Picking the best DAISY

Simon Winder
Microsoft Research
swinder@microsoft.com

Gang Hua
Microsoft Live Labs
ganghua@microsoft.com

Matthew Brown
University of British Columbia
mbrown@cs.ubc.ca

Abstract

Local image descriptors that are highly discriminative, computational efficient, and with low storage footprint have long been a dream goal of computer vision research. In this paper, we focus on learning such descriptors, which make use of the DAISY configuration and are simple to compute both sparsely and densely. We develop a new training set of match/non-match image patches which improves on previous work. We test a wide variety of gradient and steerable filter based configurations and optimize over all parameters to obtain low matching errors for the descriptors. We further explore robust normalization, dimension reduction and dynamic range reduction to increase the discriminative power and yet reduce the storage requirement of the learned descriptors. All these enable us to obtain highly efficient local descriptors: e.g, 13.2% *error at 13 bytes storage per descriptor, compared with* 26.1% *error at 128 bytes for SIFT.*

1. Introduction

Local feature matching has become ubiquitous in vision for recognition and registration. In recognition it is often combined with vector quantization to create "visual words" for searching large image databases and for object class recognition [14, 15, 24, 7, 4]. When combined with interest points [11, 12], it facilitates point matching without initialization for image stitching and structure from motion [3, 16, 19]. The extensive research on designing local image descriptors has always been toward image descriptors which are highly discriminative, computational efficient and can be stored in only a few bytes.

Building on our prior descriptor learning work [23], we pursue such image descriptors by learning the optimal descriptors with a DAISY configuration [20] using a new training data set of match/non-match image patches. This new data set is improved from our previous data-set [23, 8] in the sense that each image patch is now centered on a real interest point, and therefore it is not necessary to introduce synthetic jittering noise during training and testing.

We focus particularly on the DAISY configuration for two reasons: First, we have demonstrated that the DAISY configuration (a.k.a, the polar Gaussian pooling approach which has origins in geometric blur [2]) is one of the best for designing discriminative local image descriptors [23]; second, Tola *et al.* [20] demonstrated that such DAISY descriptors can be computed very efficiently both sparsely and densely. In addition to learning the optimal parameters for different DAISY configurations, we further leverage robust normalization, dimension reduction and dynamic range reduction to increase the discriminative power and simultaneously reduce the memory footprint of the learned descriptors. Hence our contributions are the following:

- We use a new ground-truth training set which is based on patches centered on real interest points which have been matched using dense stereo data.
- We test multiple configurations of low-level filters and DAISY pooling and optimize over their parameters.
- We investigate the effects of robust normalization.
- We apply PCA dimension reduction and dynamic range reduction to compress the representation of performant descriptors.
- We discuss computational efficiency and provide a list of recommendations for descriptors that are useful in different scenarios.

2. Related work

A number of good descriptors have been described in the literature [13] although researchers still tend to rely on hand crafted algorithms. Recently there has been a move to learn descriptor parameters for matching tasks and to explore a range of algorithms. Lepetit and Fua [10] showed that randomized trees based on simple pixel differences could be an effective operation. Shotton *et al.* [18] demonstrated a related scheme for object class recognition. Babenko *et al.* [1] applied boosting to learn point-based feature matching representations. Winder and Brown [23] introduced an image descriptor pipeline where combinations of algorithms

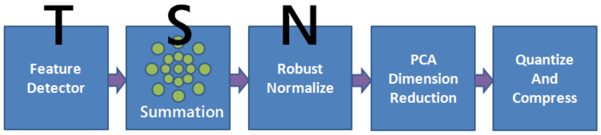

Figure 1. Processing stages in the descriptor algorithm.

were interchanged and each combination was optimized on a matching task by maximizing ROC area.

Since descriptors often have large dimensionality, various authors have studied dimension reduction. In PCA-SIFT, Ke and Sukthankar applied PCA dimension reduction on gradient patches to form local descriptors [9]. Mikolajczyk and Schmid [13] introduced the GLOH descriptor and found good results for PCA dimension reduction. Recently Hua *et al.* [8] used discriminative embedding techniques to find linear projections that actively discriminate between match and non-match classes.

Dimensionality reduction is not the end of the story however, as our eventual aim is to generate descriptors that use as few bits as possible. This is imperative for internet scale recognition, as has been demonstrated by Torralba et al [21] in the context of internet image search. Tuytelaars [22] has also shown successful object recognition performance by quantizing SIFT descriptors to just 4 bits per dimension.

Our work is similar to [23] in that we build a descriptor pipeline and attempt to optimize its parameters using a training set consisting of matching and non-matching image patches that relate to interest points. However we extend this approach by introducing a more realistic and more challenging ground truth data set which avoids the need for synthetic interest points and perturbations. We also add stages for dimension reduction and dynamic range reduction. We focus exclusively on the DAISY footprint, extensively testing its combination with a number of the most promising feature algorithms and, unlike [20], we machine optimize its parameters to obtain the best matching performance.

