Towards large-scale geometry indexing by feature selection

Giorgos Tolias, Yannis Kalantidis, Yannis Avrithis, Stefanos Kollias

National Technical University of Athens
Iroon Polytexneiou 9, Zografou, Greece

Abstract

We present a new approach to image indexing and retrieval, which integrates appearance with global image geometry in the indexing process, while enjoying robustness against viewpoint change, photometric variations, occlusion, and background clutter. We exploit shape parameters of local features to estimate image alignment via a single correspondence. Then, for each feature, we construct a sparse spatial map of all remaining features, encoding their normalized position and appearance, typically vector quantized to visual word. An image is represented by a collection of such *feature maps* and RANSAC-like matching is reduced to a number of set intersections. The required index space is still quadratic in the number of features. To make it linear, we propose a novel feature selection model tailored to our feature map representation, replacing our earlier hashing approach. The resulting index space is comparable to baseline bag-of-words, scaling up to one million images while outperforming the state of the art on three publicly available datasets. To our knowledge, this is the first geometry indexing method to dispense with spatial verification at this scale, bringing query times down to milliseconds.

Keywords: image retrieval, geometry indexing, feature maps, feature selection, spatial matching

1. Introduction

Geometry is essential in many problems of computer vision like feature correspondence, image registration, wide baseline stereo matching, object recognition, and retrieval. And it has been more so in early years when features were non-discriminative, *e.g.* points. With the advent of more discriminative features and descriptors, discarding geometry altogether has been an "easy" way to deal with viewpoint change and occlusion. The success of the *bag-of-words* (BoW) model, largely due to its very low computational cost, has come as quite a surprise to many, for instance in the seminal work of Sivic and Zisserman [1].

In order to boost performance at large scale however, geometry is still essential. Even if weaker or stronger geometric models are feasible in tasks like registration or recognition, this is clearly not the case for image retrieval. State of the art approaches are still based mostly on appearance in the *filtering* stage, while geometric or spatial constraints typically come as a second, *re-ranking* or *spatial verification* stage. The former is carried out by an inverted file and is non-exhaustive; only a small percentage of the database is accessed during scoring. The latter is practically the most time consuming task. The need for including spatial information in the index itself is identified *e.g.* in Philbin *et al.* [2].

Even in more recent work, such indexing has only been achieved in the form of weak geometric constraints as in Jegou *et al.* [3], local geometry, as in Chum *et al.* [4], or representations that are not fully invariant to geometric transformations, such as Wu *et al.* [5]; a detailed account is provided in section 2. On the other hand, global geometry indexing is at least as old as *geometric hashing* by Lamdan and Wolfson [6], where features are non-discriminative. To our knowledge, our earlier method of [7] has been the first to index appearance and global geometry under invariance, but index space requirements have limited it to 50K images. We attempt here a solution towards large scale image retrieval.

One of our starting points is [2] where spatial matching is performed as a special case of RANSAC [8]. Shape parameters of local features are used to generate each hypothesis using a single feature correspondence. The idea stems from Lowe [9] but has been studied in more depth only recently, *e.g.* in Köser *et al.* [10]. We go a step further and for each feature we encode the normalized position and appearance of all remaining features in a sparse histogram that we call a *feature map*. One may think of feature map as a local descriptor that globally describes the entire image in a local coordinate frame. This has strong connections to *shape context* [11], geometric hashing [6], and previous work which is discussed in section 2. Under this novel representation, spatial matching is further reduced to a collection of inner product or set intersection operations.

The feature map representation is quadratic in the number of features. In [7], *feature map hashing* (FMH) is used to make it linear. A locality sensitive hashing (LSH) framework is adopted and min-wise independent permutations [12] are extended to collections of sets to derive a similarity measure for feature map collections. The images returned by the inverted file used in the filtering stage are not only ranked according to similarity, but are also associated with a rough estimate of the relevant geometric transformation, as in [4]. Full spatial matching is thus reduced to a single local optimization step of LO-RANSAC [13]. For the same processing time, the number of images we verify

is effectively increased by an order of magnitude. The retrieval performance of FMH without re-ranking is superior to bag-of-words with re-ranking, on three publicly available datasets.

Despite the hashing framework applied in [7], the memory requirements of FMH are still high enough to prohibit its use for datasets in the order of 10^6. Hence, in the present work, we choose to substitute hashing with a novel *feature selection* model. Through an automated and unsupervised learning process, we select and index only the most informative features for each image and thus keep memory requirements comparable to the baseline bag-of-words model. We experiment on three publicly available retrieval datasets of size up to 10^6 with excellent performance. Most importantly, we find that spatial verification and re-ranking is no longer needed as the database gets larger. Hence queries can be restricted to the filtering stage alone with query times in milliseconds.

Following the related work in section 2, section 3 provides a background on a number of related problems in shape matching, feature correspondence and indexing. Section 4 derives our feature map representation along with the associated matching process. These first two sections provide a detailed account of the work first introduced in our earlier work [7]. Feature selection, introduced here, is presented in section 5 and implementation details in section 6. Experiments and discussion are provided in sections 7 and 8, respectively.

2. Related work

A more detailed account of related bibliography is presented here by topic. In each case, limitations and differences from our work are highlighted.

2.1. Local coordinate frames

Normalizing a set of planar points in a reference coordinate frame defined by a number of reference points is quite common. Examples are *Bookstein* and *Kendall coordinates* [14], where the first two points are arbitrarily chosen as reference, effectively removing up to similarity transformations. To deal with point correspondence and outliers, *geometric hashing* [6] does the same for every possible combination of reference points in the original set. Larger sets of reference points are also considered to remove more complex transformations, *e.g.* 3-point combinations for affine. Positions are quantized as in our work. The complexity is such that it is typically applied to a small number of prototypes for recognition.

A single feature is enough to define each reference coordinate frame in our work, so we can effectively decompose all images in the database and the query image at query time as well. Chum and Matas [15] also implement geometric hashing with a single feature defining each reference frame, but for each feature they encode local shape rather than appearance. Our representation is different in that it takes local shape into account only when rectifying—on the other hand, we integrate appearance in our joint codebook, rendering a feature map very discriminative.

2.2. Feature context

A feature map, seen as a local descriptor, is a concept very close to *shape context* [11], where the position of all neighboring points is quantized in a log-polar map. The local coordinate frame is only normalized using global information, so that invariance is lost under partial matching. The *proximity distribution kernel* [16] is also quite relevant. For each pair of visual words in the codebook, it records the proximity distribution of relevant feature pairs in an image. It is less discriminative because feature correspondence and exact position is lost; it is also not invariant to scale change or affine transformations. Assuming only one reference frame, *spatial pyramid matching* [17] is related, in the sense that appearance and position is jointly matched. Geometric invariance is lost again.

Numerous variants of local feature context models exist. Berg *et al.* [18] minimize a total cost based on descriptors, deriving a set of relaxed linear programming problems. Jiang and Yu [19] work on feature positions instead of descriptors, deriving a single linear programming problem. Leordeanu and Hebert [20] cast the problem as spectral clustering and derive a greedy algorithm that involves the principal eigenvector of a sparse affinity matrix. The above methods are most useful in object recognition and are too costly for indexing. Using a codebook however, we will see how context can help in a very simplified form.

2.3. Indexing appearance and geometry

As discussed in the introduction, Philbin *et al.* [2] approximate RANSAC based on single correspondence hypotheses in the second, ranking stage of retrieval. Even further, Perdoch *et al.* [21] vector-quantize local shapes for memory efficiency, without significant loss in precision. We rather precompute rectified feature maps and integrate them in the index. Spatial matching is now performed in the first, filtering stage, which is orders of magnitude faster.

Chum *et al.* [4] make a similar attempt for local geometry via *geometric min-hashing*. They construct sketches where one reference feature is chosen uniformly at random among features of unique visual word (similarly to our *origins*) and a number of other features are chosen in the neighborhood of the reference one (similarly to our *rectified features*). Exact position is lost and geometry is imposed on local neighborhoods only, with application to small object mining. On the other hand, the latter application resembles our mining process for origin selection.

Jegou *et al.* [3] make another attempt to integrate geometry in the index via *weak geometric consistency* (WGC). They extend bag-of-words (BoW) voting by separately recoding log-scale and orientation differences between features. Local shape is thus taken into account, though it is not possible to extend to affine transformations; feature position is lost altogether. WGC is considered in our experiments for comparisons along with Baseline BoW with different re-ranking options; see section 7.

Another recent work is [22], where Zhang *et al.* index both the visual word and the quantized location of each feature. Voting is carried out in the space of relative translations as opposed to the relative log-scale and orientations of [3]. A coarse spatial

grid provides robustness, but apparently this approach remains invariant only with respect to translation. The space of complete relative transformations is employed in our *Hough pyramid matching* [23] for voting, but it has not been possible to extend this matching scheme to indexing.

Wu *et al.* [5] group local features that are detected inside MSER regions. They index the geometry of the resulting *bundled* features by encoding intra-bundle feature orderings. Matching is performed between bundles and similarity is penalized by the number of incorrectly ordered feature correspondences. Once more, only local geometry is taken into account. Moreover, orderings are along the horizontal and vertical direction and thus not rotation invariant. Zhou *et al.* [24] use a binarization scheme on a similar representation, with similar limitations.

Finally, Cao *et al.* [25] encode geometric information in terms of orderings as well. They extract a series of quantized linear and circular orderings of the image features using many different parameters, apply histogram calibration and equalization for invariance and then learn the best spatial configurations. However, relying on training images makes the approach impractical for large scale retrieval.

