Efficient and Distinct Large Scale Bags of Words

Thomas Pönitz[1,2], Julian Stöttinger[1,2], René Donner[3], Allan Hanbury[4]

[1] Institute of Computer Aided Automation, Vienna University of Technology

[2] CogVis Ltd., Vienna

[3] Computational Image Analysis and Radiology Lab
Department of Radiology, Medical University of Vienna

[4] Information Retrieval Facility, Vienna

Abstract
Due to the increasing flood of digital images and the overall increase of storage capacity, large scale image databases are common these days. Managing such a vast number of digital images is not trivial. This work deals with the problem of finding replicas in image databases containing more than 100 000 images. The bag of visual words method, in analogy to the bag of words method in text search applications, is examined in detail. Every image can be seen as a set of visual words. Computation of these visual words requires clustering of a huge amount of data in a fast yet accurate way. This is the most time consuming part in such an application. A clustering algorithm is developed that has linear runtime and can be carried out in parallel. We observe that with increasing size of the database, the problem of decreasing discrimination between high frequency images arises. Features of images with natural repetitive texture become similar to other images and show up in most of the search results. This problem is addressed by developing an asymmetric Hamming distance measurement for bags of visual words. It allows better discrimination power in large databases, while being robust to image transformations such as rotation, crop, or change of resolution and size.

1. Introduction

This paper deals with the problem of detecting image replicas in large scale image databases. In this work we denote a replica as a copy or reproduction of a work of art, especially one made by the original artist, or a copy or reproduction, especially one on a scale smaller than the original. In terms of computer vision the meaning of image replica is slightly different. Following [10] we refer not only to a bit exact copy of a given original image as replica, but also to modified versions of the image after certain manipulations, as long as these manipulations do not change the *perceptual* meaning of the image content (compare Fig.1). In particular, replicas include all variants of the original. These include image obtained after common image processing manipulations such as compression, filtering, adjustments of contrast, or geometric manipulations.

The paper is organized as follows. The following section gives related work, Section 3. describes our proposed method, Section 4. gives experimental validation of the method while Section 5. concludes.

(a) Query image (b) Rank 1 (c) Rank 2 (d) Rank 3 (e) Rank 4

Figure 1. Results of finding scanned print advertisements with the proposed framework (10 000 images).

2. Related work

For description of visual context, feature extraction is carried out with either global or local features. Global features lack invariance against occlusions and cropping but are a powerful tool in certain applications including image retrieval (e.g. [16, 17]) and provide a fast and efficient way of image representation. They render the task of deciding which parts of the image are used for further processing and which parts are discarded right away unnecessary. Recently, dense sampling of local features achieved good performance especially for the bags of words approach and robust learning systems [11, 19]. Visual descriptors are categorized in three classes: The distribution of certain properties of the image (e.g. SIFT), spatial frequency (e.g. wavelets) or other differentials (e.g. local jets) [12]. Clustering is mainly done for signature generation, feature generalization, vocabulary estimation or assignment of descriptions to a subsets of categories. There are hierarchical and partitional approaches to clustering. Due to the excessive memory and run-time requirements of hierarchical clustering [5], partitional clustering, such as the kMeans, is the method of choice in creating feature signatures.

Classification of images is the phase of finding correspondences and decisions based on the extracted features. Image descriptors are compared with previously learnt and stored models. This is computed by a similarity search or by building a model based on supervised or unsupervised learning techniques. Classification approaches need feature selection to discard irrelevant and redundant information [4, 6, 14]. It is shown that a powerful matching stage can successfully discard irrelevant information and better performance is gained with increased number of features [19]. However, training and clustering are the most time consuming stages of state of the art recognition frameworks. Clustering of a global dictionary takes several days for current benchmark image databases, becoming infeasible for online databases resulting in several billion features [15]. Moreover, there is a upper limit for the size of meaningful descriptions using global dictionaries: Words that appear often in descriptions become meaningless and worsen the performance of the whole systems. Several approaches address this problem (e.g. [2, 18]) examining the occurrences of words on global scope. We extend that idea for the special properties of near duplicate detection. Two algorithms are developed in this paper: The parallel kShifts algorithm for the estimation of the global vocabulary and an asymmetric distance which focuses on the differences between similar images taking the global occurrence into account. We introduce a similiarity measurement, which is robust to highly frequent visual words without the knowledge of the whole dictionary in the following.

