Real-time Bag of Words, Approximately

J.R.R. Uijlings
ISLA, Informatics Institute
University of Amsterdam
Science Park 107, 1098 XG
Amsterdam, The Netherlands
J.R.R.Uijlings@uva.nl

A.W.M. Smeulders
ISLA, Informatics Institute
University of Amsterdam
Science Park 107, 1098 XG
Amsterdam, The Netherlands
ArnoldSmeulders@uva.nl

R.J.H. Scha
Insitute for Logic, Language
and Computation
University of Amsterdam
Amsterdam, The Netherlands
Scha@uva.nl

ABSTRACT

We start from the state-of-the-art Bag of Words pipeline that in the 2008 benchmarks of TRECvid and PASCAL yielded the best performance scores. We have contributed to that pipeline, which now forms the basis to compare various fast alternatives for all of its components: *(i)* For descriptor extraction we propose a fast algorithm to densely sample SIFT and SURF, and we compare several variants of these descriptors. *(ii)* For descriptor projection we compare a k-means visual vocabulary with a Random Forest. As a preprojection step we experiment with PCA on the descriptors to decrease projection time. *(iii)* For classification we use Support Vector Machines and compare the χ^2 kernel with the RBF kernel. Our results lead to a 10-fold speed increase without any loss of accuracy and to a 30-fold speed increase with 17% loss of accuracy, where the latter system does real-time classification at 26 images per second.

Categories and Subject Descriptors

I.4.7 [**Image Processing and Computer Vision**]: Feature Measurements; I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis

General Terms

Algorithms, Measurement

Keywords

Bag of Words, Computational Efficiency, Image Retrieval, Feature Extraction, Random Forest

1. INTRODUCTION

For the past few years, systems based on a Bag of Words framework produced the best results on several large scale content based image and video retrieval benchmarks, such as the Pascal VOC challenge [2] and the TRECvid Video Retrieval task [18]. Upon examination of these systems it becomes apparent that more is better [10, 19]: Densely sampling many regions outperforms interest point detectors sampling fewer regions. Using more descriptor variants is better. Larger visual vocabularies give higher performance. Larger datasets give better results. As a result standard Bag of Words systems have become computationally expensive.

This paper investigates various faster alternatives to the standard Bag-of-Words pipeline for all of its components: obtaining region descriptions of an image, projecting these descriptions on a visual vocabulary, and classification. Our main contributions are the following: *(a)* We provide a modified way to compute (dense) SIFT for the descriptor step. *(b)* In the same step we turn SURF [1] into a densely computed feature and prove its effectiveness in classification. *(c)* We show that the Random Forests [14] used in combination with Principle Component Analysis are an efficient and effective alternative to k-means for the projection step. We evaluate performance on the Pascal VOC 2007 dataset and achieve real-time classification.

2. RELATED WORK

Jurie and Triggs [7] showed that sampling patches on a regular dense grid outperforms the use of interest points as used for example in the evaluation of Zhang *et al.* [21]. We exploit the regularity of this dense sampling method to reduce computation time for calculating the region descriptors.

Mikolajczyk and Schmid [13] did a comparison of different descriptors and found SIFT [9] or SIFT-like descriptors to be the best for matching under different invariances. Mikolajczyk *et al.* [11] then showed that SIFT also performs best for object recognition.

To speed up the calculation of SIFT, Grabner *et al.* [5] proposed to remove the Gaussian weighting around the origin of the descriptor, which allows the use of integral images which are fast in combination with interest point detectors. We will remove this Gaussian weighting scheme to exploit the spatial nature of SIFT by reusing its components.

Bay *et al.* [1] proposed SURF, a spatial descriptor similar to SIFT based on Haar wavelet responses rather than oriented gradients. Haar wavelets are cheaper to compute than the Gaussian derivatives that are used in SIFT. As with SIFT, we will exploit the spatial nature of SURF in combination with the dense sampling strategy to reuse its components.

Large visual vocabularies created with unsupervised k-means clustering gives good performance (e.g. [21, 10, 20]) and we will use this as our baseline.

CIVR '09, July 8-10, 2009 Santorini, GR
Copyright 2009 ACM 978-1-60558-480-5/09/07 ...$5.00.

Several tree-based algorithms have been proposed to speed up the projection step, [12, 15, 16], allowing for a logarithmic rather than a linear projection time in the number of visual words. The most interesting seems the work on a supervised random forest of Moosmann *et al.* [15]: besides a computational advantage it is the only method also reporting a higher accuracy on their four-class dataset. Intuitively, trees can be less tailored to the specific target classes if the number of these classes increase. This paper will verify whether their results extends to more classes than four.

Support Vector Machines (SVMs) are a very popular classifier due to its robustness against large feature vectors and sparse data. They are successfully used in Bag of Words methods. The choice of SVM-kernel has a large impact on performance. Both Zhang *et al.* [21] and Jiang et al. [6] determined that the χ^2-kernel gives the best accuracy. We will redo part of their experiments but also take computational efficiency into account.

Lazebnik *et al.* [8] proposed the spatial pyramid, introducing a weak form of spatial information by increasingly subdividing the image and obtain a codebook frequency histogram for each region separately. This results in a 5-10% performance increase on the Pascal VOC dataset (data not shown) at a very limited computational projection cost. However, resulting codebook frequency histograms and with that classification time will increase with a factor n, where n is the number of image regions. Because we are focusing on speed in this paper and because the spatial pyramid seems intuitively equally powerful for all descriptors and projection methods we do not have to include this method in our experiments.

