Descriptor Learning for Efficient Retrieval

James Philbin[1], Michael Isard[3], Josef Sivic[2], and Andrew Zisserman[1]

[1] Visual Geometry Group, Department of Engineering Science, University of Oxford
[2] INRIA, WILLOW, Laboratoire d'Informatique de l'Ecole Normale Superieure, Paris
[3] Microsoft Research, Silicon Valley

Abstract. Many visual search and matching systems represent images using sparse sets of "visual words": descriptors that have been quantized by assignment to the best-matching symbol in a discrete vocabulary. Errors in this quantization procedure propagate throughout the rest of the system, either harming performance or requiring correction using additional storage or processing. This paper aims to reduce these quantization errors *at source*, by learning a projection from descriptor space to a new Euclidean space in which standard clustering techniques are more likely to assign matching descriptors to the same cluster, and non-matching descriptors to different clusters.

To achieve this, we learn a non-linear transformation model by minimizing a novel margin-based cost function, which aims to separate matching descriptors from *two* classes of non-matching descriptors. Training data is generated automatically by leveraging geometric consistency. Scalable, stochastic gradient methods are used for the optimization.

For the case of particular object retrieval, we demonstrate impressive gains in performance on a ground truth dataset: our learnt 32-D descriptor without spatial re-ranking outperforms a baseline method using 128-D SIFT descriptors with spatial re-ranking.

1 Introduction

We are interested in the problem of efficiently retrieving occurrences of a particular object, selected by an image query, in a large unorganized set of images. Typically, methods in particular object retrieval take a text-retrieval approach to the problem in order to achieve fast retrieval at run time [1,2,3,4]. Interest points and descriptors are found in every dataset image and the descriptors are then clustered (usually by k-means or some variant) and quantized to give a visual word representation for each image in the corpus.

Whilst being ostensibly similar to textual words, visual words as generated through clustering suffer from a lot more noise and dropout compared to text. This is caused partly by errors and failures in interest point detection and description, but also by quantization – descriptors that lie close to a Voronoi boundary after clustering being assigned to the "wrong" visual word. Previous work attempted to overcome quantization errors by compensating for mis-clustered descriptors using additional information in the retrieval index, for example by soft-assigning descriptors [5,6,7], or by performing more work at query time [1,8].

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part III, LNCS 6313, pp. 677–691, 2010.

Instead, the goal of this work is to reduce these errors at source, by constructing a projection from the raw descriptor space to a new Euclidean space in which matching descriptors are more likely to land in the same cluster, and non-matching descriptors are more likely to land in different clusters. By removing the initial quantization errors, we keep the indexes small (for example, they become less sparse when soft-assignment is used) and the query times fast. Optionally, our method can also reduce the dimensionality of the projected descriptors resulting in smaller storage requirements for features and increased clustering and quantization speeds during pre-processing.

There have been several recent applications of distance learning to classification problems [9,10,11,12,13,14,15], however these methods assume clean, labelled data indicating pairs of points that belong to the same class and pairs that belong to different classes. In our task, even when the same object appears in two images, the images typically have different backgrounds and there is a non-trivial transformation between the views of a common object, so we cannot simply classify *images* as being matching or non-matching. At the same time the number of individual descriptors per image and the complexity of the correspondence problem between them means that manually labelling the sets of matching and non-matching descriptors would be unacceptably burdensome. Therefore, in this work, we introduce a new method for generating training data from a corpus of unlabelled images using standard techniques from multi-view geometry. In contrast to Hua *et al.* [16], who also generated training pairs from unlabelled image data via patches matched by the Photo Tourism system [17], here we adopt a much cheaper pairwise image measure which doesn't require us to compute a global bundle adjustment over many image pairs. Thus, we can train on patches of objects that appear in as few as two images.

Previous works in distance learning use two categories of point pairs for training: "matching" and "non-matching", typically derived from known class labels. In this work, we show that we can significantly improve performance by forming two "non-matching" categories: random pairs of features; and those which are easily confused by a baseline method. We adopt a margin-based cost function to distinguish these three categories of points, and show that this gives improved performance more than using non-margin-based methods [14,16].

To optimize this cost function, a fast, stochastic, online learning procedure is used that permits the use of millions of training pairs. We will show that non-linear projection methods, previously used for hand-written digit classification [13], perform better than the linear projections previously applied to computer vision distance learning [9,10,11,12].

The next section motivates the distance learning task by showing that retrieval performance is significantly worse using standard quantized descriptors than when a much slower, exhaustive search procedure is applied to the raw SIFT descriptors – this indicates the potential gain achievable from better clustering. After describing in Section 3 how we automatically generate our training data, we set out our learning methods in Section 4 and then conclude with results and

a discussion. Improved performance is demonstrated over SIFT descriptors [18] on standard datasets with learnt descriptors as small as 24-D.

