Improving SURF Image Matching Using Supervised Learning

Hatem Mousselly Sergieh*, Elöd Egyed-Zsigmond*, Mario Döller†, David Coquil†, Jean-Marie Pinon*and Harald Kosch†
*INSA de Lyon
7, Avenue Jean-Capelle, 69621 Villeurbanne, France
Email:(firstname.lastname)@insa-lyon.fr
†Universität Passau
Innstrasse 43, 94032 Passau,Germany
Email:(firstname.lastname)@uni-passau.de

Abstract—**Keypoints-based image matching algorithms have proven very successful in recent years. However, their execution time makes them unsuitable for online applications. Indeed, identifying similar keypoints requires comparing a large number of high dimensional descriptor vectors. Previous work has shown that matching could be still accurately performed when only considering a few highly significant keypoints. In this paper, we investigate reducing the number of generated SURF features to speed up image matching while maintaining the matching recall at a high level. We propose a machine learning approach that uses a binary classifier to identify keypoints that are useful for the matching process. Furthermore, we compare the proposed approach to another method for keypoint pruning based on saliency maps. The two approaches are evaluated using ground truth datasets. The evaluation shows that the proposed classification-based approach outperforms the adversary in terms of the trade-off between the matching recall and the percentage of reduced keypoints. Additionally, the evaluation demonstrates the ability of the proposed approach of effectively reducing the matching runtime.**

I. Introduction

Image matching based on local features such as SIFT [1], and SURF[2] has tremendously improved the quality of content-based image retrieval (CBIR) applications. State-of-the-art algorithms are now able to reliably discover image correspondences under different conditions, such as perspective, illumination and scale changes. However, their performance is unsuitable for online applications. Indeed, finding keypoint correspondences implies comparing thousands of high dimensional descriptor vectors, which results in a serious performance bottleneck.

While in many applications, such as CBIR and searched-based image annotation, image features can be extracted offline, finding images similar to a query image must still be done online. To improve the performance of this process, a good solution consists of reducing the number of keypoints used for the matching. Indeed, it has been shown that a good performance could still be achieved when considering a selected subset of the extracted local features [3][4].

In the literature, characterizing and pruning SIFT keypoints was the focus of several works (for example: [3][5][6]), while a less effort has be done on characterizing SURF keypoints. SURF is a keypoint extraction and description algorithm that provides a comparable alternative for SIFT and requires much less processing time for detecting and matching keypoints [7]. This is due to using integral image technique and the smaller descriptor size [2] (typically, the SURF descriptor consists of 64-bins which is half the size of the SIFT descriptor). A further step to increase the runtime efficiency of SURF-based image matching, can be achieved by identifying the characteristics of the keypionts which contribute more to the matching.

In this paper, we consider further improving the efficiency of SURF-based image matching. For this purpose, we propose a machine learning approach for keypoint pruning which aims to identify the keypoints that are important for the matching. Consequently, image matching can be reduced to finding keypoint correspondences in the subsets of important keypoints. To achieve this, we build a binary classifier that is able to distinguish between two classes of keypoints: *significant* and *insignificant*. For this purpose, a Random Forest classifier [8] is fed with training data consisting of labeled keypoints extracted from large collection of images spanning various categories. The decision on the importance of a keypoint is performed by running the feature vector of a test keypoint on the learned model to predict its usefulness for the matching.

The proposed solution should be very effective in the context of image retrieval. Indeed, the computational overhead caused by the training and prediction processes only has minimal impact since they can be performed offline. The only online component, matching local features, should be much more efficient as only a small number of important descriptors needs to be processed.

To verify the efficiency of the proposed approach, we compare it to other approaches for keypoint pruning that also investigate the SURF feature [4][9][10]. These approaches follow a common method for reducing the number of generated SURF keypoint, which is based on visual attention and saliency maps [11]. The basic idea in these approaches is to identify conspicuous regions in the image by simulating the way in which humans perceive images. Correspondingly, a saliency map is generated in which image regions are stressed differently according to their visual importance to a human viewer. Consequently, the number of keypoints extracted from an image can be reduced by applying the extraction algorithm only on image regions with high saliency values.

The effectiveness of the introduced keypoint pruning approach is evaluated in terms of the trade-off between the loss in matching recall and the ratio of pruned keypoints. The results show that our approach clearly outperforms the saliency maps approach. Furthermore, we show the achieved matching accuracy as well as the runtime improvement based on a dataset that is totally different from the training dataset.

The rest of the paper is organized as follows: in the next section, related work is reviewed. In Section III keypoint pruning based on Random Forest classifier is presented. In Section IV keypoint pruning using saliency maps is briefly discussed. Several evaluation studies are provided in Section V. We conclude and discuss future work in the last Section.

II. Related Work

Applications for finding similar images have proven very successful since the introduction of the groundbreaking SIFT feature in 2004 [1]. Earlier works focused on efficiently indexing the huge number of produced descriptors. For instance, Ke et al. [5] proposed using locality-sensitive hashing to index SIFT descriptors. Recently, the Bag of Words (BoW) representation has seen much use, as it provides a faster matching [12][13].