3. Method

To avoid the occurrence of ambiguous features in large image data-sets, we choose to improve the approach of bags of visual features simply by the improving the distance measure between image signatures. We develop a more specific decision criteria: Frankly, the problem of near duplicate detection is solved by this approach already. Unfortunately, this does not hold for very large data-sets, where images become more and more similar to each other. Moreover, certain images tend to be similar to all the others as they contain almost every feature. We define this as the *Kirschbaum* problem as images of small, non-repetitive texture (e.g. a close image of a cherry tree in full bloom, but also water, grass or sand, compare Fig. 2) tend to show up in every image query when using bags of words on large data-sets. We solve this problem by developing a dedicated classification technique outperforming standard approaches.

Figure 2. The Kirschbaum problem: High frequency textures are problematic in conjunction with scale invariant local features. They tend to lose their distinction to other images in large databases.

For the acquisition of local visual features SIFT with the best performing parameters and the implementation from [20] is used. One major challenge using bags of visual features is the generation of a global codebook on large data-sets (e.g. [7], [13]). We manage to solve this task by iteratively approximating the desired result in linear time. In the following, the proposed method is described in more detail. The main steps and phases of the approach are explained.

3.1. Clustering

For the commonly used kMeans algorithm, many more efficient solutions exist (e.g. [8]). However, as the most important constraints in this work were considered execution time, parallel execution and linear run-time we propose a new algorithm, the fast clustering algorithm *kShifts*. It is basically a random sampling algorithm that can be used in over- or under-sampling mode (compare Alg. 1). From a set of given data samples $S = (s_1, s_2, \ldots, s_n)$ a multiset P is generated that contains i random permutations of S, where i is also the number of iterations ($P = (p_1, p_2, \ldots, p_m)$ with $m = i * n$). This multiset is then processed in linear order (algorithm 1). The results are vectors in feature space $c \in C$ with a predefined cardinality $|C| = k$. The function $nearestCenter$ assigns a sample to it's nearest cluster center based on a specified distance function (quadratic Euclidean distance in this case). It is carried out in parallel. The $weightSample$ and $weightCenter$ functions are designed so that the weight shifts gradually from the sample to the center. In other words: the influence of the samples decreases over time, thus allowing the centers to *settle*. Figure 3(a) shows this for 2d random

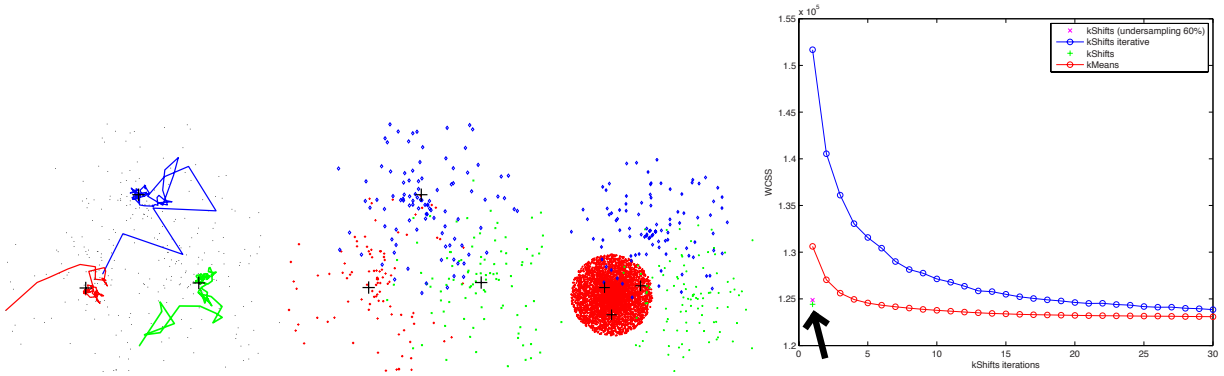
Data: permutation $P = (p_1, p_2, \ldots, p_m)$, number of cluster centers k with $k <= m$
Result: cluster centers $C = (c_1, c_2, \ldots, c_k)$
for $i = 1, 2, \ldots, m$ **do**
    $p = P(i)$;
    $c = nearestCenter(C, p)$;
    $c = weightSample(i, m) * p + weightCenter(i, m) * c$;
end

Algorithm 1: The basic kShifts algorithm.

