Efficient Representation of Local Geometry for Large Scale Object Retrieval

Michal Perďoch, Ondřej Chum and Jiří Matas
Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University in Prague
{perdom1,chum,matas}@cmp.felk.cvut.cz

Abstract

State of the art methods for image and object retrieval exploit both appearance (via visual words) and local geometry (spatial extent, relative pose). In large scale problems, memory becomes a limiting factor – local geometry is stored for each feature detected in each image and requires storage larger than the inverted file and term frequency and inverted document frequency weights together.

We propose a novel method for learning discretized local geometry representation based on minimization of average reprojection error in the space of ellipses. The representation requires only 24 bits per feature without drop in performance. Additionally, we show that if the gravity vector assumption is used consistently from the feature description to spatial verification, it improves retrieval performance and decreases the memory footprint. The proposed method outperforms state of the art retrieval algorithms in a standard image retrieval benchmark.

1. Introduction

Very large collections of images are becoming available both due to commercial efforts [4] and photo sharing of individual people [6, 5]. Image retrieval and object recognition methods suitable for large datasets must not only have running time that grows slowly with the size of the collection, but must be very efficient in the use of memory. As soon as the representation of the complete collection fails to fit into dynamic memory, running time jumps by orders of magnitude (to 15–35s per query) as reported in [2].

In the paper, we propose a method for highly memory-efficient representation of local geometry associated with visual words. Geometric verification has been shown essential in recent state of the art retrieval approaches [2, 18, 7]. Local geometry represented by an affine covariant ellipse is noticeably bigger than the size of tf-idf weights and labels when stored in a naive way thus becoming a significant factor determining the limits of retrieval methods.

The proposed discretized local geometry representation is learned by minimization of average reprojection error in the space of ellipses. The minimization process leads to an approximately four fold compression of local geometry memory requirement and the representation requires only 24 bits per feature without loss of performance in a challenging retrieval problem when compared with the exact representation (modulo floating point precision), making geometry more compactly represented than visual appearance.

The proposed representation is designed, besides compactness, to seamlessly support the assumption that the "gravity vector" is useful for fixing the orientation ambiguity of affine covariant points. In a retrieval experiment, the gravity vector is exploited consistently in the process of obtaining image representation unlike in [2, 18] where the assumption is enforced in spatial verification.

In the second part of the paper, the discretized representation of local geometry is integrated in an image retrieval method [19] and evaluated on two public datasets (Oxford buildings, INRIA Holidays) according to a standard protocol. For the Oxford dataset, the performance measured by mean average precision is superior to the state of the art. On the INRIA dataset, results are comparable to those reported in the literature.

Related work. Besides [20, 2, 18, 19], we are not aware of work focusing on the use of local geometry in very large scale retrieval problems. Although the literature on image retrieval is huge, very few methods exploit local geometry. Methods based on global descriptors [3, 21] and on the bag-of-words, *i.e.* global histograms of local descriptors [9], dominate the field.

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

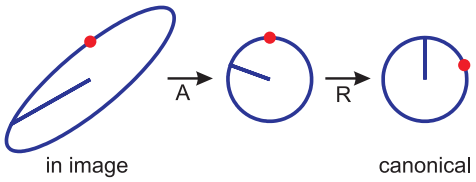

Figure 1. The ellipse normalization transformation ($\mathbf{A}$) preserves the gravity vector direction (red dot), $\mathbf{R}$ rotation to canonical position.

Object recognition methods that include geometric verification such as [12, 10] have focused mainly on recall and precision on small databases where memory issues are irrelevant. Only recently, recognition methods handling thousands of object emerged[15, 17]. In [18, 19], Philbin et al. have shown the importance of local geometry and reported a significant drop in performance when the memory limit was reached. We show that the limit can be pushed back by a factor five without loss of retrieval performance.

The rest of the paper is structured as follows. Section 2 introduces a novel method for efficient representation of local geometry. In Section 3 we detail out the implementation improvements of detector and the use of gravity vector, that significantly improved the overall performance. Finally the proposed geometry representation and gravity vector is evaluated in Section 4.

2. Learning Efficient Ellipse Representation

An efficient geometry representation of elliptical regions has to allow generation and verification of global (or semi-local) geometric constraints, while using minimal number of bits for the geometric information. In this section we first consider two representations of local geometry: ellipses and local affine frames (local coordinate systems) that combine an ellipse and a distinguished point or a dominant orientation. We then define a similarity measure in the space of affine transformations with important geometric interpretation; and finally, introduce k-means based minimization scheme.