An additional step to speed up image matching is made by works that deal with the problem of reducing the number of generated keypoints. Foo et al. [3] proposed a strategy for reducing the number of SIFT keypoints based on their contrast (intensity) values. They showed that an efficient matching can be achieved by ranking the keypoints according to their contrast values and selecting the top N. Another method for identifying if a feature is useful for the matching was introduced by Turcot and Lowe [14]. The usefulness of a SIFT feature of a query image is determined through its counterparts in other matching images. This is achieved by using RANSAC to determine if the feature is geometrically consistent. This method reduces the number of features by more than an order of magnitude without compromising the quality of the matching. In more recent work on near-duplicates, Dong et al. [6] showed that SIFT features with near-empty regions are a major source of false positives, and argued that therefore only features with rich internal structure should be used for near-duplicate selection. Thus, they proposed to apply entropy-based filtering of SIFT features. Experiments showed that a single match of filtered keypoints suffices to decide that two images are near-duplicates. However, the authors did not discuss the impact of their approach on speeding up the matching. Indeed, no information were given on the percentage of the keypoints that can be discarded from the matching when using their approach.

Using visual attention to provide faster image matching was considered by several works. In [4] the authors use saliency maps based on Itti's model [11] to identify salient SURF keypoints. They then considered the calculated saliency values to decide whether two keypoints match. In [9] the authors propose another method for generating a retina-optical saliency map by using local phase information of the input data. The saliency maps were used to filter SIFT and SURF features for the purpose of matching video sequences of a robot-navigation system. Another method for generating saliency maps was proposed by [10]. In this work phase Fourier transform is used to construct the saliency map. To enable faster scene matching, SURF keypoints are extracted only from the salient image regions.

Characterizing local image features for object recognition and tracking using supervised learning was addressed by Le Petit and his colleagues [15][16]. For this purpose, a training dataset was created by generating different views of the same object and extracting keypoints in each view. The authors used multi-label classification based on randomized trees to extract keypoint signatures that enable tracking the keypoints of an object under different view changes.

The work presented in this paper is distinguished from the above reviewed works in several points. In contrast to other approaches which consider the SIFT feature as in [3] [14] [6], the main focus of our work is on characterizing SURF keypoints and pruning them accordingly. Furthermore, we follow a different approach which is based on classification using a generic training dataset.

The proposed classification-based keypoint pruning share the idea of using supervised learning with the work presented in [15]. In that work a multi-class classifier is used to enable object identification in video scenes, while in our work we apply binary classification. We claim that binary classification is more convenient for the case of image matching since we are concerned in identifying similar images disregarding the particular objects where the visual similarity is identified. Additionally, in our approach we use Random Forest classification while in [15] randomized trees are used. Moreover, in our work we consider characterizing SURF keypoints using other global image features in place of the SURF descriptor and provide different evaluation studies.

The adversary saliency maps approach is in line with the proposal found in [4]. However, we further investigate the effect of varying the saliency thresholds on the trade-off between the ratio of reduced keypoint and the matching recall.

III. Supervised Keypoint Pruning

Our goal in this paper is to speed up image matching based on the SURF feature by classifying the extracted keypoints in two categories: 1) Significant keypoints, which are highly salient to distortions; correspondences between such keypoints in different images are strong indicators of similarity. 2) Insignificant keypoints, which are less salient, do not contribute much for establishing visual similarity, and can be practically excluded from the matching process.

This problem can be addressed by creating a binary classifier that distinguishes between the two categories of keypoints. Formally, suppose we have an input image I. A set of keypoints K can be extracted from I using an algorithm such as SURF. Each keypoint $k_i \in K$ can be described by a set of features F extracted from a patch of width w centered on the keypoint and denoted as $Q_w^F(k_i)$. For this work, we investigated F that consists of only one feature (This includes the SURF descriptor as well as other global image feature. See III-D). Furthermore, suppose that there exists a perfect labeling function that assigns a label for each keypoint based on the associated feature patch: $Y(Q_w^F) \in L$ and $L = \{-1, 1\}$, where -1 corresponds to the insignificant class and 1 to the significant one. Y cannot be directly derived, but can be approximated by a classifier $\hat{Y}$ that can predict the label of a keypoint with a minimum bias ϵ from the perfect labeling function:

$$P(Y(Q_w^F) \neq \hat{Y}(Q_w^F)) < \epsilon$$

To build the classifier $\hat{Y}$ a training dataset is required. A training instance corresponding to a keypoint k is the tuple $k(x, y, Q_w^F, Label)$ in which (x, y) are the coordinates of the keypoint in the image, Q_w^F is the feature patch and $Label$ is the class of the keypoint. Once the classification model has been learned, it can be used to determine the usefulness of a test keypoint k' for the matching. For this purpose, the same set of features which is used in the training process of the classifier is extracted from a patch ${Q'}_w^F$ centered on k' and passed to the classifier. k' will be considered by the matching if $Y({Q'}_w^F) = 1$ and discarded when $Y({Q'}_w^F) = -1$.

In the following sections the process of building a keypoint classifier and the associated challenges are discussed in detail.