data and three cluster centers. In this work the weight functions for the i-th of m samples were defined as

$$weightCenter(i, m) = \frac{f(i, m) - 1}{f(i, m)}$$
$$weightSample(i, m) = \frac{1}{f(i, m)}$$
$$f(i, m) = 1 + a \cdot \frac{i}{m}$$

with $a = 42$, as determined by empirical tests with relevant data. One drawback of this algorithm is the same as with other density base clustering algorithms - sparse regions can be under-sampled and thus discriminative data is neglected as seen in figure 3(c). Another claim is that the results depend on the order in which the sample points are processed. In comparison to kMeans the results with normal sampling are slightly worse but can be improved with oversampling. Figure 3(d) shows the within-cluster sum of squares (WCSS). For this figure 100 000 SIFT vectors of 125 random images were clustered. For the undersampling only 60% of the features were used. For oversampling 30 random permutations of the features were concatenated. The runtime of standard kMeans is the same as for kShifts ($\theta(n) = n * k * i$, n number of samples, k number of centers and i number of iterations), however for fewer iterations and most important for one iteration, the results of kShifts are significantly better, although the error measurement used is not directly transferable to the quality of the resulting visual code book.



(a) Traces (b) Clusters (c) Density (d) Sampling

Figure 3. Movement of cluster centers per iteration (a) and their final locations (b). Within-cluster sum of squares (WCSS) on 100 000 SIFT features (d). Note the performance of $m = 30$ (kshift iterative) and $m = 1$ (kshift). (c): Under-sampling of sparse regions.

3.2. Feature quantization

Assignment of the visual features of an image to their corresponding visual codebook element is time consuming and can quickly become a bottleneck of a replica detection system. Given the high dimensionality of the data nearest neighbor searches that were designed for 2d or 3d space are not efficient enough [1]. In this work the codebook is relatively small and not frequently changed. Therefore a search structure is created that cannot be altered, but recreated fast if needed. To speed up the search process a simple tree structure was used were the data is clustered hierarchically into hyper-spheres. Each sphere encloses all data points from it's ancestors. A search is than initiated at a datapoint, which should be assigned to a visual word, with a search sphere of specified size. The size of the sphere is then iteratively increased clipping the other spheres and sub-spheres until the first data point is enclosed by it. The distances to all data points that lie within clipped spheres are calculated and the nearest point is selected. This leads to an exact result while keeping the number of distance calculations low. The complexity of this algorithms depends on the number of spheres clustered for each level. In an ideal case (data points equally distributed between spheres) it would be $\theta(m * log_m(n))$, where n denotes the number of centers and m the number of spheres per level.

3.3. Classification

As the bag of visual words method from [3] is inspired by the bag of words representation for text categorization, we use a text related distance method as well. In text search applications one typically searches with a set of keywords. The frequency of these keywords in a document is then used to generate a ranking. Typically a document that contains fewer other keywords than the query text is ranked higher as it can be considered more specifically relevant for the search.

For image replica detection we altered this approach. An image is represented by a bag (a histogram) of visual words. Given the allowed image transformations so that an image is seen as a replica and under the assumption of a perfect feature detection and extraction, there are only two transformations that can remove visual words from an image description: Cropping and forgery. Additionally only forgery can add new visual words. As the intention of image forgery is only to change the semantic meaning of an image and not the overall appearance it can be assumed that the amount of new visual words in an image replica is not substantial. In contradiction to cropping where the number of lost visual words can be higher than those remaining. To account for this we divided our distance measurement in two parts and treated the query words and the database words differently, thus making it an unsymmetric distance. Given two bags $\mathbf{b_{query}} = (q_1, q_2, \ldots, q_n)$ and $\mathbf{b_{database}} = (d_1, d_2, \ldots, d_n)$ where q and d are normalized frequencies of words, the difference is calculated as

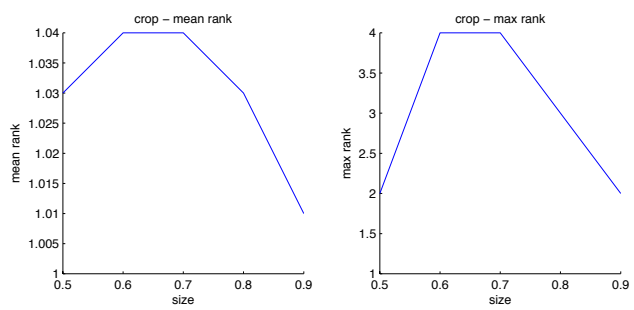
$$d = 1 - \mathbf{b_{query}} \cdot \mathbf{b_{database}} + \sum_{q_i \in b_{query}, q_i = 0} d_i f_{lost} w_i + \sum_{d_i \in b_{database}, d_i = 0} q_i f_{new} w_i$$

with $f_{lost} = 10$ and $f_{new} = f_{lost} \cdot 10$ empirically determined to account for cropping. To increase the influence of less frequent words a weight based on overall database occurrence w_i is added, $\cdot$ denotes the dot product. The first part of the distance becomes fully effective if two images have the same visual words and discriminates them based on the word frequencies. When there are no visual words in common, the last two parts have impact on the distance measure only. This distance helps to overcome the "Kirschbaum Problem" on two levels: *New* visual words (query image only) are punished. Database images with different visual words will not likely show up in the final tracking result. Second, we remain robust to images with many *lost* visual words (visual words that show up in the trained database only), focusing on the common visual words.