Ellipses. Points $\mathbf{x}$ on an ellipse satisfy

$$(\mathbf{x}-\mathbf{x}_0)^\top \mathbf{E}\,(\mathbf{x}-\mathbf{x}_0) = 1,$$

where $\mathbf{x}_0$ is the center of the ellipse and $\mathbf{E}$ is a 2×2 positive definite matrix. It is convenient to represent an ellipse by a transformation $\mathbf{A}$ mapping points on the ellipse to points on a unit circle. Such transformation $\mathbf{A}$ satisfies $\mathbf{E} = \mathbf{A}^\top\mathbf{A}$. The decomposition of $\mathbf{E}$ is not unique and is defined up to an arbitrary rotation $\mathbf{Q}$, since $(\mathbf{QA})^\top(\mathbf{QA}) = \mathbf{A}^\top\mathbf{A}$. To remove the ambiguity, we choose to restrict the normalization transformation $\mathbf{A}$ to lower triangular matrices

$$\mathbf{A} = \begin{pmatrix} a & 0 \\ b & c \end{pmatrix}. \tag{1}$$

Lower triangular matrices form a group, closed under inversion and composition. Also, an affine transformation of the chosen type has one eigenvector equal to $(0,1)^\top$, which is exploited and related in Section 2.1 to a gravity vector assumption.

Local affine frames. The center of the ellipse and its shape provides an affine frame up to an unknown rotation. To resolve for the rotation, either a dominant orientation [12] or a distinguished point on the ellipse [16] must be provided with the ellipse. We represent this additional information by a rotation $\mathbf{R}$, so that transformation $\mathbf{RA}$ transforms an ellipse represented by lower triangular matrix $\mathbf{A}$ to a canonical position. The process is demonstrated in Figure 1.

Let $(\mathbf{A}_i, \mathbf{R}_i, \mathbf{x}_i)$ and $(\mathbf{A}_j, \mathbf{R}_j, \mathbf{x}_j)$ represent two corresponding features, where $\mathbf{x}_i, \mathbf{x}_j$ are centers of ellipses. The affine transformation $\mathbf{H}$ mapping coordinates of one frame to another is a composition of normalization of one frame to canonical, followed by a denormalization to the other frame

$$\mathbf{H} = \mathbf{A}_j^{-1}\mathbf{R}_j^\top\mathbf{R}_i\mathbf{A}_i. \tag{2}$$

In this shortened notation, the translation is omitted for the sake of clarity and we denote $(\mathbf{A}_i, \mathbf{R}_i, \mathbf{x}_i)$ as $(\mathbf{A}_i, \mathbf{R}_i)$. Clearly, the translation is given by the translation from $\mathbf{x}_i$ to $\mathbf{x}_j$.

Compacting representation by discretization. To reduce the memory requirements of storing the geometric information, we aim at representing a set of similar elliptical regions with local affine frames $(\mathbf{A}_i, \mathbf{R}_i)$ by a good approximation $(\mathbf{B}, \mathbf{R}_i)$.

Ideally for each $\mathbf{A}_i$ a prototype $\mathbf{B}_i$ is found such that

$$\mathbf{B}_i^{-1}\mathbf{R}_i^\top\mathbf{R}_i\mathbf{A}_i = \mathbf{B}_i^{-1}\mathbf{A}_i = \mathbf{I}$$

where $\mathbf{I}$ is identity. In practice, some error $\mathcal{E}$ will remain as a result of the quantization

$$\mathbf{B}^{-1}\mathbf{A}_i = \mathbf{I} + \mathcal{E}.$$

Here $\mathbf{B}$ is again lower triangular matrix uniquely representing an elliptical shape. Please note that the normalization transformation $\mathbf{A}_i$ is represented by $\mathbf{B}$ while the orientation $\mathbf{R}_i$ is fixed for all $\mathbf{A}_i$ (as explained in Section 2.1).

