Geometric consistency checks for kNN based image classification relying on local features

Giuseppe Amato
ISTI-CNR
via G. Moruzzi, 1
Pisa, Italy
g.amato@isti.cnr.it

Fabrizio Falchi
ISTI-CNR
via G. Moruzzi, 1
Pisa, Italy
f.falchi@isti.cnr.it

Claudio Gennaro
ISTI-CNR
via G. Moruzzi, 1
Pisa, Italy
c.gennaro@isti.cnr.it

ABSTRACT

Applications of image content recognition, as for instance landmark recognition, can be obtained by using techniques of kNN classifications based on the use of local image features, such as SIFT or SURF. Quality of image classification can be improved by defining geometric consistency check rules based on space transformations of the scene depicted in images. However, this prevents the use of state of the art access methods for similarity searching and sequential scan of the images in the training sets has to be executed in order to perform classification. In this paper we propose a technique that allows one to use access methods for similarity searching, such as those exploiting metric space properties, in order to perform kNN classification with geometric consistency checks. We will see that the proposed approach, in addition to offer an obvious efficiency improvement, surprisingly offers also an improvement of the effectiveness of the classification.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.1 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

Image indexing, image classification, recognition, landmarks, local features

1. INTRODUCTION

An emerging challenge that is recently attracting attention in the field of multimedia information retrieval is that of landmark recognition [22]. It consists in automatically recognizing the landmark (a building, a square, a statue, a monument, etc.) appearing in a non annotated picture. Landmark recognition is particularly appealing for instance in applications for mobile devices, where one wants to obtain information on monuments by simply taking a picture, or automatic annotation of media published on social network services.

The problem of landmark recognition is typically addressed by leveraging on techniques of automatic classification, as for instances kNN Classification [11], applied to image local features, such as SIFT [17] and SURF [8].

In Computer Vision, an interesting problem, that scientists address using local features, is that of automatically locating an object in a test image containing many other objects. To this goal, a transformation able to map the model image on the test image is evaluated on the basis of candidate matches obtained comparing the local features and using various transformation estimation algorithms.

These transformation estimation techniques can also be used with kNN specification to perform a geometric consistency check with the purpose of improving the quality of the image classification. Given an image to be classified, a kNN classifier compares it against the images of a training set, in order to identify the most similar images and consequently the correct class. Geometric consistency checks, as discussed in the rest of the paper, can be used to create image similarity functions that are more effective in deciding that the same landmark is contained in two images. We will see, in fact, that geometric consistency checks offer a performance boost with respect to classification based solely on the presence of interest points in images.

The problem of using similarity functions that are based on the geometric consistency checks is that classification of an image should be performed exhaustively by comparing the image to be classified and all the images in the training set. This is due to the fact that the similarity functions based on geometric consistency checks do not offer nice properties like the metric properties, for instance.

In this paper we will show that kNN classification with geometric consistency checks can be reformulated as a problem of similarity searching executed at the level of the individual local features, rather than entire image. Similarity functions between individual local features are generally metric functions and in most cases are also defined as Euclidian distances. This makes it possible to capitalize on the research and the results obtained in the field of similarity searching in metric spaces [21] to make the kNN classification with geometric consistency checks efficient and scalable.

We will also see that the reformulation that we propose in this paper, in addition of offering higher efficiency and scalability, surprisingly also offers improvement of effectiveness over the exhaustive kNN classifier.

The structure of the paper is as follows. Section 2 presents

SISAP '11 , June 30 - July 01, 2011, Lipari, Italy.

some related work. Local features are introduced in Sections 3. In Sections 5, 6, 7 various approaches and similarity measures are presented. Sections 8 and 9 presents the experimental results.

2. RELATED WORK

In the last few years the problem of recognizing landmarks have received growing attention by the research community. In [18] methods for placing photos uploaded to Flickr on the World map was presented. In the proposed approach the images were represented by vectors of features of the tags, and visual keywords derived from a vector quantization of the SIFT descriptors. In [22], Google presented its approach to building a web-scale landmark recognition engine. Most of the work reported was used to implement the Google Goggles service [1]. The approach makes use of the SIFT feature. The recognition is based on best matching image searching, while our novel approach is based on local features classification.

In [12], various MPEG-7 descriptors have been used to build kNN classifier committees. However local features were not considered. In [10] a survey on mobile landmark recognition for information retrieval is given. Classification methods reported as previously presented in the literature include SVM, Adaboost, Bayesian model, HMM, GMM. The kNN based approach which is the main focus of this paper is not reported in that survey.