A. Training Data Set

We build a training dataset from a collection of 1100 groups of images taken from the Object Recognition Benchmark dataset [12]. Each group consists of 4 images that are visually similar, i.e., depicting the same objects but taken from different perspectives and under different illumination conditions (Figure 1).

Labels for the keypoints are assigned by matching the corresponding images according to the SURF algorithm. For each group, an image is selected randomly and matched with the other three images in the same group. A keypoint is labeled as *significant* if it has a correspondence in each one of the other three images. Other keypoints that do not fulfill this condition are labeled as *insignificant*.

B. Diversity of the Training Set

While building the training dataset, it is very likely that feature patches will contain redundancy if the corresponding keypoints, which are extracted from the same image, are spatially close to each other. To avoid this, a distance based filtering is applied. Let I denote an image and K_I the set of keypoints extracted from I. The training keypoints were selected randomly. For every pair of keypoints $k_1, k_2 \in K_I$ with the same label, we ensured that the distance between them is higher than a threshold d, i.e., $dist(k_1, k_2) > d$, where $dist$ is the distance function. In this work we used the Euclidean distance and we set d to 5 pixels as proposed in [17].

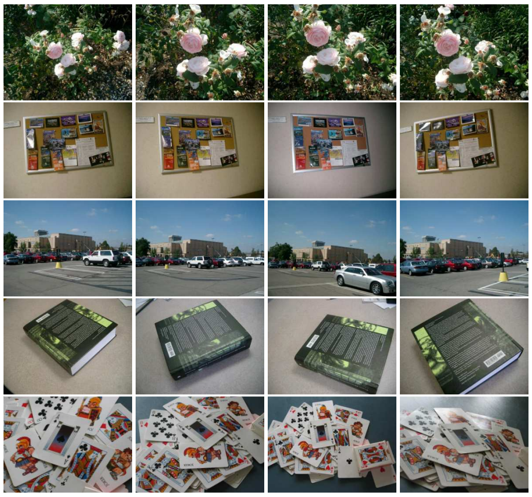

Fig. 1. Examples of images [12] used for creating the training dataset

C. Dealing with Imbalance

Labeling keypoints using the above described approach leads to an imbalanced training dataset in which the number of insignificant instances greatly exceeds the number of significant ones. In our training dataset less than 5% of the training instance were labeled as significant. Classification based on an imbalanced training dataset leads to poor classification results [18] because the classifier tends to assign all instances to the majority class (in our case the insignificant class).

To address this problem we created a balanced training dataset by sampling the original one. We investigated different sampling configurations to create a training sample. We realized that classifiers trained with a dataset created by using *random sampling without replacement* provide the best performance. In fact, creating a balanced sample without allowing replacement leads to a training sample of smaller size than the original training dataset. If all instances of the minority class are included, the size of the training sample will be at most twice the number of instances in the minority class. In contrast, random sampling with replacement allows larger training datasets, and generally, leads to a classifier with a smaller generalization error compared to the case in which no replacement was performed. However, not allowing replacement ensures that the training is performed on real instances and not on pseudo ones as in other replacement approaches, which can lead to a more accurate classifier. Indeed, our experiments with real test scenarios showed that the classifiers trained with random sample without replacement provided the best results.

D. Classification Features

The training instances must be described according to distinctive features before they can be fed into the learning phase. The quality of the features has a direct influence on the performance of the classifier. For this purpose, we investigated different kinds of image features extracted from squared patches centered on the keypoints. The set of features that we investigated includes the SURF descriptor itself, as well as other color and texture features, which we describe below.

SURF and Reduced SURF: The SURF descriptor has 64 dimensions and is generated by calculating Haar wavelet responses in 4×4 oriented square sub-regions centered on the keypoint [2]. We used the SURF descriptor to characterize the keypoints and extended it with four additional attributes: 1) the strength of the keypoint which represents the intensity of the corresponding blob. Positive values indicate dark blobs while negative values indicate light ones, 2) the Gaussian scale at which the keypoint was discovered, 3) the trace of the Gaussian matrix used to discover the keypoint blob and 4) the orientation of the keypoint. The final feature vector has 68 components.

Furthermore, we applied attribute selection on the generated SURF descriptor to identify the most important classification attributes. For this purpose, we used the RELIEF-F [19] method for feature selection. RELIEF-F is considered one of the most effective feature selection algorithms. The basic idea is to select instances at random from the training set and calculate their nearest neighbors, and adjust a feature weighting vector so that a higher weight is given to features that discriminate the instance from neighbors of different classes. The algorithm identified 48 distinctive attributes of the SURF descriptor. We coined the new feature as Reduced SURF and used it to train a keypoint classifier.

Color Histogram: it represents the color distribution of an image (or a region of it). It is calculated by first defining a number of ranges (bins) for each color component of the considered color space and second by counting the number of pixels falling in each bin. We used the RGB color space and for each of the three color channels we used 8 bins. The final histogram contains $8^3 = 512$ bins in total.