Since the geometric information (of a pair $\mathbf{A}_i$,$\mathbf{A}_j$) is used to estimate an image to image affine transformation (Eqn. (2)), in the end, the quality of the combined transformation $\mathbf{H}$ should be optimized. The quality of an affine transformation is well captured by integrating the reprojection error e over a (unit) circle

$$e = \int\limits_{\|\mathbf{x}\|^2=1} \|\mathbf{I}\mathbf{x} - (\mathbf{I}+\mathcal{E})\mathbf{x}\|^2 = \int\limits_{\|\mathbf{x}\|^2=1} \|\mathcal{E}\mathbf{x}\|^2.$$

Using simple manipulations we show that the integrated reprojection error e is a monotonic function of the Frobenius norm $\|\mathcal{E}\|_F$ of the error matrix $\mathcal{E}$

$$e = \int\limits_{\alpha=0}^{2\pi} \left\| \mathcal{E} \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix} \right\|^2 = \pi \cdot \|\mathcal{E}\|_F^2. \qquad (3)$$

Selecting the best prototypes. Given a large training set of elliptic shapes represented by lower triangular matrices $\mathcal{A} = \{\mathbf{A}_1, \ldots, \mathbf{A}_N\}$ we aim to find a best set of prototypes $\mathcal{B} = \{\mathbf{B}_1, \ldots, \mathbf{B}_K\}$ of clusters $\mathcal{A}_j \subset \mathcal{A}$. Having developed an error measure in the space of transformations of form (1), we can apply a standard k-means algorithm on the set $\mathcal{A}$.

The k-means algorithm is parameterized by the number K of ellipse clusters, represented by transformations $\mathbf{B}_j \in \mathcal{B}$. In the assignment step of each iteration, for all ellipses $\mathbf{A}_i \in \mathcal{A}$ the best prototype $\mathbf{B}_{f(i)}$, where $f(i)$ denotes the assignment, is found by

$$f(i) = \operatorname*{argmin}_j \|\mathbf{B}_j^{-1}\mathbf{A}_i - \mathbf{I}\|_F^2.$$

In the refinement step, a new optimal prototype $\mathbf{B}_j$ is computed from all ellipses in the j-th cluster $\mathcal{A}_j = \{\mathbf{A}_i, f(i) = j\}$. This is achieved by optimizing

$$\mathbf{B}_j = \operatorname*{argmin}_B \sum_{\mathbf{A}_i \in \mathcal{A}_j} \|\mathbf{B}^{-1}\mathbf{A}_i - \mathbf{I}\|_F^2. \qquad (4)$$

Minimization of Eqn. (4) leads to a system of three linear equations, that can be solved in closed form. Finally, each elliptical region $\mathbf{A}_i$ is represented by its final assignment $j = f(i)$, which is an index into the list of prototypes $\mathbf{B}_j$. This can be thought of as a geometric "vocabulary" (Fig. 2).

Separate representation of scale It is well known that the scale is independent of the elliptical shape of a feature. Therefore, it might be interesting to separate the effects of scale and the remaining factor of the *normalizing* transformation $\mathbf{A}$

$$s = \sqrt{\det(\mathbf{A}^{-1})} \quad \mathbf{A}' = s\mathbf{A}.$$

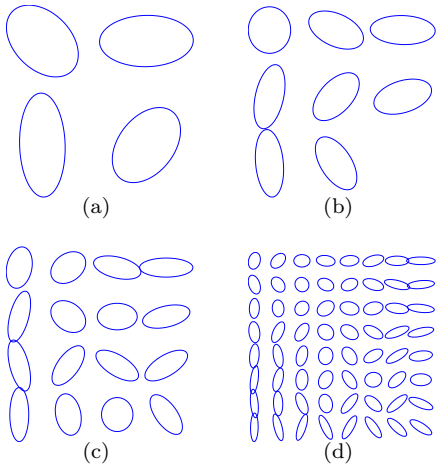

Figure 2. Learned geometric "vocabularies". Examples of ellipse prototypes (with scale removed) for $K = 4, 8, 16$ and 64.

To separate the scale, log-scale ($log(s_i)$) is uniformly split into L intervals, each interval $\bar{s}_1, \ldots, \bar{s}_L$ is represented by its average value $\bar{s}_l$. The resulting scale $\bar{s}_l$ is removed from the normalizing affine transformation $\mathbf{A}_i$. Afterwards, the k-means clustering is performed on the set of scale-normalized transformations $\mathbf{A}'_i$ as before.

The separation of scale allows to reduce the number of prototypes $\mathbf{B}_j$ necessary to cover the space of all shapes $\mathcal{A}$ and thus reduces the computational cost of assignment to shape prototypes. An example set of corresponding ellipses in two images represented using different values of L and K is shown in Figure 3 (denoted by SxEy for L=2^x, K=2^y).