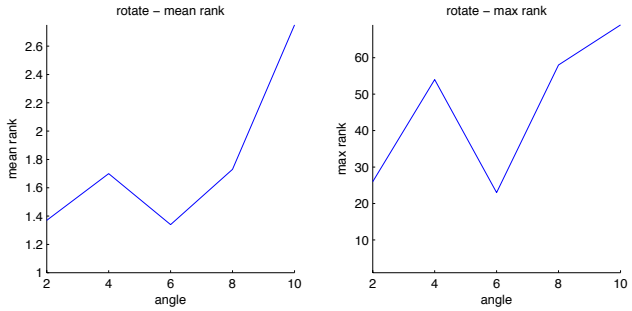
4. Experiments

In the following, the experiments for replica detection are presented. For the dataset, we used 103 452 press images provided by the IT department of the Austrian Press Agency[1]. The pictures are of varying quality and context. Prior to feature acquisition they were resampled to a maximum size of 800×800 pixels. The following image transformations were applied to the query images: blur, noise, rotation and cropping, shrinking and cropping. Due to copyright restrictions no examples can be shown.

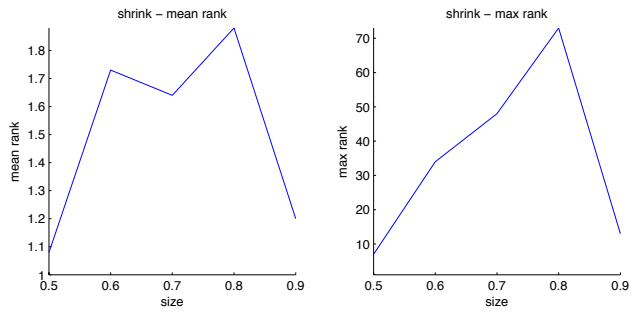
First features were extracted from all images in the database (103 452) using SIFT [9]. Due to time constraints only 60% of all images were randomly selected and their features used for clustering with kShifts, resulting in 48 070 925 features in total. These features were than quantized as described in 3.2. For testing, we chose 100 random pictures as input images. The images were changed with increasing distortion to test the robustness of the application.



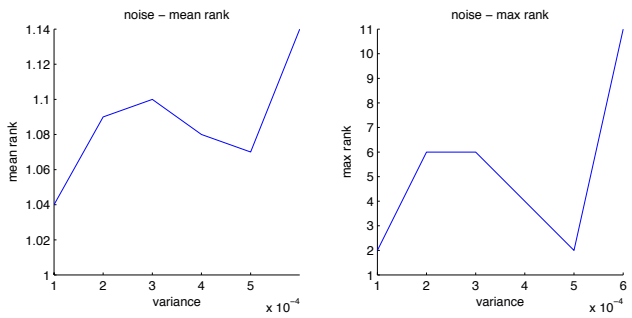
(a) Iteratively cropped query image. Size gives



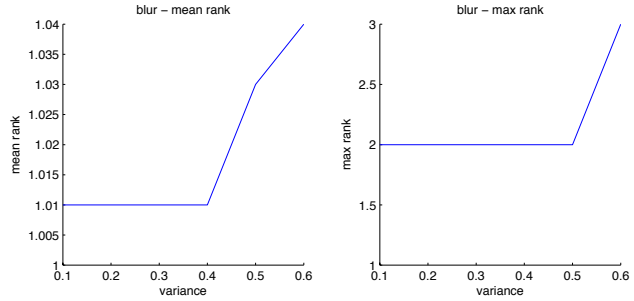
(b) Rotation of query image



(c) Shrink



(d) Noise



(e) Blur

Figure 4. Results of replica detection in 103 452 press images. Rank denotes the position of retrieved images in the final classification, (a) - (e) give results for the applied transformations on the query images.

