(31)}$	$40.15_{(27)}$	$\mathbf{31.10}_{(32)}$	$39.46_{(31)}$	$35.09_{(128)}$
Notre D.	Yosemite	$45.43_{(61)}$	$32.53_{(45)}$	$34.61_{(25)}$	$\mathbf{31.27}_{(19)}$	$33.38_{(20)}$	$33.19_{(46)}$	$35.04_{(17)}$	$28.50_{(128)}$
Notre D.	Liberty	$51.63_{(97)}$	$41.66_{(45)}$	$40.75_{(18)}$	$\mathbf{36.54}_{(19)}$	$39.95_{(20)}$	$42.68_{(46)}$	$41.46_{(17)}$	$35.09_{(128)}$
Synthetic	Notre D.	$43.78_{(66)}$	$24.04_{(29)}$	$26.25_{(29)}$	$24.86_{(26)}$	$26.10_{(33)}$	$\mathbf{22.94}_{(30)}$	$26.05_{(34)}$	$26.10_{(128)}$

TABLE III

Best subspace method over all T-blocks.

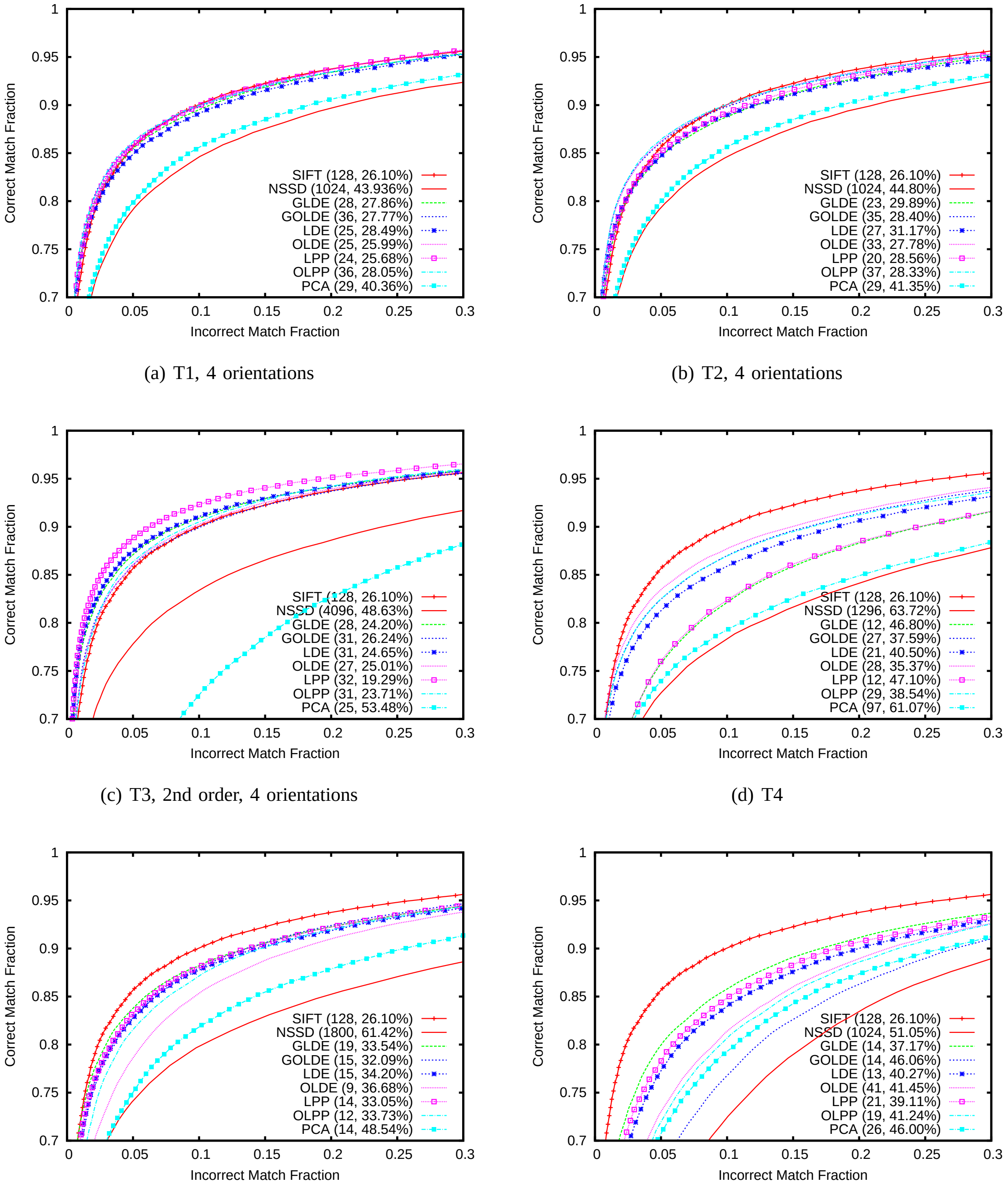
obtained using PCA on T3h-S4-25 (rectified steerable filters with DAISY-like Gaussian summation regions) and for the other half, the best results were from T3j-S2-17 plus PCA (rectified steerable filters and log-polar GLOH-like summation regions). The best results here gave less than half the error rate of SIFT, using about 1/4 of the number of dimensions. See “best of the best” table V.