3. Descriptor Pipeline

Our descriptor pipeline is shown in Figure 1 and is similar to [23] except that we have added extra blocks for dimension reduction and quantization. In addition we focus on a specific range of algorithms that we have found to be promising for each stage.

Descriptors can be sampled densely in an image for applications such as stereo reconstruction or face recognition, or else can be computed from scale and rotation normalized patches sampled from the vicinity of interest points for location matching and 3D reconstruction. In both cases the input to our algorithm is a square image patch and the goal is to produce a reduced dimension vector which uniquely characterizes the region while being robust to common imaging distortions.

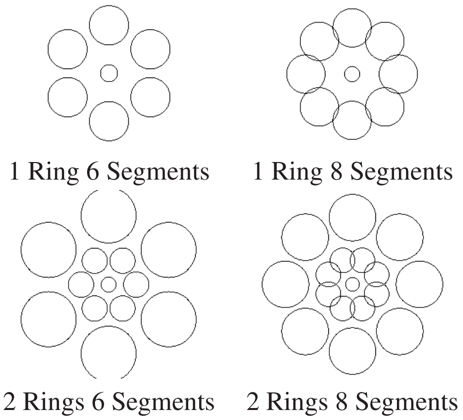

Figure 2. Typical DAISY descriptor Gaussian summation regions learned by our algorithm for steerable filter T-blocks. Circles indicate 1 standard deviation. Best results were obtained by offsetting concentric rings by $180/n$ degrees, where n is the number of segments.

T-block This block takes the pixels from the image patch and transforms them to produce a vector of k non-linear filter responses at each pixel. The elements of the vectors are designed to have positive values. Block T1 involves computing gradients at each pixel and bilinearly quantizing the gradient angle into k orientation bins as in SIFT [11]. Block T2 rectifies the x and y components of the gradient to produce a vector of length 4: $\{|\nabla_x| - \nabla_x; |\nabla_x| + \nabla_x; |\nabla_y| - \nabla_y; |\nabla_y| + \nabla_y\}$, or alternatively length 8 by concatenating this with the 4-vector resulting from rotating the gradient by $45°$ and using the same approach. Both T1 and T2 include a Gaussian pre-smoothing stage to set the gradient scale. Block T3 uses steerable filters [5] evaluated at a number of different orientations. The filters can have odd, even or dual phase and their responses are rectified into positive and negative parts which are then carried by different vector elements in the same way as for T2. For dual phase (quadrature) filters, the vector dimensionality is $k = 4n$ where n is the number of orientation channels. The filter scale was varied by changing the kernel sampling rate [5]. All scale parameters were machine optimized jointly with other descriptor parameters. Further details of all these T-blocks can be found in [23].

S-block This stage spatially accumulates weighted filter vectors to give N linearly summed vectors of length k and these are concatenated to form a descriptor of kN dimensions. For this block we use normalized Gaussian summation regions arranged in a series of concentric rings (called S4 by [23] and the DAISY descriptor by [20]). Typical configurations are shown in Figure 2. The sizes of the Gaussians and the radii of the rings are parameters that we optimize (see below). The total number of dimensions at this stage $D = k\,(1 + \mathit{rings} \times \mathit{segments})$.

N-block The N-block normalizes the complete descriptor to provide invariance to lighting changes. One possibility is to use simple unit-length normalization. We use a form of threshold normalization with the following stages: (1) Normalize the descriptor 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) Scale the vector to a byte range. [11] This procedure has the effect of reducing the dynamic range of the descriptor and creating a robust function for matching. We learn the best threshold value κ.

Dimension Reduction Previously we used discriminative learning such as locality preserving projections for dimension reduction [8]. However, other authors [17], and our own experiments have found that applying principal components analysis (PCA) to image filter responses without class labels can be just as effective if the high-dimensional representation is already discriminative. Here we use PCA in this manner. To learn PCA projections, we first optimize the parameters of the descriptor and then compute the matrix of principal components based on all descriptors computed on the training set. Next we find the best dimensionality for reduction by computing the error rate on random subsets of the training data while progressively increasing the dimensionality by adding PCA bases until a minimum error is found. This gives us the final reduced transformation matrix for the descriptor pipeline. Additionally, we always normalize the length of descriptor vectors following the dimension reduction stage.