In [9] the effectiveness of NN image classifiers has been proved and an innovative approach based on Image-to-Class distance that is similar in spirit to our approach has been proposed.

The bag of visual words model was initially proposed in [19]. In [20] the use of term weighting techniques and classical distances from text retrieval in the case of images has been explored. The experiments show that the effectiveness of a given weighting scheme or distance is strongly linked to the dataset used. In the case of large and varied image collections, the noise in descriptor assignation and the need to use larger vocabularies tend to make all distances and weights equivalent.

An alternative approach to RANSAC for geometry consistency checks based on interest points position has been presented in [16, 15]. Basically, a proximity-based order-respecting intersection is performed after searching in the whole set of local features the most similar to the one extracted from the query.

In [5], we presented four local features based image classification algorithms. These algorithms classify an image in two steps: first each local feature is classified considering the local features of a training set; second the whole image is classified considering the label assigned to each local feature and the confidence of these classifications. In this paper we will not consider this approach because it is very difficult to define any geometric consistency check algorithm on top of them. However, a direct comparison with the results obtained in this paper is given in the Experimental Results settings.

3. LOCAL FEATURES

The approach described in this paper focuses on the use of image local features. Specifically, we performed our tests using the SIFT [17] and SURF [8] local features. In this section, we briefly describe both of them.

The Scale Invariant Feature Transformation (SIFT) [17] is a representation of the low level image content that is based on a transformation of the image data into scale-invariant coordinates relative to local features. Local feature are low level descriptions of keypoints in an image. Keypoints are interest points in an image that are invariant to scale and orientation. Keypoints are selected by choosing the most stable points from a set of candidate location. Each keypoint in an image is associated with one or more orientations, based on local image gradients. Image matching is performed by comparing the description of the keypoints in images. For both detecting keypoints and extracting the SIFT features we used the public available software developed by David Lowe [3].

The basic idea of Speeded Up Robust Features (SURF) [8] is quite similar to SIFT. SURF detects some keypoints in an image and describes these keypoints using orientation information. However, the SURF definition uses a new method for both detection of keypoints and their description that is much faster still guaranteeing a performance comparable or even better than SIFT. Specifically, keypoint detection relies on a technique based on a approximation of the Hessian Matrix. The descriptor of a keypoint is built considering the distortion of Haar-wavelet responses around the keypoint itself. For both detecting interest points and extracting the SURF features, we used the public available noncommercial software developed by the authors [4].

For both SIFT and SURF the Euclidean distance is typically used as measure of dissimilarity between two features [17, 8].

3.1 Local Features Matching

A useful aspect that is often used when dealing with local features is the concept of local feature matching. In [17], a distance ratio matching scheme was proposed that has also been adopted in [8] and many others.

Let us consider a local feature f_i belonging to an image d_i (i.e. $\mathrm{f}_i \in d_i$) and an image d_j. First, the feature $\mathrm{f}_j \in d_j$ that best matches f_i, based on a distance δ, is referred to as the first nearest neighbor (in the remainder $NN_1(\mathrm{f}_i, d_j)$) and is selected as candidate match. Then, the distance ratio $\sigma(\mathrm{f}_i, d_j) \in [0, 1]$ between second-closest and closest neighbors of f_i in d_j is considered. The distance ratio is defined as:

$$\sigma(\mathrm{f}_i, d_j) = \frac{\delta(\mathrm{f}_i, NN_1(\mathrm{f}_i, d_j))}{\delta(\mathrm{f}_i, NN_2(\mathrm{f}_i, d_j))} \quad (1)$$

Finally, f_i and $NN_1(\mathrm{f}_i, d_j)$ are considered matching if the distance ratio $\sigma(\mathrm{f}_i, d_j)$ is smaller than a given threshold. Thus, the set of candidate local features matches between image d_i and d_j is:

$$C^d_{d_i,d_j} = \{(f_i, f_j) \mid f_i \in d_i, f_j \in d_j, \sigma(\mathrm{f}_i, d_j) < c\} \quad (2)$$

In [17] $c = 0.8$ was proposed reporting that this threshold allows to eliminate 90% of the false matches while discarding less than 5% of the correct matches. In [5] an experimental evaluation of classification effectiveness varying c that confirms the results obtained by Lowe, is reported. In the following we will use $c = 0.8$ for both SURF and SIFT.

