Histograms for Texture based Image Retrieval[1)]

Christian Wolf [1], Jean-Michel Jolion [2] and Horst Bischof [1]

[1] Vienna Univ. of Technology, Pattern Recognition and Image Proc. Group Favoritenstr.9/1832, 1040 Wien, Austria. [2] INSA de Lyon, Laboratoire

Reconnaissance de Formes et Vision
20, Avenue Albert Einstein, 69621 Villeurbanne cedex, France.

Abstract:
Content based image retrieval is the task of searching images from a database, which are visually similar to a given example image. Since there is no general definition for visual similarity, there are different possible ways to query for visual content. In this work we present methods for content based image retrieval based on texture similarity using interest points and Gabor features. Interest point detectors are used in computer vision to detect image points with special properties, which can be geometric (corners) or non-geometric (contrast etc.). Gabor functions and Gabor filters are regarded as excellent tools for texture feature extraction and texture segmentation. We present methods how to combine these methods for content based image retrieval and to generate a histogram based texture description of images. Experimental results of the query system on test image databases are given.

1 Introduction

Content based image retrieval systems use the contents of a query image provided by the user to search for similar images in a possibly large database. All of the known methods for pre-attentive search emphasize the need for descriptions of images and powerful metrics to compare these descriptions. Most common approaches are based on colour [10], structure [4], textures [7] or combinations [9].

Histograms have already been used extensively for image indexation, especially colour histograms [10] and structure based methods [4]. They provide an effective and efficient means to compare image contents. However, they rely on raw image data like colour or gray values, which are not available for textures without pre-processing. In this paper we describe a method based on histograms filled with local texture features extracted on interest points using a Gabor filter bank.

[1)]This work was supported in part by the Austrian Science Foundation (FWF) under grant S-7002-MAT

2 Local Texture Features

Unlike colour, texture is not a property which can be extracted from a single pixel. A single pixel does not have a texture. A common definition is the repetition of basic texture elements. Therefore, widely used properties are frequency, direction, phase etc. However, these properties are dependent on the scale the image is analysed. A nearly constant area could be textured with a big constant texture or a very fine (infinite) texture. A pullover could show textures on two scales: The printed pattern of squares at a higher scale and the structure of the wool at a lower scale.

For this reason a Gabor filter bank is a suitable tool to extract texture features. The filters of a Gabor filter bank are designed to extract different orientations and frequencies. They have already been widely used for texture analysis and retrieval [5][7]. A Gabor filter bank consists of several filters based on the family of Gabor functions: Each filter responds to a specific orientation and a specific frequency interval.

Extracting local textures features on all points of the images is computational too expensive and not necessary to properly describe the image, especially when most of the image contains non relevant background. Instead we apply an interest point detector in a first step to detect key locations. Texture features are then extracted in the neighbourhood of these locations.

Interest point detectors should deliver pre-attentively "interesting" points in an image. For man-made objects, corners provide valuable information about the scene, therefore the first developed interest point detectors were mainly corner detectors [2]. However, recently other detectors gathering points more suitable to indexation purposes have been developed [1] [6]. The methods described in this paper do not rely on a specific detector. Similar results are obtained by the detectors of Jolion [1], Loupias [6] and Harris and Stephens [2].

The basic idea of our method is to extract a fixed number (e.g. $N = 200$) of regions centered around a detected interest point. Each interest region is input to a Gabor filter bank of $S \times K$ filters, K being the number of orientations (e.g. $K = 8$) and S the number of scales (e.g. $S = 3$). This will give us $N \cdot K \cdot S$ filter responses to process, where a response is a 2D function. For efficient image indexation we have to decrease this amount of data to process.

3 Image representation by Histograms

The result of the feature extraction step outlined above is a set of filter responses. Each response corresponds to one of N interest points, to one of K directions and to one of S scales. The most important description of the filter response is the maximum amplitude, from now on referred to as *amplitude* only. It tells how strong this interest region responds to the filter applied to it. Literally spoken we could say that it specifies how much structure of the given orientation in the given scale can be found in that region. Depending on the ordering of the filter responses we can build different image descriptors. In this paper we will introduce two different types:

- An ordered set of histograms storing amplitude distributions
- An ordered set of histograms storing differences of amplitudes.