2.1. The Gravity Vector

In the previous section, we have assumed that the local affine frame $(\mathbf{A}_i, \mathbf{R}_i, \mathbf{x}_i)$ can be reduced to $(\mathbf{A}_i, \mathbf{I}, \mathbf{x}_i)$, i.e. we assume that characteristic orientations $\mathbf{R}_i$ can be ignored (set to $\mathbf{I}$). This assumption can be interpreted, together with our choice of $\mathbf{A}_i$ which preserves orientation of vertical axis, as existence of so-called *gravity vector*, i.e. existence of a vector in an image pointing down in the gravity direction which is preserved (as well as vector pointing upwards). The idea of gravity vector in geometric verification was introduced by Philbin et al. in [18] who proposed to use the gravity vector in spatial verification instead of computed orientation $\mathbf{R}_i$ of local affine frame to get better estimate of global affine transformation and showed that the assumption of the gravity vector is satisfied more often than expected.

We propose to use the gravity vector already in the

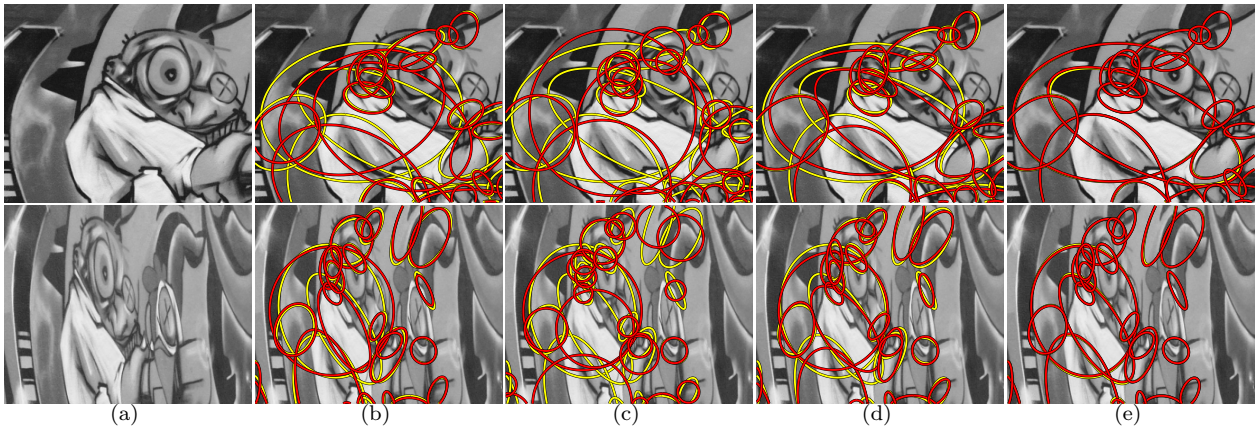

Figure 3. The precision of geometry representation (presented on a few corresponding ellipses detected in images 1 and 4 of the Graffiti sequence). In yellow: exact ellipses. In red: ellipses represented by b) 256 shape prototypes with scale (S0E8), c) 256 scales, 1 shape prototype (S8E0), d) 16 scales, 16 shape prototypes (S4E4), e) 16 scales and 4096 shape prototypes (S4E12).

description of an affine point. The above-mentioned local affine frame $(\mathbf{A}_i, \mathbf{I}, \mathbf{x}_i)$ is computed for each affine feature and used to normalize a small neighborhood of the point. Note that the orientation was fixed $\mathbf{R}_i = \mathbf{I}$ (*c.f.* Figure 1). The SIFT descriptor is then computed on this normalized neighborhood and vector quantized in the standard way [18].

Thus, the assumption of the stability of the gravity vector is enforced consistently both in the description and geometric verification. The gravity vector assumption also allows keeping one orientation/description per affine point further compacting the representation by factor of 1.3 to 1.5. We show in experiments that the use of gravity vector in description improves the image retrieval performance, which is expected: if the assumption is true (*c.f.* Figure 4), the method benefits from its full use.

3. Implementation

Various technical aspects and parameters of our implementation that further improved the overall performance are detailed in this section.

The features and local geometry were obtained by a modified version of Hessian-Affine detector proposed by K.Mikolajczyk[1]. We have changed the initial scale selection which is in our version based on the scale-space maxima of the Hessian operator. As shown by Lindeberg in [11] Hessian operator has similar scale-selection properties as Laplace operator, however with the Hessian we can simultaneously localize scale-space maxima both in location and scale. Finally, an affine adaptation procedure as described in [13] is applied on scale covariant "Hessian-Hessian" points. We have observed that this choice improves the retrieval performance.