3. FAST BAG-OF-WORDS COMPONENTS

3.1 Fast Dense Sampling

This section introduces our novel, fast, and simple way of calculating densely sampled descriptors. Both SIFT and SURF are spatial descriptors: each descriptor is constructed out of four by four subregions whose pixel-wise responses are summed. In the case of SIFT the responses are oriented gradients calculated using image convolutions, for SURF these are Haar wavelet responses calculated using simple summations and subtractions.

First we observe that if the Dense Sampling rate is the same as the size of a subregion we can reuse these subregions for the other descriptors. For the original 4 by 4 SURF and SIFT descriptors this means a factor 16 speed improvement for doing the summations over the pixel responses.

The original SIFT uses a Gaussian weighting over the complete image patch, attributing greater importance to the middle region of the descriptors. This is incompatible with the reuse of subregions. Unlike when using interest points however, the middle region of the descriptor does not seem more important than the rest when using Dense Sampling. We will therefore omit this Gaussian weighting.

We now present an efficient way to sum the responses within each subregion by using two matrix multiplications: One to sum in the row direction and the other to sum over the column direction. For example, consider that we calculated the pixel-wise responses R from an image. Now we want to sum these responses over subregions of 3 by 3 pixels. We can calculate this by the matrix multiplication ARB, where A sums over elements in the row direction and has the form

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 \end{pmatrix}.$$

Matrix B sums over the column direction and looks like the transpose of A but has a different size depending on R.

For robustness against small shifts in position of the descriptor, SIFT uses a linear weighting to divide the responses over neighbouring subregions. We do the linear weighting by slightly modifying A (and B') to

$$\begin{pmatrix} 1 & 1 & ^2/_3 & ^1/_3 & 0 & 0 & 0 & 0 & 0 & \cdots \\ 0 & 0 & ^1/_3 & ^2/_3 & 1 & ^2/_3 & ^1/_3 & 0 & 0 & \cdots \\ 0 & 0 & 0 & 0 & 0 & ^1/_3 & ^2/_3 & 1 & ^2/_3 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix},$$

where the top left number is 1 rather than $^2/_3$ due to the boundary effect. A similar variation enables a Gaussian weighting scheme.

Doing the summation of the regions with these two matrix multiplications is highly efficient: As opposed to a simple for-loop over the regions we obtain a ten-fold speed increase for images of 300 by 500 pixels which are in our dataset.

3.2 Random Forest

We create a Random Forest [14] using the extremely randomized trees algorithm of Geurts *et al.* [3]. Unlike k-means, this algorithm is *supervised*. To learn a tree we use 250,000 labelled descriptors D from our training set, where the labels come from the global image annotations (*i.e.* annotations at image level). Learning is done recursively. At each node n, s random splits are proposed by choosing a random dimension of the descriptors and a random threshold t. This splits the set of descriptors D_n at node n in D_a and D_b. Each split is evaluated using the information gain, defined as [17]

$$\Delta E = -\frac{|D_a|}{|D_n|}E(D_a) - \frac{|D_b|}{|D_n|}E(D_b), \qquad (1)$$

where $E(D_a)$ is the Shannon Entropy of the class labels of D_a. The split with the highest information gain is then adopted. Training then continues with D_a and D_b and stops if a specific depth is reached or if a set is empty.

4. EXPERIMENTAL SETUP

Our paper compares various alternatives to the traditional Bag of Words pipeline to obtain better computational efficiency at a minimal loss of classification accuracy. For convenience we divide our experiments into three categories which represent different parts of the Bag-of-Words pipeline:

Descriptors The features which describe the extracted local image patches.

Projection The projection of these descriptors onto the visual vocabulary, resulting in what we call a codebook frequency histogram.

Classification The classification of these codebook frequency histograms.

All experiments were done on the Pascal VOC 2007 challenge, which is divided into predefined training and set sets of respectively 5011 and 4952 images. There are 20 different object classes and some images contain multiple classes. To measure accuracy we use the Mean Average Precision (MAP) over all classes. The Average Precision is defined as

$$\frac{1}{m}\sum_{i=1}^{n}\frac{f_c(x_i)}{i}, \tag{2}$$

where n is the number of images, m is the number of images of class c, and x_i is the i-th image in the ranked list $X = \{x_1, \cdots, x_n\}$. Finally, f_c is a function which returns the number of images of class c in the first i images if x_i is of class c, and 0 otherwise. This measure gives a number in range $(0, 1]$ where a higher number corresponds to a better performance.

Computational efficiency is measured in milliseconds per image. Reported classification time is over all 20 classes. All our measurements on computational efficiency were done on a single core of a 2.53 Ghz Intel Core Duo E7200 processor.

4.1 Baseline

Our baseline Bag-of-Words system is modelled after the best systems of the Pascal VOC challenge 2007 and 2008 [10, 19].

We use the intensity based SIFT descriptor extracted by our fast Dense Sampling strategy, termed D-SIFT from now on. We sample subregions of 6 by 6 pixels each 6-th pixel and use the original spatial configuration of 4 by 4 subregions.