Quantization In image compression, such as in JPEG, it is common to transform image data into another space, e.g., using DCTs, and then to quantize these transformed coefficients before Huffman coding. Here we employ a similar dynamic range quantization with an aim to reduce memory requirements when large databases of descriptors are stored. Descriptor elements (either signed when PCA reduction is used or unsigned when it is not) are quantized into L levels. For example with signed descriptor elements v_i and L odd, quantized elements $q_i = \lfloor \beta L v_i + 0.5 \rfloor$, where $q_i \in \{-(L-1)/2, \dots, (L-1)/2\}$ and β is a single common scalar which is optimized to give the best error rate on the training data. For even numbers of levels we use $q_i = \lfloor \beta L v_i \rfloor$ with $q_i \in \{-L/2, \dots, L/2 - 1\}$. For this paper, we quantized all PCA-reduced dimensions to the same number of levels despite their differences in variance. This was motivated by common practice in encoding transform coefficients for image and video compression but could be an area of experimentation.

4. Training and Testing

Recent advances in wide base-line matching and structure from motion allow reconstructing 3D points and cameras for data sets containing thousands of images [19]. Furthermore advances in multi-view stereo allow dense surface models to be obtained despite greatly varying imaging conditions [6]. We use these 3D reconstructions as a source of training data. Previous work [23] used re-projections of 3D point clouds to act as synthetic interest points around which known corresponding patches could be sampled. This has the disadvantage that it does not capture the real statistics of interest point noise and results in data sets which are not sufficiently demanding. We therefore use dense surface models obtained via stereo matching to establish correspondences between real interest points. Multi-view constraints allow us to generate accurate correspondences that would be very challenging for unconstrained 2D matching.

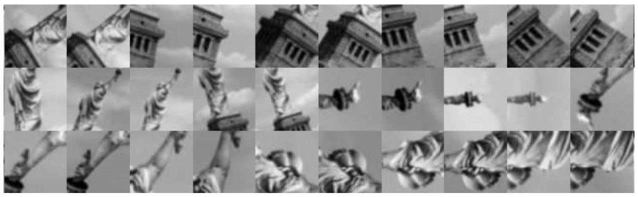

Figure 3. Example image patches from our Liberty data set.

We make use of camera calibration and dense multi-view stereo data for three datasets—Yosemite, Liberty, and Notre Dame—containing over 1000 images provided by [19]. We detect Difference of Gaussian interest points with associated position, scale and orientation [11] in each image and we extract scale and orientation normalized patches around these points and store them in a database. To determine ground truth matches we make use of the provided depth maps to transfer a local dense sampling of points around each interest point into a second image and then use least squares to estimate the expected position, scale and orientation of the projected interest point. We check to see if the interest point would be visible in the second image by using visibility maps from [6]. We then declare the nearest true interest point in the second image to be a match if it is detected within 5 pixels of position, 0.25 octaves of scale and $\pi/8$ radians of angle. All interest points falling outside twice these ranges are defined to be non-matches. Interest point detections lying between these ranges are deemed to be ambiguous and are not used in training or testing. Changing these design points would allow us to trade off invariance and discrimination in any descriptors that we learn. Figure 3 shows example patches from the data set. [1]

In order to optimize descriptor parameters we use exactly the approach described in [23]. In general we find that machine optimization of parameters is crucial and produces far better error rates than trying to guess them by hand. We retained one data set of 100,000 random patch pairs with 50% matches (Yosemite) for training and used two 100,000 pair datasets for testing (Liberty and Notre Dame). To learn parameters we maximize the area under the ROC curve

[1] http://www.cs.ubc.ca/∼mbrown/patchdata/patchdata.html

	1 Ring		2 Rings	
Segments:	6	8	6	8
T1-4	34.43	34.24	29.05	28.64
T1-8	27.89	26.52	23.28	22.94
T1-12	26.55	26.19	22.85	22.57
T1-16	26.93	26.28	22.59	22.75
T2-4	35.77	35.62	30.21	29.38
T2-8	35.01	34.85	28.35	28.29
T2-8a	26.96	26.16	23.03	22.57
SIFT	35.09			

Table 1. Error rates for gradients: Liberty

	1 Ring		2 Rings	
Segments:	6	8	6	8
T1-4	26.10	25.66	20.91	20.82
T1-8	21.09	20.33	16.92	15.62
T1-12	21.26	20.73	16.96	16.23
T1-16	21.33	20.45	16.95	16.77
T2-4	27.51	27.08	21.59	21.35
T2-8	25.97	25.99	19.56	19.82
T2-8a	21.20	20.79	16.58	16.60
SIFT	26.10			

Table 2. Error rates for gradients: Notre Dame

by using Powell's conjugate gradient method that operates without the need for derivatives. At each step of gradient descent we loop over the data set of match or non-match patch pairs computing descriptor-space distances. These distances are then accumulated into match and non-match histograms from which an ROC curve and its area can be computed. Once parameters have stabilized, the final error rates are evaluated on the test sets. Typical descriptors had from 5 to 15 parameters, and in general training was reliable and repeatable from different initial conditions.