To aid in the dissemination of these results, we have created a document detailing parameter settings for the most successful DAISY configurations, as well as details of the recognition performance/computation time tradeoffs. This can be found on the same website as our patch datasets: http://www.cs.ubc.ca/∼mbrown/patchdata/tutorial.pdf.

We also used this approach to perform dimensionality reduction on SIFT itself, the results are shown in Figure 6(b). We were able to reduce the number of dimensions significantly (to around 20), but the matching performance of the LDA reduced SIFT descriptors was only slightly better than the original SIFT descriptors (∼1% error).

D. Comparisons with Synthetic Interest Point Noise

Previous work [31], [12] used synthetic jitter applied to image patches in lieu of the position errors introduced in interest point



(a) T1, 4 orientations

(b) T2, 4 orientations

(c) T3, 2nd order, 4 orientations

(d) T4

(e) Normalized gradient

(f) Bias-gain normalized image

Fig. 10. Testing of linear discriminant descriptors trained on Yosemite and tested on Notre Dame. The optimal number of dimensions and the associated 95% error rate is given in parentheses. NSSD: Normalized sum squared difference computed on the output of the T-block directly without embedding.

Training	Test	PCA	GLDE	GOLDE	LDE	OLDE	LPP	OLPP	SIFT
Yosemite	Notre D.	$\mathbf{11.98}_{(29)}$	$19.12_{(39)}$	$13.64_{(49)}$	$18.03_{(60)}$	$12.48_{(71)}$	$16.77_{(52)}$	$14.07_{(36)}$	$26.10_{(128)}$
Yosemite	Liberty	$\mathbf{18.27}_{(29)}$	$26.92_{(32)}$	$19.88_{(49)}$	$25.20_{(60)}$	$18.70_{(71)}$	$25.39_{(32)}$	$20.33_{(36)}$	$35.09_{(128)}$
Notre D.	Yosemite	$\mathbf{13.55}_{(36)}$	$25.25_{(87)}$	$15.67_{(67)}$	$21.78_{(35)}$	$15.04_{(99)}$	$22.30_{(48)}$	$15.56_{(86)}$	$28.50_{(128)}$
Notre D.	Liberty	$\mathbf{16.85}_{(36)}$	$30.38_{(28)}$	$20.01_{(53)}$	$26.48_{(45)}$	$19.80_{(49)}$	$26.78_{(48)}$	$19.47_{(48)}$	$35.09_{(128)}$

TABLE IV

BEST SUBSPACE METHODS FOR COMPOSITE DESCRIPTORS.