The description of affine covariant points is computed on affine-normalized local neighborhoods. Transformations $\mathbf{A}_i$ were computed for all affine points using lower-triangular Choleski-like decomposition that preserves orientation of vertical axis, i.e. $\mathbf{A}_i$ of the form (1). The gravity vector assumption was applied to fix the rotation $\mathbf{R}_i = \mathbf{I}$, i.e. with above-mentioned decomposition, rotation $\mathbf{R}_i$ was simply ignored. Then SIFT descriptors [12] were computed on the affine normalized patches with measurement region of radius $r = 3\sqrt{3}s$, where $s = (\det \mathbf{A}_i)^{-1/2}$ is the scale of the point[2].

4. Performance Evaluation

4.1. Image Retrieval Framework

The performance of geometry compression and the influence of the gravity vector assumption is demonstrated on image retrieval.

The method used is similar to the approach proposed in [2, 19]. In short, affine covariant points are detected using modified version of Hessian-Affine [14] in all database images, described by affine covariant SIFT descriptors. SIFT descriptors are vector quantized using k-means with approximate nearest neighbor. The standard tf-idf weighting with L_2 norm is computed and images ranked as in [20]. The Spatial verification (SP) is performed on the top ranked images according

[1] http://www.robots.ox.ac.uk/~vgg/research/affine/

[2] chosen such that for an isotropic Gaussian blob with variance σ^2, $s = \sigma$

Figure 4. Matches (yellow ellipses) in the INRIA and Flickr5M datasets with the gravity vector assumption. Although the images are slightly rotated or with heavy perspective effects, a correct transformation was found.

to the tf-idf score: first, a set of tentative correspondences is formed between the query image and each of the top images. Then, all hypotheses (from all corresponding pairs) of affine transformation are computed. The best five hypotheses (with the highest number of inliers), for each top ranked image are refined using local optimization [1]. Top images are then re-ranked according the number of inliers of the best hypothesis of affine transformation. Optionally, a query expansion step (QE) is applied as in [2]; the top ranked spatially verified images are taken and the visual words of back projected features lying in the query image (or query bounding box) are used in new, enhanced query. We also report results using soft assignment (SA) [19]. Storing multiple visual words for each feature in database [19] multiplies the storage size required for the inverted file. Therefore, we perform soft assignment only on the query side as in [8]. We use five nearest neighbours in the experiments with soft assignment, which results in approximately five times longer query time. The validation of the method was performed on image retrieval benchmarks proposed in [18, 7]. The first experiment focuses on the choice of geometry representation that provides good trade-off between the memory footprint and the performance. The performance of the method is compared with other state of the art methods. Finally, the execution time and memory footprint statistics are given.

4.2. Datasets

Oxford5K and 105K. The Oxford5K dataset was first presented in [18]. It contains 5062 images downloaded from Flickr with Oxford related tags, correctly (sky-is-up) oriented. Additional, ≈100k images of 75 most popular Flickr tags are provided as distractors. These two sets together form Oxford105K dataset.

Paris. The Paris dataset introduced in [19] consists of approximately 6000 images of tourist spots in Paris. It was used to train an independent visual vocabulary to mimic the real-life situation where the sought images are not known to the retrieval system beforehand.

Flickr 5M. In order to show the performance on a large collection of images, we collected a dataset from Flickr that consists of more than 5 million images with tags related to man-made objects such as buildings, favorite landmarks, tourist spots, major cities etc. We have also included images from the Oxford105K dataset.

Holidays dataset. The Holidays dataset was presented in [7]. A set of 1491 personal holiday images in 500 image groups (scenes) is provided together with the ground truth. Most of the images are correctly oriented (or can be with the help of EXIF orientation tag). However about 5%-10% of the images, spread over the groups, are rotated (unnaturally for a human observer). We report the performance on two versions of the dataset, original *Holidays* and *Holidays rotated* where we manually rotated (by 90°, 180° or 270°) outdoor images. In the latter, the correct (sky-is-up) orientation was obvious.

4.3. Evaluation Protocol

The image retrieval benchmark follows the protocol used for the Oxford5K dataset, described in [18]. There are 55 predefined queries (5 per each of 11 landmarks) with ground truth results: images are labeled as *good, ok, junk, bad* depending on the visibility of query object in the image.