5. Results

For each trained descriptor we computed ROC curves and obtained % error rates when 95% of all correct matches were obtained. We show results for training on the Yosemite data set and testing on the 100,000 patch-pair Liberty and Notre Dame sets.

5.1. Gradient-based Descriptors

Tables 1 and 2 show results for descriptors that make use of image gradients in their T-blocks. T1 involves soft histogramming of the gradient angle into k bins while T2 involves direct use of the rectified 1D derivatives. We varied the number of T-block orientation bins and tested four configurations of DAISY pooling. Error rates fall from 4 (T1-4) to 8 (T1-8) orientations for T1 but beyond that show little change, so larger numbers of orientation bins are unnecessary.

	1 Ring		2 Rings	
Segments:	6	8	6	8
T3-2nd-2	30.40	30.50	26.88	26.71
T3-2nd-4	28.05	27.72	23.22	23.39
T3-2nd-6	27.50	28.02	23.12	22.94
T3-2nd-8	27.61	27.96	23.53	22.90
T3-4th-2	42.81	42.25	37.25	36.27
T3-4th-4	33.23	32.97	28.60	28.82
T3-4th-6	31.61	32.08	28.15	27.73
T3-4th-8	31.71	31.88	27.67	27.76
SIFT	35.09			

Table 3. Error rates for steerable filters: Liberty

	1 Ring		2 Rings	
Segments:	6	8	6	8
T3-2nd-2	21.54	21.34	17.22	17.24
T3-2nd-4	18.38	18.45	14.79	14.16
T3-2nd-6	18.09	18.37	14.44	14.09
T3-2nd-8	18.26	18.21	14.63	14.33
T3-4th-2	33.78	32.81	28.15	27.44
T3-4th-4	33.23	24.33	20.20	20.04
T3-4th-6	23.54	23.15	19.88	19.03
T3-4th-8	23.33	22.67	19.44	18.64
SIFT	26.10			

Table 4. Error rates for steerable filters: Notre Dame

T2 performs slightly less well than T1 for the same dimensionality. Adding extra dimensions when going from T2-4 to T2-8 show less reduction in error than going from T1-4 to T1-8. This is probably because the orientation selectivity of T2 is much wider and the T2-8 vector elements are therefore more correlated than the T1-8 elements. To test this, we modified T2-8 to include a stage which narrows selectivity by subtracting the mean in a manner similar to biological cross-orientation inhibition: $v_i' = \max(v_i - \frac{\alpha}{k}\sum v_j, 0)$. This resulted in the significantly improved error rates shown as T2-8a and $\alpha \approx 2.5$ was found to be optimal. T2 is less computationally expensive than T1 because it avoids polar conversion and bilinear weighting operations. In fact if one is satisfied with the error rate of SIFT, then this can be approximately matched by **T2-4-1r6s** with very low complexity and only 28 dimensions.

Our gradient results also show that for spatial summation of filter vectors, two DAISY rings give significantly better error rates than a single ring. We found this result to be consistent across all our descriptors. Additionally we found minor but consistent improvement when moving from 6 to 8 segments per ring. Overall descriptor dimensionality varies from 28 (**T1-4-1r6s**) to 272 (**T1-16-2r8s**) in these tables.

5.2. Steerable Filters

We extensively tested combinations of steerable filters with different arrangements of DAISY spatial pooling. In Tables 3 and 4, 2nd and 4th order filters are used with dif-

Segments:	6	8	12	
Orientations:	6	4	6	Rings
T3-2nd-odd	31.51	31.36	32.32	1
T3-2nd-even	33.36	34.29	34.31	1
T3-2nd-dual	27.50	27.72	28.81	1
T3-4th-odd	34.36	35.33	34.73	1
T3-4th-even	34.15	35.69	34.40	1
T3-4th-dual	31.61	32.97	32.19	1
T3-2nd-odd	25.33	25.33	25.28	2
T3-2nd-even	28.20	28.43	27.01	2
T3-2nd-dual	23.12	23.39	23.22	2
T3-4th-odd	29.43	30.24	28.77	2
T3-4th-even	28.95	30.16	28.53	2
T3-4th-dual	28.15	28.82	28.00	2
T3-2nd-odd	24.30	24.10	23.58	3
T3-2nd-even	28.06	27.83	27.48	3
T3-2nd-dual	23.06	22.73	22.49	3