3.1 Amplitude distribution

Considering the output of our Gabor filter bank the main information we gathered is the amplitude information for different orientations and scales of the Gabor filters per interest region. If we re-order the data we get a distribution of responses (i.e. amplitudes) per scale and orientation. The filter responses for one filter can be filled into a histogram, which therefore models the response of the image to this filter with the given orientation and the given scale. Thus, the feature data of the whole image, i.e. of all filters of the filter bank, can be stored in an ordered set of $K \times S$ histograms, where S is the number of scales and K is the number of orientations.

Filling the histograms this way we ignore the spatial coherence of the different interest regions, i.e. the locations of the regions are not represented in the 1D-histogram. In order to add the spatial coherence to our representation we use two-dimensional histograms instead of one-dimensional ones. We need couples of data instead of single values to increase the histogram bins. These couples are created by performing a n-nearest neighbour search for every interest point. The entries, which increase the histograms, are pairs of interest points. All interest points of the image are traversed and taken as first points of these pairs. The second point is the neighbour found by a n-nearest neighbour search. Hence, for each interest point we create n pairs of points. The amplitudes of the couples are entered into the histogram: The amplitude of the first point is used to calculate the x-index of the bin, the amplitude of the second point (the neighbour) to calculate the y-index.

Figure 1 shows an example image and two 2D histograms out of its $K \times S$ histograms. The image contains mainly horizontal structures (orientation 0). Therefore, the histogram for orientation 0 shows strong responses, *i.e.* high bins from indices 4 to 6. The histogram for orientation index $\frac{K}{4}$, which corresponds to structures in orientations around 45 degrees, shows only one high bin at index $(0, 0)$, *i.e.* almost no response.

The comparison of two images is based on already known distance measurements of histograms and the means of these distances. For the distances between two single histograms we used the Battacharrya distance, which performed slightly better than the L_2 distance, thus confirmed

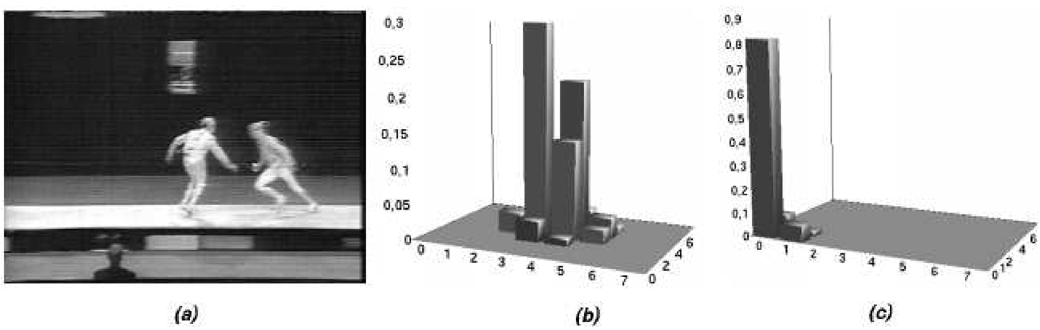

Figure 1: Image (a) and the histograms (amplitude $\times$ amplitude) for 0^o (b) and 45^o (c)

results of Huet and Hancock [4]. The distance formula is given by

$$D_{Batt}(H_A, H_B) = -ln \sum_i \sum_j \sqrt{H_A(i,j) \cdot H_B(i,j)}$$

We included some compensation for rotation by comparing each histogram not only with its corresponding histogram but as well with the immediate neighbours of the same scale. We can represent our ordered set of 3×8 histograms as a set of 3 vectors, $\mu(i)\,, i = 1..3$, each containing 8 histograms. By cyclic permutation of each of these vectors and taking the minimum of the comparison and a comparison with one rotated vector, we get the final distance:

$$d(A,B) = \frac{1}{S} \sum_{i=1...S} d(\mu(i), \nu(i)) \qquad\qquad d(\mu,\nu) = \min \left\{ \begin{array}{c} d(\mu,\nu) \\ d(perm(\mu),\nu) \\ d(\mu, perm(\nu)) \end{array} \right\}$$

where $\mu(i)$ is the histogram vector of set A for scale i, and $\nu(i)$ is the histogram vector of set B for scale i. The function $perm(\mu)$ permutates the vector cyclically.