Color and Edge Directivity Descriptor (CEDD) combines both color and texture features in one histogram [20]. It allows fast image retrieval since it is limited to 54 bytes per image. CEDD is created by splitting the image in a predefined number of blocks and calculating the color histogram of each block over the HSV color space. A 24-bins histogram of five different colors is generated for every block. After that, 5 filters are used to extract the texture information related to the edges present in the image; the extracted edges are classified in vertical, horizontal, 45-degree diagonal, 135-degree diagonal and non-directional edges. The final descriptor consists of 144 bins.

Fuzzy Color and Texture Histogram (FCTH) also combines color and texture information in one quantized histogram [21]. First, a 10-bins histogram is extracted for 10 colors in the HSV color space. These colors are preselected based on the positions of the vertical edges in each channel. The histogram is extended to 24 bins by separating each color in 3 hues: dark color, color and light color. Then, Haar Wavelet transform is applied to the luminance component Y of YIQ color space. Finally, the extracted texture information is used to enrich the feature resulting in feature histogram of 192 bins.

Joint Composite Descriptor (JCD) exploits the fact that the color information of FCTH and CEDD are extracted from the same fuzzy system and combines the texture area of the two features [22]. JCD is built from 7 texture areas with each area made up of 24 sub regions corresponding to color areas. The feature histogram consists of 168 bins.

E. Classification Using Random Forest

Random Forest (RF) [8] is a state-of-the-art machine learning method that belongs to *ensemble learning* algorithms, which aggregates the predictions of many classifiers. The effectiveness of RF can be compared to that of other powerful classifiers, such as support vector machines (SVMs). Moreover, RF avoids overfitting the training data and generates an unbiased estimate of the generalization error.

Briefly, an RF classifier works as follows: a "forest" is built from a collection of n tree classifiers. Each tree is built from bootstrapped sample of the training data. In contrast to traditional classification trees [23], in which the best split for a tree is selected from all provided predictors, trees in RF are grown by choosing the best split predictor out of a random selection of m predictors. The leaf nodes of each classification tree contains the posterior distribution of the classes.

Formally, let $T = \{T_i\}_{i=1}^{n}$ denote the set of trees in the forest and $N_i = \{N_{i,j}\}_{j=1}^{k_i}$ the set of leaf nodes of the i^{th} tree where k_i is the number of the corresponding leaf nodes. Furthermore, let C be the set of available classes. Now, a leaf node $N_{i,j}$ contains the distribution of the classes at that node $P_{N_{i,j}}(C)$. To predict the class of a new feature vector v, it is dropped over each tree in the forest until it reaches a leaf node $N_{i,v}$ in each tree. The class C of the feature vector v is determined by a majority voting of the n trees and is given by the following conditional probability:

$$P(C|v) = \underset{C}{\operatorname{argmax}} \frac{1}{n} \sum_{i=1}^{n} P_{N_{i,v}}(C)$$

RF classifiers have two interesting features. First, they provide a robust generalization error estimate. This is achieved during the training phase as follows: while building the trees, about one third of the instances of the training sample are left out; these are called out-of-bag (OOB) instances. In turn, OOB instances are used as a testing sample for the corresponding tree classifier. The proportion of OOB instances which are misclassified indicates the classification error of the corresponding tree. By averaging the OOB errors over the whole set of tree classifiers, a generalization error of the RF can be obtained. According to this characteristics, RF requires no cross-validation since this is already performed while building the forest. The second feature is that RF classifiers have the advantage of ease of configuration. Only two parameters have to be set: the number of trees n and the number of predictors m from which a split for the tree is randomly selected.

IV. Saliency based Keypoint Pruning

Visual attention [11][24] is a model that simulates human perception of visual scenes. When analyzing a visual scene, humans start by selecting regions that seem most relevant to them (salient), and then conduct finer/higher-level activities (such as object recognition) on these regions only. Inspired by this, Itti et al. [11] proposed a computational model for visual attention composed of the following stages. First, simple color, intensity and orientation features of an input image are extracted using the center-surround technique. After that, for each feature a corresponding feature map is created. Finally, feature maps are fused in a saliency map which topographically codes the conspicuous locations in the input image.

We used an implementation[1] of Itti's model to create saliency maps of images. The maps are used to reduce the number of generated keypoints by applying the keypoint extraction algorithm on the regions where the saliency values exceed a predefined threshold.

V. Evaluation

In the following sections the proposed keypoint pruning approach is evaluated from different perspectives. First, we show the classification accuracy of the used Random Forest classifiers. Next, the performance of image matching using our keypiont pruning approach is evaluated and compared to other approaches. Finally, a runtime analysis is provided.

A. Classification Performance

We built a training dataset consisting of about 10K keypoint instances distributed uniformly on the significant and insignificant classes. Each keypoint was characterized by each of the features presented in Section III-D. The features were extracted from squared patches centered on the keypoints. The width of the patch has a great effect on the classification results. Too small patches make the extracted features miss information about the corresponding keypoints while the inverse happens when too big patches are used. To address this problem, we identified the size of the patch dynamically and followed the method used for generating the SURF descriptor [2]. We set the width of the patch to 6 times the scale σ at which the keypoint has been discovered.