Train	Test	Parametric	Non-parametric	Composite	SIFT
Yosemite	Notre Dame	$14.43_{(400)}$	$19.29_{(32)}$	$\mathbf{11.98}_{(29)}$	$26.10_{(128)}$
Yosemite	Liberty	$20.48_{(400)}$	$31.10_{(32)}$	$\mathbf{18.27}_{(29)}$	$35.09_{(128)}$
Notre Dame	Yosemite	$15.91_{(544})$	$31.27_{(19)}$	$\mathbf{13.55}_{(36)}$	$28.50_{(128)}$
Notre Dame	Liberty	$21.85_{(400)}$	$36.54_{(19)}$	$\mathbf{16.85}_{(36)}$	$35.09_{(128)}$

TABLE V

"BEST OF THE BEST" RESULTS.

detection. In order to evaluate the effectiveness of this strategy, we tested a number of descriptors that were trained on a dataset with synthetic noise applied ([31]).

For results, see the last rows of tables I, II and III. Here, "synthetic" means that synthetic scale, rotation and position jitter noise was applied to the patches, although the actual patch data was sampled from real images as in [31]. For the parametric descriptors, there is a clear gain of 5-10% from training using the new non-synthetic dataset. For the LDA based methods smaller gains are noticeable.

E. *Learning Descriptors for Harris Corners*

Using our multi-view stereo ground truth data we can easily create optimal descriptors for any choice of interest point. To demonstrate this, we also created a dataset of patches centred on multi-scale Harris corner points (see Figure 12). The left column shows the projections learnt from Harris corners and the right column from DOG interest points, for normalized image patches. The projections learnt from the two different types of interest points share several similarities in appearance. They are all centre focused, and look like Gaussian derivatives [16] combined with geometric blur [22]. We also found that the order of the performance of the descriptors learnt from the different embedding methods are similar to each other across the two datasets.

F. *Effects of Normalization*

As demonstrated in [35], the post-normalization step is very important for the performance of the non-parametric descriptors learnt from synthetically jittered data-set. We observe a similar phenomenon in our new experiments with the new data.

The higher performance of the parametric descriptors when compared to the non-parametric descriptors is in some part attributable to the use of SIFT-style clipping normalization versus simple unit-length normalization for these. Since parametric descriptors maintain a direct relation between image-space and descriptor coefficients compared with coefficients after PCA reduction, SIFT-style clipping, by introducing a robustness function, can mitigate differences due to spatial occlusions and shadowing which affect one part of the descriptor and not another. For this reason applying SIFT-style normalization prior to dimension reduction seems appropriate.

Figure 13 shows the effect of changing the threshold of clipping for SIFT normalization. Error rates are significantly improved when the clipping threshold are equal to around $1.6/\sqrt{D}$ when tested on a wide range of parametric descriptors with different dimensionality. This graph shows the drastic reduction in error rate compared with simple unit normalization.

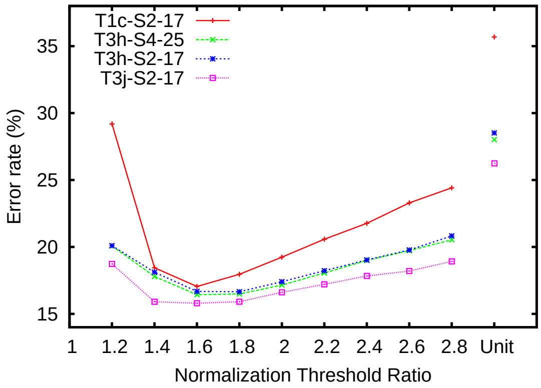


Fig. 13. Change in error rates as the normalization clipping threshold is varied for parametric descriptors. The threshold was set to $r/\sqrt{D}$ where r is the ratio and D is the descriptor dimensionality. Unit: unit length normalization without clipping.