The performance in all retrieval experiments is measured using a mean average precision (mAP), the area under precision-recall curves. Precision is defined as the ratio between the number of retrieved positive (*good* and *ok*) images and number of all retrieved images except the images labeled as *junk* (*junk* are treated as if they were not present in the dataset). The recall is then the ratio between the number of retrieved positive and all positive images in the database.

Parameters			Oxford5K vocab.		Paris vocab.	
Method	QE	bits	Ox5K	Ox105K	Ox5K	Ox105K
w/o SP		0	0.717	0.568	0.558	0.423
S0E4		20	0.766	0.708	0.611	0.551
S0E4	✓	20	0.884	0.846	0.771	0.714
S2E2		20	0.767	0.708	0.613	0.557
S2E2	✓	20	0.887	0.846	0.765	0.710
S4E0		20	0.782	0.720	0.630	0.564
S4E0	✓	20	0.890	0.844	0.780	0.723
S0E8		24	0.788	0.725	0.634	0.574
S0E8	✓	24	0.901	0.856	0.784	0.728
S4E4		24	0.787	0.724	0.628	0.563
S4E4	✓	24	0.893	0.848	0.781	0.723
S8E0		24	0.782	0.719	0.633	0.568
S8E0	✓	24	0.892	0.850	0.783	0.726
S4E12		32	0.789	0.726	0.635	0.572
S4E12	✓	32	0.901	0.855	0.783	0.727
Exact		160	0.786	0.723	0.635	0.572
Exact	✓	160	0.900	0.852	0.782	0.725

Table 1. Mean Average Precision (mAP) of different representations of local geometry on Oxford5K and Oxford105K datasets. SxEy encodes the number of bits used for representing of 2^x scales and 2^y shapes. S0Ey denotes representations without separated scale parameter, QE - query expansion.

4.4. Comparison of Geometry Representations

To show the achievable amount of compression, we compare the performance of spatial verification with exact geometry and different setups of proposed geometry representation. Each geometry representation is denoted with parameters x and y of its geometric vocabulary as SxEy, where $L = 2^x$ is the number of scale representatives $\bar{s}_l$ and $K = 2^y$ the number of shape prototypes $\mathbf{B}_j$. The exact geometry data were stored as five single precision numbers – image coordinates and parameters a, b, c of the normalizing transformation.

Geometry representations were learnt on a random subset of 10M features detected in the Oxford5K dataset. Image coordinates were stored as a 16bit index of the nearest node in a regular grid over the image for all compressed representations.

The performance is reported on the Oxford5K and 105K datasets using two different vocabularies trained on the Oxford5K and Paris datasets each with 1M visual words. Additionally, to demonstrate the behavior of separating the scale parameter, we compared a number of geometric vocabularies. The results on all datasets are summarized in Table 1. It is clear that from the S0E8 downwards (more than 24bits for the geometry information per feature) there is an almost negligible difference in the performance for all setups. The shape of an ellipse can be compressed *without a visible drop in the performance* to as few as 8bits. The achievable ratio of compression to the naive representation of exact geometry is more than 6.5. We observed that for a small number of bits it is more important to keep the correct scale of the ellipse and use all bits for encoding of the scale.

Method	Oxford5K vocab.			Paris vocab.		
	Matched	Hyp.	LO	Matched	Hyp.	LO
S0E4	1667	48.35	78.16	1084	19.77	30.51
S2E2	1703	50.52	74.33	1131	20.29	30.06
S0E8	1822	66.30	81.47	1281	23.32	30.05
S4E4	1819	64.08	81.74	1283	24.25	30.12
S4E12	1835	75.37	82.15	1323	26.26	29.72
Exact	1844	75.81	81.92	1328	26.54	29.72

Table 2. Performance comparison of different representations of local geometry in Spatial Verification. Matched - number of matching ground truth pairs with more than 3 inliers (out of 2785 possible), Hyp. - average number of inliers in top five initial hypothesis, LO - average number of inliers after Local Optimization.

The results with S8E0 geometry compression (features represented only by scale and no affine shape) are only marginally worse than results achieved using the affine shape. The results may suggest that affine covariant features are not necessary for image retrieval (on this dataset) and that similarity features would be sufficient. We conclude that the affine shape is not crucial for geometric verification. This observation is not surprising – in many successful image matching approaches, features geometry is represented by a single point. The corrupted geometric information is also alleviated by application of the local optimization step [1]. The merit of the affine covariant features needs to be established in further experiments.