Table 5. Error rates for steerable filters: Liberty

Segments:	6	8	12	
Orientations:	6	4	6	Rings
T3-2nd-odd	22.17	22.21	23.23	1
T3-2nd-even	24.64	24.78	24.19	1
T3-2nd-dual	18.09	18.45	18.47	1
T3-4th-odd	26.78	27.07	25.89	1
T3-4th-even	25.82	27.98	26.06	1
T3-4th-dual	23.54	24.33	23.67	1
T3-2nd-odd	17.38	17.40	16.87	2
T3-2nd-even	19.21	18.72	18.15	2
T3-2nd-dual	14.44	14.16	14.58	2
T3-4th-odd	21.08	21.48	20.23	2
T3-4th-even	20.67	21.84	19.78	2
T3-4th-dual	19.88	20.04	19.22	2
T3-2nd-odd	15.78	15.71	14.98	3
T3-2nd-even	18.89	18.37	17.46	3
T3-2nd-dual	14.43	14.53	14.19	3

Table 6. Error rates for steerable filters: Notre Dame

ferent numbers of orientation channels. Each filter orientation involves a quadrature pair which is rectified into four T-block vector elements (Section 3). Similar to the gradient results, we found that the error rate reduced as the number of orientations increased up to a point. For 2nd order filters, 4 orientations are probably sufficient, but 6 or more are required for 4th order filters before there is a plateau in error rate. This is most likely due to the narrower orientation bandwidth of 4th order filters. Contrarily to the results of [23] we found that 4th order filters performed significantly less well than 2nd order filters. This is probably due to the more challenging and realistic data sets that we used which provide a clear cut separation of descriptor performance data.

As with gradients, moving from one to two rings produced a large reduction in error rate while moving from 6 to 8 segments produced marginal improvements. These descriptors had from 56 (**T3-2nd-2-1r6s**) to 544 dimensions (**T3-2nd-8-2r8s**). The error rates for T3 steerable filters were better than for T1 gradients but this difference was only apparent for testing on the Notre Dame data set.

In [23] it was shown that it is important to maintain feature phase. But is it necessary to use a quadrature pair of steerable filters? To test this we compared the performance of quadrature pairs with the performance when only odd or even symmetric filters were used. These results are presented in Tables 5 and 6 and clearly show that using both phases produces a significantly better error rate than odd or even filters alone. It seems that the information carried by even and odd filter responses is sufficiently independent to boost performance when used together. For 2nd order, odd filters were found to give better results than even filters. It could be that odd filters allow better discrimination among the edges that are prevalent in natural images. Fourth order filters are less selective for wide-band features and this could be why error rates for odd versus even 4th order filters are similar.

These results also show that doubling the dimensionality of the descriptor by using two filter phases instead of one is a better plan than doubling the number of DAISY segments from 6 to 12. Large numbers of segments seem not to improve the result since the Gaussian regions start to overlap and over-sample the descriptor footprint. However, computing 12 segments with 6 filter orientations or computing 8 segments with 4 filter orientations has the advantage that the resulting descriptors can be trivially rotated in steps of $30°$ or $45°$ respectively, simply by permuting the order of descriptor dimensions at the S-block output.

These tables also show that more rings are better: The improvement from 2 to 3 rings would probably increase if we were not limited by the 64×64 patch size. During training, we observed that the size constants of the steerable filters and the footprint of the DAISY spatial pattern tended to increase jointly until the DAISY was limited by the bounds of the patch.

5.3. Dual-Band Descriptors

Since many applications make use of multi-scale pyramids, we decided to test the idea of combining descriptors at two spatial scales. Steerable filters have a band-pass response so it is reasonable to expect that more information is available if two filter banks are used which are tuned to different spatial scales. We concatenated the descriptors resulting from two parallel T and S-block channels and learned the relative size constants for the filters and the two DAISY footprints. Tables 7 and 8 show the results for quadrature 2nd and 4th order filters having 4 or 6 orientations paired with a 2 ring, 6 segment S-block. Fixed filter scale ratios were tried that correspond to typical pyramid scale inter-

		Without PCA		With PCA	
T-Block	Scale	Error	Dims	Error	Dims
T3-2nd-4	×1.26	22.04	416	-	-
T3-2nd-4	×1.41	21.61	416	-	-
T3-2nd-4	×2.00	19.36	416	18.36	182
T3-2nd-4	Learn	18.75	416	17.24	37
T3-4th-4	Learn	21.39	416	18.55	144
T3-2nd-6	×1.26	22.09	624	-	-
T3-2nd-6	×1.41	21.15	624	-	-
T3-2nd-6	×2.00	20.02	624	19.08	27
T3-2nd-6	Learn	19.12	624	17.14	42
T3-4th-6	Learn	21.32	624	18.85	166
SIFT	-	35.09	128	-	-

Table 7. Error rates for 2 ring 6 segment two-scale steerable filter descriptors: Liberty