G. *Minimizing Bits*

For certain applications, such as scalable recognition, it is important that descriptors are represented as efficiently as possible. A natural question is: "what is the minimum number of bits required for accurate feature descriptors?". To address this question we tested the recognition performance of our parametrized descriptors as the number of bits per dimension was reduced from 8 to 1. The results are shown in Figure 14 for the parametric descriptors. Surprisingly, there seems to be very little benefit to using any more than 2 or 3 bits of dynamic range per dimension, which suggests that it should be possible to create local image descriptors with a very small memory footprint indeed. In one case (T1c-S2-17), the performance actually degraded slightly as more bits were added. It could be that in this case quantization caused a small noise reduction effect. Note that this effect was small (1% in error rate), and not shown for the other descriptors, where the major change in performance came from 1 to 2 bits per dimension, which gave around 16% change in error rate. Whilst it would also be possible to quantize bits for dimension reduced (embedded) descriptors, a variable number of bits per dimension would be required as the variance on each dimension can differ substantially across the descriptor.

VII. LIMITATIONS

Here we address some limitations of the current method and suggest ideas for future work.

A. *Repetitive image structure*

One caveat with our learning approach scheme is that distinct 3D locations are *defined* to be different classes, when in the

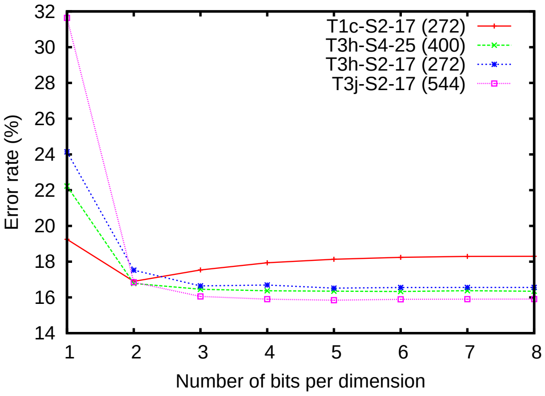


Fig. 14. Results of limiting the number of bits in each descriptor dimension. Not many more than 2 bits are required per dimension to retain a good error rate.

real world, they can often have the same visual appearance. One common example would be repeated architectural structures, such as windows or doors. Such repetitions typically cause false positives in our matching schemes (see Figure 15). For the Notre Dame dataset, false positives occur due to translational repetition (e.g. the stone figures) as well as rotational repetitions (e.g. the rose window).

B. Multi-view Stereo Data

Although there have been great improvements in stereo in recent years [30], using multi-view stereo to train local image descriptors has its limitations. Noise in the stereo reconstruction will inevitably propagate through to the set of image correspondences, but probably a bigger issue is that certain image correspondences, i.e., in regions where stereo fails, will not be present at all. One way around this problem would be to use imagery registered to LIDAR scans as in [42].

VIII. Conclusions

We have described a scheme for learning discriminative, low-dimensional image descriptors from realistic training data. These techniques have state-of-the-art performance in all our test scenarios. The techniques described in this paper have been used to design local feature descriptors for a robust structure from motion application called Photosynth[1] and an automatic panoramic stitcher named ICE[2] (Image Compositing Editor).

Recommendations

To summarize our work, we suggest a few recommendations for practitioners in this area:

- **Learn parameters from training data** Successful descriptor designs typically have many parameter choices that are difficult to optimize by hand. We recommend using realistic training datasets to optimize these parameters.
- **Use foveated summation regions** Pooling regions that become larger away from the interest point are generally found to have good performance. See [38] for an efficient implementation approach.
- **Use non-linear filter responses** Some form of non-linear filtering before spatial pooling is essential for the best performance. Steerable filters work well if the phase is kept. Rectified or angle-quantized gradients are also a good and simple choice.
- **Use LDA for discriminative dimension reductions** LDA can be used to find discriminative, low dimensional descriptors without imposing a choice of parameters. However, if a discriminative representation has already been found, PCA can work well for reducing the number of dimensions.
- **Normalization** Thresholding normalization often provides a large boost in performance. If dimension reduction is used, normalization should come before the dimension reduction block.