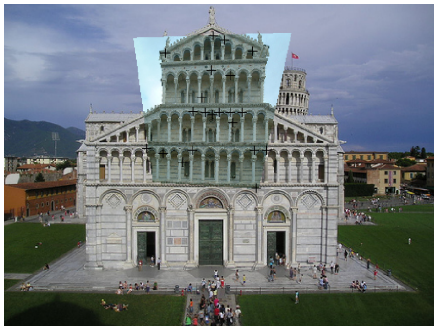
Please note, that this parameter will be used in defining the image to image based similarity measures of Section 5 while it is not necessary for the similarity search approach presented in Section 6.

Image 1

Image 2

Rotation, Scale and Translation

Affine



Homography

Figure 1: Results of searching for matching a portion of Image 2 on Image 1 using various type of transformations found relying on local features and RANSAC.

3.1.1 Geometric Consistency Checks

Each local feature is extracted considering a point of interest in the image and a region around it. The coordinates of the point of interest are associated to the description together with the scale and orientation of the region in the image. In fact, the description of the region itself is scale and orientation invariant because it has been defined for searching similar regions despite changes in scale and/or orientation.

The coordinate of the interest point and the scale and orientation information related to the region can be used to perform consistency checks of the candidate matches. Moreover this information can be used for estimating the transformation able to map one image on top of the other (e.g. for image stitching).

The algorithm used to estimated such a transformation are typically the Random Sample Consensus (RANSAC) [13] and Least Median of Squares. However, fitting methods such as RANSAC or Least Median of Squares perform poorly when the percent of correct matches falls much below 50%. Fortunately, much better performance can be obtained by clustering features in scale and orientation space using the Hough transform.

Hough Transform is used to cluster matches in groups that agree upon a particular model pose. Hough transform identifies clusters of feature by using each feature to vote for all object poses that are consistent with the feature. When clusters of features are found that vote for the same pose of an object, the probability of the interpretation being correct is much higher than for any single feature. In our experiments, we create a Hough transform entry predicting the model orientation, and scale from the match hypothesis. A pseudo-random hash function is used to insert votes into a one-dimensional hash table in which collisions are easily detected. The Hough transform is typically used for increasing the percentage of inliers before estimating a transformation (typically using RANSAC). However, the number of matches in the greater cluster can be considered as an estimation of the actual matches.

Considering the clusters of matches created by the Hough transform it is possible to estimate a transformation able to map the points of an image on the other. Estimating a transformation using RANSAC is a process of: random selecting the requested number of matches for the given transformation estimation; evaluating the transformation itself; and selecting the matches which are consistency with it.

In the following we report the most common types of transformation that can be searched for. In Figure 1 we report the results of transformation estimation for the various types of transformation on a pair of photos of the cathedral of St. Mary in Pisa.

A **Rotation, Scale and Translation** (RST) transformation can be formalized as follows:

$$\begin{bmatrix} p'_x \\ p'_y \end{bmatrix} = \begin{bmatrix} s * cos(\sigma) & -sin(\sigma) \\ sin(sigma) & s * cos(\sigma) \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (3)$$

where σ is the angle of the counter clock rotation, s is the scaling and $\vec{t}$ is the translation. Estimating this transformation requires two couples of matching points ($\vec{p}$ and $\vec{p'}$).

An **Affine** transformation is a linear transformation (rotation, scaling and shear) followed by a translation.

$$\begin{bmatrix} p'_x \\ p'_y \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (4)$$

Please note that a RST transformation is a special type of

a general affine transformation. Affine allows also shearing which leaves fixed all points on one axis and other points are shifted parallel to the axis by a distance proportional to their perpendicular distance from the axis. Estimating this transformation requires three couples of matching points.

A **Homography** is an invertible projective transformation from the real projective plane to the projective plane that maps lines to straight lines. Any two images of the same planar surface in space are related by a homography.

$$\begin{bmatrix} wp'_x \\ wp'_y \\ w \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ 1 \end{bmatrix} \quad (5)$$

where w is a scale parameter. Please note that an affine transformation is a special type of a general homography whose last row is fixed to $h_{31} = 0, h_{32} = 0, h_{33} = 1$. Estimating this transformation requires four couples of matching points.

3.1.2 Isotropic scaling

Typically, the coordinates of the points reported by local features extraction softwares describe the pixel in the original image. However, a normalization not only is useful but can improve the effectiveness of transformation estimation. The most used normalization is the isotropic scaling [14] in which the set of points belonging to an image are translated so as to bring the centroid of the set to the origin, and the coordinates are also scaled so that on the average a point lie a distance $\sqrt{2}$ from the origin.