For each individual feature, we trained a Random Forest classifier using 100 trees. Additionally, we set the number of randomly selected features to log_2*(number of feature attributes)*. The accuracy of the classifiers was evaluated by the OOB error rate. The results shows that the classifier which was trained using the Reduced SURF feature provided the smallest OOB rate of about 24%. The other classifiers which were trained by each of the remaining features (SURF, CH, CEDD, FCTH, and JCD) separately, provided a comparable OOB values around 36%.

B. Effectiveness of Keypoint Pruning

Before listing the evaluation results we discuss some considerations related to using classification as well as saliency maps for keypoint pruning.

1) Keypoint Classification Considerations: For a given keypoint (described by a feature patch), the classifier tells with which probability the keypoint belongs to each class. Consequently, a keypoint is assigned to the class with the highest probability (in binary classification this corresponds to the class with a probability higher than 50%). However, due to the classification errors, the classifiers cannot provide perfect predictions. Therefore, the probability value at which a decision is made to assign a certain class to a keypoint is crucial for the matching performance. To understand this, let us assume that the classifier predicts that a great part of the keypoints of an image belongs with 50.1% to the insignificant class. This means that most of the keypoints will be discarded from the matching although the classifier is not quite sure about the membership of the keypoints (only for 50.1%). To address this problem, the matching performance is evaluated under different decision thresholds. Hereby, a keypoint is assigned to a certain class if the output of the classifier tells that the keypoint belongs to that class with a probability higher than a threshold t. Otherwise, the keypoint is considered to belong to the other class.

2) Saliency Map Considerations: A saliency map is a matrix which contains for each pixel in the image a corresponding saliency value falling in the range [0,1]. A saliency value of 1 indicates that the corresponding pixel is very conspicuous while a value of 0 indicates inconspicuous pixel. The corresponding keypoint pruning uses the saliency value of the pixel of the input keypoints to decide whether it should be used for the matching process. The saliency value (threshold) at which a keypoint is considered significant has a direct effect on the number of pruned keypoints, and so affects the matching performance. We investigated different saliency thresholds in order to identify the significant keypoints and reported the corresponding matching accuracy and the ratio of reduced keypoints.

3) Results: The effectiveness of the proposed approach for keypoint pruning is evaluated according the achieved image matching performance. The results are compared to the performance of 3 other matching approaches. First, a baseline method that randomly reduces the number of the keypoints of each image pair before performing the matching. Second, a matching approach that uses the whole set of generated keypoints (Full matching). Finally, by matching the images on a reduced set of keypoints using saliency maps (SM).

The accuracy of the matching is measured in terms of precision and recall. To evaluate the matching precision, we used a subset of the Object Recognition Dataset [12] and created two non-overlapping groups of images: the *query* and the *document* groups which contain 100 and 200 images respectively. We matched every image from the query group to every image in the document group without pruning the keypoints (Full), by pruning the keypoints using saliency maps

[1] http://www.klab.caltech.edu/~harel/share/gbvs.php

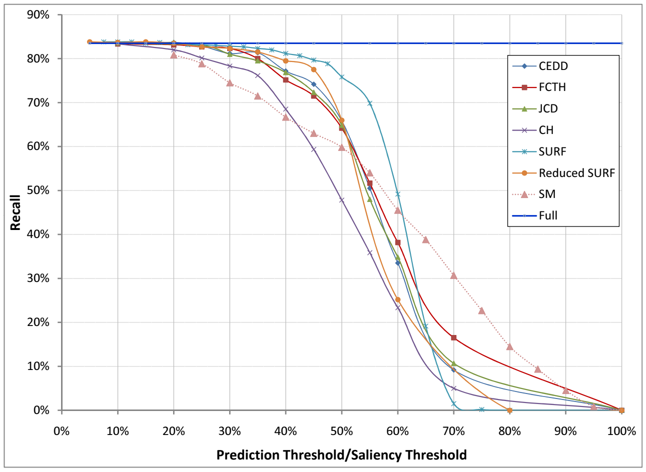

Fig. 2. Recall of different keypoint pruning approaches according to different decision/saliency thresholds. The test dataset includes 200 groups from the Object Recognition dataset

(SM) and by pruning the keypoints using our classification approach. During the matching, two images were considered similar if they shared at least 3 keypoints. We calculated the precision for each query image and took the average. The results showed that a high precision of 99% is achieved by all approaches. This leads us to the conclusion that the precision plays no important role in assessing the effectiveness of image matching that applies keypoint pruning (the precision maintains a constant value under different approaches). In contrast, the matching recall as well as the keypoint reduction ratio (after a pruning method is applied) are crucial factors in judging the quality of the matching. On the one hand, the matching recall indicates the ability of the matching algorithm of retrieving relevant images. While on the other hand, the keypoint reduction ratio gives a clue on the extent to which the matching can be made faster.

The matching recall as well as the ratio of reduced keypoints were evaluated using a test set consisting of 200 groups (different from the training set) of images taken from the Object Recognition dataset. From each group an image was randomly selected and matched to the other three images in the same group (Figure 1). In the case of classification-based keypoint pruning, the matching recall as well as the ratio of reduced keypoint were calculated using different decision thresholds on the significant class. For saliency maps different values for the saliency threshold were investigated.