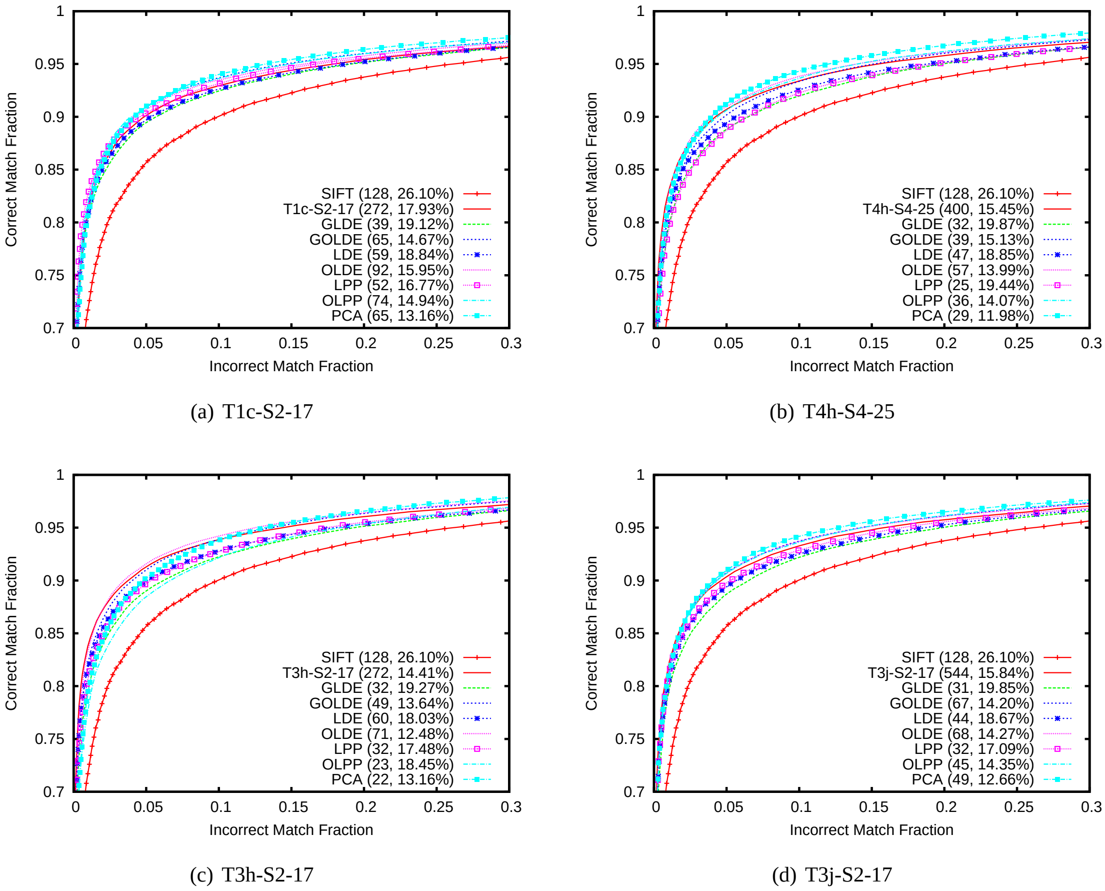
Acknowledgment

The authors would like to thank Michael Goesele and Noah Snavely for sharing their 3D reconstruction data with us. We'd also like to thank David Lowe, Rick Szeliski and Sumit Basu for helpful discussions.

