On Using Histograms of Local Invariant Features for Image Retrieval

Alaa Halawani and Hans Burkhardt
Albert-Ludwigs-University of Freiburg
Chair of Pattern Recognition and Image Processing
79110 Freiburg, Germany
{halawani|burkhardt}@informatik.uni-freiburg.de

Abstract

In this paper we employ different methods for constructing histograms from invariant features that are computed locally around a set of salient points. These points represent, together with their neighborhood, the most important visual information in an image. The features used for constructing the histograms are evaluations of Haar integrals with nonlinear kernel functions. The resulting histograms are able to preserve the local structure of the image in addition to the fact that they are invariant to Euclidean motion. We study and compare the performance of the different histogram construction methods for a database that consists of 15000 images.

1 Introduction

Invariant image features based on Haar integral were introduced by Schulz-Mirbach in [6]. These features have been used successfully in texture-classification [5], pollen-recognition [4], and image retrieval. [10, 1]. For image retrieval, Siggelkow et al. [10] have used color and texture histograms that are based on Euclidean-invariant integral features. Unlike the ordinary histogram, the invariant integral features have the advantage of capturing the local structure held in the image. Experimental results have shown that these features demonstrate a very good capability in retrieving images. However, the main disadvantage is that the computation of the invariant features over the whole image is time consuming. In order to reduce the computation complexity, Siggelkow and Schael [9] have estimated the invariant features using the Monte-Carlo method. They have computed the features for a set of randomly generated points and directions.

Recently we have compared in [1] the work based on the Monte-Carlo approximation of the invariant features with the extraction of these features from areas of high relevance in the image under consideration. These areas are image patches that are centered around the so-called *salient points*. We have used the salient point extraction algorithm introduced in [2] in order to determine these patches. It was found that the computation of invariant features around the salient points enhances the performance of the retrieval system and makes it more robust for cases like object scaling and viewpoint changes. This previous work has concentrated on using a single kernel function to extract features and build the histogram. However, there is a possibility to use more than one kernel function to build up a feature space and use it for image retrieval.

In this paper, we study both cases (using a single or a set of kernel functions) for the construction of the invariant-feature histogram for the purpose of content-based image retrieval. We concentrate on extracting the features around the salient points only. We use the HSV color space as it was found that it performs better than the RGB color space [1]. In the case of using a set of kernels to extract feature vectors around the points, we distribute the resulting vectors in several clusters and construct a histogram of the cluster numbers rather than the feature vectors themselves. This is done in order to overcome the high dimensionality of the feature vectors (which makes histogram construction difficult). We consider both one-dimensional and two-dimensional cluster-number histograms. The latter reflects the spatial relationship between the pattern at each point and the patterns of its neighboring points based on cluster numbers assigned to these patterns.

The paper is organized as follows: In section 2 we explain the process of calculating the invariant features. A brief description of the different ways used to construct the histograms is given in section 3. A summary of the experimental results is presented in section 4. Finally, a conclusion is given in section 5.

2 Invariant features based on Integration

Following is a brief description of the calculation of the rotation- and translation-invariant features. Details can be found in [6].

The idea of constructing invariant features is to apply a nonlinear kernel function $f(\mathbf{I})$ to a gray-valued image, $\mathbf{I}$, and to integrate the result over all possible rotations and translations (Haar integral over the Euclidean motion); i.e.,

$$IF(\mathbf{I}) = \frac{1}{2\pi MN} \int_{r=0}^{M} \int_{c=0}^{N} \int_{\theta=0}^{2\pi} f(g(r,c,\theta)\mathbf{I}) d\theta dc dr \quad (1)$$

where $IF(\mathbf{I})$ is the invariant feature of the image, M, N are the dimensions of the image, and g is an element in the transformation group G (which consists here of rotations and translations).

Because of the discrete nature of the image, IF is approximated by choosing r and c to be integers and by varying θ in a discrete manner producing q samples:

$$IF(\mathbf{I}) \approx \frac{1}{qMN} \sum_{r=0}^{M-1} \sum_{c=0}^{N-1} \sum_{j=0}^{q-1} f(g(r,c,\theta = j\frac{2\pi}{q})\mathbf{I}) \quad (2)$$

Bilinear interpolation is applied when the samples do not fall onto the image grid.