4. KNN CLASSIFIER

Given a set of documents D and a predefined set of *classes* (also known as *labels*, or *categories*) $C = \{c_1, \ldots, c_m\}$, *single-label document classification* (SLC) [11] is the task of automatically approximating, or estimating, an unknown *target function* $\Phi : D \rightarrow C$, that describes how documents ought to be classified, by means of a function $\hat{\Phi} : D \rightarrow C$, called the *classifier*, such that $\hat{\Phi}$ is an approximation of Φ.

A popular SLC classification technique is the *Single-label distance-weighted kNN*. Given a training set Tr containing various examples for each class c, it assigns a label to a document in two steps. Given a document d_i (an image for example) to be classified, it first executes a kNN search between the objects of the *training set*. The result of such operation is a list $kNN(d_i)$ of labeled documents d_j belonging to the *training set* ordered with respect to the decreasing values of the similarity $s(d_i, d_j)$ between d_i and d_j. The label assigned to the document d_i by the classifier is the class $c_y \in C$ that maximizes the sum of the similarity between d_i and the documents labeled c_y, in the kNN results list $kNN(d_i)$

Therefore, first a score $z(d_i, c_j)$ for each label is computed for any label $c_j \in C$:

$$z(d_i, c_j) = \sum_{d_y \in kNN(d_i) \,:\, \Phi(d_y) = c_j} s(d_i, d_y) \ .$$

Then, the class that obtains the maximum score is chosen:

$$\hat{\Phi}^s(d_i) = \arg\max_{c_j \in C} z(d_i, c_j) \ .$$

It is also convenient to express a degree of confidence on the answer of the classifier. For the *Single-label distance-weighted kNN* classifier described here we defined the confidence as 1 minus the ratio between the *score* obtained by the second-best label and the best label, i.e,

$$\nu_{doc}(\hat{\Phi}^s, d_i) = 1 - \frac{\arg\max_{c_j \in C - \hat{\Phi}^s(d_i)} z(d_i, c_j)}{\arg\max_{c_j \in C} z(d_i, c_j)} \ .$$

This classification confidence can be used to decide whether or not the predicted label has an high probability to be correct.

5. IMAGE TO IMAGE COMPARISON

In order the kNN search step to be executed, a similarity function between images should be defined. Global features, generally, are defined along with a similarity (or a distance) function. Therefore, similarity between images, is computed as the similarity between the corresponding global features. On the other hand, a single image has several local features. Therefore, computing the similarity between two images requires combining somehow the similarities between their numerous local features.

Local features have been used in Computer Vision to identify the same points in two distinct photos of the same object even changing the point of view. Thus, the similarity measure between two images can be easily defined as the percentage of local features in one image that have a match in the other one. Thus, given a set of candidate matches between two images C_{d_i,d_j}, we define the similarity as:

$$s(d_i, d_j) = \frac{|C_{d_i,d_j}|}{|d_i|} \quad (6)$$

In the following we define 5 matching criteria (C^D, C^H, C^R, C^A, C^H) that used in conjunction with Equation 6 result in 5 similarity measures.

Distance ratio matches – C^D
: This set, defined in Equation 2, is used in most of the literature as the first candidate set of matches evaluated on the basis of the local features similarities.

Hough transform matches – C^H
: As mentioned in Section 3.1, an Hough transform is often used to search for keys that agree upon a particular model pose. We define C^H as the subset of matches in C^D related to the most voted pose in terms of orientation and scale. For the experiments, we used the same parameters proposed in [17], i.e. bin size of 30 degrees for orientation, a factor of 2 for scale, and 0.25 times the maximum model dimension for location.

As described in Section 3.1.1, various transformation can be estimated using RANSAC on the clusters of matches identified by the Hough transform. Once a transformation has been estimated, the matches that are not consistent with it are rejected. Typically a threshold e on the distance between the expected (given the transformation) and actual match is used to identify inlier and outliers. Given the normalized coordinate space mentioned in Section 3.1.2, we set $e = 0.1$. Between all the transformation estimated, the one having the greater number of consistent matches is retained.

RST Transform Matches – C^R
: are the matches in C^D that are consistent with the estimated RST transformation (Equation 3).