2.4. Feature Selection

Feature selection has recently become a popular way of reducing space requirements for image retrieval. Schindler *et al.* [26] and Li and Kosecka [27] have been among the first approaches, both for location recognition. Given a dense street-view geotagged database, Schindler *et al.* use the concept of information gain to select *informative* features, *i.e.* features which occur in all images of some specific location, but rarely or never elsewhere. Similarly, Li and Kosecka obtain an information content probability for each feature with respect to location identification.

Knopp *et al.* [28] also use a geo-tagged database and exploit the fact that photos taken at faraway locations should not match. They densely compute a local confusion score for image regions in a sliding window scheme and remove features inside regions with high confusion score for all database images. Gammeter *et al.* [29] start from image clusters extracted using spatial proximity and visual similarity and use feature matching statistics to estimate a bounding box around the foreground object of each image. Although they only index the features lying inside this bounding box, a high percentage (around 66%) is still kept.

All the aforementioned feature selection models are supervised and make use of location information. On the other hand, the model presented in this work is unsupervised. Closely related is the approach of Turcot and Lowe [30], who issue each database image as a query and select *useful* features, *i.e.* features that appear as inliers during spatial verification. Only inliers of the best hypothesis are taken into account, as opposed to our model that works on all hypotheses and selects the best for each feature independently. Most importantly, we apply a different selection strategy for features used to generate hypotheses and features that are just matched as inliers. The feature selection scheme of [30] is another method that we compare to in our experiments; see section 7.

Tolias *et al.* [31] have recently proposed a feature selection scheme for *single* images, *i.e.* images in the dataset that have no matching pair and therefore [30] cannot be applied. They geometrically match the image with itself and its mirrored counterpart in order to select repeating or symmetric feature configurations.

3. Background

We start by examining a number of simple models for matching sets of features that are based on geometry, appearance, or both. We observe how these models can provide solutions for alignment, correspondence, and outliers and derive a single model that we will attempt to further simplify in the following sections. Throughout this work we assume that an image is represented by a set of local features. For each feature x in image X, we denote by $p(x) \in \mathbb{R}^2$ its position in the image, and by $d(x)$ its descriptor in some arbitrary descriptor space, encoding its appearance. In the following models, features may be equipped with position, descriptor, or both.

3.1. Shape matching

Let X, Y be two images. The features are taken as non-discriminative, that is, only their positions are known. Assume for the moment that $|X| = |Y|$ and that there is a known one-to-one mapping $\pi : X \to Y$. In the statistical theory of shape [14], one of the most well studied problems is the estimation of the optimal geometric transformation aligning the two sets

$$S_T(X, Y; r) = \max_{B,t} \sum_{x \in X} r(Bp(x) + t, p(\pi(x))), \qquad (1)$$

where $r : \mathbb{R}^2 \times \mathbb{R}^2 \to [0, 1]$ is an arbitrary spatial similarity (proximity) measure, assumed a non-increasing function of some distance metric, and $B \in \mathbb{R}^{2\times2}, t \in \mathbb{R}^2$ are respectively the scale/rotation/skew and translation component of an affine transform. In this context, the points are referred to as *landmarks* and may as well be manually specified by experts. This problem never appears as such in our case, due to unknown feature correspondence and outliers.

3.2. Feature correspondence

We now drop the known mapping assumption and rather assume discriminative features specified only by their descriptors. We also drop assumption $|X| = |Y|$. Ignoring positions for now, the following *generalized assignment* problem can deal with unknown correspondence and, partially, outliers:

$$S_A(X, Y; s) = \max_{a} \sum_{x \in X} \sum_{y \in Y} a_{x,y} s(x, y) \qquad (2)$$

$$\text{s.t.} \sum_{x \in X} a_{x,y} \leq 1, \quad \forall y \in Y \qquad (3)$$

$$\sum_{y \in Y} a_{x,y} \leq 1, \quad \forall x \in X \qquad (4)$$

$$a_{x,y} \in \{0, 1\}, \quad \forall x \in X, y \in Y \qquad (5)$$

where $s : X \times Y \to [0, 1]$ is an arbitrary similarity measure in the descriptor space, and a may be seen either as a $|X| \times |Y|$ zero-one matrix or a mapping $a : X \times Y \to \{0, 1\}$. Through a, each point $x \in X$ can then be assigned up to one point $y \in Y$, and vice versa. Note that as in Dong *et al.* [32], we formulate the problem as similarity maximization whereas most authors minimize distance; the two are equivalent.

Werman *et al.* [33] were among the first to study this problem in computer vision. Rubner *et al.* [34] generalized it from sets to distributions and from correspondence to flow, defining the *earth mover distance* (EMD) as a solution to a *transportation* problem. Several hierarchical approximations and greedy algorithms have been studied, for instance the work of Grauman and Darrell [35] on the *pyramid match kernel* (PMK).

3.3. Bag-of-words

Let $\mathcal{V}$ be a visual *vocabulary* or *codebook*, *i.e.* a finite subset of the descriptor space with $|\mathcal{V}| = k_v$ elements or *visual words*, derived *e.g.* by clustering or vector quantization on a training set. Given image X, let $v(x) \in \mathcal{V}$ be the quantized version of descriptor $d(x)$ for each feature $x \in X$. One may then construct a *bag-of-words* representation or *histogram* of X over $\mathcal{V}$ by letting $H_b(X) = \{x \in X : v(x) = b\}$ be the features of X mapped to visual word (bin) $b \in \mathcal{V}$, and $h_b(X) = |H_b(X)|/|X|$ their frequency. The histogram, denoted as $h_{\mathcal{V}}(X)$, is a vector in $\mathbb{R}^{|\mathcal{V}|}$, and may be represented as

$$h_{\mathcal{V}}(X) = \sum_{b \in \mathcal{V}} h_b(X)\mathbf{e}_b = \frac{1}{|X|} \sum_{x \in X} \mathbf{e}_{v(x)}, \tag{6}$$

where $\{\mathbf{e}_b \in \mathbb{R}^{|\mathcal{V}|} : b \in \mathcal{V}\}$ is the standard basis of $\mathbb{R}^{|\mathcal{V}|}$. It is then natural to define the discrete visual similarity measure

$$s_{\mathcal{V}}(x, y) = \delta_{v(x),v(y)} = \begin{cases} 1, & \text{if } v(x) = v(y) \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

so that features x, y in two images X, Y are similar iff assigned the same visual word. It is straightforward to see [32] that

$$S_A(X, Y; s_{\mathcal{V}}) = \sum_{b \in \mathcal{V}} \min(h_b(X), h_b(Y)), \tag{8}$$

that is, the *histogram intersection* of the bag-of-words representations of X and Y. In an analogous way, we may replace the *one-to-one* matching scheme of (2)-(5) with an *one-to-many* voting scheme

$$S_M(X, Y; s) = \sum_{x \in X} \sum_{y \in Y} s(x, y) \tag{9}$$

and confirm that the similarity measure of the histograms is equal to an *inner product* [3],

$$S_M(X, Y; s_{\mathcal{V}}) = \sum_{b \in \mathcal{V}} h_b(X)h_b(Y) = \langle h_{\mathcal{V}}(X), h_{\mathcal{V}}(Y) \rangle. \tag{10}$$

Since histograms are normalized, this is the well-known *cosine similarity* measure used in information retrieval. Either way, combined *e.g.* with an *inverted file* structure to exploit sparsity, this is a simple and fast method that is very common in the filtering stage of retrieval.

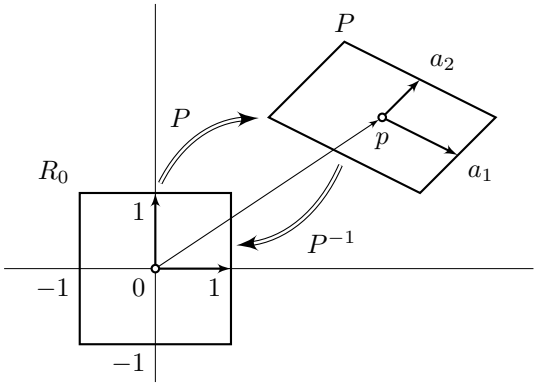


Figure 1: Patch P of a local feature, centered at position p. P may also be seen as an affine transform; the patch may then be rectified to R_0 via the inverse transform, P^{-1}.

3.4. Towards RANSAC

One-to-many matching may give unexpected results according to our perception of similarity [34]. It is however easier to estimate, especially when using a codebook. Following (9), let us start with a set of *tentative correspondences*, either defined in a nearest neighbor sense, or by

$$\mathcal{C}(X, Y; s) = \{(x, y) \in X \times Y : s(x, y) > \epsilon_s\}. \tag{11}$$

When we do use a codebook that is large enough, tentative correspondences (11) do not differ much from the one-to-one scheme (2)-(5) .

Given a specific set of correspondences $\mathcal{C}$, we may return to alignment model (1) and maximize w.r.t. affine transform (B, t) over a finite set of *hypotheses* $\mathcal{H}$:

$$S_R(X, Y; \mathcal{C}, r) = \max_{(B,t) \in \mathcal{H}} \sum_{(x,y) \in \mathcal{C}} r(Bp(x) + t, p(y)). \tag{12}$$

When hypotheses are selected at random following a specific strategy and spatial similarity is defined by a uniform kernel

$$r_\epsilon(p, q) = \begin{cases} 1, & \text{if } \|p - q\|_2 < \epsilon \\ 0, & \text{otherwise,} \end{cases} \tag{13}$$

for $p, q \in \mathbb{R}^2$ that just counts inliers, the above model is not too different from RANSAC. Given appropriate correspondences, it can jointly solve for alignment and outliers.