2 Datasets and the mAP Performance Gap

To learn and evaluate, we use two publicly available datasets with associated ground truth: (i) the *Oxford Buildings* dataset [19]; and (ii) the *Paris Buildings* dataset [20]. We show that a significant performance gap (the *mAP-gap*) is incurred by using quantized descriptors compared to using the original descriptors. It is this gap that we aim to reduce by learning a descriptor projection.

2.1 Datasets and Performance Measure

Both the Oxford (5.1K images) and Paris (6.3K images) datasets were obtained from Flickr by querying the associated text tags for famous landmarks, and both have an associated ground truth for 55 standard queries: 5 queries for each of 11 landmarks in each city. To evaluate retrieval performance, the Average Precision (AP) is computed as the area under the precision-recall curve for each query. As in [3], an Average Precision score is computed for each of the 5 queries for a landmark. These scores are averaged (over 55 query images in total for each dataset) to obtain an overall mean Average Precision (mAP) score.

Affine-invariant Hessian regions [21] are computed for each image, giving approximately $3,300$ features per image (1024×768 pixels). Each affine region is represented by a 128-D SIFT descriptor [18].

2.2 Performance Loss Due to Quantization

To assess the performance loss due to quantization, four retrieval systems (RS) are compared:

The baseline retrieval system (RS1): In this system each image is represented as a "bag of visual words". All image descriptors are clustered using the approximate k-means algorithm [3] into 500K visual words. At indexing and query time each descriptor is associated with its (approximate) nearest cluster centre to form a visual word, and a retrieval ranking score is obtained using tf-idf weighting. No spatial verification is performed. Note that each dataset has its own vocabulary.

Spatial re-ranking to depth 200 (RS2): For this system a spatial verification procedure [3] is adopted, estimating an affine homography from single image correspondences between the query image and each target image. The top 200 images returned from RS1 are re-ranked using the number of inliers found between the query and target images under the computed homography.

Spatial verification to full depth (RS3): The same method is used as in RS2, but here *all* dataset images are ranked using the number of inliers to the computed homography.

Table 1. The mAP performance gap between raw SIFT descriptors and visual words on the Oxford and Paris datasets. In the spatial cases, an affine homography is computed using RANSAC and the data is re-ranked by the number of inliers. Using raw SIFT descriptors coupled with Lowe's second nearest neighbor test [22] gives a 14% retrieval boost over the baseline method for Oxford. (i)-(iii) all use a $K = 500,000$ vocabulary trained on their respective datasets.

Item	Method	Oxford mAP	Paris mAP
i.	RS1: Baseline (visual words, no spatial)	0.613±0.011	0.643±0.002
ii.	RS2: Spatial (visual words, depth=200)	0.647±0.011	0.655±0.002
iii.	RS3: Spatial (visual words, depth=FULL)	0.653±0.012	0.663±0.002
iv.	RS4: Spatial (raw descriptors, depth=FULL)	0.755	0.672

Raw SIFT descriptors with spatial verification (RS4): Putative matches on the raw SIFT descriptors (no quantization) are found between the query and every image in the dataset using Lowe's second nearest neighbour test [18] (threshold = 0.8). Spatial verification as in RS3 is applied to the set of putative matches.

It should be noted that the methods RS3 and RS4 exhaustively match document pairs and so are infeasibly slow for real-time, large scale retrieval. RS3 is ∼10 times slower and RS4 is ∼100 times slower than RS2 even on the 5.1K Oxford dataset. These run-time gaps increase linearly for larger datasets.

The results for all four methods are shown in table 1. For methods based on visual words, the mean and standard deviation over 3 runs of k-means with different initializations are shown. Going from baseline (i) to baseline plus spatial (ii) gives moderate improvements to both datasets, but reranking significantly more documents gives little appreciable further gain. In contrast, using the raw SIFT descriptors gives a large boost in retrieval performance for both datasets, demonstrating that the *mAP-gap* is principally due to quantization errors. This implies that a lack of visual word matches contributes substantially more to missed retrievals than reranking too few documents at query time. The raw-descriptor matching procedure will be used to generate point pairs for our learning algorithm, so Table 1(iv) gives a rough upper bound to the retrieval improvement we can hope to achieve using any learning algorithm based on those training inputs.

3 Automatic Training Data Generation

In this section, we describe our method to automatically generate training data for the descriptor projection learning procedure. The training data is generated by pair-wise image matching, a much cheaper alternative to the full multi-view reconstruction used in [16,17], allowing us to generate a large number (3M+) of training pairs. In addition to positive (matched) examples, we separately collect "hard" and "easy" negative examples and show later that making this distinction can significantly improve the learnt projections.