Our visual vocabulary consists of 4096 words created using k-means clustering. This vocabulary size is kept constant throughout our experiments. New descriptors are projected to the visual vocabulary using nearest neighbour assignment. Classification of the resulting codebook frequency histograms is done using a SVM with a χ^2 kernel.

The resulting Bag-of-Words pipeline for the baseline experiment is presented in figure 1. Note that the pre-projection step is currently empty but will be used by two of our experiments.

Subsequent experiments will always affect a single element of this baseline pipeline.

4.2 Descriptors

In this experiment we compare various fast alternatives to the SIFT descriptor, where we focus on computational efficiency for both the extraction of the descriptors and the projection time because the dimensionality of the descriptors influences projection time.

We compare our implementation of the original 4x4 SIFT descriptor which includes the Gaussian weighting with our D-SIFT version which does not include this weighting. Both have 128 dimensions. We also compare this with our Dense SURF implementation, D-SURF, which has 64 dimensions. Our D-SURF descriptor differs from the normal SURF in that, as in SIFT, we include the linear weighting over the subregions to improve robustness against small changes in location of the descriptor. Preliminary results showed this weighting results in slightly better classification accuracy (data not shown).

We also experiment with the spatiality of the descriptors. The original SIFT and SURF descriptors consist of 4 by 4 subregions. For D-SIFT and D-SURF we also test a 2 by 2 version (*i.e.* 2×2) and a non-spatial (*i.e.* 1×1) version. Speed improvements for these descriptors will mainly occur in the projection phase due to their lower dimensionality. Arguably a more fair comparison of the spatiality would be if the exact pixel regions are used for calculating the descriptors. We included this for the 2 by 2 descriptors which we denote as 2×2*. An overview of the dimensionality and region sizes of the descriptors is given in table 1.

4.3 Projection

The projection time when using the standard nearest neighbour assignment is dependent on three factors: the size of the visual vocabulary, the number of descriptors generated per image, and the dimensionality of the descriptors. In this paper we discuss experiments on the number and dimensionality of the descriptors. We do not experiment with the size of the visual vocabulary: these experiments were already done extensively in other papers (*e.g.* [14, 6]). Furthermore a larger vocabulary also means increased classification time for the Support Vector Machine.

We decrease the number of descriptors per image by a random sub-sampling strategy.

The size of the descriptors is reduced by Principle Component Analysis. We drop the translation component of PCA as it has no influence on the resulting codebook frequency histograms but has a negative influence on projection time.

We also experiment with the Random Forest [14] which is a fast alternative to k-means and nearest neighbour projection. Interestingly enough, each node of a tree acts on a single value of the descriptor so unlike nearest neighbour projection computation time should not be influenced by the dimensionality of the descriptor.

4.4 Classification

We will compare the RBF with the χ^2 kernel for the Support Vector Machine (SVM) that we use as classifier. Earlier, Zhang *et al.* [21] and Jiang *et al.* [6] showed that χ^2 gives the best classification performance. This paper will also take computational efficiency into account.

4.5 Implementation Details

For the SIFT and SURF descriptors we sum responses over subregions of 6 by 6 pixels. The responses for SIFT are calculated using an oriented Gaussian derivative filter with a sigma of 1. For SURF we calculate for each 2nd pixel a Haar Wavelet response of 4 by 4 pixels. In total we generate about 4500 descriptors per image.

One visual vocabulary is created using k-means on 250,000 descriptors. Because both SIFT and SURF are normalized to unit vectors, distances are proportional to the angles between these vectors, which are in turn proportional to the inproduct of the vectors. So we project the descriptors onto the visual vocabulary using the maximum inproduct rather than the minimum distance. Results are exactly the same but we obtain a speed improvement of about a factor 2. Note that the inproduct used in conjunction with PCA will only work if the translation component is ignored.

We learn the Random Forest using 250,000 descriptors. We set the number of proposed random splits s to 15, roughly in accordance with [3]. A Random Forest of four trees gave good results for Moosmann *et al.* [14], so our forests are made of four trees of depth ten resulting in 4096 visual words.

All our implementation is done using highly efficient Mat-

Descriptor	Region size	#dimensions
SIFT 4×4	24 by 24	128
D-SIFT 4×4	24 by 24	128
D-SIFT 2×2*	24 by 24	32
D-SIFT 2×2	12 by 12	32
D-SIFT 1×1	6 by 6	8

Descriptor	Region size	#dimensions
-	-	-
D-SURF 4×4	24 by 24	64
D-SURF 2×2*	24 by 24	16
D-SURF 2×2	12 by 12	16
D-SURF 1×1	6 by 6	4

Table 1: Region size and dimensionality of the various descriptors used.

lab code making heavily use of its fast matrix manipulations. Only for the χ^2 distance and the Random Forest projection we created a C++ implementation (MEX-file), and for calculating the diagonal gradients for SIFT we used the implementation of Geusebroek *et al.* [4].

5. RESULTS

5.1 Baseline

Our baseline pipeline is illustrated in figure 1. Its Mean Averate Precision (MAP) is 0.448. This performance is comparable to [10, 20]. Descriptor extraction takes 138 milliseconds (ms) per image. Projection takes 1028 ms per image. Projection time can be further subdivided: Calculating the inproduct takes 710 ms, taking the maximum of this matrix per column takes 265 ms and finally counting the number of assignments per visual word takes 53 ms. Finally, classification takes 97 ms for all classes. Classification can again be subdivided into calculating the SVM kernel matrix, which takes 89 ms, and the actual classification for all 20 classes, which takes 8 ms per image.