4. Feature maps

In studying a number of different models we have seen how quantizing descriptors or separating correspondence from alignment provide for significant simplification. We will use this insight to derive further simplification here.

4.1. Local patches

We assume here that each local feature x is additionally associated with an image patch $P(x)$, representing local shape apart

as well as position. Following Rothganger *et al.* [36], the patch is a parallelogram represented by matrix

$$P(x) = \begin{bmatrix} a_1(x) & a_2(x) & p(x) \\ 0 & 0 & 1 \end{bmatrix}, \tag{14}$$

where $p(x)$ is now the center of the patch and $a_1(x), a_2(x) \in \mathbb{R}^2$ are the vectors from $p(x)$ to the midpoints of the two sides, as in Figure 1. The *rectified* patch R_0 is represented by the identity matrix I_3 and is transformed to the patch via P, while the patch is rectified back to R_0 via P^{-1}. So P stands either for a patch or an affine transform. This formulation is equivalent to *local affine frames* [15].

4.2. Single correspondence hypotheses

Given a patch correspondence $P(x) \leftrightarrow P(y)$ between features x, y in images X, Y, the transform from one patch to the other is $P(y)P(x)^{-1}$. Lowe [9] has been among the first to observe that each correspondence may provide a transformation hypothesis, locally approximating the global, higher order transformation between the two images. Philbin *et al.* [2] exploit this fact to speed-up the re-ranking process. In fact, the set of hypotheses is now specified by the set of correspondences. Given a one-to-one matching scheme or a discriminative enough vocabulary, the latter are $O(|\mathcal{C}|)$ and we can enumerate them all:

$$\mathcal{H}(\mathcal{C}) = \{P(y)P(x)^{-1} : (x, y) \in \mathcal{C}\}. \tag{15}$$

In particular, we say that hypothesis $P(y)P(x)^{-1} \in \mathcal{H}(\mathcal{C})$ is *generated* by correspondence $(x, y) \in \mathcal{C}$. Matching two images then boils down to identifying the hypothesis with the largest support,

$$S_H(X, Y; \mathcal{C}, r) = \max_{A \in \mathcal{H}(\mathcal{C})} \sum_{(x,y)\in\mathcal{C}} r(A\mathbf{p}(x), \mathbf{p}(y)). \tag{16}$$

Here $\mathbf{p}(x) = [p(x)^{\mathrm{T}} 1]^{\mathrm{T}}$ denotes the position of x in *homogeneous coordinates*, that is, a 3×1 vector in projective space $\mathbb{P}^2$, while $A \in \mathbb{R}^{3\times3}$ is an affine transform that includes translation, unlike B in (12). Still, computation of (16) is quadratic in $|\mathcal{C}|$.

4.3. Feature set rectification

Instead of constructing transforms $P(y)P(x)^{-1}$ for all correspondences (x, y) and performing spatial matching at query time like Philbin *et al.* [2], we *extrapolate* each local transform to the entire image frame and rectify the entire set of features *in advance*. In particular, given a feature $\hat{x}$ in image X called an *origin*, we rectify all features $x \in X$ with respect to $\hat{x}$ as follows.

Let $p^{(\hat{x})}(x) \in \mathbb{R}^2$ be the Euclidean counterpart (the 2×1 vector with the first two elements) of $P^{-1}(\hat{x})\mathbf{p}(x)$. We then say that *feature x rectified with respect to $\hat{x}$* is a new feature $x^{(\hat{x})}$ with position $p(x^{(\hat{x})}) = p^{(\hat{x})}(x)$ and descriptor $d(x^{(\hat{x})}) = d(x)$. We also say that $x^{(\hat{x})}$ is a *rectified feature*. We do not associate $x^{(\hat{x})}$ with a patch; the role of $P(x)$ is restricted to using x as an origin. Finally, let $X^{(\hat{x})} = \{x^{(\hat{x})} : x \in X\}$ be the entire feature set X rectified with respect to $\hat{x}$. Figure 2 shows a random feature set, a transformed and distorted version, and the rectified

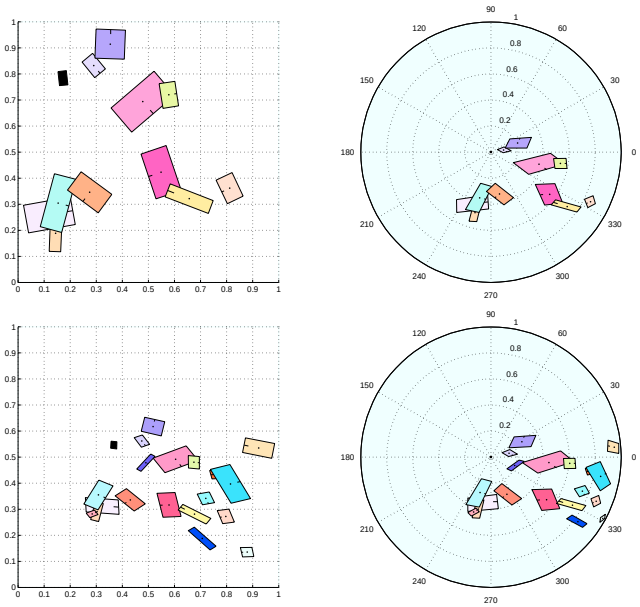


Figure 2: Top left: A random set of patches. Bottom left: The same set under affine transform, where patch position and local shape are contaminated with noise, and more patches are added. Right: Rectified counterparts; origins are the two black patches on the left. The polar grid specifies the spatial bins for feature maps, with $\tau = 0.95$, $k_\rho = 5$ and $k_\theta = 12$ (see section 6).

counterparts with their origins in correspondence—notice how the latter are roughly aligned.

Under this formulation, the same set of correspondences $\mathcal{C}$, obtained solely via descriptor matching, is used both for inlier counting and aligning:

$$\hat{S}_H(X, Y; \mathcal{C}, r) = \max_{(\hat{x},\hat{y})\in\mathcal{C}} \sum_{(x,y)\in\mathcal{C}} r(p(x^{(\hat{x})}), p(y^{(\hat{y})})) \tag{17}$$

$$= \max_{(\hat{x},\hat{y})\in\mathcal{C}} I(\mathcal{C}; \hat{x}, \hat{y}, r), \tag{18}$$

where $I(\cdot; \hat{x}, \hat{y}, \cdot)$ can be seen as the total count of inliers for hypothesis $(\hat{x}, \hat{y})$. This similarity measure is not the same as (16), but in fact it is more appropriate because it is measured in a *rectified coordinate frame* and is symmetric.

4.4. Quantization

Observe that unlike model (16), features are aligned in the summand of (17) due to rectification. This allows us to apply spatial quantization without sacrificing invariance. We adopt the visual codebook scheme of section 3 and further define a finite *spatial codebook* $\mathcal{U} \subseteq \mathbb{R}^2$ with $|\mathcal{U}| = k_u$ bins. Quantization can be uniform in this case. However, encoding all possible rectified positions in a finite set is not trivial and is discussed in section 6.

For any feature x, let $u(x)$ be the quantized version of its position $p(x)$, and $w(x) = (v(x), u(x))$ the corresponding joint visual-spatial bin. Further, define the *joint codebook* $\mathcal{W} = \mathcal{V} \times \mathcal{U}$ with $|\mathcal{W}| = k_v k_u = k$ bins. Then, for any feature set $\hat{X}$ rectified with respect to any origin, construct a *joint histogram* over $\mathcal{W}$ by letting $H_b(\hat{X}) = \{x \in \hat{X} : w(x) = b\}$ be the set of rectified features mapped to bin $b \in \mathcal{W}$, and $h_b(\hat{X}) =$

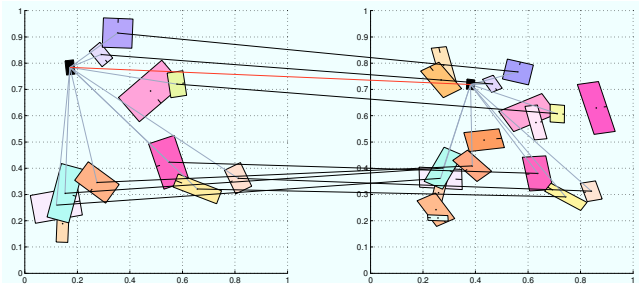


Figure 3: Inliers between two sets of features. Each inlier corresponds to a non zero term of the inner product between corresponding feature maps. Black lines connect inliers. Red line connects the origins. Grey lines connect origins with inlier features.

$|H_b(\hat{X})|/|\hat{X}|$ their frequency. The joint histogram, denoted as $h_{\mathcal{W}}(\hat{X})$, is a vector in $\mathbb{R}^k$, represented similarly to the bag-of-words histogram by (6). In fact, seen as distributions, $h_{\mathcal{V}}(X)$ is a *marginal* of $h_{\mathcal{W}}(\hat{X})$; hence we use the same symbol.