Affine Transform Matches – C^A
are the matches in C^D that are consistent with the estimated Affine transformation (Equation 4).

Homography Transform Matches – C^H
are the matches in C^D that are consistent with the estimated Homography transformation (Equation 5).

6. A SIMILARITY SEARCH APPROACH

The similarity measures defined in Section 5, which is a direct application of the techniques developed by the Computer Vision community, require the direct comparison of each pair of images. In fact, the distances are not metric and not even symmetric and the complexity of the distance evaluation does not allow any sort of indexing. Thus, given a query, searching for the k nearest images to a given query image require a complete sequential scan of the archive. The first step of all these distances for comparing two images is selecting candidate matches searching for each local feature in one image the $2NN$ in the other one. The candidate matches are then pruned considering the distance ratio defined in Equation 1.

In this section we propose to identify the candidate matches searching for $\bar{k}NN$ between all the local features in all the images in the dataset (D). Please note that $\bar{k}$ used for this NN search is different from the one eventually used for the whole image kNN search. At the end of this process we have for each local feature f_q in the query image d_q a list of candidate matches ($\bar{k}NN(f_q, D)$). Please note that the local features in $\bar{k}NN(f_q, D)$ can belong to distinct images and that the same f_q could have more than one match in the same image. However, having only the best match for each couple of f_q and image d_i is preferable. Thus, we can define the candidate matches between the query image d_q and any $d_i \in D$ as:

$$\bar{C}^D_{d_q,d_i} = \{(f_q, f_i) \mid \mathrm{f}_q \in d_q, \mathrm{f}_i \in \bar{k}NN(f_q, D) \cap d_i, \\ \delta(f_q, f_i) \leq \delta(f_q, f_j),\ \forall f_j \in \bar{k}NN(f_q, D) \cap d_i\} \quad (7)$$

Please note that $\bar{C}^D$ is equivalent, in this scenario, to C^D of Equation 2. Thus, starting from $\bar{C}^D$, it is possible to redefine the five matching criteria of Section 5 that, in conjunction with Equation 6, result in five new similarity measures.

7. THE BAG OF FEATURES APPROACH

In the last few year several object and image retrieval systems which have directly taken a text-based approach to the problem of local features matching have been proposed. Starting from [19], the "visual word" paradigm was introduced which is based on assigning each local feature to a visual word of a predefined vocabulary. At search time, two local features assigned to the same visual words will be considered as matching. The first step to describe images using visual words is to select some visual words creating a vocabulary. The visual vocabulary is typically built grouping local descriptors of the dataset using a clustering algorithm such as *k-means*. The second step is to describe each image using the words of the vocabulary that occur in it.

At the end of the process, each image is described as a set of visual words. Thus, standard text retrieval approaches can be used. In particular the cosine similarity and TF-IDF approaches have been used (e.g. [20]). Using this similarities functions, traditional inverted files can be used for efficiently searching nearest neighbor images.

The bag of features approach can also be used to define a set of candidate matches that can be used as a basis for the geometric consistency checks described in Section 3.1.1:

$$\dot{C}^D_{d_i,d_j} = \{(f_i, f_j) \mid bag(f_i) = bag(f_j)\} \quad (8)$$

$\dot{C}^D$ is equivalent, in this scenario, to C^D of Equation 2 and $\bar{C}^D$ of Equation 7. Thus, starting from $\dot{C}^D$, it is possible to redefine the five matching criteria of Section 5 that, in conjunction with Equation 6, result in five new similarity measures. Please note in this case it is not possible to avoid multiple matches of the same f_i.

8. EXPERIMENTAL SETTINGS

8.1 The Dataset

The dataset that we used for our tests is publicly available and composed of 1,227 photos of 12 landmarks located in Pisa and was used also in [7, 5, 6]. The photos have been crawled from Flickr, the well known on-line photo service. The IDs of the photos used for these experiments together with the assigned label and extracted features can be downloaded from [2].

In order to build and evaluating a classifier for these classes, we divided the dataset in a *training set* (Tr) consisting of 226 photos (approximately 20% of the dataset) and a *test set* (Te) consisting of 921 (approximately 80% of the dataset). The image resolution used for feature extraction is the standard resolution used by Flickr i.e., maximum between width and height equal to 500 pixels. In other words, uploaded photos were originally all bigger than 500 pixels on the maximum side and they have all been resized to 500.

The total number of local features extracted by the SIFT and SURF detectors were about 1,000,000 and 500,000 respectively. The number of local features per image varies between 113 and 2816 for SIFT and 50 and 904 for SUFR.