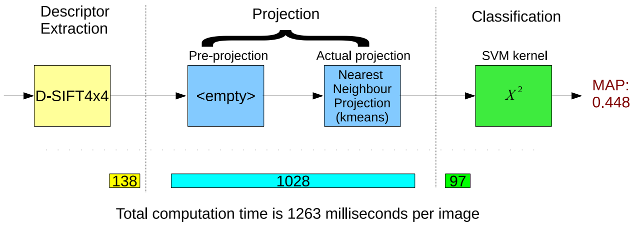


Figure 1: An overview of the Bag-of-Words classification pipeline as used in the baseline experiment.

Note that the projection time provided here is obtained by using the nearest neighbour assignment of section 4.5. Using a fast vectorized squared Euclidean distance function instead of the inproduct takes 1496 ms instead of 710 ms, almost doubling projection time. A simple C++ implementation of the Euclidean distance takes up to 10 seconds(!) per image.

Note that the matrix multiplication order is important. If for the resulting inproduct matrix the maximum should be taken over non-sequential elements in the memory, taking this maximum takes 560 ms rather than 265 ms.

5.2 Descriptors

We first observe that there is no significant difference in classification accuracy between the original SIFT with the Gaussian weighting and our implementation: SIFT 4 by 4 has a Mean Average Precision (MAP) of 0.443 and D-SIFT 4 by 4 a MAP of 0.448. However, the difference in time for extraction is a factor 5. This result is more positive than those of Grabner *et al.* [5], who reported a slight loss of accuracy in removing the Gaussian weighting in the context of descriptor *matching*.

D-SURF4×4 has a MAP of 0.441, almost the same as D-SIFT 4×4. But the time to extract the descriptors is about 6 times as fast for D-SURF. The projection time is 27% faster which can be attributed to its lower dimensionality.

The 2×2 versions of D-SIFT are equally good as the 4×4 version. This also holds for the D-SURF $2 \times 2*$. However, D-SURF 2×2 is slightly worse with a MAP of 0.419. The projection times of the 2×2 versions are better than their 4×4 counterparts: projection time for D-SIFT is reduced from 1082 ms per image to 702 ms, for D-SURF this is reduced from 786 ms to 632 ms.

5.3 Projection

5.3.1 *Random Forest*

We used the Random Forest on all descriptors of our previous experiment. As can be seen in figure 3, Projection time for the Random Forest is 40-60 times faster than nearest neighbour assignment. Interestingly enough, although theoretically the Forest should not be too sensitive to the number of dimensions, projection time still decreased when using fewer dimensions: D-SIFT 4×4 takes 29 ms per image to project and D-SIFT $2 \times 2*$, which has a quarter of the number of dimensions, takes 13 ms per image. While the number of operations are exactly the same (same number of descriptors with the same number of tests), this means that memory access takes quite some time.

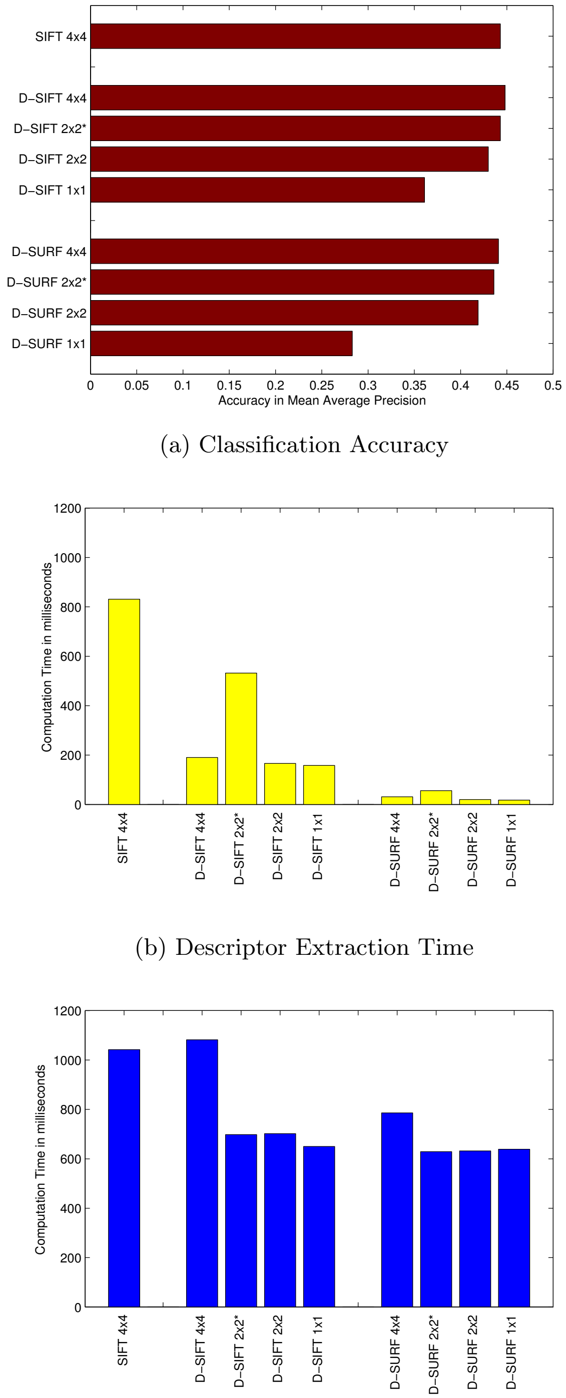
We can see from figure 3(a) that most of the time nearest neighbour projection is slightly better than the Random Forest. However, given the large increase in computational efficiency Random Forests would be a good choice in Bag-of-Words approaches.