Similarly to the discrete visual similarity (7), we define the discrete spatial similarity measure $r_{\mathcal{U}}(x, y) = \delta_{u(x),u(y)}$, so that x, y are similar iff assigned to the same spatial bin. Then, matching model (17) becomes

$$\hat{S}_H(X, Y; \mathcal{C}, r_{\mathcal{U}}) = \max_{(\hat{x},\hat{y})\in\mathcal{C}} \sum_{w\in\mathcal{W}} h_w(X^{(\hat{x})}) h_w(Y^{(\hat{y})}). \quad (19)$$

Using a visual codebook means that correspondences in $\mathcal{C}$ are features x, y belonging to the same visual word,

$$\mathcal{C}(X, Y) = \{(x, y) \in X \times Y : v(x) = v(y)\}; \quad (20)$$

this specializes tentative correspondences (11) for discrete visual similarity (7). Now, let $V(X) = \{b \in \mathcal{V} : H_b(X) \neq \emptyset\}$ be the set of visual words present in feature set X and $V(X, Y) = V(X) \cap V(Y)$ the common visual words of feature sets X, Y. Then, if $f^{(\hat{x})}(X) = h_{\mathcal{W}}(X^{(\hat{x})})$ is the histogram of X's counterpart that is rectified with respect to $\hat{x}$, the overall image similarity measure becomes

$$\hat{S}_F(X, Y) = \max_{b\in V(X,Y)} \max_{\substack{\hat{x}\in H_b(X)\\ \hat{y}\in H_b(Y)}} \langle f^{(\hat{x})}(X), f^{(\hat{y})}(X)\rangle. \quad (21)$$

We have thus derived a new image representation and a corresponding matching process expressed by (21). In the following we discuss their properties.

4.5. Feature maps

We call $f^{(\hat{x})}(X)$ the *feature map* of X with *origin* $\hat{x}$. The set $F(X) = \{f^{(\hat{x})}(X) : \hat{x} \in X\}$, that is, the set of feature maps of X with origin ranging over all its features, is respectively the *feature map collection* of X. Visually, a feature map may be understood as the assignment of rectified features to spatial bins, as on the right of Figure 2. There is a different map for each origin: we may then think of each origin's map as a *local* descriptor, that encodes the *global* feature set rectified in a local coordinate frame. Well aligned feature sets are likely to have maps with a high degree of overlap.

Returning to the example of Figure 2, inliers of the two rectified feature sets are the features lying in the same bins of the joint histogram. These inliers are explicitly shown as correspondences by black lines in Figure 3. Each inlier corresponds to a nonzero term in the inner product of (21). In fact, Figure 3 illustrates all correspondences of the two feature maps having the two black patches as origins. Taking the maximum over all origins $\hat{x}, \hat{y}$ yields our image similarity of (21), where potential origin pairs are constrained to the same visual word.

We may see in (21) a clear separation between (a) *correspondence* based on visual information, in the collection of potential origin pairs $\hat{x} \in H_b(X), \hat{y} \in H_b(Y)$ for each common visual word b, and (b) *alignment* via inlier count based on spatial information, in the inner product of the two feature maps $f^{(\hat{x})}(X)$, $f^{(\hat{y})}(Y)$.

The inner product operation in (21) is reminiscent of our one-to-many choice in (11); we could equally use a histogram intersection, that we have seen to be more appropriate. However, since the joint histogram is sparse and takes values in $\{0, 1\}$ with high probability, we choose to binarize all histogram values. This is known as *max-pooling* [37]. This choice makes inner product and intersection equivalent operations and saves memory.

The time required for the intersection or inner product operation is proportional to the true size of feature maps, that is $O(n)$, where n is the size of a feature set. When the visual codebook is large enough, the maximum is taken over $O(j)$ combinations of features where j is the average number of common visual words. The total operation is typically $O(nj)$, and $O(n^2)$ in the worst case. Space requirements are $O(n)$ for a feature map and $O(n^2)$ for a collection in worst case. Savings can be made by *spatial proximity* constraints, or *feature selection*, respectively. The former is implemented via range parameter τ defined in section 6, while the latter is thoroughly discussed in section 5.

4.6. Summing up

Several pairs of feature maps may be aligned between two similar images. This fact is not captured by the max operator in (21), which returns the inlier count of the best aligned pair only. One may expect the *sum* over all pairs of feature maps to better discriminate relevant from non-relevant images. This indeed turns out to be the case. We thus define *feature map similarity* (FMS) as

$$S_F(X, Y) = \sum_{b\in V(X,Y)} \sum_{\substack{\hat{x}\in H_b(X)\\ \hat{y}\in H_b(Y)}} \langle f^{(\hat{x})}(X), f^{(\hat{y})}(Y)\rangle. \quad (22)$$

Like RANSAC and its generalized model (12), similarity measure (21) only keeps the best transformation hypothesis to count inliers. By summing over all hypotheses however, FMS (22) is similar to one-to-many voting scheme (9). Even if noisy inliers thus increase similarity in non-relevant images, true ones are counted over several origin pairs, boosting similarity in relevant images. Experiments show that the benefit of the latter boosting effect is higher.

An additional benefit is that summing over all maps introduces a flexibility to the similarity measure, exactly as in *hough*

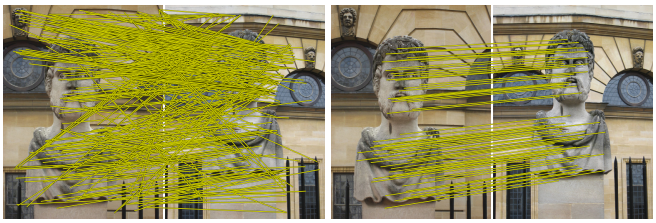


Figure 4: Left: Tentative correspondences with SURF features and a 100K vocabulary. Right: Inliers using FastSM. As revealed in Figure 5, there are two different matching surfaces between the two images, and two different relative transformations. FastSM captures only one.

Figure 5: Inliers using FMS for different origin pairs, for the example of Figure 4. Origins are shown in green circles with the scale and orientation of the relevant SURF feature. Inliers are shown in yellow lines. Each origin pair contributes to the sum of (22) and may capture either of the two matching surfaces.

pyramid matching (HPM) [23]. In HPM, votes from several origin pairs get grouped in the transformation space via a mode seeking process, without ever counting inliers. In both cases, multiple matching surfaces, deformable objects or 3D scene geometry of a scene can be captured. This is visualized in the example of Figures 4 and 5. Like RANSAC, *fast spatial matching* (FastSM) [2] can capture inliers of a single matching surface only, while FMS aggregates contributions from all.

In our prior work [7], origins are only chosen among features that *map uniquely to visual words*, as in [4]. Precisely, constraint $|H_v(X)| = |H_v(Y)| = 1$ is added in the outer $\max$ operator of (21). In this work, we drop this constraint and allow any feature to be an origin, provided that it is *selected* as such according to the selection process of section 5. In practice, speed is increased due to feature selection. Finally, the *inverse document frequency* (idf) voting scheme can be incorporated into our similarity score by adjusting the joint histogram construction. Given a rectified feature set $\hat{X}$, we simply let

$$h_{\mathcal{W}}(\hat{X}) = \sum_{w=(v,u)\in\mathcal{W}} h_w(\hat{X})\,\mathrm{idf}(v)\mathbf{e}_w, \qquad (23)$$

where $\mathrm{idf}(v)$ is the idf value of visual word v for each joint bin $w = (v, u)$.

5. Feature selection

We have seen that the size of a feature map is quadratic in the number of features in an image. This holds of course if all features are used both within a feature map (as rectified features), and as origins. On this account, in our earlier work of *feature map hashing* (FMH) [7], we have employed a combination of vocabulary learning and random selection. In particular, we select origins depending on the visual word, where the most appropriate visual words are chosen through unsupervised learning, and we keep a fixed size subset of rectified features for each feature map, using random permutations [38].

Still, experiments in [7] have shown that indexing space is the main bottleneck of the entire approach, reaching a scale of 50K images only. It is worth noting that random permutations are also used in *geometric min-hashing* (GmH) [4] to select *central* features, as well as *secondary* features from spatial neighborhoods. This option offers better scaling but suffers from low recall.

In this work, we develop a novel feature selection scheme. What is interesting is that selection is tailored to our feature map representation, in the sense that we introduce different criteria for the selection of *origins* and *rectified features*: the former is based on alignment properties, while the latter on both matching properties and locality. Unlike [7], both criteria are based on an offline, unsupervised *learning* process on a large, unlabeled image collection. Each feature is chosen independently, either as an origin, or as a rectified feature in a feature map.

The learning objective is to identify features robust enough to be matched correctly across different views of the same object. For this purpose we set up a baseline retrieval system over the entire image collection, based on bag-of-words representation, inverted file indexing, tf-idf voting and spatial verification by FastSM [2]. We then issue each image X in the collection as a query, yielding a *response* $\mathcal{R}(X)$. The response is assumed to contain all images depicting some object in common with the query, under varying viewing conditions. We select features in X according to repeating patterns over $\mathcal{R}(X)$. The overall strategy is similar to Turcot and Lowe [30], but our selection criteria are different.

Selection of origins and rectified features is discussed in sections 5.1 and 5.2, respectively. The above is of course feasible only for objects with at least two instances in the image collection, that is, for images with nonempty response. These are called *matched* images, and the remaining ones are called *single*, depicting a unique instance of a particular object in the collection. Selection is based on low-level feature properties in this case, as discussed in section 5.3.

5.1. Origins

In the single correspondence hypothesis model (16), as well as in rectified feature set matching (17)-(18), each tentative correspondence $(\hat{x}, \hat{y}) \in \mathcal{C}$ determines a transformation hypothesis. Each hypothesis is evaluated in terms of inliers,

$$I(\mathcal{C}; \hat{x}, \hat{y}, r) = \sum_{(x,y)\in\mathcal{C}} r(p(x^{(\hat{x})}), p(y^{(\hat{y})})), \qquad (24)$$

that is, in terms of correspondences $(x, y) \in \mathcal{C}$ for which features x, y are aligned when rectified with respect to origins $\hat{x}, \hat{y}$ respectively. A high number of inliers indicates that origins $\hat{x}, \hat{y}$ are aligned in terms of both local shape and position, as represented by patches $P(\hat{x}), P(\hat{y})$. On the other hand, inlier features x, y are aligned in position only. The former is thus appropriate to the selection of origins, and the latter to the selection of rectified features. Appearance is desirable in both cases, as tentative correspondences $\mathcal{C}$ are chosen according to visual similarity (11) or by visual word (7) when using a codebook.