Figure 2 shows that for the different classification approaches, the matching recall increases to reach that of the full matching with a decreasing prediction thresholds[2]. A lower prediction threshold means that more keypoints will be used by the matching and therefore a higher recall can be achieved. The same applies for the saliency threshold since smaller saliency thresholds means that less keypoints will be pruned, and so a higher recall can be achieved.

The reverse can be noticed regarding the ratio of reduced keypoints. With increased decision/saliency thresholds more keypoints are discarded, resulting in higher keypoint reduction ratios (the corresponding graph were omitted for space consideration).

To get a better insight on the relation between the matching recall and the ratio of reduced keypoints, Figure 3 shows the plot of the the recall, relative to the recall of full keypoint matching, versus the ratio of reduced keypoints (For better visibility only a subset of the investigated keypoint classifiers is shown). First, it can be observed that the presented keypoint pruning methods lead to a much better matching performance than a random keypoint reduction. At a very low keypoint reduction ratio of 5% the random keypoint reduction causes a drop in the matching recall to more than 30%. In contrast, the best performance is achieved by pruning the keypoints using a classifier trained with the Reduced SURF feature closely followed by a classifier trained with the full SURF feature. Both approaches provide a high keypoint reduction ratio of more than 40% while at the same time the corresponding drop in the recall stays below 5%. Regarding the other classification features, keypoint pruning using classifiers trained with FCTH, CEDD and JCD features provide comparable performance. They reduce the amount of detected keypoints to more than 30% with a drop in the recall for less than 10%. A similar results were reported when saliency maps were used to prune the keypoints. Finally, the classifier which is trained with the color histogram feature (CH) provides the lowest *keypoint reduction ratio-recall* performance.

C. Runtime Evaluation

Finding visual similarity between two images using the classification approach consists of the following steps: 1) detecting keypoints in each image, 2) describing the keypoints by features extracted from a patch centered on the keypoints, 3) filtering the keypoint based on their predicted labels and 4) matching a reduced set of the keypoints. Similarly, when saliency maps are used the following steps are followed: 1) generating the saliency map, 2) detecting keypoints in salient

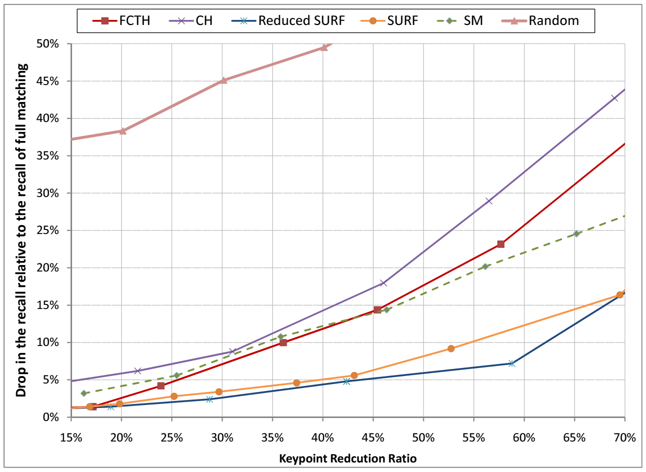

Fig. 3. The drop in the matching recall according to keypoint reduction ratio in proportion to the recall achieved by a full matching. The results are obtained from 200 groups of the Object Recognition dataset

[2] Due to space consideration and since the decision and the saliency threshold values fall in the same range, the two thresholds were shown on the same axis

regions, 3) filtering the keypoint according to their saliency values and 4) matching a reduced set of the keypoints.

For many applications, the first three steps of both approaches can be performed offline. For example with CBIR, these tasks can be executed as background processes before inserting a new image into the database. On the contrary, the matching step must be done online.

The benefit of keypoint pruning on reducing the matching runtime can be estimated analytically as follows. Suppose that we have two images I and J. Let N and M be the numbers of keypoints extracted from I and J respectively.

For a matching algorithm such as the Linear Nearest Neighbor (LNN), finding correspondences between I and J implies calculating the distance between the descriptor vectors of each keypoint pair. Let us denote C the cost of matching measured in the number of descriptor comparisons. With LNN, this cost is given by: $C = N \times M$.

Now, suppose that the **average** keypoint reduction ratio of a keypoint pruning approach is $\bar{k}$. Then the matching cost after applying the pruning C' is calculated as: $C' = (1 - \bar{k})N \times (1 - \bar{k})M$. Consequently, the order in which the matching with keypoint pruning is faster than the full matching is calculated as:

$$\text{Runtime Ratio} = \frac{C}{C'} = \frac{1}{(1 - \bar{k})^2} \tag{1}$$

Experimentally, we evaluated the runtime requirements of each of the presented approaches using a ground truth dataset created from a personal collection of images with image resolution up to 3264×1840 pixels and about 1 MB per image on average. This dataset was selected for two reasons. First, it enables us to investigate runtime requirements with a ”realistic” dataset, which was not specifically designed for a particular retrieval task. Second, using it we can determine how our classification-based approach generalizes to image datasets unrelated to the one that was used in the training. The dataset contains 27 groups and each group have 7 images on average. Images in the same group depict the same scene from different perspectives, at different scales and under different illumination conditions

For each image in the collection we extracted and pruned the keypoints using our approach as well as saliency maps. Table I summarizes the runtime taken by each of the pre-matching processing steps averaged on all images in the ground truth. The test is done using a computer with Intel(R) CORE i5 and 8GB RAM.