Because both Random Forests and SURF are not a defacto standard, we will include the Random Forest and D-SURF 4×4 descriptor in our subsequent experiments.

5.3.2 *Subsampling*

As expected, by using a random sub-sampling strategy classification accuracy as well as the projection speed decreases which can be seen in figure 4. The projection speed decreases linearly. The classification accuracy goes down more or less logarithmically: at 70% of the descriptors there is less than 5% performance loss for all tested pipelines. For k-means the sub-sampling seems a viable strategy to reduce total computation time. But if speed is an issue it is better to resort to a Random Forest, and for the Random Forest the speed increase is only marginal compared to the whole Bag of Words pipeline. This makes sub-sampling a poor strategy for reducing computation time.

5.3.3 *PCA*

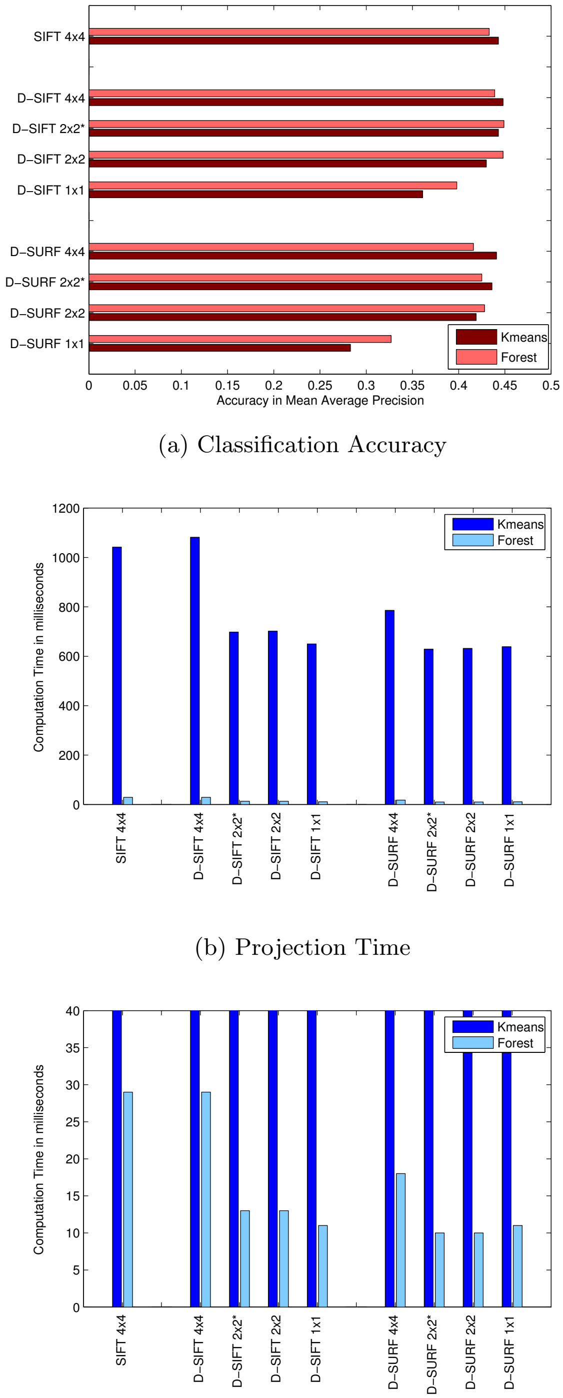


(a) Classification Accuracy

(b) Descriptor Extraction Time

(c) Projection Time

Figure 2: Classification accuracy and computation speeds for various descriptors. Speed is for both descriptor extraction and projection using nearest neighbour with a k-means visual vocabulary.