In another experiment, we have measured the number of geometrically verified features for each query and ground truth image pair. In all 55 queries on the Oxford5K dataset, there are 2785 of possibly matching pairs. A set of tentative correspondences was formed for each pair as in standard spatial verification step. Inliers to all geometry hypotheses of RANSAC were computed with each of the geometry representations. The pairs with more than 3 inliers were marked as *Matched* (these can cause re-ranking of the image after tf-idf scoring). For such pairs, the number of inliers of the top five hypothesis was accumulated and averaged over all queries. Additionally, local optimization [1] (final step of spatial verification) was performed for the top five hypotheses on each pair and the number of inliers was accumulated separately. Results are shown in Table 2. We can observe a small drop in the number of *matched* pairs for setups below S0E8 and signifi-

Params		Oxford5K vocab.		Paris vocab.	
GV	QE	Ox5K	Ox105K	Ox5K	Ox105K
		0.772	0.687	0.592	0.501
	✓	0.887	0.844	0.733	0.637
✓		0.786	0.723	0.635	0.572
✓	✓	0.900	0.852	0.782	0.725

Table 3. Comparison of retrieval performance with and without the gravity vector assumption on Oxford datasets, GV - gravity vector assumption, QE - query expansion.

cantly higher than the drop of setups in rows 1 and 2. The average number of inliers ("average quality of hypotheses") also drops slowly for the first phase (Hyp.) of spatial verification, but the drop is negligible after local optimization step (LO). The local optimization step takes all the inliers to the initial hypothesis and re-estimates the affine transformation from all of them. The impact of quantization is minimized, since a large number of features is used.

4.5. Gravity Vector in Image Retrieval

The gravity vector assumption is equivalent to fixing the vertical vanishing point. Assuming that the important objects in the photographs are on vertical planes, the gravity vector assumption disambiguates the rotation of the affine covariant features. In order to solve for feature rotation without the gravity vector assumption, multiple dominant directions are extracted [12]. The experiment summarized in Table 3 shows that the feature rotation obtained from the gravity vector assumption (GV) is more stable than estimate of dominant orientation. This is reflected in better retrieval results in rows 3 and 4 of Table 3. In this experiment, 1.5 dominant directions per feature are detected on average, which naturally leads to 1.5 times larger memory footprint (c.f. Table 6). Overall, the gravity vector assumption both improves the precision of the retrieval and reduces the memory requirements.

In Table 4 we compare the results of our method (S0E8 with gravity vector) with most recent state of the art methods [19, 7, 8] on the Oxford5K and Oxford105K datasets. Results shows that our method with query expansion (and soft assignment) achieves best results on both datasets and with both vocabularies.

4.5.1 Evaluation on a Holidays dataset

The test protocol for the Holidays dataset is based on mAP, but does *not* count the query image. This is necessary on a dataset with a very low average recall according to ground truth – in many cases there are only two images in a group. We report the performance

Method	Oxford5K vocab.		Paris/Other*	
	Ox5K	Ox105K	Ox5K	Ox105K
S0E8	0.788	0.725	0.634	0.574
S0E8+SA	0.846	0.779	0.725	0.652
S0E8+QE	0.901	0.856	0.784	0.728
S0E8+SA+QE	**0.916**	**0.885**	**0.822**	**0.772**
Oxford SP	0.653	0.565	0.460	0.385
Oxford SP+QE	0.801	0.708	0.654	0.562
Oxford SP+SA+QE	0.825	0.719	0.718	0.605
INRIA	-	-	0.547*	-
INRIA TR	-	-	0.610*	-

Table 4. Comparison with state of the art methods. Oxford - mAP results from Table 5 in [19], SP - spatial verification, QE - query expansion, SA - soft assignment. INRIA and INRIA TR, the best results on Oxford5K dataset in [7] resp. [8], *please note that a different vocabulary was used.

Method	Holidays	Holidays rot.
S0E8	0.715	0.765
S0E8+QE	0.736	0.783
S0E8+SA	0.769	0.811
S0E8+QE+SA	0.780	**0.828**
INRIA HE+WGC	0.751	-
INRIA TR (MA+HE+WGC)	**0.810**	-

Table 5. Comparison of performance on INRIA dataset with *Oxford 5K* visual vocabulary. Values are modified mAPs (the query image is not counted). For INRIA and INRIA TR, the best results on Holiday dataset in [7] resp. [8] were taken.

on two datasets, original *Holidays* and *Holidays rotated* in Table 5. We see that even on the original dataset our approach performs reasonably well. On the Holidays rotated dataset, we have achieved results that exceeds the most recent technical report of Jegou et al. [8].