We proceed as follows: (i) An image pair is chosen at random from the dataset; (ii) A set of *putative* matches is computed between the image pair. Each putative

match consists of a pair of elliptical features, one in each image, that pass Lowe's second nearest neighbour ratio test [18] on their SIFT descriptors; (iii) RANSAC is used to estimate an affine transform between the images together with a number of inliers consistent with that transform. Point pairs are only taken from image matches with greater than 20 verified inliers. The ratio test ensures that putative matches are distinctive for that particular pair of images. This procedure generates three sets of point pairs, shown in Figure 1, that we treat distinctly in the learning algorithm:

1. **Positives:** These are the point pairs found as inliers by RANSAC.
2. **Nearest neighbour negatives (nnN):** These are pairs marked as outliers by RANSAC—they are generally close in descriptor space as they were found to be descriptor-space nearest neighbors between the two images, but are spatially inconsistent with the best-fitting affine transformation found between the images.
3. **Random negatives (ranN):** These are pairs which are not descriptor-space nearest neighbours, i.e. random sets of features generally far apart in the original descriptor space.

A histogram of SIFT distances for the three different sets of point pairs on the Oxford dataset is shown in Figure 2(b). As expected, the original SIFT descriptor easily separates the random negatives from the positive and NN negative point pairs, but strongly confuses the positives and NN negatives. Section 5 will show that the best retrieval performance arises when the positive and NN negative pairs are separated whilst simultaneously keeping the random negative pairs distant. It is important to note that, due to the potential for repeated structure and the limitations of the spatial matching method (only affine planar homographies are considered), some of the nnN point pairs might be incorrectly labelled positives – this can lead to significant noise in the training data. We collect 3M training pairs from the Oxford dataset split equally into positive, NN negative and random negative pairs, and we also have a separate set of 300K pairs used as a validation set to determine regularization parameters.

4 Learning the Descriptor Projection Function

Our objective here is to improve on a baseline distance measure that partially confuses some pairs of points that should be kept apart (the nearest neighbor negatives pairs) with those that should be matched (the positive pairs), as shown in figure 2(b). There is a danger in learning a projection using *only* these training points that are confused in the original descriptor space: although we might learn a function to bring these points closer together, the projection might (especially if it is non-linear) "draw in" other points so that a particular pair of points are no longer nearest neighbours. Being a nearest neighbour explicitly depends on all other points in the space, so great care must be exercised when ignoring other points.

Here, we aim to overcome these problems by incorporating the distances between a large set of random point pairs directly into our cost function. These

Fig. 1. Gathering training point pairs. Three groups of point pairs are shown: (a) inliers to an affine homography found using RANSAC (positives); (b) outliers which are nevertheless nearest neighbors in SIFT space (nnN); and (c) random pairs of points which are usually distant in descriptor space (ranN).

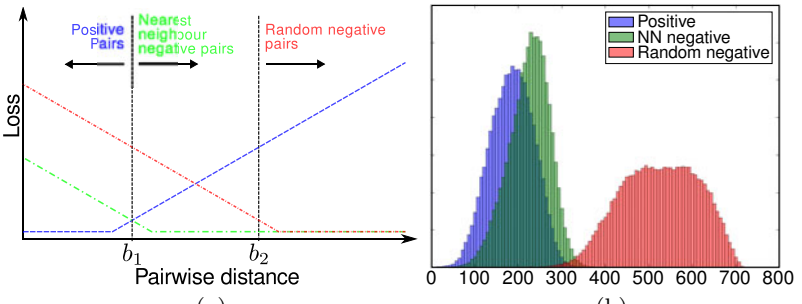

Fig. 2. Multiple margins. (a) Schematic of the multiple margin loss functions. This encourages the ordering on point pairs to be satisfied as per Equation 1. (b) Histograms of the raw 128-D SIFT distances for the three types of point pairs.

are precisely the pairs which can "crowd in" and tend to reduce the precision of clusters during vocabulary building if they are not explicitly considered. This effect has previously been ignored. It will be shown that, if this third set (the random negatives) is not explicitly considered, then a learnt mapping can reduce the confusion between positive and NN negative training pairs, but this simultaneously reduces the distance between random negative point pairs, leading to increased confusion. The solution we propose here is to add an additional loss function to prevent this confusion (and we quantify its benefit in Section 5).

More formally, given a set of positive training pairs $\mathbf{P}$, NN negative training pairs **nnN**, and random negatives **ranN**, our aim is to learn a projection function $T : \mathbb{R}^D \rightarrow \mathbb{R}^M$, where D is the dimension of the original descriptor space (e.g. $D = 128$ for SIFT) and M is the dimension of the projected descriptor, such that:

$$d(T(p_i), T(p_j)) < d(T(p_k), T(p_l)) \text{ and } d(T(p_i), T(p_j)) \ll d(T(p_m), T(p_n)) \quad (1)$$

for $p_i, p_j \in \mathbf{P}$, $p_k, p_l \in \mathbf{nnN}$ and $p_m, p_n \in \mathbf{ranN}$.