References

[1] R. Szeliski, "Image alignment and stitching: A tutorial," Microsoft Research, Tech. Rep. MSR-TR-2004-92, December 2004.
[2] M. Brown and D. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 2007.
[3] M. Pollefeys, L. V. Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch, "Visual modeling with a hand-held camera," *International Journal of Computer Vision*, vol. 59, no. 3, pp. 207–232, 2004.
[4] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo tourism: Exploring photo collections in 3D," in *SIGGRAPH Conference Proceedings*. New York, NY, USA: ACM Press, 2006, pp. 835–846.
[5] D. Nistér and H. Stewénius, "Scalable recognition with a vocabulary tree," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, June 2006, pp. 2161–2168.
[6] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR07)*, 2007.
[7] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International Journal of Computer Vision*, vol. 73, no. 2, pp. 213–238, June 2007.
[8] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *Proceedings of the IEEE Interntaion Conference on Computer Vision*, Bejing, October 2005.
[9] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proceedings of the Interntaional Conference on Computer Vision and Pattern Recognition*, 2003.
[10] K. Mikolajczyk and C. Schmid, "Scale and affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 1, no. 60, pp. 63–86, 2004.
[11] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 27, pp. 1615–1630, 2005.
[12] V. Lepetit and P. Fua, "Keypoint recognition using randomized trees," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1465–1479, 2006.
[13] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Internaional Conference on Computer Vision and Pattern Recognition*, June 2008.
[14] B. Babenko, P. Dollar, and S. Belongie, "Task specific local region matching," in *International Conference on Computer Vision (ICCV07)*, Rio de Janeiro, 2007.
[15] J. M. D. Martin, C. Fowlkes, "Learning to detect natural image boundaries using local brightness, color and texture cues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 5, May 2004.

[1] http://www.photosynth.com
[2] http://research.microsoft.com/ivm/ice.html



(a) T1c-S2-17

(b) T4h-S4-25

(c) T3h-S2-17

(d) T3j-S2-17

Fig. 11. ROC curves for composite descriptors trained on Yosemite and testing on Notre Dame.

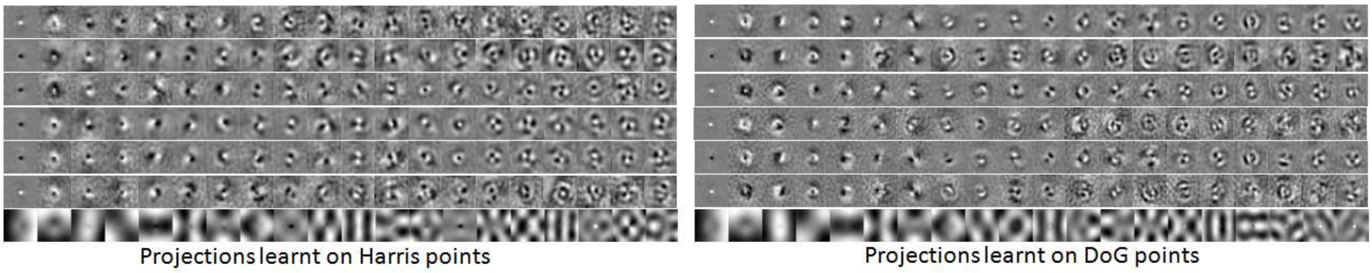

Projections learnt on Harris points
Projections learnt on DoG points

Fig. 12. Comparison of projections on patches centred on Harris corner points (left column), and DOG points (right column), respectively. From top to the bottom, we present projections learnt using the embedding blocks of E2, E3, E4, E5, E6, E7 and E1, respectively.