4.6. Time Complexity and Memory Footprint

In this experiment we have measured different properties of the image retrieval system that uses the proposed geometry representation. To achieve even better memory footprint, we implemented a simple label compaction algorithm.

Label Compaction. In our implementation, visual words (indices to visual vocabulary), are stored twice. Once in the inverted file (IF) and once as a list of visual words in a document (along with the geometry in the GL file), which seems to be unavoidable. The inverted file is required for fast scoring, where the list of all documents containing a certain visual words is required. For query expansion, access to all features in the retrieved document is needed too. Without query expansion, it would be sufficient to store only the inverted file. Both the list of documents in the inverted file and the list of visual words in a document are sorted. We com-

Dataset	GV	#imgs	feats	IF(MB)	GL(MB)	B/feat.
Ox5K		5062	18.2	25.01	73.33	5.66
Ox5K	✓	5062	12.5	18.75	51.15	5.84
Ox105K		104933	335.8	389.12	1357.55	5.51
Ox105K	✓	104933	234.3	290.49	960.76	5.60
Holidays	✓	1491	4.8	8.95	19.27	6.21
Holidays r.	✓	1491	4.9	9.10	20.35	6.29
Flickr5M	✓	5050505	9645.7	12247.27	39801.70	5.65

Table 6. Dataset statistics for geometry representation S0E8. GV - gravity vector, feats - number of features (in *millions*), IF - length of inverted file, GL - length of geometry and visual word labels.

Dataset	Machine	SP[s]	SP+QE[s]
Oxford5K	4x3.0Ghz	0.238	0.458
Oxford105K	4x3.0Ghz	0.247	0.509
Flickr5M	8x2.2Ghz	0.727	1.639

Table 7. Average query times without and with query expansion for 55 queries of the Oxford5K benchmark.

bine delta coding with efficient Huffman compression. This results in 11bits per feature in the inverted file on average. The overall memory requirements for each of the datasets are summarized in Table 6. For the largest dataset Flickr5M, we have achieved approximately 46bits per feature with the S0E8 representation.

Finally we have measured the average query time of the 55 queries in Oxford5K benchmark for different datasets (*c.f.* Table 7). Two machines were used, first with 1×Intel 3.0Ghz QuadCore with 32GB memory and second with AMD Opteron 2×2.2Ghz QuadCore with 64GB memory. On the latter machine, even the Flickr5M dataset fits easily into memory.

5. Conclusions

We have proposed a novel method for learning discretized local geometry based on the minimization of the average reprojection error in the space of ellipses. We have shown that the minimization produces a highly compact representation. With 24 bits representing position, elliptical shape and scale (*i.e.* affine transformation modulo rotation) of each feature, image retrieval performance is almost as good as with exact representation of local geometry. We show that the representation naturally incorporates the gravity vector assumption.

Additionally, we have shown that if the gravity vector assumption is used consistently in all stages of image retrieval from feature description to spatial verification, performance is improved and memory footprint is reduced. A method exploiting the local geometry representation outperforms state of the art retrieval algorithms in a standard image retrieval benchmark.

References

[1] O. Chum, J. Matas, and J. Kittler. Locally optimized RANSAC. In *DAGM*, pages 236–243, 2003.

[2] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, 2007.

[3] J. Hays and A. A. Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3), 2007.

[4] http://maps.google.com/help/maps/streetview/.

[5] http://www.flickr.com/.

[6] http://www.panoramio.com/.

[7] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008.

[8] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometry consistency for large scale image search - extended version. Technical report, INRIA, 2008.

[9] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, pages 2169–2178, 2006.

[10] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *CVPR*, volume 2, pages 775–781, 2005.

[11] T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 30(2):79–116, 1998.

[12] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[13] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *ECCV*, pages 128–142, 2002.

[14] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *IJCV*, 65(1-2):43–72, 2005.

[15] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[16] S. Obdrzalek and J. Matas. Object recognition using local affine frames on distinguished regions. In *BMVC*, pages 113–122, 2002.

[17] S. Obdrzalek and J. Matas. Sub-linear indexing for large scale object recognition. In *BMVC*, 2005.

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

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

[20] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

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

The authors were supported by Czech Science Foundation Project 102/07/1317 and by EC project FP6-IST-027113 eTRIMS.