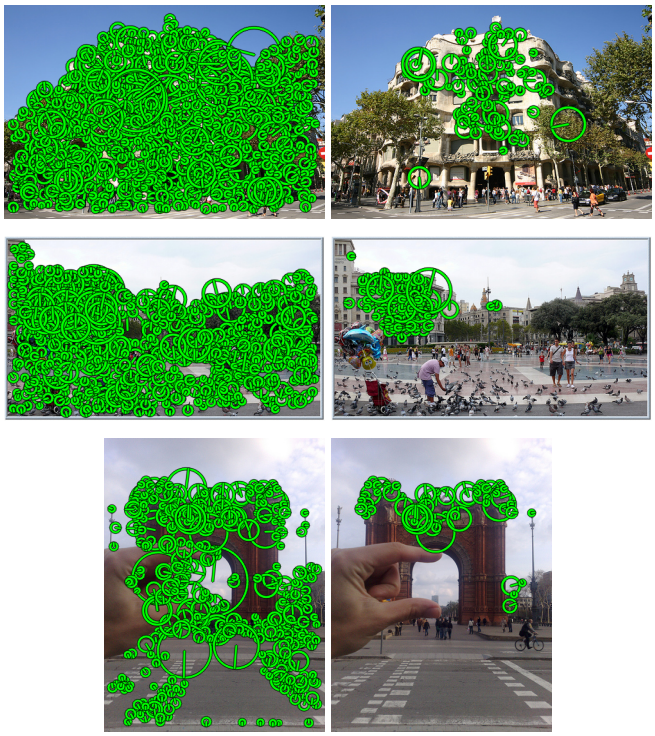


Figure 6: Left: Sample images and initially detected SURF features. Right: Features selected as origins for each image.

Focusing on origin selection using a codebook, let $\mathcal{C}(X, Y)$ be the set of tentative correspondences of images X, Y, as given by (20). Each feature z in image X may participate in multiple hypotheses across images $Y \in \mathcal{R}(X)$ found to depict the same object with X. We would like to select it as an origin if it participates in at least one successful hypothesis. Hence we consider the maximum number of inliers over all hypotheses, defining the *origin support* of $z \in X$ as

$$\alpha_X(z) = \max_{Y \in \mathcal{R}(X)} \max_{\substack{(\hat{x},\hat{y}) \in \mathcal{C}(X,Y) \\ \hat{x}=z}} I(\mathcal{C}(X, Y); \hat{x}, \hat{y}), \qquad (25)$$

where we have omitted r and have assumed that spatial similarity is measured by the uniform kernel of (13). Note that instead of measuring alignment of two patches directly, we rather measure the number of inliers they generate. This better reflects matching under feature rectification (18) and FMS (22).

Given image X, we select origins according to their support as

$$\alpha(X) = \{z \in X : \alpha_X(z) > \tau_\alpha\}. \qquad (26)$$

The feature map collection then becomes $F(X) = \{f^{(\hat{x})}(X) : \hat{x} \in \alpha(X)\}$. Figure 6 depicts sample images and the features selected as origins. These features appear on static foreground objects of the image, particularly buildings, and not on background, moving objects or persons. This is because such foreground objects tend to repeat across different images.

5.2. *Rectified features*

For a rectified feature it is important that it appears as an inlier in some hypothesis between two images. Given the selected origins $\alpha(X), \alpha(Y)$ of images X, Y, let $\mathcal{A}(X, Y) = \mathcal{C}(\alpha(X), \alpha(Y))$ be the set of origin correspondences based on a visual codebook according to (20). Each feature $z \in X$ may turn up as an inlier to multiple hypotheses generated by correspondences in $\mathcal{A}(X, Y)$, across multiple images $Y \in \mathcal{R}(X)$. We would like to select it as a rectified feature if it is an inlier to at least one hypothesis. According to the definition of inliers in (24), we consider the minimum distance between z and any corresponding feature y in any image $Y \in \mathcal{R}(X)$, after rectifying with respect to any hypothesis in $\mathcal{A}(X, Y)$,

$$\delta_X(z) = \min_{Y \in \mathcal{R}(X)} \min_{\substack{(\hat{x},\hat{y}) \in \mathcal{A}(X,Y) \\ (x,y) \in \mathcal{C}(X,Y) \\ x=z}} ||p(x^{(\hat{x})}) - p(y^{(\hat{y})})||_2, \qquad (27)$$

where we have assumed that spatial similarity r is a decreasing function of the ℓ_2 norm. The rationale is that a feature z with low distance $\delta_X(z)$ indicates that z has been aligned to at least one feature in a different image, and this provides evidence that z may indeed be an inlier to a hypothesis in a new query. We then define the *inlier support* of feature $z \in X$,

$$i_X(z) = \exp\left\{-\frac{\delta_X(z)^2}{2\sigma_i^2}\right\}, \qquad (28)$$

as a decreasing function of the minimum distance $\delta_X(z)$. Parameter σ_i is a measure of spatial proximity of two matching features in a rectified frame, so it is naturally related to the inlier threshold ϵ of (13), and to the spatial bin size.

Now, a second objective is to encourage features close to the origin. One reason is that nearby features are likely to belong to the same surface or rigid object, hence also likely to follow the same geometric transformation. A second reason is that a transformation hypothesis is deduced from local feature shape, hence is an approximation that is better near the origins. Therefore, given an origin $\hat{x}$ of image X, the *locality strength* $\ell^{(\hat{x})}(z)$ of feature $z \in X$ with respect to $\hat{x}$ is a non-increasing function of the distance $\rho^{(\hat{x})}(z)$ of z from the origin when rectified. In particular, we choose

$$\ell^{(\hat{x})}(z) = \exp\left\{-\frac{\rho^{(\hat{x})}(z)^2}{2\sigma_\ell^2}\right\}, \qquad (29)$$

where parameter σ_ℓ determines the balance between local and global geometry. The complete *rectified feature support* of feature $z \in X$ with respect to origin $\hat{x} \in X$ is

$$\beta_X^{(\hat{x})}(z) = i_X(z)\ell^{(\hat{x})}(z), \qquad (30)$$

Figure 7: Several pairs of matching feature maps and the corresponding inliers after rectified feature selection, following the example of Figures 4 and 5. Inliers are less than in Figure 5, but still enough to match the images, especially since inliers are summed over all feature maps. Note that these are not the only pairs of maps with inliers for the images shown.

favoring features with high inlier support that are close to the origin as well. It is clear that the first factor is the one that involves learning over the image collection, hence uses X as a query, while the second refers to image X alone, but is particular to each feature map originating at $\hat{x}$. This selection scheme is flexible in that it allows different features in each feature map. Given image X and origin $\hat{x} \in X$, we first select features according to rectified feature support, as

$$\beta^{(\hat{x})}(X) = \{z \in X : \beta_X^{(\hat{x})}(z) > \tau_\beta\}, \tag{31}$$

and then construct the corresponding feature map $f^{(\hat{x})}(\beta^{(\hat{x})}(X))$. An example of inliers using FMS after rectified feature selection are shown in Figure 7.

5.3. Single images

Images for which the response of matched images is empty, are unique views of an object in the entire database. Unfortunately, such unique views are the majority in practice. Without any other source of information, we resort to the *strength* available by the feature detector. This choice is in contrast to [30], where selection is based on the detected *scale*. We have experimentally found strength to be superior, which is in agreement with findings of [31]. As origins we keep a fixed number of top-ranking features in descending order of detector strength. For rectified features, we use a non-increasing function of detector strength to replace the inlier support of (28), and then combine it with locality strength as in (30). A particular choice of function is considered in section 6. Again, we keep a fixed number of features with the highest rectified feature support.

6. Implementation

6.1. Spatial quantization

In a rectified coordinate frame, we encode positions in polar coordinates (ρ, θ). To ensure that sensitivity to origin scale and orientation errors is independent of distance from the origin, log-polar coordinates are typical, as in [11]. In our case however, due to sparsity and binarization of joint histograms, it is more important to ensure uniform distribution with respect to radius ρ. As shown in Figure 8, the distribution of ρ is found experimentally close to a Weibull distribution $f_{\lambda,\kappa}(\rho)$ with λ and κ being the scale and shape parameters, respectively, estimated from samples via maximum likelihood [39]. Then, non-linear transformation with the Weibull CDF

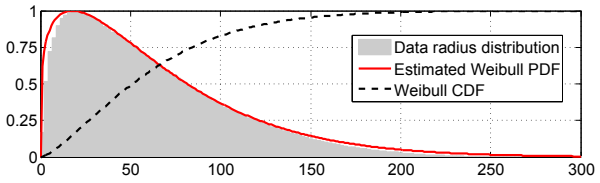


Figure 8: Distribution of radius ρ over 40K rectified feature sets from 200 images of the *European Cities* dataset, containing 8M features; see section 7). ML fitting of Weibull distribution yields $\lambda = 68.7$ and $\kappa = 1.23$.

Figure 9: Rectified features for a specific origin and range parameter $\tau = 0.2, 0.4, 0.6, 0.8$. The origin is shown in green with scale and orientation, and rectified features with red circles.

$$F_{\lambda,\kappa}(\rho) = 1 - e^{-(\rho/\lambda)^\kappa} \tag{32}$$

makes the distribution roughly uniform in $[0, 1]$.