Furthermore, we evaluated the matching accuracy as well as the runtime ratio on the new dataset. Similar to the process followed in the last section, we randomly selected an image from each group and matched it to the other images in the same group.

TABLE I
RUNTIME OF THE PRE-MATCHING PHASES

Task	Runtime (sec)
Keypoint Extraction	4.14
SM Generation and Filtering	2.07
SURF Prediction and Filtering	3.95
CEDD Prediction and Filtering	5.89
FCTH Prediction and Filtering	5.75
JCD Prediction and Filtering	6.03

Figure 4 shows the trade-off between the drop in the recall and the percentage of the reduced keypoints achieved by our matching approach as well as saliency maps. We performed the test by using 8 different configurations of our approach (in Figure 4 the numbers on the right of the classification features correspond to the used decision thresholds). Furthermore, we compared our approach to saliency map keypoint pruning with saliency thresholds values of 0.2 and 0.3 (denoted as SM-0.2 and SM-0.3 in Figure 4). First, it can be observed that for the new dataset, the trade-off between the keypoint reduction ratio and the drop in the matching recall is in accordance with the results obtained from using a subset of the Object Recognition dataset (Figure 3). Similarly, at keypoint reduction ratio between 30% to 40% the drop in the recall does not exceed 10% disregarding the applied pruning method. Furthermore, the results show the superiority of the Reduced SURF and the SURF features. For example, Reduced SURF-0.4 (a classifier trained with the Reduced SURF feature and uses a threshold of 0.4 as a decision threshold) could achieve a keypoint reduction ratio of more than 45% with a drop of only 3% in the matching recall. Moreover, the results emphasize again that our approach outperforms the saliency map approach. For instance, the configurations SM-0.3 and SURF-0.5 (shown in italic in Figure 3) both select less than 50% of the keypoints for the matching. However, the saliency map approach (SM-0.3) decreases the matching recall twice as much as our approach (SURF-0.5). As a final remark, based on our evaluations, the best trade-off value between the keypoint reduction ratio and the matching recall can be achieved by using a classifier trained using the SURF descriptor and its reduced variant, namely Reduced SURF.

In the same experiment we also reported the average runtime taken while matching the images in each group. After that, we calculated the ”practical” runtime ratio achieved by performing image matching using our approach as well as saliency maps. The practical runtime ratio was calculated by dividing the time

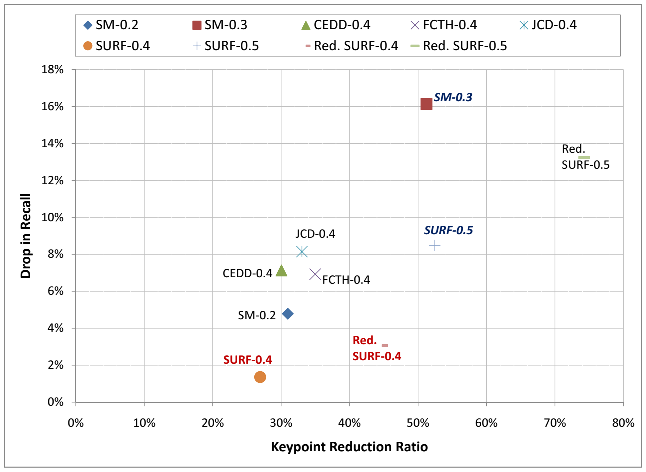

Fig. 4. The drop in the recall according the keypoint reduction ratio achieved by our approach as well as saliency maps.

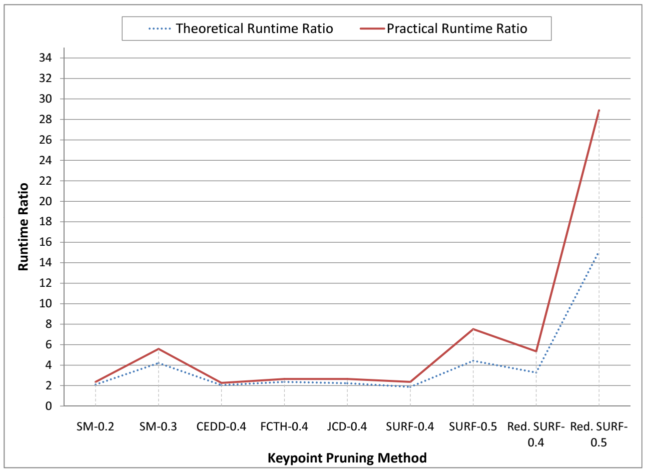

Fig. 5. Runtime ratio achieved by our approach as well as saliency maps. The dotted line corresponds to the theoretical runtime ratio while the solid line represent the practical runtime ratio