In practice, it is not possible to fully separate these pairwise distances because of noise in the training data and restricted model complexity, so instead a margin based approach will be used which encourages the distance between the three classes of point pairs to separate without enforcing the distance ordering as a hard constraint. The loss function for this situation is illustrated in Figure 2(a). The first margin aims to separate the positive and NN negative point pairs confused by SIFT in the original space. The second margin applies a force to the random negatives to keep them distant from the positive pairs – ideally the overlap in histograms between the positive and random negative point pairs should be small. This motivates learning the projection function by minimizing the cost function:

$$\begin{aligned} f(\lambda, W) = &\sum_{x,y \in P} \mathcal{L}\big(b_1 - d_W(x,y)\big) + \sum_{x,y \in nnN} \mathcal{L}\big(d_W(x,y) - b_1\big) \\ &+ \sum_{x,y \in ranN} \mathcal{L}\big(d_W(x,y) - b_2\big) + \frac{\lambda}{2}\|W\|^2 \end{aligned} \quad (2)$$

where $\mathcal{L}(z) = \log(1+\exp(-z))$ is the logistic-loss, a smooth approximation to the hinge loss which is more suitable for learning with gradient-based optimization, $d_W(x,y) = \|T(x;W) - T(y;W)\|_2$ is the standard Euclidean distance between the projected points, and W are the parameters of the projection function T.

The first three terms in (2) give the loss for the three different margins used, and the fourth is a regularization term, controlled by λ, which is used to limit the model complexity and stop over-fitting on the training data. b_1 and b_2 are the positions of the left-hand and right-hand margin biases in projected distance space. $f(\lambda, W)$ can be differentiated w.r.t. W by repeated application of the chain rule provided T is also differentiable. The absolute values of b_1 and b_2 are unimportant due to the scaling freedom in the projection functions – it is the ratio b_1/b_2 which is important.

4.1 Projection Function Models

We consider two different forms for the projection function T: a linear model of the form Wx; and a non-linear form for T based on a deep belief network (DBN). For the linear model, the projection function T is parameterized as $T(x;W) = Wx$, with derivative $\frac{\partial T_i}{\partial W_{ij}} = x_j$, where W is a real valued $D' \times 128$ matrix and so projects x linearly to a D'-dimensional space. This is equivalent to learning a Mahalanobis matrix $M = W^\top W$, therefore the linear model is equivalent in power to that used in [9,10,16]. Because W is real valued, M is positive semi-definite, and by learning W directly one can avoid the complications of adding semi-definiteness constraints into the learning routine. Though the projection function T is linear, the cost function (2) is not convex in W due to the square roots in $d(\cdot)$. However, previous work [13,23] has shown that, in

practice, optimizing this cost over W does not lead to serious problems with poor local minima.

For the non-linear model the projection function is based on a DBN [24] using a series of restricted Boltzmann machines (RBM). In this case W contains the projection parameters and biases for all the layers of the DBN. For one hidden layer, the projection function is of the form:

$$T(x; W_1, W_2, W_3, h_0, h_1, h_2) = W_3\sigma(W_2\sigma(W_1\sigma(x + h_0) + h_1) + h_2)$$

where σ is an element-wise logistic sigmoid function, W_i are matrices and h_i are column vectors (the h_i act as per-layer biases for the transformation). For a DBN projecting a 128-D SIFT descriptor to a 32-D descriptor with a single hidden layer of size 384-D, the number of parameters is $= 128 \times 384 + 384 \times 384 + 384 \times 32 + 128 + 384 + 384 = 209,792$. We adopt a DBN architecture because we expect non-linearities to allow the distance function to adapt itself depending on the statistics of the local neighborhood of the features being considered, and so improve the separation in distances between matching and non-matching point pairs. While a kernel method might be thought of as a natural alternative mechanism to introduce non-linearity, this would rule out the direct mapping of descriptors that we seek. DBNs have previously proven successful for distance learning in simple vision tasks, such as handwritten digit classification [13].

One potential problem with DBNs is that again (2) is not a convex function of the parameters W. Nevertheless, with a large amount of training data and good stochastic learning routines (see below), we find solutions which empirically seem to generalize well to unseen data.

4.2 Optimization

The task is to minimize the cost function, $f(\lambda, W)$, w.r.t. to the parameterized weights, W. In this work we use stochastic gradient descent (SGD) methods to optimize the loss function, for two main reasons. First, stochastic gradient methods scale linearly with the number of training points and have constant memory requirements. This makes them attractive in online learning or when the amount of training data is very large as in our case—here we use 3M training pairs, though more could easily be generated. Second, although stochastic gradient methods may require a large number of steps to converge, they often learn models which generalize well to unseen data [25].