		Without PCA		With PCA	
T-Block	Scale	Error	Dims	Error	Dims
T3-2nd-4	×1.26	13.01	416	-	-
T3-2nd-4	×1.41	12.30	416	-	-
T3-2nd-4	×2.00	10.50	416	9.77	182
T3-2nd-4	Learn	10.03	416	9.71	37
T3-4th-4	Learn	12.00	416	10.00	144
T3-2nd-6	×1.26	13.02	624	-	-
T3-2nd-6	×1.41	12.25	624	-	-
T3-2nd-6	×2.00	10.73	624	11.60	27
T3-2nd-6	Learn	10.05	624	9.49	42
T3-4th-6	Learn	11.70	624	9.75	166
SIFT	-	26.10	128	-	-

Table 8. Error rates for 2 ring 6 segment two-scale steerable filter descriptors: Notre Dame

vals, as well as simply learning the best ratio. Error rates obtained using this method were excellent, dropping to around 10% for the Notre Dame data set. We found that a factor of two was close to optimal between the scale of the two filter banks and between their associated DAISY footprints (learned values were ≈ 2.2). In addition we noticed that there was more improvement for 4th order filters than for 2nd, presumably because two parallel filter banks were able to make up for the narrower spatial frequency bandwidth in the case of 4th order. This suggests that it would be interesting learn a parametric T-block filter and optimize the frequency bandwidth directly.

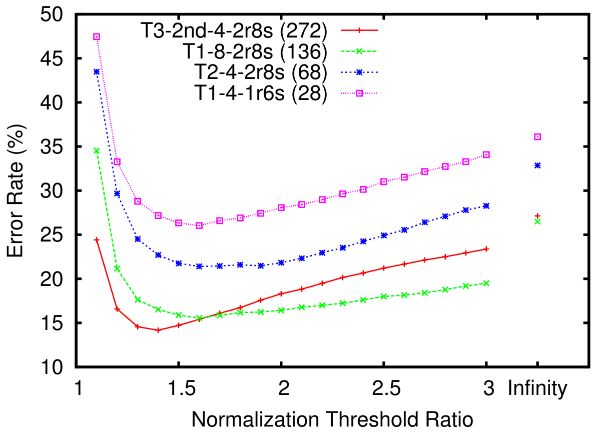

Figure 4. Variation of error rate with normalization threshold (Notre Dame). The threshold was set to $r/\sqrt{D}$ where r is the ratio and D is the descriptor dimensionality (given in brackets).

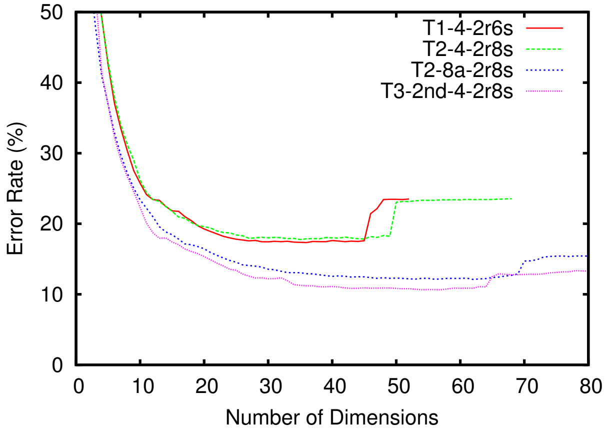

Figure 5. Variation of error rate with the number of PCA bases.

5.4. Normalization

In [23] it was noted that SIFT-style clipping normalization performed better than simple unit vector normalization. We decided to investigate this more thoroughly. Since the descriptors so far maintain a direct relation between image-space and descriptor coefficients at the S-block output, 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.

Figure 4 shows the typical effect of changing the clipping threshold for our normalization. Error rates are significantly improved when the clipping thresholds are equal to around $1.6/\sqrt{D}$ when tested on a wide range of descriptors with different dimensionality D. This graph shows the drastic reduction in error rate compared with simple unit normalization ("Infinity" on the graph).

5.5. Dimension Reduction

Various authors have sought to apply PCA and other dimensionality reduction methods to descriptors [13, 9, 8]. We therefore applied PCA techniques to reduce the dimensionality of our learned descriptors. The matrix of principal components was computed using descriptors from the Yosemite training set. Figure 5 shows how the error rate on the training set changes as the number of dimensions is increased by progressively adding PCA bases. We use these curves to determine the best dimensionality for lowest error, although there is typically a wide choice to trade off be-