3.2 Differences of amplitudes

The histograms described in the last section were built by raw image data, like amplitudes. The data was fed pairwise, but still there was no relation between the points. Huet and Hancock used histograms filled with geometric attributes of line segment pairs to index large databases of line patterns [4]. Popescu adopted this idea and derived geometric attributes from the locations of the interest points [8]. Instead of pairs, triples of interest points are taken, and the angle of the points is used to create the histograms.

The main idea of these algorithms is the storage of relative feature data. We use a similar approach based on the output of the Gabor filter bank. Like in the last section an image is characterised by an ordered set of $S \times K$ histograms. For each filter response we traverse all points and search the n nearest neighbours. To calculate the bin index of the first dimension of the two dimensional histogram, we use the difference of amplitudes for each pair interest point and its neighbour. The bin index of the second dimension is defined by the ranking of the neighbourhood. Comparison is done by the same method as discussed in the last section.

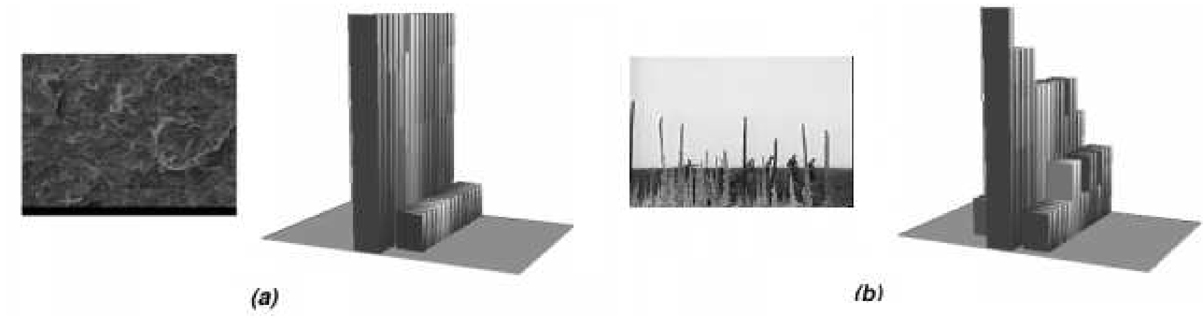

Figure 2: Two images and the histogram (differences of amplitude $\times$ ranking) for 0^o

Each histogram holds the information how the amplitudes for this filter change between the interest points to their neighbours. The y dimension of the histogram stores the neighbourhood ranking of the n-nearest neighbourhood search, thus the bins with higher y indices store the differences of amplitudes between interest points which are further away than couples of interest points stored in the bins with lower y indices. Figure 2 shows two example images of one of our test databases and their histograms for scale index 0 and orientation index 0, which corresponds to an orientation of 0^o. The histogram of the texture example (Figure 2.a) shows an almost uniform distribution along the y coordinate of the histogram. This can be explained by the periodic nature of the texture example, where the differences in the amplitudes are almost constant across the different distances between two interest points. Considering the histogram of the natural image on the other hand (2.b), we remark the difference in the bin sizes along the y coordinate of the histogram. The reason is the different type of the image, which contains sharp changes in structure. In this image the distance between two interest points determines the difference in their responses to a specific filter, hence the difference in amplitudes.

4 Experimental Results

Our test database contains 609 images grabbed from a French television channel. The images are all of the same format (384x288 pixels) and coded in JPEG with 75% quality. The contents differs from outdoor activities (reports of sports) to talk shows, full scope shots of people, weather forecasts, logos and advertisements. To be able to measure a query performance a manual clustering of the image set was necessary. The clusters contain images of successive sequences. In fact, the pictures of one cluster mostly are taken from the same program and sometimes even from the same scene. We used the following parameters in our experiments:

We extract $N = 200$ interest regions of size 32×32 pixels. The Gabor filter bank contains $K \times S = 8 \times 3$ filters.

Since there is no general definition for visual similarity between images, measuring retrieval performance is a difficult task and depends on the purpose. A single query uses one image out of a cluster C, which contains d images. The system answers with c images of which r are from the original cluster C. We use a measure which is widely used for indexation systems: precision.

$$P = \frac{r}{c}$$

As the name suggests, the precision of the result of a single query denotes how precise the result set responds to the desires of the user. By changing the size of the result set we get a performance curve for this query. We calculate the final curve for a retrieval method by averaging the curves for all single queries using different query images.