SGD incrementally minimizes a cost function f by examining just a few data points at a time. If $f(X, W)$ is the function to minimize, and X is the data, the SGD update is:

$$W_{t+1} = W_t - \Theta_t \nabla_w f(X_{m..n}, W_t)$$

The parameter vector W is updated according to the negative gradient of the cost computed on just a few examples $X_{m..n}$. Θ_t is a learning rate which should decrease over time to ensure convergence. Here, we use "mini-batches" of 200 point pairs (with labels positive, NN negative, and random negative) per parameter update step. In each mini-batch there are about equal numbers of the

three types of point pairs. In practice SGD can converge slowly so, to speed up convergence, we use a pseudo second order method known as Stochastic Meta Descent (SMD) [26]. SMD uses a per-parameter learning rate based on an approximation of the local curvature. One difference from the method used in [26] is that here, we estimate the Hessian-vector product using finite-differences: $Hv \approx (\nabla f(W_t + \epsilon v) - \nabla f(W_t))/\epsilon$, rather than using an analytical approach.

4.3 DBN Implementation Details

The weights in the model are initialized using a generative procedure that proceeds layer-by-layer, optimizing weights using contrastive divergence [24]. This performs an initialization that empirically speeds up convergence of the subsequent discriminative training. The generative training is run layer-by-layer for one pass of the training data and takes around 30 minutes on a modern processor.

After this initialization, W is learnt discriminatively by optimizing (2): each point pair of a mini-batch is pushed through the network to give the transformed descriptors. The differentiable cost function (2) is then used to compute the gradient on the output layer based on all the points in the minibatch. Back-propagation is used to compute the gradient for the other layers in the network.

Once the gradient has been computed for all the DBN weights and hidden biases, the parameters W are updated using SMD. This is done once per mini-batch of points for a number of iterations over the dataset. For a DBN with one hidden layer of size 384-D projecting to 32-D, training one iteration of 3M point pairs takes just under 35 minutes on a single core of a modern processor. Training is performed for 50 iterations over the training data – after this we see rapidly diminishing returns.

Table 2. Comparison of several different retrieval methods. The results for the proposed methods are shown under the "Learnt" descriptor. The results for the baseline and raw-matching methods are duplicated from table 1 for completeness. DNF[1]: RS4 is too slow to be run on this dataset.

	Descriptor	Notes	Descriptor size	Dataset	mAP		
					RS1	RS2	RS4
(i)	SIFT		128	Oxford	0.613	0.647	0.755
(ii)	Learnt	Linear	32	Oxford	0.599	0.634	
(iii)	Learnt	Linear	64	Oxford	0.636	0.665	
(iv)	Learnt	Non-linear	24	Oxford	0.606	0.649	
(v)	Learnt	Non-linear	32	Oxford	0.644	0.681	
(vi)	Learnt	Non-linear	64	Oxford	**0.662**	**0.707**	
(vii)	SIFT		128	Paris	0.655	0.669	0.683
(viii)	Learnt	Non-linear	32	Paris	0.669	0.680	
(ix)	Learnt	Non-linear	64	Paris	**0.678**	**0.689**	
(x)	SIFT		128	Oxford-100K	0.490	0.541	DNF[1]
(xi)	Learnt	Non-linear	32	Oxford-100K	0.524	0.592	
(xii)	Learnt	Non-linear	64	Oxford-100K	**0.541**	**0.615**	

5 Results

Our objective is to reduce the *mAP-gap*, and so our principal evaluation measure will be the mAP of the retrieval system. However, as mAP is computed after many steps of processing (and also involves a weighting function, tf-idf, in two of the retrieval systems), in some cases we also show a simpler measure which is the *cluster true positive rate* (CTPR): this is simply the proportion of true positive validation pairs which cluster to the same visual word – for a fixed vocabulary size it is a measure of the recall of each word, and is closer to the informal goal stated in the introduction. In practice the two measures are closely correlated. In the following we learn the projection function on the Oxford dataset. Where results are stated with error bars, these are computed as a standard deviation over 3 vocabularies learnt from different initializations of the k-means clustering, with $k = 500,000$. For the baseline system RS1, CTPR is 0.336 ± 0.005. The regularization parameter, λ, is optimized on the validation set. Other than the generalization experiments, all results are produced on the Oxford dataset.

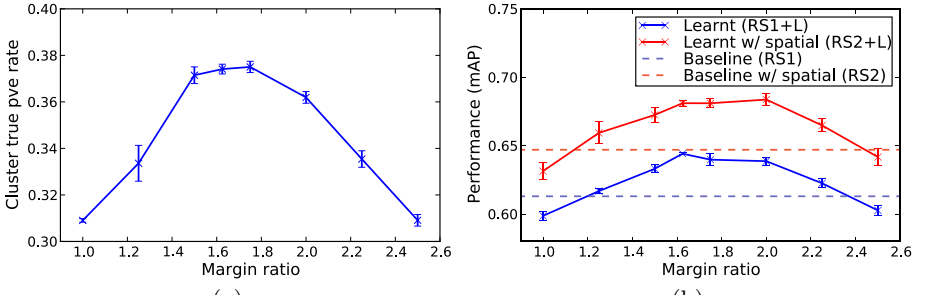

Fig. 3. Adjusting the margin ratio. The (a) CTPR, and (b) mAP retrieval performance as a function of the margin ratio, b_2/b_1 (see Equation 2). The hidden layer dimension and final dimension are 384 and 32 respectively. "+L" indicates that a learnt model is used.