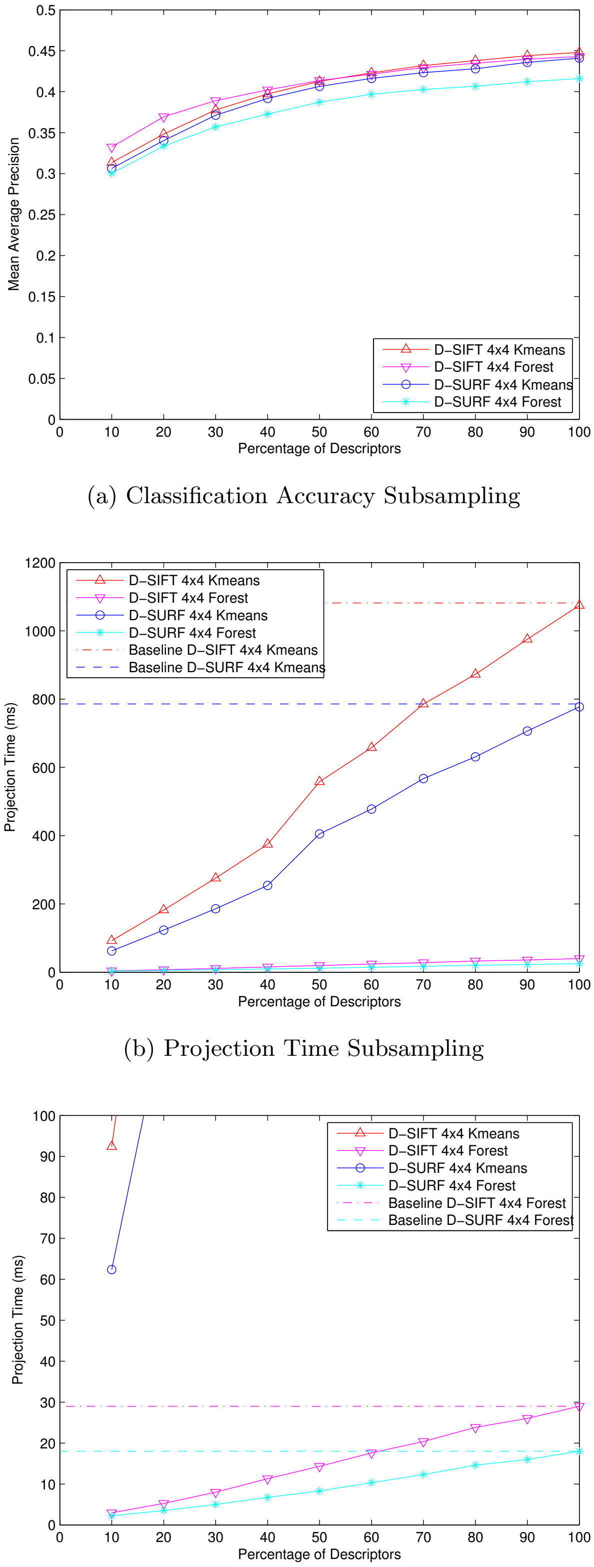


(a) Classification Accuracy

(b) Projection Time

(c) Projection Time (close-up of 3(b))

Figure 3: Random Forests versus k-means nearest neighbour projection: Classification accuracy and projection speeds for various descriptors.

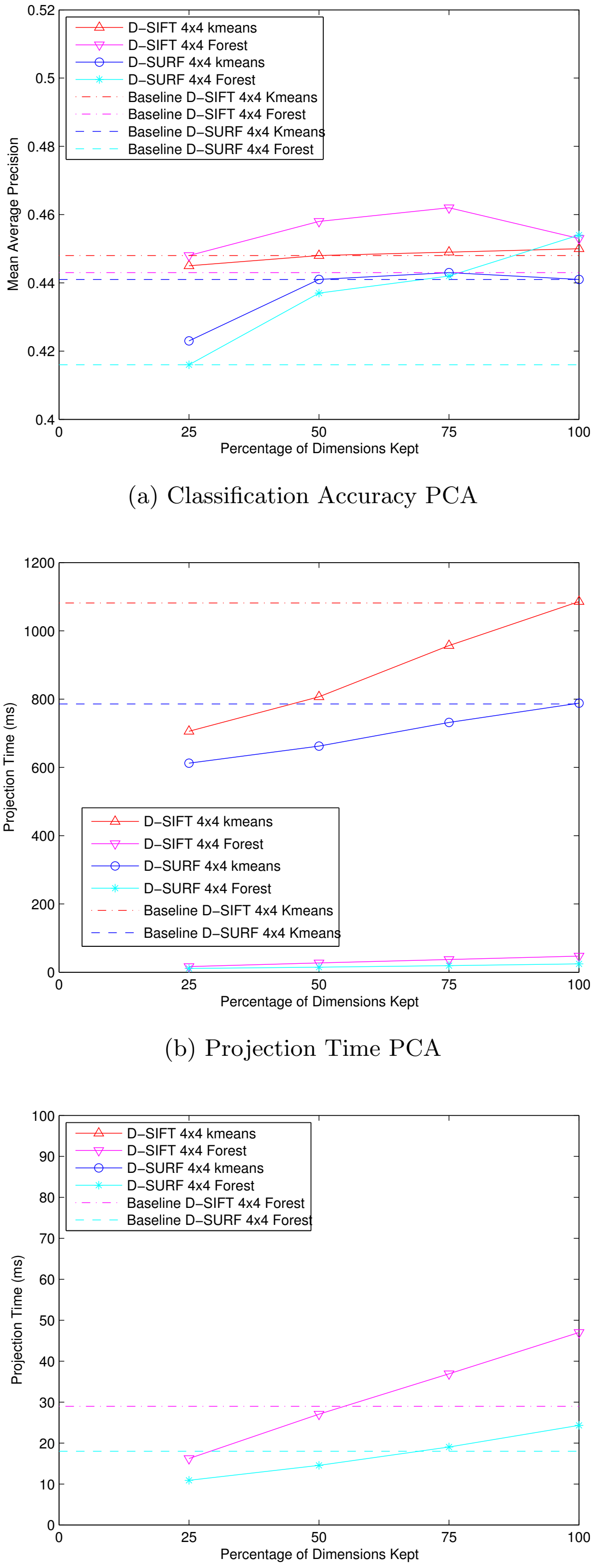


(a) Classification Accuracy Subsampling

(b) Projection Time Subsampling

(c) Projection Time Subsampling (close-up of 4(b))

Figure 4: Classification Accuracy and Projection Speed when using fewer descriptors of the image. Dashed lines denote the baseline scores.