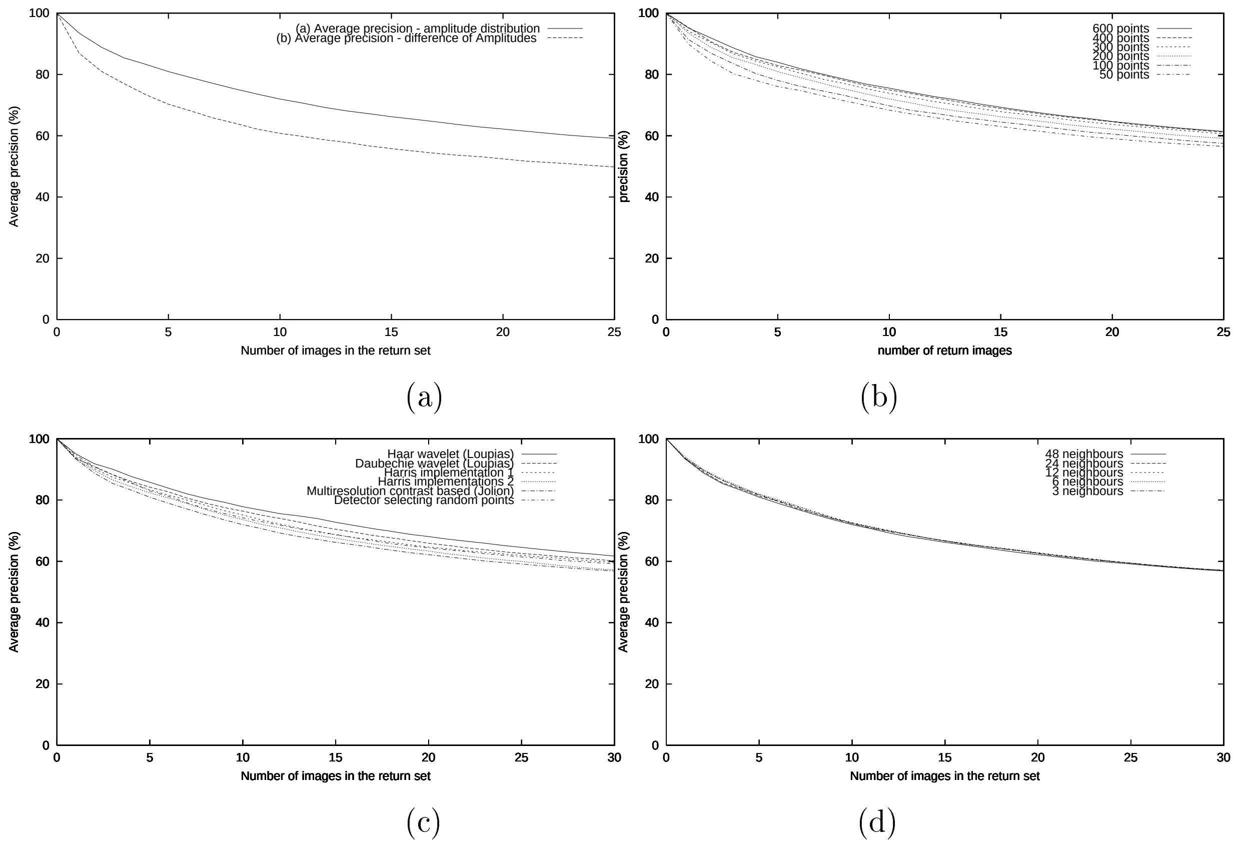

Figure 3: The performance curves for both histogram types (a) for different counts of interest points (b) for different interest point detectors (c) and for different counts of neighbours (d)

Figure 3.a shows two curves measured for the two histogram types. For a typical result set of 10 pictures 74 % of the images are relevant, for 20 pictures 62 %. The histogram set storing the amplitude distribution performs slightly better than the version storing amplitude differences and neighbourhood ranking. We conclude, that the absolute amplitude information is more descriptive than the relative information.