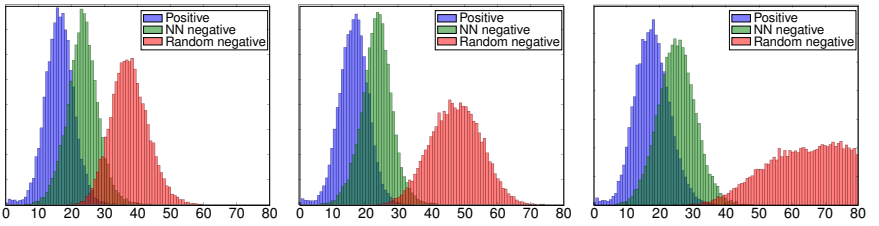

Fig. 4. Histograms of pair distances. The distance histograms of validation pairs after training for (a) $b_2/b_1 = 1.0$ (b) $b_2/b_1 = 1.6$ (c) $b_2/b_1 = 2.5$. The histograms show the positive, NN negative and random negative point pairs.

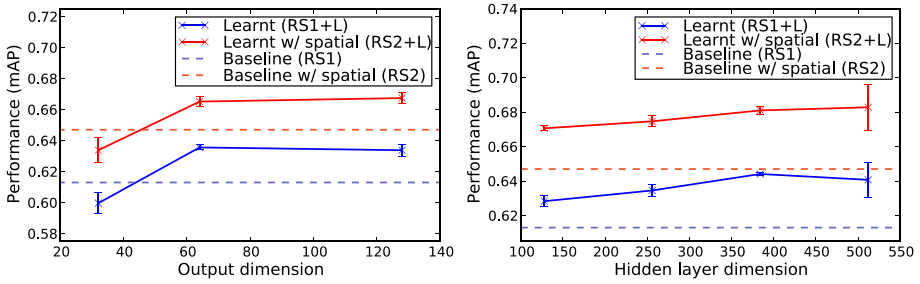

Fig. 5. (a) **Linear model**: mAP performance as the final dimension D' is varied. (b) **Non-linear model**: mAP performance as the hidden layer dimension is varied. The output dimension is fixed to 32.

Choosing the margin ratio: Figure 3 examines the retrieval performance as a function of the margin ratio b_2/b_1 for a non-linear model with one hidden layer of size 384 projecting down to 32-D. This ratio controls the extent to which the random negative pairs should be separated from the positive pairs. At $b_2/b_1 = 1.0$, both margins are the same, which mimics previous methods that use just two types of point pairs: if the ratio is set too low, the random negative pairs start to be clustered with the positive pairs; if it is set too high then the learning algorithm focuses all its attention on separating the random negatives and isn't able to separate the positive and NN negative pairs. Distance histograms for different margin ratios are shown in Figure 4. As the ratio is increased, there is a peak in performance between 1.6 and 1.7. In all subsequent experiments, this ratio is set to 1.6 with $b_1 = 20.0$. These results clearly demonstrate the value of considering both sets of negative point pairs.

Linear model: Results for the linear model are given in Table 2 and are shown in Figure 5(a). Performance increases only up to 64-D and then plateaus. At 64-D the performance without spatial re-ranking is 0.636 ± 0.002, an improvement of 3.4% over RS1. With spatial re-ranking the mAP is 0.665 ± 0.003, an improvement of 1.8% over RS2. Therefore, a learned linear projection leads to a slight but significant performance improvement, and we can reduce the dimensionality of the original descriptors by using this linear projection with no degradation in performance.

We compare to the linear discriminant method of Hua *et al.* [16], using a local implementation of their algorithm on our data. For this method, we used the ranN pairs as the negatives for training (performance was worse when nnN pairs were used as the negatives). Using 1M positive and 1M random negative pairs, reducing the output dimension to 32-D, gives a performance of 0.585 without spatial re-ranking; and 0.625 with spatial re-ranking. This is slightly worse than our linear results which gives an mAP of 0.600 and 0.634 respectively. The difference in performance can be explained by our use of a different margin-based cost function and the consideration of both the nnN and ranN point pairs.