8.2 Performance Measures

For evaluating the effectiveness of the classifiers in classifying the documents of the *test set* we use the micro-averaged *accuracy* and micro- and macro-averaged *precision*, *recall* and F_1.

Micro-averaged values are calculated by constructing a global contingency table and then calculating the measures using these sums. In contrast macro-averaged scores are calculated by first calculating each measure for each category and then taking the average of these. In most of the cases we reported the micro-averaged values for each measure.

Precision is defined as the ratio between correctly predicted and the overall predicted documents for a specific class. *Recall* is the ratio between correctly predicted and the overall actual documents for a specific class. F_1 is the harmonic mean of *precision* and *recall*.

Note that for the *single-label* classification task, micro-averaged *accuracy* is defined as the number of documents correctly classified divided by the total number of documents of the same label in the *test set* and it is equivalent to the micro-averaged *precision*, *recall* and F_1 scores.

			Image to image comparison					Similarity search approach $\bar{k}$=10				
			d-ratio	Geometric consistency check				$\bar{C}^{D}_{d_q,d_i}$	Geometric consistency check			
				Hough	RST	Affine	Hom.		Hough	RST	Affine	Hom.
***k*=1**	**Accuracy**	**SIFT**	0.877	0.912	0.931	**0.939**	0.922	0.880	**0.948**	0.941	0.943	0.931
		SURF	0.807	0.870	0.907	**0.920**	0.905	0.859	0.909	0.928	**0.935**	0.893
	F_1 Macro	**SIFT**	0.864	0.899	0.924	**0.935**	0.843	0.876	**0.946**	0.938	0.868	0.865
		SURF	0.788	0.845	0.902	**0.917**	0.835	0.842	0.903	**0.924**	0.856	0.836
***k*=10**	**Accuracy**	**SIFT**	0.864	0.901	0.936	0.929	0.923	0.875	0.939	0.940	0.946	0.933
		SURF	0.841	0.871	0.899	0.911	0.908	0.850	0.903	0.928	0.928	0.898
	F_1 Macro	**SIFT**	0.843	0.879	0.929	0.923	0.837	0.854	0.932	0.937	0.868	0.866
		SURF	0.818	0.849	0.888	0.899	0.841	0.823	0.892	0.922	0.846	0.839
Best	**Accuracy**	**SIFT**	0.877	0.916	0.936	**0.941**	0.923	0.877	**0.948**	0.943	0.947	0.936
		SURF	0.851	0.887	0.910	**0.920**	0.912	0.861	0.905	0.928	**0.936**	0.898
	F_1 Macro	**SIFT**	0.864	0.904	0.929	**0.937**	0.843	0.868	**0.946**	0.939	0.868	0.868
		SURF	0.828	0.867	0.904	**0.917**	0.845	0.844	0.897	**0.924**	0.858	0.840
Best *k*	**Accuracy**	**SIFT**	*1*	*2*	*8*	*3*	*9*	*2*	*1*	*9*	*7*	*12*
		SURF	*20*	*21*	*4*	*1*	*4*	*3*	*4*	*1*	*2*	*3*
	F_1 Macro	**SIFT**	*1*	*2*	*8*	*3*	*1*	*2*	*1*	*9*	*9*	*12*
		SURF	*18*	*21*	*4*	*1*	*4*	*3*	*4*	*1*	*3*	*3*

Figure 2: Image Similarity Based Classification Results using the image to image and similarity search approaches for $\bar{k} = 10$.

9. EXPERIMENTAL RESULTS

In Figure 2 we report the results obtained by both the image to image comparison and similarity search approaches. Accuracy and macro averaged F_1 are reported for both SIFT and SURF. Given that the kNN require a parameter k we report the results obtained for $k = 1, 10$ and the best results obtained for $k \in [1, 100]$.

Comparing the results obtained by the various similarity functions for the image to image comparison approach, we can see that geometric consistency checks are able to significantly improve the quality of the classification process. The best results are obtained by searching for an affine transformation. This is consistent with the fact that both SIFT and SURF are affine invariant. It is worth to say that the benefits of consistency checks are more relevant for SURF even if its overall performance remains below the one obtained using SIFT. Regarding the local features used and the computational cost, note that say that the number of local features detected by the SIFT extractor is twice the ones detected by SURF. Thus, on one hand SIFT has better performance while on the other hand SURF is more efficient.