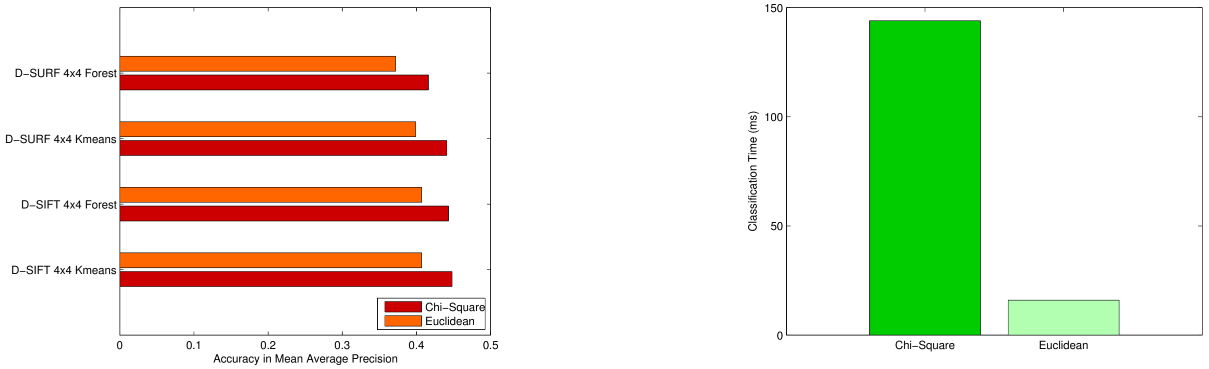


(a) Classification Accuracy PCA

(b) Projection Time PCA

(c) Projection Time PCA (close-up of 5(b))

Figure 5: Classification Accuracy and Projection Speed when using Principle Component Analysis to reduce the number of dimensions. Dashed lines denote the baseline scores.



(a) Classification Accuracy

(b) Classification Time

Figure 6: Comparison of χ^2 and an Euclidean distance matrix for classification Accuracy and classification speed. Classification speed is measured for a single image for all twenty classes. Classification speed is dependent only on the size of the visual vocabulary which is the same for all methods.

Results for the Principle Component Analysis experiment are given in figure 5.

Interestingly enough, results for the Random Forest go up while using PCA. The decorrelation of the dimensions have a positive influence on the decision boundaries of the Random Forest. This is especially noticeable for D-SURF 4×4 whose performance goes up 9% from 0.416 to 0.454.

PCA has a positive influence on projection speed. Kmeans projection time can go down 15-20% without losing any accuracy. Without any dimensionality reduction the projection speeds of Random Forests is slower due to the extra costs of the PCA conversion. But better results are achieved at the same projection speed by dropping some dimensions: drop 50% of the dimensions for D-SIFT4×4 and 25% for D-SURF4×4 . So it is always beneficial to perform PCA.

5.4 Classification

In this experiment we compare the RBF kernel with the χ^2 kernel. Results are presented in figure 6.

Classification time for the χ^2 kernel takes 97 milliseconds per image for all twenty classes. For the RBF kernel it takes 13 milliseconds, 7.5 times as fast. As mentioned earlier, computation time of the classification can be further divided into calculating the kernel matrix and doing the classification itself. For χ^2 the calculation of the total kernel matrix for our test set takes 441 seconds, which amounts to 89 milliseconds per image. The RBF kernel takes 25 seconds to compute, which is 5 milliseconds per image. For both kernel matrices, classification takes 8 milliseconds for all 20 classes, which is less than half a millisecond per class per image.

While the RBF kernel is ten times as fast, it is also 10% less accurate as was reported before [6, 21]. Hence the RBF kernel is only a good choice if speed is essential.

The experiments presented here are done without using the spatial pyramid [8]. Using the spatial pyramid in the same way as the top Pascal VOC performers [10, 19] increases the size of the final codebook frequency histogram and hence the calculation time of the kernel matrix with a factor 8. This makes the calculation of the kernel matrix the current bottleneck of the presented Bag-of-Words pipeline.

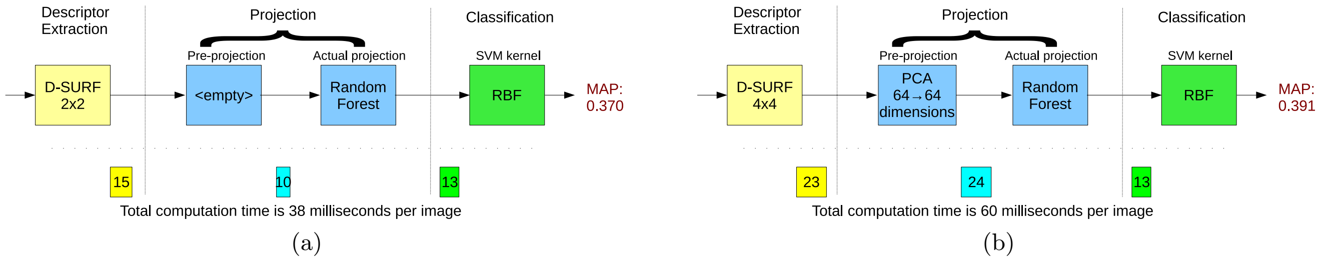
6. CONCLUSIONS

We recommend D-SURF4×4 descriptors and a RBF kernel if speed is essential. Two good choices of a fast Bag-of-Words pipeline are given in figure 7. D-SURF 2×2 descriptors, Random Forest projection and the RBF kernel as presented in figure 7(a) give a Bag of Words pipeline which does classification at a speed of 38 milliseconds per image at a Mean Average Precision (MAP) of 0.370. At 26 images per second this amounts to a real-time classification system. Using D-SURF 4×4, PCA retaining all dimensions, Random Forest projection and a RBF kernel takes 60 milliseconds per image at a MAP of 0.391.