		Without PCA		With PCA	
T-Blk	R/S	Error	Dim	Error	Dim
T1-4	1r6s	34.4 26.1	28	28.0 20.5	27
T1-4	1r8s	34.2 25.7	36	31.1 22.4	17
T1-4	2r6s	29.1 20.9	52	23.5 15.4	44
T1-4	2r8s	28.6 20.8	68	23.9 16.1	25
T1-8	1r6s	27.9 21.1	56	24.0 17.2	41
T1-8	1r8s	26.5 20.3	72	22.9 17.0	23
T1-8	2r6s	23.3 16.9	104	19.5 13.1	62
T1-8	2r8s	22.9 15.6	136	18.9 12.3	53
T2-4	1r6s	35.8 27.5	28	28.7 20.3	26
T2-4	1r8s	35.6 27.1	36	32.8 24.2	15
T2-4	2r6s	30.2 21.6	52	23.8 15.3	31
T2-4	2r8s	29.4 21.4	68	23.8 15.9	29
T2-8a	1r6s	27.0 21.2	56	24.6 18.4	19
T2-8a	1r8s	26.2 20.8	72	22.0 16.5	49
T2-8a	2r6s	23.0 16.6	104	18.6 12.9	67
T2-8a	2r8s	22.6 16.6	136	19.9 13.5	35
T3-6	1r6s	27.5 18.1	168	21.5 12.8	45
T3-4	1r8s	27.7 18.5	144	21.5 13.2	46
T3-6	1r12s	28.8 18.5	312	24.7 16.0	21
T3-6	2r6s	23.1 14.4	208	18.0 10.9	33
T3-4	2r8s	23.4 14.2	272	19.3 12.2	26
T3-6	2r12s	23.2 14.6	600	19.8 12.8	25
SIFT	-	35.1 26.1	128	-	-

Table 9. Error rates for descriptors with PCA. Error rate figures show Liberty then Notre Dame test set results. R/S - Rings / Segments. Dim - Dimensions. T3 uses 2nd order steerable filters.

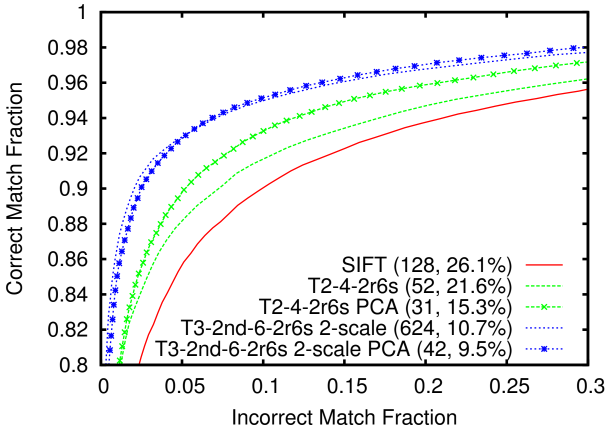

Figure 6. ROC curves for selected descriptors: Notre Dame. Figures in brackets show dimensionality and error rates.

tween the two. It can be seen that PCA is not just useful in reducing the dimensionality—it is also beneficial in reducing the error rate still further by removing noise dimensions which often contribute considerably to the error. Tables 7, 8, and 9 show error rates for selected descriptors. In all cases it can be seen that PCA is able to both reduce the error rate and reduce the number of dimensions required. For the Notre Dame set, 9.7% error is possible with only 37 dimensions compared to 26.1% and 128 dimensions for SIFT. Particularly large reductions in dimensionality and error rate are obtained for the T2 based descriptors, e.g., 24.2% at 15 dimensions for **T2-4-1r8s**, which is satisfying because they are simple to compute.

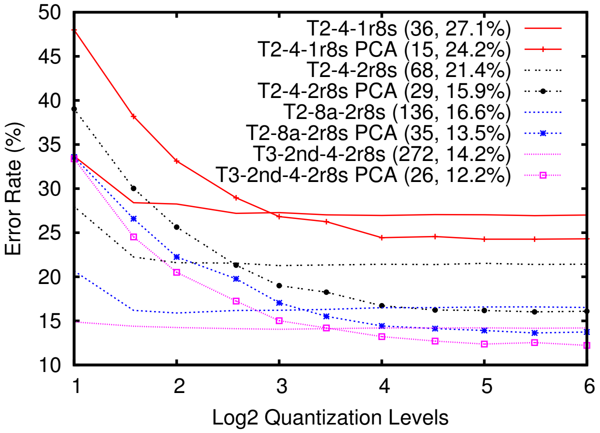

Figure 7. Quantization of descriptor dynamic range: Notre Dame. Error rate reduces rapidly as the number of levels used to represent the descriptor elements is increased from 2^1 to 2^6. Figures in brackets show dimensionality and error rates.