In Figure 2 we also report the results obtained by the similarity search approach using $\bar{k} = 10$, i.e., performing a $10NN$ for each local feature in the query over the local features in the training set. In our experiments we also tested $\bar{k} = 30, 50, 100$ obtaining comparable but worst results. Surprisingly the similarity search approach often performs better than the image to image comparison. The intuition is that the $\bar{k}NN$ search performed between all the local features in the training set is able to reduce the number of false matches. Please note, that this approach is also more efficient because the local features compared using the Euclidean distance are indexable while the whole image are not.

In the case of the similarity search approach the choice of the geometric consistency check is more problematic. In particular both Homography and Affine reveal a big loss in F_1 while Hough perform significantly better because of the loss noisy first step matches. However, the overall best is RST. The intuition is that we have less first step matches but less noise resulting in better results with a geometric consistency check that only require two matches for the transformation evalution.

In Figure 3 we report the results obtained by the bag of features approach, described in Section 7 using a vocabulary of $100k$ features selected using the k-means algorithm. As known in the literature, typically the more the words, the better the results. In our experiments we are dealing with a dataset of about 1 million features. Thus, $100k$ of visual words is the highest value for which it does make sense to perform a clustering algorithm. The results are worst than one obtained before. Moreover, the geometric consistency checks do not allow significantly gains in performance especially considering F_1. The intuition is that the candidate matches found using the bag of features approach are too much noisy. Standard cosine and TF-IDF similarity measure are more suitable for this scenario. It is worth to note, that the k-means algorithm for selecting the $100k$ words was performed over the whole dataset while it would have been more correct to only consider the training images. In fact, the test images should not be used during any training phase. However, we preferred to compare our approach

			Bag of features					
			cosine	cosine TF-IDF	Geometric consistency check			
					Hough	RST	Affine	Hom.
k=1	Accuracy	SIFT	0.863	0.875	0.877	0.878	0.812	**0.882**
		SURF	0.853	0.845	0.849	0.851	0.820	**0.857**
	F_1 Macro	SIFT	**0.879**	0.869	0.869	0.870	0.805	0.800
		SURF	**0.839**	0.829	0.750	0.750	0.725	0.787
k=10	Accuracy	SIFT	0.856	**0.888**	0.866	0.868	0.817	0.887
		SURF	0.851	**0.862**	0.872	0.868	0.857	0.855
	F_1 Macro	SIFT	0.845	0.849	0.852	**0.854**	0.804	0.804
		SURF	0.835	**0.848**	0.778	0.769	0.754	0.785
Best	Accuracy	SIFT	0.885	**0.896**	0.878	0.882	0.832	0.891
		SURF	0.858	0.863	0.876	**0.878**	0.828	0.859
	F_1 Macro	SIFT	0.871	**0.873**	0.869	0.873	0.821	0.811
		SURF	0.843	**0.849**	0.779	0.783	0.758	0.789
Best k	Accuracy	SIFT	*7*	*8*	*2*	*4*	*4*	*6*
		SURF	*3*	*9*	*15*	*7*	*13*	*2*
	F_1 Macro	SIFT	*3*	*3*	*1*	*2*	*4*	*6*
		SURF	*7*	*9*	*15*	*7*	*13*	*2*

Figure 3: Classification Results using the Bag of Features approach with a vocabulary of 100k features.

in this scenario even if the bag of features performance are actually overestimated.

In [5], we presented four local features based image classification algorithm that classify an image in two steps: first each local feature is classified considering the local features of a training set; second the whole image is classified considering the label assigned to each local feature and the confidence of these classifications. The results obtained are very similar to the best results obtained here. In particular, the Weighted LF Distance Ratio Classifier, which is the best performing algorithm in [5], obtained 0.928 in *accuracy* and 0.922 in F_1 using SURF, which are slightly worst than the values obtained by the similarity search approach proposed in this paper considering the RST consistency check. Regarding, SIFT the results obtained by the Weighted LF Distance Ratio Classifier were 0.952 in *accuracy* and 0.947 in F_1 which are slightly better than the measures obtained by the similarity search approach with RST. It is worth to note that even if the results are very similar, the geometric consistency check also results in a transformation estimation that could be necessary in some scenarios as, for instance, in augmented reality. It would be interesting to add geometric consistency check to the algorithms proposed in [5], but their local feature classification approach results in very difficult geometric consistency check definition.