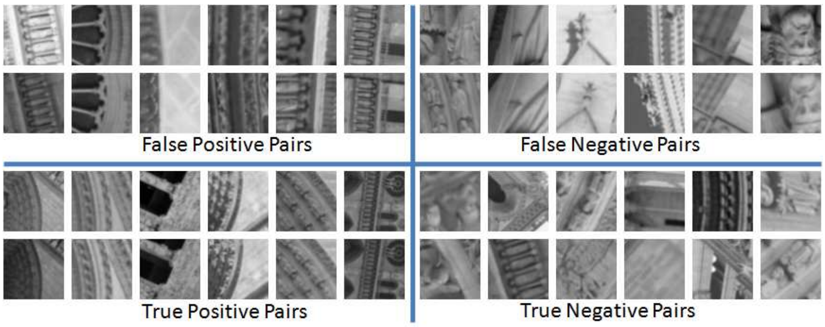


Fig. 15. Some of the false positive, false negative, true positive and true negative image patch pairs when testing on the new Notre Dame dataset using E-blocks learnt from the new Yosemite dataset. We used a combination of T3 (steerable filters) and E2 (LPP) in this experiment. Each row shows 6 pairs of image patches and the two image patches in each pair are shown in the same column. Note that the two images in the false positive pairs are indeed obtained from different 3D points but their appearances look surprisingly similar.

[16] C. Schmid and R. Mohr, “Local grayvalue invariants for image retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 530–535, May 1997.

[17] C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy, “Canonical frames for planar object recognition,” in *European Conference on Computer Vision*, 1992, pp. 757–772.

[18] D. Lowe, “Object recognition from local scale-invariant features,” in *International Conference on Computer Vision*, Corfu, Greece, September 1999, pp. 1150–1157.

[19] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[20] D. Hubel and T. Wiesel, “Brain mechanisms of vision,” *Scientific American*, pp. 150–162, September 1979.

[21] S. Belongie, J. Malik, and J. Puzicha, “Shape context: A new descriptor for shape matching and object recognition,” in *Advances in Neural Information Processing Systems*. MIT Press, Cambridge, MA, 2000.

[22] A. Berg and J. Malik, “Geometric blur and template matching,” in *International Conference on Computer Vision and Pattern Recognition*, 2001, pp. I:607–614.

[23] Y. Ke and R. Sukthankar, “PCA-SIFT: a more distinctive representation for local image descriptors,” in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, vol. 2, July 2004, pp. 506–513.

[24] P. Belhumeur, J. Hespanha, and D. Kriegman, “Eigenfaces VS Fisherfaces: Recognition using class specific linear projection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[25] X. He, S. Yan, Y. Hu, P. Niyogi, and H. Zhang, “Face recognition using Laplacianfaces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328–340, March 2005.

[26] H. Chen, H. Chang, and T. Liu, “Local discriminant embedding and its variants,” in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, vol. 2, San Diego, CA, June 2005, pp. 846–853.

[27] J. Duchene and S. Leclercq, “An optimal transformation for discriminant and principle component analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 978–983, 1988.

[28] P. Moreels and P. Perona, “Evaluation of feature detectors and descriptors based on 3D objects,” in *Proceedings of the International Conference on Computer Vision*, vol. 1, 2005, pp. 800–807.

[29] M. Goesele, S. Seitz, and B. Curless, “Multi-view stereo revisited,” in *International Conference on Computer Vision and Pattern Recognition*, New York, June 2006.

[30] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. Seitz, “Multi-view stereo for community photo collections,” in *International Conference on Computer Vision*, Rio de Janeiro, October 2007.

[31] S. Winder and M. Brown, “Learning local image descriptors,” in *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR07)*, Minneapolis, June 2007.

[32] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge University Press, 1992.

[33] M. Brown and D. Lowe, “Unsupervised 3D object recognition and reconstruction in unordered datasets,” in *5th International Conference on 3D Imaging and Modelling (3DIM05)*, 2005.

[34] N. Snavely, S. Seitz, and R. Szeliski, “Modeling the world from internet photo collections,” *International Journal of Computer Vision*, vol. 80, no. 2, pp. 189–210, 2008.

[35] G. Hua, M. Brown, and S. Winder, “Discriminant embedding for local image descriptors,” in *Proceedings of the 11th International Conference on Computer Vision (ICCV07)*, Rio de Janeiro, October 2007.