Figure 3.b shows performance curves of the amplitude distribution method with different counts of interest points collected. As we can see the performance of the algorithm is weakly dependent on the number of interest points. If enough area of the image is covered by interest regions, and if the histograms are filled enough — i.e. the number of non-zero bins of the normalised histogram is sufficiently high — then increasing the number of interest points does not increase query performance.

For our experiments we used different interest point detectors: Two different implementations of the Harris corner detector[2], two different versions of the Loupias wavelet based interest point detector[6] based on the Haar and the Daubechie wavelet respectively, and the Jolion multiresolution contrast based interest point detector[1]. To evaluate the dependency of our algorithms on the choice of the detector, we additionally implemented an "interest point detector" selecting a fixed number of random points in an image.

Figure 3.c shows performance curves of the amplitude based algorithm and different detectors. We remark, that the differences in performance between the various detectors are not significant. However, surprising is the good performance of the algorithms in experiments where the random interest point detector is used. The performance of the random points detector is equal to the performance of the other detectors. This can be explained with the richness of our image features. The feature data collected on the interest points has enough descriptive power, which is not improved by a stable interest point detector choosing points "appropriate" for our type of features.

The number of neighbours n for the n-nearest neighbour search determines how much influence of the spatial coherence of the feature data is represented in the histogram. However, the number has to be adjusted to the data. If the algorithm does not search enough neighbours, then the spatial coherence is under represented in the histogram. If the algorithm searches too many neighbours on the other hand, then the information about the spatial coherence is lost, since the found neighbours tend to be the same for points near to each other. We conducted experiments using 3, 6, 12, 24 and 48 neighbours. Surprisingly there is almost no difference in the query performance (Figure 3.d), although counts of $n = 12$ or even $n = 48$ are far too high to store any spatial information in the histograms, considering that $N = 200$ interest points are collected on each image. We conclude, that the additional information about the spatial coherence does not change the descriptiveness of the histograms.

5 Conclusion and Outlook

In this paper we presented histogram based methods to use interest point detectors and a Gabor filter bank to create texture descriptions for image indexing. The image representation introduced holds a rough texture description of images. The similarity measure is able to distinguish groups of images of the same type, i.e. images having similar content without considering many details. Typical applications could be e.g. databases of television broadcast stations, which need to find screenshots of similar scenes or shots of the same telecast in a large set of television screenshots. Experiments with our test image databases prove the good performance for this task. Further details on the algorithms and comparisons to other representations based on interest points and Gabor filters are given in [11]. Further work currently pursued is to join this texture based approach with methods based on colour, structure and shape into one weighted indexation system, which uses feedback of the user to recalculate the weights of the system [3].

References

[1] S. Bres and J.M. Jolion. Detection of interest points for image indexing. In *3rd Int. Conf. on Visual Inf. Systems, Visual 99*, pages 427–434. Springer, Lecture Notes in Computer Science, 1614, June 1999.

[2] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings 4th Alvey Visual Conference*. Plessey Research Roke Manor, UK, 1988.

[3] A. Heinrichs, D. Koubaroulis, B. Levienaise-Obadia, P. Rovida, and J.M. Jolion. Image indexing and content-based search using pre-attentive similarities. In *6th RIAO Conference, PARIS*, pages 1616–1631, April 12-14 2000.

[4] B. Huet and E.R. Hancock. Cartographic indexing into a database of remotely sensed images. In *Third IEEE Workshop on Applications of Computer Vision (WACV96)*, pages 8–14, Sarasota, Dec 1996.

[5] A.K. Jain and F. Farrokhina. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[6] E. Loupias, N. Sebe S. Bres, and J.M. Jolion. Wavelet-based salient points for image retrieval. In *International Conference on Image Processing, Vancouver, Canada*, 2000.

[7] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8), August 1996.

[8] O. Popescu. Utilisation des points d'intérêt pour l'indexation d'images. Technical Report RR 99.07, Laboratoire Reconnaissance de Formes et Vision., 1999.

[9] C. Schmidt and R. Mohr. Local gray value invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5), May 1997.

[10] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[11] Ch. Wolf. Content based image retrieval using interest points and texture features. Technical Report PRIP-TR-061, Pattern Recognition and Image Processing Group, Institute of Computer Aided Automation, April 2000.