5.6. Descriptor Quantization

We sought to reduce the storage requirements for the descriptors still further by quantizing each dimension independently as described in Section 3. We found that typically only a few bits of dynamic range are required (Figure 7). This is especially true for descriptors without PCA where extremely aggressive quantization is possible. In particular, we found that for many of the higher dimensional descriptors it is only necessary to keep 1 bit per dimension while still maintaining good error rates. This is due in part to the thresholding normalization which already results in near binarization of these descriptors. When PCA is used, the error rate comes down more slowly as more quantization levels are added and typically reaches a plateau at around 4 bits per dimension. Since PCA often reduces dimensionality and error rates substantially, this still translates into a bit reduction over non-PCA descriptors for the same error, albeit with higher computational cost.

Examples from Figure 7 show that **T2-4-1r8s** combined with PCA reduction to 15 dimensions can be quantized at 4 bits per dimension to give 7.5 bytes in total at 24.4% error, and **T3-2nd-4-2r8s** with PCA can be compressed to 13 bytes at 13.2% error. These numbers compare favorably with 128 bytes and 26.1% error for SIFT. It may be possible that additional compression could be achieved by using variable length Huffman codes, but we did not try this experiment. In addition, for PCA, it would be interesting to test the effects of quantizing different dimensions at different numbers of levels or with different β gains.

6. Discussion

In this paper we have demonstrated a number of descriptors with low error rate, low computation burden and low storage footprint. Parameters for all our descriptors are available from the authors. They were optimized for matching around interest points but we have observed them to perform well in various related applications. Three scenarios are of interest when selecting from the range of descriptors available: Real-time, e.g., for mobile devices; highly discriminative, e.g., object class recognition; and large databases, e.g., image search or geolocation from images.

In a real time mobile device application we first favor low computational burden and perhaps also small descriptors. The T2-4 blocks with one or two rings are particularly cheap to compute and have low dimensionality. They can also be quantized to 2–3 bits per dimension without PCA. To compute them at fixed rotation, one needs four gradient maps over the whole image and must compute either two or three Gaussian blurs on each [20]. After this the descriptors can be point sampled where needed and then threshold normalized.

For applications that require good discrimination, the descriptors with lowest error make use of second order steerable filters at two spatial scales and apply PCA to remove nuisance dimensions. Examples are given in Table 8.

Large data-base applications require a descriptor with very low storage requirements and relatively low computational burden. **T3-2nd-4-2r8s** and **T2-4-1r8s** with PCA are good candidates which take up only a few bytes.

Although all our descriptors can be computed on rotated and scaled patches, computational benefit results from using approximate discrete rotations and scales by employing a scale pyramid, permuting the T-block output and rotating the DAISY point sampling pattern, or else simply permuting the descriptor after normalization in the case where the number of T-block orientations is suitably matched with the number of DAISY segments. Descriptors with this convenient rotation property include T3-6 with 12 segments and T1-8, T2-8a or T3-4 with 8 segments. Further work should focus on the reliability of matching and data-base lookup using this scenario since this rotation/scale discretization is a characteristic of the fast method of Tola *et al.* [20].

7. Acknowledgements

The authors thank Michael Goesele for kindly providing stereo data, and also Matthew Uyttendaele, Drew Steedly, and David Nister for helpful conversations.

References

[1] B. Babenko, P. Dollar, and S. Belongie. Task specific local region matching. In *ICCV*, 2007.

[2] A. C. Berg and J. Malik. Geometric blur for template matching. In *CVPR*, pages 607–614, 2001.

[3] M. Brown and D. Lowe. Automatic panoramic image stitching using invariant features. *IJCV*, 74(1):59–73, 2007.

[4] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[5] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE PAMI*, 13:891–906, 1991.

[6] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. Seitz. Multi-view stereo for community photo collections. In *ICCV*, 2007.

[7] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, Bejing, October 2005.

[8] G. Hua, M. Brown, and S. Winder. Discriminant embedding for local image descriptors. In *ICCV*, 2007.

[9] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *CVPR*, pages 506–513, 2004.

[10] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE PAMI*, 28(9):1465–1479, 2006.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004.

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

[13] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE PAMI*, 27:1615–1630, 2005.

[14] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, pages 2161–2168, 2006.

[15] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[16] M. Pollefeys, L. van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *IJCV*, 59(3):207–232, 2004.

[17] H. Shan and G. W. Cottrell. Looking around the backyard helps to recognize faces and digits. In *CVPR*, 2008.

[18] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[19] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *ACM Transactions on Graphics*, volume 25, pages 835–846, 2006.

[20] E. Tola, V. Lepetit, and P. Fua. A fast local descriptor for dense matching. In *CVPR*, 2008.

[21] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large databases for object recognition. In *CVPR*, 2008.

[22] T. Tuytelaars and C. Schmid. Vector quantizing feature space with a regular lattice. In *ICCV*, 2007.

[23] S. A. J. Winder and M. Brown. Learning local image descriptors. In *CVPR*, 2007.

[24] J. G. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *IJCV*, 73(2):213–238, 2007.