If speed is of lesser importance, the question is what pipeline will result in the maximum accuracy with a minimal computational effort. Two such pipelines are presented in figure 8. Figure 8(a) presents a pipeline with D-SURF 4×4, PCA retaining all dimensions, Random Forest Projection and a χ^2 kernel, resulting in a classification speed of 144 milliseconds per image for a MAP of 0.454. Figure 8(b) presents a pipeline with D-SIFT4×4 , PCA while keeping 75% of the dimensions, Random Forest projection and a χ^2 kernel, resulting in a MAP of 0.462. These pipelines are 5 and 10 times faster than our baseline and have a better accuracy.

In the proposed Bag-of-Word pipeline the primary bottleneck is the calculation of the χ^2 kernel, especially considering that the spatial pyramid [8] was not even included in our experiments. Because the calculation of the χ^2 kernel seems highly suitable for parallelisation, a possible solution would be to implement it on a computer graphics card.

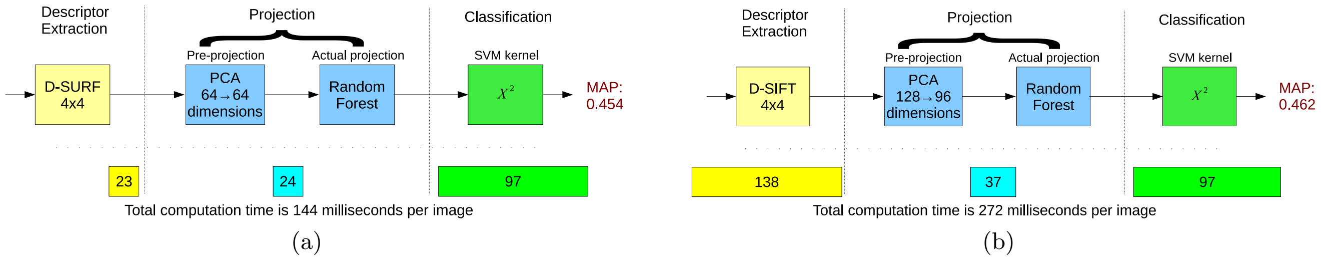
To summarize, our paper presented a fast way to obtain both SIFT and SURF descriptors on a densely sampled grid. Furthermore we showed that Random Forests combined with PCA perform as good as a k-means vocabulary with nearest neighbour projection on this dataset and is 20-30 times faster. The RBF kernel is 7.5 times as fast as the χ^2 kernel but results in about 10% accuracy loss and is therefore only advisable when speed is of primary importance.



(a)

(b)

Figure 7: Two good Bag-of-Words pipelines when focus lies on speed.



(a)

(b)

Figure 8: Two good Bag-of-Words pipelines for obtaining maximum accuracy with minimal computational effort.

Our results led us to propose several fast Bag-of-Words pipelines in figure 7 and 8, one of them which achieves real-time classification.

7. REFERENCES

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110:346–359, 2008.

[2] M. Everingham and J. Winn. The Pascal VOC challenge 2007 development kit. Technical report, University of Leeds, 2007.

[3] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, 2006.

[4] J. M. Geusebroek, A. W. M. Smeulders, and J. van de Weijer. Fast anisotropic gauss filtering. *IEEE Transactions on Image Processing*, 12:938–943, 2003.

[5] M. Grabner, H. Grabner, and H. Bischof. Fast approximated SIFT. In *ACCV*, 2006.

[6] Y.G. Jiang, C.W. Ngo, and J. Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *CIVR*, 2007.

[7] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *ICCV*, 2005.

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

[9] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004.

[10] M. Marszałek, C. Schmid, H. Harzallah, and J. van de Weijer. Learning representations for visual object class recognition. Pascal VOC 2007 challenge workshop. ICCV, 2007.

[11] K. Mikolajczyk, B. Leibe, and B. Schiele. Local features for object class recognition. In *ICCV*, 2005.

[12] K. Mikolajczyk and J. Matas. Improving descriptors for fast tree matching by optimal linear projection. In *ICCV*, 2007.

[13] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27:1615–1630, 2005.

[14] F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *PAMI*, 9:1632–1646, 2008.

[15] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*, pages 985–992, 2006.

[16] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[17] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[18] A.F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and TRECVID. In *MIR*, 2006.

[19] M.A. Tahir, K. van de Sande, J. Uijlings, F. Yan, X. Li, K. Mikolajczyk, J. Kittler, T. Gevers, and A. Smeulders. Uva and surrey @ pascal voc 2008. Pascal VOC 2008 challenge workshop. ECCV, 2008.

[20] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. A comparison of color features for visual concept classification. In *CIVR*, 2008.

[21] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *IJCV*, 73(2):213–238, 2007.