taken by full kepyoint matching to that taken by matching with keypoint pruning (the solid line in Figure 5). The graph shows that our approach as well as saliency maps are able to speed up image matching by at least two times. Indeed, the improvement in the runtime is directly proportional to the ratio of reduced keypoints. We also compared the practical runtime ratio to the theoretical runtime ratio as defined in Formula 1. The figure shows that in all cases the practical runtime ratio is higher that the theoretical one, and at low values their graphs almost overlap. With increasing runtime ratio the difference between the two graphs increase to the advantage of the practical runtime ratio. This can be justified, according the Laplacian sign check which is performed while matching SURF descriptors. The SURF algorithm avoids comparing two descriptors when they have different signs, so that faster matching can be achieved. The theoretical analysis provided above does not consider this case and assumes that all descriptors will be compared disregarding their signs, which results in a lower runtime ratio than that of the practical case.

VI. Conclusions

In this paper we presented a machine learning approach for identifying SURF keypoints that are important for image matching. For this purpose, we investigated different image features for characterizing SURF keypoints. Furthermore, we compared the performance of our approach to another approach for keypoint pruning based on saliency maps. The results show that our approach outperforms the adversary and is able to provide high keypoint reduction ratio with a slight drop in the matching recall. Furthermore, the evaluation shows that the execution time of image matching can be efficiently improved using the presented approach. As future work we will consider combining different image features to improve the classification accuracy. Additionally, we will consider investigating techniques for reducing the complexity of the preprocessing phases.

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[2] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[3] J. J. Foo and R. Sinha, "Pruning sift for scalable near-duplicate image matching," in *Proceedings of the eighteenth conference on Australasian database - Volume 63*, 2007, pp. 63–71.

[4] V. Pimenov, "Fast image matching with visual attention and surf descriptors," in *Proceedings of the 19th International Conference on Computer Graphics and Vision*, 2009, pp. 49–56.

[5] Y. Ke, R. Sukthankar, and L. Huston, "An efficient parts-based near-duplicate and sub-image retrieval system," in *Proceedings of the 12th annual ACM international conference on Multimedia*. ACM, 2004.

[6] W. Dong, Z. Wang, M. Charikar, and K. Li, "High-confidence near-duplicate image detection," in *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*. ACM, 2012, pp. 1:1–1:8.

[7] L. Juan and O. Gwun, "A comparison of sift, pca-sift and surf," *International Journal of Image Processing (IJIP)*, vol. 3, no. 4, 2009.

[8] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, Oct. 2001.

[9] F. Lòpez-García, X. R. Fdez-Vidal, X. M. Pardo, and R. Dosil, *Scene Recognition through Visual Attention and Image Features: A Comparison between SIFT and SURF Approaches*. InTech, 2011, pp. 185–198.

[10] S. Chen, C. dong Wu, X. sheng Yu, and D. yue Chen, "Fast scene recognition based on saliency region and surf," in *Intelligent Control and Information Processing (ICICIP), 2011 2nd International Conference on*, vol. 2, july 2011, pp. 863 –866.

[11] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 11, pp. 1254–1259, 1998.

[12] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, 2006.

[13] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[14] P. Turcot and D. Lowe, "Better matching with fewer features: The selection of useful features in large database recognition problems," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2109–2116.

[15] V. Lepetit and P. Fua, "Keypoint recognition using randomized trees," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 9, pp. 1465 –1479, sept. 2006.

[16] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua, "Fast keypoint recognition using random ferns," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 3, pp. 448 –461, march 2010.

[17] M. Calonder, V. Lepetit, and P. Fua, "Keypoint signatures for fast learning and recognition," in *Proceedings of the 10th European Conference on Computer Vision: Part I*, ser. ECCV '08. Springer-Verlag, 2008.

[18] F. Provost, "Machine learning from imbalanced data sets 101," in *Proceedings of the AAAI2000 Workshop on Imbalanced Data Sets*, 2000.

[19] H. Liu and H. Motoda, *Computational Methods of Feature Selection (Chapman & Hall/Crc Data Mining and Knowledge Discovery Series)*. Chapman & Hall/CRC, 2007.

[20] S. A. Chatzichristofis and Y. S. Boutalis, "Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval," in *Proceedings of the 6th international conference on Computer vision systems*, ser. ICVS'08. Springer-Verlag, 2008, pp. 312–322.

[21] S. Chatzichristofis and Y. Boutalis, "Fcth: Fuzzy color and texture histogram - a low level feature for accurate image retrieval," in *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS '08. Ninth International Workshop on*, may 2008, pp. 191 –196.

[22] S. Chatzichristofis, Y. Boutalis, and M. Lux, "Selection of the proper compact composite descriptor for improving content based image retrieval," in *Signal Processing, Pattern Recognition and App.*, 2009.

[23] L. Rokach and O. Maimon, "Top-down induction of decision trees classifiers - a survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 35, no. 4, nov. 2005.

[24] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry." *Hum Neurobiol*, vol. 4, no. 4, 1985.