[1] www.apa-it.at

Features were extracted and processed in the same way as for the database images. We define the rank of an image by the position it shows up in the result page. Rank 1 is the first and most similar image, rank 100 is the 100th similar image in the data-set. An image in the first 100 ranks is categorized as found. The left diagrams of figure 4 show the mean rank over the experiment under increasing distortion of the picture information. The diagrams in on the right give the worst performance of all the 100 images (maximum rank). In diagram 4(e), the 100 input images are consecutively blurred with a Gaussian kernel up to a variance of 0.6. It is shown, that blurring does not change the retrieval performance of the application: Up to a variance of 0.4, a perfect retrieval result is provided, gathering all duplicates on rank 1 in the dataset. Increasing the blur up to a variance of 0.6, we lose 2 pictures up to rank 3. Diagram 4(d) shows the successive adding of Gaussian noise to the image. Wrong image information is introduced to the image. Throughout the whole experiment, all the images stay in the top 10 ranks. Interestingly, after a local maxima of the mean ranks, the performance becomes more stable again and the mean rank decreases. Rotation and cropping leads has more impact on the final retrieval result (see diagram 4(b)) as solely cropping the images (see diagram 4(a)). In diagram 4(c) it is shown that the approach is robust to shrinking of the images. diminish the retrieval result slightly for certain scale factors.

5. Conclusion

The application shows that it is possible to track a single image in large scale data sets. We can distort and transform visual information in the form of cropping, blurring scaling just to mention a few: We still are able to find the right images or very similar images in a reliable way. Runtime of the application meets the requirements for an industrial use as it is possible to track 1000 images per hour. One of the future improvements of the application could be the introduction of color description to the feature space - which would lead to more discrimination power of the actual image description, but will lead to drawbacks when aiming for the tracking of greyscale images.

Acknowledgment

This work was partly supported by the Austrian Research Promotion Agency (FFG), project OMOR 815994, the Austrian National Bank projekt 13497 AORTAMOTION, the Center for Innovation and Technology project 477786 COBAMIR, and the CogVis[2] Ltd. CogVis Ltd. is not liable for any use that may be made of the information contained herein.

References

[1] Stefan Berchtold, Bernhard Ertl, Daniel A. Keim, Hans-Peter Kriegel, and Thomas Seidl. Fast nearest neighbor search in high-dimensional space. In *In Proceedings of the 14th International Conference on Data Engineering*, pages 209–218, 1998.

[2] O. Chum, J. Philbin, and A. Zisserman. Near duplicate image detection: min-hash and TF-IDF weighting. In *BMVC*, 2008.

[3] Chris Dance, Jutta Willamowski, Lixin Fan, Cedric Bray, and Gabriela Csurka. Visual categorization with bags of keypoints. In *ECCVW on Statistical Learning in CV*, 2004.

[2] `http://www.cogvis.at/`

[4] Gy. Dorko and C. Schmid. Selection of scale-invariant parts for object class recognition. In *ICCV*, pages 634–641, 2003.

[5] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Comput. Surv.*, 31(3):264–323, 1999.

[6] Frederic Jurie and Bill Triggs. Creating efficient codebooks for visual recognition. In *ICCV*, pages 604–610, 2005.

[7] Frederic Jurie and Bill Triggs. Creating efficient codebooks for visual recognition. In *ICCV*, pages 604–610, Washington, DC, USA, 2005. IEEE Computer Society.

[8] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *PAMI*, 24(7):881–892, 2002.

[9] D. Lowe. Distinctive image features from scale-invariant keypoints. In *IJCV*, pages 91–110, 2003.

[10] Yannick Maret, Spiros Nikolopoulos, Frédéric Dufaux, Touradj Ebrahimi, and Nikolaos Nikolaidis. A novel replica detection system using binary classifiers, r-trees, and pca. In *ICIP*, pages 925–928. IEEE, 2006.

[11] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *CVPR*, pages 26–36, 2006.

[12] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, 2005.

[13] Frank Moosmann, Bill Triggs, and Frédéric Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*, pages 985–992, 2006.

[14] Ryuzo Okada and Stefano Soatto. Relevant feature selection for human pose estimation and localization in cluttered images. In *ECCV*, pages 434–445, 2008.

[15] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *CVPR*, pages 1–7, 2007.

[16] S H. Srinivasan and Neela Sawant. Finding near-duplicate images on the web using fingerprints. In *ACM MM*, pages 881–884, 2008.

[17] Antonio Torralba, Rob Fergus, and Yair Weiss. Small codes and large image databases for recognition. In *CVPR*, pages 1–8, 2008.

[18] Panu Turcot and David G. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problems. In *ICCVW*, 2009.

[19] Tinne Tuytelaars and Cordelia Schmid. Vector quantizing feature space with a regular lattice. In *ICCV*, pages 1–8, 2007.

[20] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *PAMI*, (in press), 2010.