10. CONCLUSIONS

In this paper we have presented a techniques that allows performing kNN classification of images by also performing geometric consistency checks of the scenes appearing in images. The proposed approach allows executing classification efficiently relying on the use of access methods for similarity searching, such as those exploiting metric space properties. We have performed an extensive experimentation of the proposed approach and we have shown that it offers higher effectiveness than basic kNN classification that simply uses percentage of matches. In the tests we have compared various solutions for geometric consistency checks both exhaustively sequentially scanning all images in the training set and by using our method relying on similarity searching. From these tests we have also surprisingly observed that the proposed approach, based on similarity search, offers better effectiveness than the exhaustive and non scalable approach.

11. ACKNOWLEDGMENTS

This work was partially supported by the VISITO Tuscany project, funded by Regione Toscana, in the POR FESR 2007-2013 program, action line 1.1.d, and the MOTUS project, funded by the Industria 2015 program.

12. REFERENCES

[1] Google goggles. `http://www.google.com/mobile/goggles/`. last accessed on 30-March-2010.

[2] Pisa landmarks dataset. `http://www.fabriziofalchi.it/pisaDataset/`. last accessed on 3-March-2011.

[3] SIFT keypoint detector. `http://people.cs.ubc.ca/~lowe/`. last accessed on 3-March-2011.

[4] SURF detector. `http://www.vision.ee.ethz.ch/~surf`/. last accessed on 3-March-2011.

[5] G. Amato and F. Falchi. kNN based image classification relying on local feature similarity. In *SISAP '10: Proceedings of the Third International Conference on SImilarity Search and APplications*, pages 101–108, New York, NY, USA, 2010. ACM.

[6] G. Amato and F. Falchi. Local feature based image similarity functions for kNN classfication. In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence (ICAART 2011)*, pages 157–166. SciTePress, 2011. Vol. 1.

[7] G. Amato, F. Falchi, and P. Bolettieri. Recognizing landmarks using automated classification techniques: an evaluation of various visual features. In *in Proceeding of The Second Interantional Conference on Advances in Multimedia (MMEDIA 2010)*, pages 78–83. IEEE Computer Society, 2010.

[8] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *In ECCV*, pages 404–417, 2006.

[9] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *CVPR*. IEEE Computer Society, 2008.

[10] T. Chen, K. Wu, K.-H. Yap, Z. Li, and F. S. Tsai. A survey on mobile landmark recognition for information retrieval. In *MDM '09*, pages 625–630. IEEE Computer Society, 2009.

[11] S. Dudani. The distance-weighted k-nearest-neighbour rule. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-6(4):325–327, 1975.

[12] T. Fagni, F. Falchi, and F. Sebastiani. Image classification via adaptive ensembles of descriptor-specific classifiers. *Pattern Recognition and Image Analysis*, 20:21–28, 2010.

[13] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

[14] R. I. Hartley. In defence of the 8-point algorithm. In *Proceedings of the Fifth International Conference on Computer Vision*, ICCV '95, pages 1064–, Washington, DC, USA, 1995. IEEE Computer Society.

[15] T. Homola, V. Dohnal, and P. Zezula. Proximity-based order-respecting intersection for searching in image databases. In *In Proceedings of the 8th International Workshop on Adaptive Multimedia Retrieval (AMR 2010)*, 2010.

[16] T. Homola, V. Dohnal, and P. Zezula. Sub-image searching through intersection of local descriptors. In *Proceedings of the Third International Conference on SImilarity Search and APplications*, SISAP '10, pages 127–128, New York, NY, USA, 2010. ACM.

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[18] P. Serdyukov, V. Murdock, and R. van Zwol. Placing flickr photos on a map. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 484–491, New York, NY, USA, 2009. ACM.

[19] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the Ninth IEEE International Conference on Computer Vision - Volume 2*, ICCV '03, pages 1470–, Washington, DC, USA, 2003. IEEE Computer Society.

[20] P. Tirilly, V. Claveau, and P. Gros. Distances and weighting schemes for bag of visual words image retrieval. In *Proceedings of the international conference on Multimedia information retrieval*, MIR '10, pages 323–332, New York, NY, USA, 2010. ACM.

[21] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search: The Metric Space Approach*, volume 32 of *Advances in Database Systems*. Springer-Verlag, 2006.

[22] Y. Zheng, M. Z. 0003, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven. Tour the world: Building a web-scale landmark recognition engine. In *CVPR*, pages 1085–1092. IEEE, 2009.