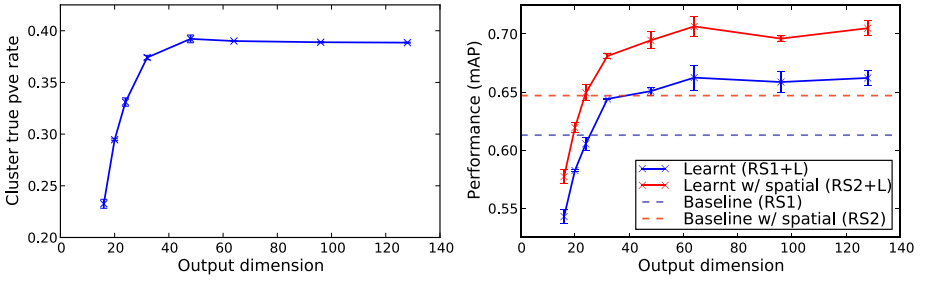

Fig. 6. Non-linear model: variation with output dimension for (a) CTPR and (b) mAP performance. The hidden layer dimension is fixed at 384. (a) the CTPR rate increases as the output dimension increases to 48, then flattens. Similarly, (b) shows that at $D = 32$ the projected descriptors without spatial re-ranking (system RS1) achieve performance equal to the original descriptors with spatial re-ranking (system RS2). Performance continues to increase to $D = 64$ and then plateaus. Spatial reranking on the projected descriptors beats RS2 by 5.9% (0.647 to 0.706 mAP).

Non-linear model: Figure 5(b) shows the results of adjusting the dimension of the hidden layer. The hidden layer dimension only affects the time taken to train the model and project the features into the new space and doesn't affect storage requirements or clustering/assignment speed. From the figure, one can see that retrieval performance increases up to around 384-D before leveling off, and for subsequent experiments we fix the hidden layer dimension at 384-D.

Figure 6 and Table 2 show the effect on performance of adjusting the output dimension of the projection function. The learnt descriptor attains the same performance as SIFT at just over 24-D, a saving in storage of over 5 times. At 32-D, but without spatial re-ranking, the learnt descriptor performs as well as using SIFT with spatial re-ranking (RS2). After 32-D, the performance gains start to level off, but still improve up to 64-D. Using a 64-D descriptor with spatial re-ranking beats RS2 by 5.9% (0.647 to 0.706 mAP). Note also that the non-linear model greatly improves performance over the linear model (0.665 to 0.706 mAP). The non-linear model substantially closes the *mAP-gap* and brings the quantized visual word method much closer in performance to the raw SIFT method. This is achieved with no increase in query times or index size.

Generalization: Here we examine the generalization of the learnt descriptor to the held-out Paris dataset. Spatial re-ranking on the raw SIFT descriptors for Paris gave a much lower performance gain than for Oxford, so there is less that our method can do. Nevertheless, we still increase performance over the baseline method. Our 64-D descriptor, learnt on Oxford, gives a score without spatial re-ranking of 0.678 (compared to 0.655) and with spatial re-ranking gives 0.689 (compared to 0.669), slightly exceeding the performance from using the raw descriptors of 0.683. This is principally due to the many non-planar queries present in the Paris dataset which is challenging for the RS4 method.

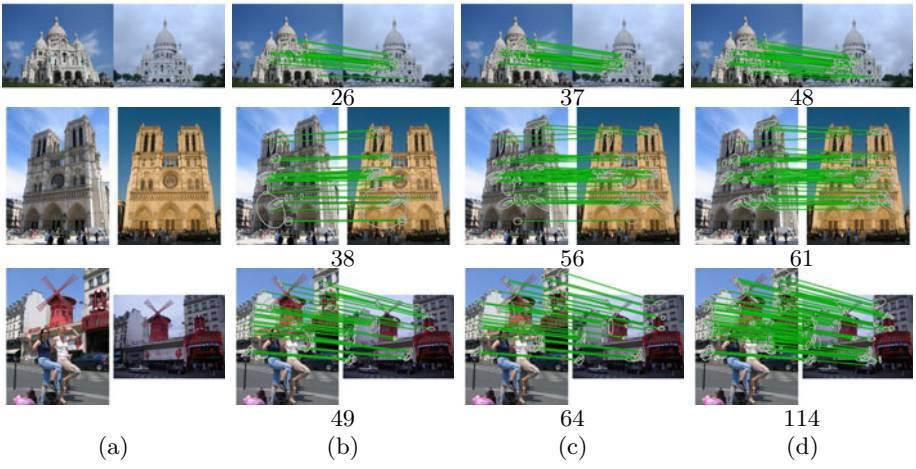

Fig. 7. Qualitative examples on the Paris dataset. Demonstrating the improvements in matching using our quantized learnt descriptor. The number of inliers found are listed beneath each image pair. The four columns are: (a) the original image pair; (b) matches found by the baseline visual words; (c) matches found by our learnt visual words; (d) matches found by the raw SIFT matching method.