Now, consider feature $z \in X$ rectified with respect to $\hat{x}$. Its inlier support is $i_X(z) \leq 1$ by (28), so if z is to be selected as a rectified feature, (30) and (31) imply that $\tau_\beta < \beta^{(\hat{x})}(x) \leq \ell^{(\hat{x})}(z)$, so that $\rho^{(\hat{x})}(z) < (-2\sigma_\ell^2 \ln \tau_\beta)^{1/2}$ by (29). This upper bound motivates truncating radius ρ, encoding it as

$$\bar{\rho} = \begin{cases} \frac{1}{\tau} F_{\lambda,\kappa}(\rho), & \text{if } F_{\lambda,\kappa}(\rho) \in [0, \tau) \\ 0, & \text{otherwise,} \end{cases} \tag{33}$$

where

$$\tau = F_{\lambda,\kappa}((-2\sigma_\ell^2 \ln \tau_\beta)^{1/2}) \tag{34}$$

is a *range parameter*, and finally discarding all features with $\bar{\rho} = 0$. Note that τ has been directly used in [7] to control the balance between local and global geometry. In contrast, keeping τ_β fixed, we control τ through σ_ℓ in this work. This permits joint control of locality and inlier support.

Finally, we uniformly quantize $\bar{\rho}$ and θ in k_ρ and k_θ bins over $[0, 1]$ and $[0, 2\pi]$ respectively, such that $k_\rho k_\theta = k_u$. The spatial mapping $(\bar{\rho}, \theta)$ is illustrated in the right part of Figure 2, where the non-linear distortion near $\bar{\rho} = 1$ is visible. The selected rectified features for a specific origin and different values of parameter τ are shown in Figure 9.

6.2. Feature selection

Since the objective is to reduce index space, we also limit the origins per image and rectified features per feature map to n_α and n_β, respectively. In particular, we rank origins $\alpha(X)$ by descending order of origin support (25) and keep the n_α top-ranking entries when $|\alpha(X)| > n_\alpha$. Similarly, for each origin $\hat{x}$, we rank rectified features $\beta^{(\hat{x})}(X)$ by descending order of rectified feature support (30), and keep the the n_β top-ranking entries when $|\beta^{(\hat{x})}(X)| > n_\beta$.

For single images, we keep the n_α^s strongest origins. Rectified features are selected with the same support measure $\beta_X^{(\hat{x})}(z)$ given by (30), but distance $\delta_X(z)$ in (27) is now replaced by a non-increasing function of the detector strength. In particular, we choose $\delta_X(z) = 1/\log g(z)$, where $g(z)$ is the detector strength of feature z. Again, only the top n_β^s rectified features are kept.

6.3. Indexing and filtering

For indexing, we pre-compute all feature map collections and store them in an inverted file structure. Two rectified features will match if they are mapped to the same visual word $v \in \mathcal{V}$, fall into the same spatial bin $u \in \mathcal{U}$ and the corresponding origins are mapped to the same visual word $\hat{v} \in \mathcal{V}$. However, storing *posting lists* for all possible triplets $(\hat{v}, u, v)$ would produce a huge and really sparse structure, since the visual codebook may be as large as 10^6.

Hence, for each combination of origin visual word $\hat{v} \in \mathcal{V}$ and spatial bin $u \in \mathcal{U}$, we store a dense mapping from pair $(\hat{v}, u)$ to a posting list of all associated rectified features in all images found in the database. Each posting list is sparse in v so we represent it as a list of pairs of the form (v, id), where v is the visual word of a rectified feature found in image with identifier id. The list is ordered in terms of v, so that given a specific visual word, the relevant list of image identifiers may be found in logarithmic time using binary search.

At query time, we compute the feature map collection of the query image and extract all triplets $(\hat{v}, u, v)$. For each pair $(\hat{v}, u)$, we access the associated posting list and retrieve the relevant image list for each v, casting a vote for each image id found. In effect, for query feature map $f^{(\hat{x})}(X)$ and database feature map $f^{(\hat{y})}(Y)$, we estimate similarity $S_F(X, Y)$ without explicitly computing any zero element of terms $\langle f^{(\hat{x})}(X), f^{(\hat{y})}(Y)\rangle$ in (22). We do not perform any kind of feature selection on the query image features that would require learning: all features are kept as origins and rectified features are only constrained by range parameter τ.

Each entry in the inverted file contains an image id and the visual word v of a rectified feature. We allocate 4 bytes for the former, and 2 bytes for the latter, using run-length encoding. The total space requirements for an image are 6 bytes per rectified feature. In contrast to [7], feature identifiers are not stored in the inverted file because it turns out that spatial verification is not necessary, as discussed in section 7. This results in further reduction of index space.

7. Experiments

7.1. Datasets

We conduct experiments on the *European Cities 1M* dataset[1] [40]. The ground truth, or test set, consists of 927 images depicting landmarks in Barcelona city; we also refer to it as the *Barcelona* dataset. Sample images from each of the 17 groups are shown in Figure 10. The distractor set consists of 908.859 Flickr images. Sample distractor images are shown in Figure 11. We do not include any other photos from Barcelona in the distractor set, to ensure no other image depicts the same scene or building as the ground truth. We further conduct experiments on the publicly available *Oxford Buildings*[2] [2] and UKB[3] [41] datasets.

Figure 10: Representative images from all 17 groups of the *European Cities 1M* test set, depicting landmarks in Barcelona.

Figure 11: Sample distractor images from the *European Cities 1M* dataset.

[1] http://image.ntua.gr/iva/datasets/ec1m/
[2] http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/
[3] http://www.vis.uky.edu/~stewe/ukbench/

7.2. Evaluation protocol

In all experiments, we use SURF features and descriptors [42] and a $k_v = 100$K generic visual codebook trained using *approximate k-means* (AKM) [2] on a set of images of urban scenes that are not part of our evaluation datasets. Baseline bag-of-words (BoW), weak geometric consistency (WGC) [3] and useful feature selection (UF) [30] are the methods we compare to. BoW and WGC are also followed by a second re-ranking phase where spatial verification is carried out with FastSM [2]. In offline feature selection and online query measurements, spatial re-ranking is performed on the 500 and 100 top-ranking images respectively, and verified images with more than 4 inliers are only kept in the response.

Our BoW implementation uses dot product similarity on L_2-normalized term frequency vectors with tf-idf. In our WGC implementation scale and orientation are quantized into 8 bins and the final similarity score is calculated with dot product on frequency vectors. In our UF implementation we keep 300 features for single images using the SURF feature detector strength, in contrast to [30] that uses scale. For spatial quantization in FMS we choose configuration $k_\rho = 4$, $k_\theta = 6$ according to the experiments in [7]. We evaluate overall performance via mean average precision (mAP), except for the UKB dataset, where a score is used that is the recall in the first four items. All times refer to single-threaded C++ implementations on a 2GHz Quad Core processor with 20GB of memory.

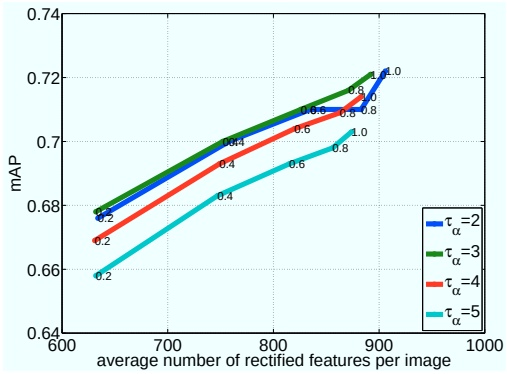
7.3. Tuning

We have experimented on the test set along with a subset of the distractor set to find the most appropriate selection parameters. Keeping more geometrically verified features increases performance given sufficient computing resources, especially index space residing in main memory. In practice however, we rather need a compromise. Performance is measured by mean average precision (mAP) over all queries in the test set. Index space is measured by the average number of rectified features over all feature maps in an image. This is equal to the average number of entries per image in the inverted file.

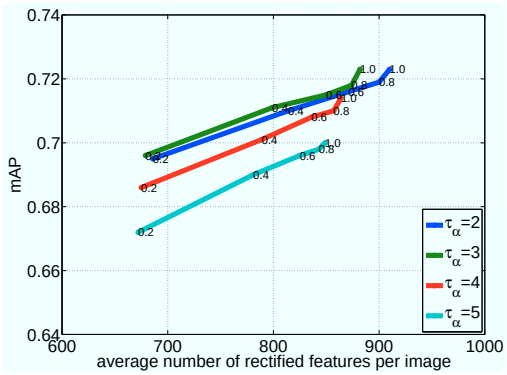
The maximum number of origins per image, n_α, and rectified features per feature map, n_β, give an upper bound on the total index space. In practice the space is limited by the thresholds on origin and rectified feature support, τ_α and τ_β respectively. It is also affected by the locality strength parameter σ_ℓ or equivalently range parameter τ, as specified by (29) and (34), respectively. This controls the balance between local and global geometry, while τ_β is kept fixed. In particular, we keep up to two standard deviations with respect to locality by setting $\tau_\beta = e^{-2}$, so that (34) becomes $\tau = F_{\kappa,\lambda}(2\sigma_\ell)$. In effect, we vary σ_ℓ but rather measure τ in our tuning experiments, which is more intuitive because $\tau \in [0, 1]$. The particular choice for τ_β means that we also keep up to two standard deviations with respect to inlier support (28), where parameter σ_i is fixed at 0.05.