[36] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, “Object recognition with cortex-like mechanisms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 411–426, 2007.

[37] W. T. Freeman and E. H. Adelson, “The design and use of steerable filters,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 891–906, 1991.

[38] E. Tola, V. Lepetit, and P. Fua, “A fast local descriptor for dense matching,” in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Anchorage, June 2008.

[39] K. Mikolajczyk and J. Matas, “Improving descriptors for fast tree matching by optimal linear projection,” in *Proceedings of the International Conference on Computer Vision*, Rio de Janeiro, 2007.

[40] D. Cai, X. He, J. Han, and H.-J. Zhang, “Orthogonal Laplacianfaces for face recognition,” *IEEE Transaction on Image Processing*, vol. 15, no. 11, pp. 3608–3614, November 2006.

[41] G.Hua, P. Viola, and S. Druker, “Face recognition using discriminatively trained orthogonal rank one tensor projections,” in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, Minneapplois, MN, June 2007.

[42] 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,” in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Anchorage, June 2008.

[43] S. Winder, G. Hua, and M. Brown, “Picking the best daisy,” in *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR09)*, Miami, June 2009.

Matthew Brown is a Postdoctoral Fellow at the Ecole Polytechnique Fédérale de Lausanne. He obtained the M.Eng. degree in Electrical and Information Sciences from Cambridge University in 2000, and the Ph.D. degree in Computer Science from the University of British Columbia in 2005. His research interests include Computer Vision, Machine Learning, Medical Imaging and Environmental Informatics. He worked with Microsoft Research as an intern in Cambridge in 2002, and in Redmond in 2003-2004. He returned to Microsoft Research Redmond as a Postdoctoral Researcher during 2006-2007. His work there focused on image segmentation, panoramic stitching and local feature design. His work on panoramic stitching has been widely adopted, and appears on the curriculum of many University courses as well as in several commercial products. He is a director and CTO of Vancouver based Cloudburst Research Inc.

Gang Hua is a Senior Researcher at Nokia Research Center Hollywood. Before that, he was a Scientist at Microsoft Live Labs Research from 2006 to 2009. He received his Ph.D. degree in Electrical and Computer Engineering from Northwestern University in 2006, the M.S. and B.S. degree in Electrical Engineering from Xi’an Jiaotong University in 2002 and 1999, respectively. He was enrolled in the Special Class for the Gifted Young of XJTU in 1994. During the summer 2005 and summer 2004, he was a research intern with the Speech Technology Group, Microsoft Research, Redmond, WA, and a research intern with the Honda Research Institute, Mountain View, CA, respectively.

He received the Richter Fellowship and the Walter P. Murphy Fellowship at Northwestern University in 2005 and 2002, respectively. When he was in XJTU, he was awarded the Guanghua Fellowship, the Eastcom Fellowship, the Most Outstanding Student Exemplar Fellowship, the Sea-star Fellowship and the Jiangyue Fellowship in 2001, 2000, 1997, 1997 and 1995 respectively. He was also a recipient of the University Fellowship from 1994 to 2001 at XJTU. He is a member of both IEEE and ACM. As of Jan, 2009, he holds 1 US patent and has 18 more patents pending.

Simon Winder is a Senior Developer in the Interactive Visual Media group at Microsoft Research. He obtained his Ph.D. in 1995 from the School of Mathematical Sciences, University of Bath, UK, studying computational neuroscience of primate vision. Prior to joining Microsoft, Simon obtained a B.Sc. in 1990 and a M.Eng. in 1991 from the University of Bath, studying Electrical and Electronic Engineering. Prior employment includes work on thermal imaging hardware at GEC Sensors, Basildon, UK, and later work on MPEG-4 video standardization for the Partnership in Advanced Computing Technologies, Bristol, UK. His current research includes feature detection and descriptors for matching and recognition with application to 3D reconstruction and real-time scene recognition, localization, and mapping. To date he has filed 20 US patents.