In figure 7, we qualitatively examine the spatially verified inliers between some image pairs for the baseline method, our quantized learnt method and the raw descriptor method. The quantized learnt descriptor gives more inliers to the computed homography and closes the gap on the raw matching method.

Table 2(x)-(xii) gives retrieval results for Oxford combined with a large set of 100K images [3]. The additional images do not contain the landmarks and so act as "distractors" for retrieval. Using the quantized learnt descriptor with D=64 and spatial re-ranking gives a substantial boost in performance from 0.541 to 0.615. Again this illustrates that the learnt projection function is able to generalize to other datasets, whilst still boosting retrieval performance.

6 Conclusion

We have shown that, by transforming descriptors prior to clustering, we can boost performance considerably over a baseline retrieval method and can produce results using visual words alone that are as good as the baseline method combined with spatial re-ranking. We have considerably closed the performance gap between the raw SIFT matching method and the much faster quantized retrieval method for both datasets considered here. This performance boost comes at zero runtime cost (though some offline cost) and with reduced data storage.

Since the descriptors are transformed *before* quantization, they can easily be used in conjunction with other recent works that have improved performance over a raw bag of visual words approach, such as [27,28].

We have illustrated the method for SIFT and for two types of projection functions, but clearly the framework of automatically generating training data and learning the projection function through optimization of (2) could be applied to other descriptors, e.g. the DAISY descriptor of [29] or even directly to image patches.

Acknowledgements. We are grateful for financial support from the EPSRC, the Royal Academy of Engineering, Microsoft, ERC grant VisRec no. 228180, ANR project HFIBMR (ANR-07-BLAN-0331-01) and the MSR-INRIA laboratory.

References

1. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008)
2. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proc. CVPR (2006)
3. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proc. CVPR (2007)
4. Sivic, J., Zisserman, A.: Video Google: A Text Retrieval Approach to Object Matching in Videos. In: Proc. ICCV (2003)
5. Boiman, O., Shechtman, E., Irani, M.: In defence of nearest-neighbor based image classification. In: Proc. CVPR (2008)
6. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: Proc. CVPR (2008)
7. van Gemert, J., Geusebroek, J.M., Veenman, C., Smeulders, A.: Kernel codebooks for scene categorization. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 696–709. Springer, Heidelberg (2008)
8. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: Automatic query expansion with a generative feature model for object retrieval. In: Proc. ICCV (2007)
9. Schultz, M., Joachims, T.: Learning a distance metric from relative comparisons. In: NIPS (2003)
10. Weinberger, K., Blitzer, J., Saul, L.: Distance metric learning for large margin nearest neighbor classification. In: NIPS (2005)
11. Kumar, P., Torr, P., Zisserman, A.: An invariant large margin nearest neighbour classifier. In: Proc. ICCV (2007)
12. Frome, A., Singer, Y., Sha, F., Malik, J.: Learning globally-consistent local distance functions for shape-based image retrieval and classification. In: Proc. ICCV (2007)
13. Salakhutdinov, R., Hinton, G.: Learning a nonlinear embedding by preserving class neighbourhood structure. In: AI and statistics (2007)
14. Mikolajczyk, K., Matas, J.: Improving descriptors for fast tree matching by optimal linear projection. In: Proc. ICCV (2007)
15. Ramanan, D., Baker, S.: Local distance functions: A taxonomy, new algorithms, and an evaluation. In: Proc. ICCV (2009)
16. Hua, G., Brown, M., Winder, S.: Discriminant embedding for local image descriptors. In: Proc. ICCV (2007)

17. Snavely, N., Seitz, S., Szeliski, R.: Photo tourism: exploring photo collections in 3D. In: Proc. ACM SIGGRAPH, pp. 835–846 (2006)
18. Lowe, D.: Object recognition from local scale-invariant features. In: Proc. ICCV (1999)
19. http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/
20. http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/
21. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. IJCV 1, 63–86 (2004)
22. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV 60, 91–110 (2004)
23. Guillamin, M., Verbeek, J., Schmid, C.: Is that you? Metric learning approaches for face identification. In: Proc. ICCV (2009)
24. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science 313, 504–507 (2006)
25. Bottou, L., Bousquet, O.: The tradeoffs of large scale learning. In: NIPS (2007)
26. Bray, M., Koller-Meier, E., Schraudolph, N.N., Van Gool, L.: Stochastic meta-descent for tracking articulated structures. In: Proc. CVPR (2004)
27. Chum, O., Perdoch, M., Matas, J.: Geometric min-hashing: finding a (thick) needle in a haystack. In: Proc. CVPR (2009)
28. Perdoch, M., Chum, O., Matas, J.: Efficient representation of local geometry for large scale object retrieval. In: Proc. CVPR (2009)
29. Winder, S., Hua, G., Brown, M.: Picking the best daisy. In: Proc. CVPR (2009)