Figure 12 shows index space and performance achieved on the *European Cities 1M* dataset with a subset $50,000$ distractor images, sampled from both single and matched images according to their distribution. Each curve corresponds to a different value of τ_α; along each curve, τ increases from 0.2 to 1.0. The highest performance is achieved for $\tau_\alpha = 3$. It is interesting that $\tau_\alpha = 2$ increases space, losing in performance at the same time. This case corresponds to at least three features being aligned, one of which serves as the origin. The true inliers are two only, which is not enough evidence for selection, justifying the loss. Now, savings can be made by either using a higher value of τ_α or a lower value of τ. Figure 12 reveals that the latter option is preferable, since higher performance is achieved with the same memory. As a compromise, we choose $\tau_\alpha = 3$ and $\tau = 0.6$.

After conducting a number of trials on the maximum number of origins and rectified features, we show in Figures 12(a) and 12(b) results for $n_\alpha = 50$, $n_\beta = 100$ and $n_\alpha = 100$, $n_\beta = 50$ respectively. We choose the second configuration, again for higher performance with the same memory. It seems that selecting more origins is better since the presence of aligned origins is crucial for FMS, while rectified features are in general easier to match. Similarly, after a set of trials on single images, the maximum number of origins per image, n^s_α, and rectified features per map, n^s_β, are set to 30 and 20 respectively in all our experiments.



(a) $n_\alpha = 50, n_\beta = 100$



(b) $n_\alpha = 100, n_\beta = 50$

Figure 12: Average number of rectified features per image and mAP measure on the *European Cities 1M* dataset with $50,000$ distractor images, sampled from both single and matched images. The value of parameter τ is overlaid on each curve.

Method	FMS			FMH [7]
	Single	Matched	Total	Total
Images	652104	257682	909786	-
Features b.s.	483.1	562.6	505.6	-
Origins	30.0	55.4	37.2	200
Rectified features	18.5	43.2	25.5	50
Features a.s.	557.6	2401.9	1079.9	10000.0

Table 1: Selection statistics for FMS and FMH. FMS statistics are measured on the *European Cities 1M* dataset, while the ones for FMH are fixed. Total number of images, average number of features per image before selection (b.s.), average origins per image, average rectified features per origin and average features per image after selection (a.s.).

	Single	Matched	Total
Images	652104	257682	909786
Features b.s.	483.1	562.6	505.6
Features a.s.	270.1	70.9	213.7

Table 2: Selection statistics for UF on the *European Cities 1M* dataset: total number of images, average number of features per image before selection (b.s.) and after selection (a.s.).

7.4. Results

For the selection parameters chosen, Table 1 gives the average number of features per image before selection, origins per image, rectified features per origin and total elements per image. Measurements are presented both separately for single and matched images, and in total. FMH [7] selection employs hashing with random permutations for rectified features and visual word statistics for origins. The number of features after selection is the actual number of entries in the inverted file per images. The proposed method has one order of magnitude less index space requirements than FMH.

Similar measurements are shown in Table 2 for UF. Observe that our approach integrates geometry in the indexing process and still needs only twice more entries than BoW in the inverted file, which has no spatial information at all. Compared to UF, it needs about 5 times more entries on average.

Figure 13 compares the proposed approach to bag-of-words (BoW), weak geometric consistency (WGC) and useful feature selection (UF) on the *European Cities 1M* dataset for a varying number of distractor images. BoW and WGC results are also re-ranked by applying FastSM to a short-list of the 100 top-ranking images. FMS clearly outperforms the other methods showing a benefit from geometry indexing, especially at larger scale. In fact, FMS outperforms BoW and WGC even with re-ranking. Most importantly, we have also attempted re-ranking on the FMS response and all mAP measurements have differed in up to the third significant digit. Hence global feature geometry is successfully indexed and no re-ranking is necessary with FMS. To our knowledge, this is achieved for the first time at this scale.

Spatial verification is typically the most time consuming query operation. As presented in Table 3, FMS has query times only 3 times higher than BoW and 4 times lower than BoW with re-ranking. As shown in our experiments and in accordance with [43], WGC increases performance but the increase in query time is also considerable. In WGC with re-ranking, query times increase further. In UF, features are selected with

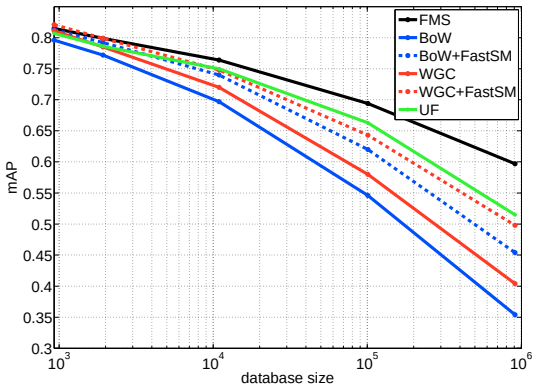


Figure 13: Mean average precision for varying database sizes on the *European Cities 1M* dataset for BoW, WGC, UF and FMS. BoW and WGC are also followed by re-ranking on the top 100 images using FastSM.

BoW	BoW+FastSM	WGC	WGC+FastSM	UF	FMS
88	1167	3221	4294	54	286

Table 3: Average query times in ms for all methods. Times for sorting scores is not included.

spatial criteria and high performance is achieved in small query times. Our method outperforms UF by including geometry not only in the off-line selection process, but also at query time.

The same experiments are conducted on the publicly available *Oxford Buildings* dataset as a test set, with the same distractor set of European Cities 1M; results are presented in Figure 14. In this case the benefit from geometry indexing only occurs at large scale: we have to include up to 10K distractor images before the benefit of FMS over UF appears, while the benefit over BoW with re-ranking only appears at $5 \cdot 10^5$ distractors. This result is in accordance with the results of [43] and [3], where, using weak geometry, WGC shows little or no benefit over BoW on the *Oxford Buildings* dataset in the absence of distractors. It is possibly due to the fact that images are larger in Oxford, thus containing larger number of features per image and yielding more correspondences. Note also that UF is consistently lower than BoW+FastSM; this is in accordance with [30], where most benefit comes from feature augmentation rather than selection.

This finding means that, despite the fact that including geometry in index makes matching more discriminative and that FMS can get support from multiple surfaces while FastSM is restricted to a single surface, FMS can still be inferior to baseline BoW+FastSM depending on the dataset and amount of distractors. This is due to the fact that a lot of information is discarded during feature selection, which is nevertheless necessary to reduce index space. It is thus possible that relevant images still rank too low to be recovered even with spatial verification. This is important in view of the fact that geometry verification can be made both faster and more precise [23]. It is however interesting that geometry indexing is possible at a scale of 1M images, and that this scale is exactly where its gain over re-ranking appears.

We have further conducted comparisons on the UKB dataset.

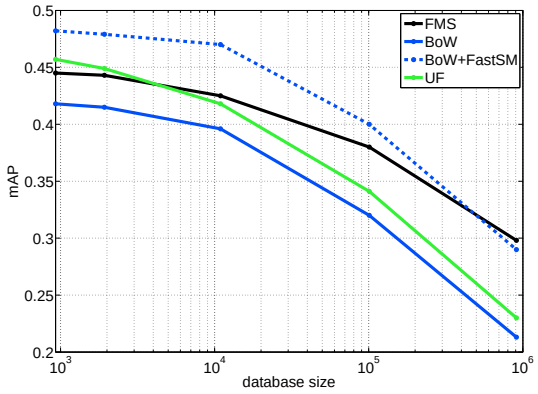


Figure 14: Mean average precision for varying database sizes on the *Oxford Buildings* dataset for BoW, FMS and UF. BoW is also combined with re-ranking on the top 100 images using FastSM.

Method	FMS	BoW	BoW+FastSM	UF
Score	**2.47**	1.90	2.1	2.34

Table 4: Score on the UKB dataset with 1M distractor images for BoW, FMS and UF. BoW is also combined with re-ranking on the top 100 images using FastSM.

This particular dataset differs in the fact that only 4 similar images appear in the dataset for each object. Results are presented in Table 4. Both UF and FMS outperform BoW+FastSM in this large scale experiment with 1M distractor images. This comes to no surprise; background features that are common in many different object classes of UKB tend to corrupt BoW scoring. Using selection, UF and FMS discard all background features that are not matched geometrically and therefore get better results from the filtering stage. Still, FMS further outperforms the UF selection.

Finally, ranking examples for a single query image are shown in Figure 15 for BoW and FMS. A lot of true positive images are lost with BoW, which appear to be ranked in top position with FMS.

8. Discussion

To our knowledge, our feature map representation introduced in [7] has been the first to integrate appearance and global geometry in the indexing stage, while being invariant to geometric transformations and robust to occlusion. However, the hashing scheme employed in that work has not been enough to keep the required index space at reasonable levels. The novel feature selection process introduced in this work enables this approach to work with a memory footprint comparable to the baseline bag-of-words model and handle image databases of size 10^6.

We consider our experiments successful, not because we achieve better overall precision in the specific datasets compared to the state of the art, but rather because we make spatial matching work at large scale and show that pairwise spatial verification in a re-ranking sense is not needed to improve performance any further at large scale. This is in contrast to other

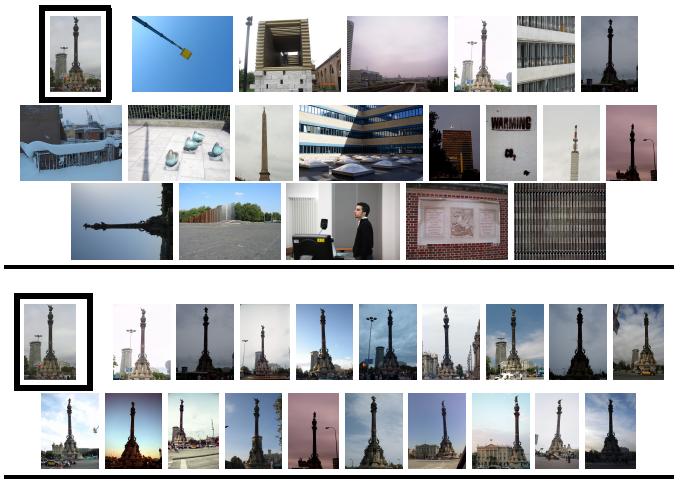


Figure 15: Sample query (with black border) and ranked retrieved images. Results using BoW (top) and FMS (bottom).

geometry indexing schemes that still need re-ranking, and it is important because re-ranking is linear in the number of images to verify and practically the most time-consuming task at query time.

The question remains open whether geometry indexing will mature to the point where re-ranking will be no longer be needed. We have made significant progress but depending on the dataset, indexing may still be inferior to re-ranking in the absence of distractors. One extreme example of re-ranking towards web scale is [44], performing exhaustive verification on billions images. Though this appears contradicting to our approach, it may be the case that the two approaches can actually be complementary, in the sense that our similarity in (22) remains an inner product, hence subject to the same optimizations as conventional BoW.

We have developed our methodology for affine transformations, and this is because state of the art feature detectors are affine covariant. In fact, we have seen that very good performance is achieved even with SURF features providing for similarity transformation only. Extending *e.g.* to homography would be straightforward, should such features mature, like Koeser and Koch [45]. We see it as a challenge for future feature detectors to achieve better alignment in shape as well as position, so that more stable origins become available.

Feature selection has been crucial in making this indexing approach scale up. The fact that we apply a different strategy for origin and rectified feature selection makes our approach more robust than [30], while still keeping query times comparable. A relaxed spatial matching model like [23] may open the way to entirely new selection schemes. In any case, selection relies on a mining process assuming multiple views. It is true however that single views are the majority in practice. Other criteria like symmetry may apply in this case [31].

The larger the scale, the more important geometry is, but keeping both query time and index size restricted is a challenge and various extensions to feature maps could be considered. For the appearance part, soft assignment [46] is a straightfor-

ward extension. For the spatial part where quantization is uniform and dimensionality low, a hierarchical approach like spatial pyramid matching [17] would be more appropriate.

We find the feature map representation the most important contribution of this work. While numerous variants have been around as discussed in section 2, feature maps are unique in being discriminant and invariant enough at the same time. We foresee a new research direction in applying this concept to problems like large scale object recognition and detection, where geometric consistency, invariance and speed are as crucial as in retrieval. More flexible geometric models would be more appropriate in this case, like the recent developments in relaxed spatial matching [23].

References

[1] J. Sivic, A. Zisserman, Video google: A text retrieval approach to object matching in videos, in: International Conference on Computer Vision, 2003. 1

[2] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: Computer Vision and Pattern Recognition, 2007. 1, 2, 5, 7, 10, 11

[3] H. Jegou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, in: European Conference on Computer Vision, 2008. 1, 2, 4, 11, 12

[4] O. Chum, M. Perdoch, J. Matas, Geometric min-hashing: Finding a (thick) needle in a haystack, in: Computer Vision and Pattern Recognition, 2009. 1, 2, 7

[5] Z. Wu, Q. Ke, M. Isard, J. Sun, Bundling features for large scale partial-duplicate web image search, in: Computer Vision and Pattern Recognition, 2009. 1, 3

[6] Y. Lamdan, H. Wolfson, Geometric hashing: A general and efficient model-based recognition scheme, in: International Conference on Computer Vision, 1988. 1, 2

[7] Y. Avrithis, G. Tolias, Y. Kalantidis, Feature map hashing: Sub-linear indexing of appearance and global geometry, in: ACM Multimedia, 2010. 1, 2, 7, 9, 10, 11, 12, 13

[8] M. Fischler, R. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395. 1

[9] D. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110. 1, 5

[10] K. Köser, C. Beder, R. Koch, Conjugate rotation: Parameterization and estimation from an affine feature correspondence, in: Computer Vision and Pattern Recognition, 2008. 1

[11] S. Belongie, J. Malik, J. Puzicha, Shape context: A new descriptor for shape matching and object recognition, in: Neural Information Processing Systems, 2000. 1, 2, 9

[12] A. Broder, Identifying and filtering near-duplicate documents, in: Symposium on Combinatorial Pattern Matching, 2000. 1

[13] O. Chum, J. Matas, J. Kittler, Locally optimized RANSAC, in: DAGM Symposium on Pattern Recognition, 2003. 1

[14] I. L. Dryden, K. V. Mardia, Statistical Shape Analysis, Wiley, 1998. 2, 3

[15] O. Chum, J. Matas, Geometric hashing with local affine frames, in: Computer Vision and Pattern Recognition, 2006. 2, 5

[16] H. Ling, S. Soatto, Proximity distribution kernels for geometric context in category recognition, in: International Conference on Computer Vision, 2007. 2

[17] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: Computer Vision and Pattern Recognition, 2006. 2, 14

[18] A. Berg, T. Berg, J. Malik, Shape matching and object recognition using low distortion correspondences, in: Computer Vision and Pattern Recognition, 2005. 2

[19] H. Jiang, S. X. Yu, Linear solution to scale and rotation invariant object matching, in: Computer Vision and Pattern Recognition, 2009. 2

[20] M. Leordeanu, M. Hebert, A spectral technique for correspondence problems using pairwise constraints, in: International Conference on Computer Vision, 2005. 2

[21] M. Perdoch, O. Chum, J. Matas, Efficient representation of local geometry for large scale object retrieval, in: Computer Vision and Pattern Recognition, 2009. 2

[22] Y. Zhang, Z. Jia, T. Chen, Image retrieval with geometry-preserving visual phrases, in: Computer Vision and Pattern Recognition, 2011. 2

[23] G. Tolias, Y. Avrithis, Speeded-up, relaxed spatial matching, in: International Conference on Computer Vision, 2011. 3, 7, 12, 13, 14

[24] W. Zhou, Y. Lu, H. Li, Y. Song, Q. Tian, Spatial coding for large scale partial-duplicate web image search, in: ACM Multimedia, 2010. 3

[25] Y. Cao, C. Wang, Z. Li, L. Zhang, L. Zhang, Spatial-bag-of-features, in: Computer Vision and Pattern Recognition, 2010. 3

[26] G. Schindler, M. Brown, R. Szeliski, City-scale location recognition, in: Computer Vision and Pattern Recognition, 2007. 3

[27] F. Li, J. Kosecka, Probabilistic location recognition using reduced feature set, in: International Conference on Robotics and Automation, 2006. 3

[28] J. Knopp, J. Sivic, T. Pajdla, Avoiding confusing features in place recognition, in: European Conference on Computer Vision, 2010. 3

[29] S. Gammeter, L. Bossard, T. Quack, L. V. Gool, I know what you did last summer: Object-level auto-annotation of holiday snaps, in: International Conference on Computer Vision, 2009. 3

[30] P. Turcot, D. Lowe, Better matching with fewer features: the selection of useful features in large database recognition problems, in: International Conference on Computer Vision, 2009. 3, 7, 9, 11, 12, 13

[31] G. Tolias, Y. Kalantidis, Y. Avrithis, Symcity: Feature selection by symmetry for large scale image retrieval, in: ACM Multimedia, 2012. 3, 9, 13

[32] W. Dong, Z. Wang, M. Charikar, K. Li, Efficiently matching sets of features with random histograms, in: ACM Multimedia, 2008. 4

[33] M. Werman, S. Peleg, A. Rosenfeld, A distance metric for multidimensional histograms, Computer, Vision, Graphics, and Image Processing 32 (3) (1985) 328–336. 4

[34] Y. Rubner, C. Tomasi, L. Guibas, The earth mover's distance as a metric for image retrieval, International Journal of Computer Vision 40 (2) (2000) 99–121. 4

[35] K. Grauman, T. Darrell, The pyramid match kernel: Discriminative classification with sets of image features, in: International Conference Computer Vision, 2005. 4

[36] F. Rothganger, S. Lazebnik, C. Schmid, J. Ponce, 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints, International Journal of Computer Vision 66 (3) (2006) 231–259. 5

[37] L. Liu, L. Wang, X. Liu, In defense of soft-assignment coding, in: International Conference on Computer Vision, 2011. 6

[38] A. Broder, M. Charikar, A. Frieze, M. Mitzenmacher, Min-wise independent permutations, in: ACM Symposium on Theory of Computing, 1998. 7

[39] A. Cohen, Maximum likelihood estimation in the weibull distribution based on complete and on censored samples, Technometrics 7 (4) (1965) 579–588. 9

[40] Y. Avrithis, Y. Kalantidis, G. Tolias, E. Spyrou, Retrieving landmark and non-landmark images from community photo collections, in: ACM Multimedia, 2010. 10

[41] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: Computer Vision and Pattern Recognition, 2006. 10

[42] H. Bay, T. Tuytelaars, L. Van Gool, SURF: Speeded up robust features, in: European Conference Computer Vision, 2006. 11

[43] H. Jegou, M. Douze, C. Schmid, Improving bag-of-features for large scale image search, International Journal of Computer Vision 87 (3) (2010) 316–336. 12

[44] H. Stewénius, S. H. Gunderson, J. Pilet, Size matters: exhaustive geometric verification for image retrieval, in: European Conference on Computer Vision, 2012. 13

[45] K. Köser, R. Koch, Perspectively invariant normal features, in: International Conference on Computer Vision, 2007. 13

[46] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Lost in quantization: Improving particular object retrieval in large scale image databases, in: Computer Vision and Pattern Recognition, 2008. 13