The above equation suggests that invariant features are computed by applying a nonlinear function, f, on the neighborhood of each pixel in the image, then summing up all the results to get a single value representing the invariant feature. Using several different functions finally builds up a feature space.

Much of the local information is lost by summing up the local results. This makes the discrimination capability of the features weak. In order to preserve the local information, Siggelkow et al. [10] replaced the summation ($\sum_r \sum_c$) by histogramming:

$$IF(\mathbf{I}) = \text{hist}\left(\left\{\frac{1}{q}\sum_{j=0}^{q-1} f(g(r,c,\theta = j\frac{2\pi}{q})\mathbf{I}) \,\middle|\, \begin{array}{l} r = 0,\cdots,M-1, \\ c = 0,\cdots,N-1, \end{array}\right\}\right) \quad (3)$$

It is also possible to replace all the summations by a histogram operation [8], i.e.,

$$IF(\mathbf{I}) = \text{hist}\left(\left\{f(g(r,c,\theta = j\frac{2\pi}{q})\mathbf{I}) \,\middle|\, \begin{array}{l} r = 0,\cdots,M-1, \\ c = 0,\cdots,N-1, \\ j = 0,\cdots,q-1 \end{array}\right\}\right) \quad (4)$$

Invariant features can be either color or texture features, depending on the chosen kernel function. Invariant color features can be computed by applying the "*monomial kernels*" which have the form:

$$f(\mathbf{I}) = \left(\prod_{p=0}^{P-1} \mathbf{I}(x_p, y_p)\right)^{\frac{1}{P}} \quad (5)$$

In order to construct texture features, a "*relational kernel*" function [5] is to be applied. This kernel has the form:

$$f(\mathbf{I}) = rel(\mathbf{I}(x_1, y_1) - \mathbf{I}(x_2, y_2)) \quad (6)$$

Where

$$rel(\gamma) = \begin{cases} 1 & \text{if } \gamma < -\epsilon \\ \frac{\epsilon-\gamma}{2\epsilon} & \text{if } -\epsilon \leq \gamma \leq \epsilon \\ 0 & \text{if } \epsilon < \gamma \end{cases} \quad (7)$$

This kind of kernels was introduced in [5] and is based on the Local Binary Pattern (LBP) texture features [3], which map the relation between a center pixel and its neighborhood pixels into a binary pattern. Equation 7 extends the LBP operator to give values that fall in $[0, 1]$.

3 Constructing Histograms from Local Invariant features

The salient point extraction algorithm that we use was introduced by Loupias and Sebe [2]. We have chosen to use this detector as it has more information content and better repeatability compared with the well-known Harris detector [7]. We employ the following ways of histogram construction using invariant features extracted around the salient points:

1. **Single kernel function with averaging over rotation:** Applying one kernel function around the salient point and then constructing a histogram using equation 3.

2. **Single kernel function without averaging over rotation:** Applying one kernel function around the salient point and then constructing a histogram using equation 4. Getting rid of the integration over rotation keeps more local information which is expected to lead to better retrieval results.

3. **Set of kernel functions with one-dimensional cluster-number histogram:** A problem of dimensionality arises when using a set of kernel functions to extract feature vectors from the points. Applying n kernel functions around each point for the three channels of the color space and averaging over rotation yields a feature vector of length $3n$. Constructing a $3n$-dimensional histogram, $n > 1$, is difficult and becomes prohibitive for larger n values. In order to overcome this problem, we map the feature vectors into scalars by distributing the feature vectors extracted from all points in all images of the database in set of k clusters and then dealing with the clusters rather than with the feature vectors. The clusters are numbered $1, 2, ...k$. For each image, a k-bin histogram, h, of cluster numbers is constructed instead of a histogram of the feature vectors themselves (see Fig.1). If SP is the set of salient points in an image and $CN(i)$ is the cluster number assigned to the feature vector at point i, then:

$$h(l) = \sum_{i \in SP} \delta\left(CN(i) - l\right), \qquad l = 1, 2 \cdots k \quad (8)$$

4. **Set of kernel functions with two-dimensional cluster-number histogram:** This method is meant to show the local co-occurrence of feature vectors by reflecting the spatial relationship between cluster number assigned to the vector at each point and cluster numbers assigned to the vectors at its p-neighboring points. Essentially, for each point we search for its p-closest points and determine to which clusters their vectors belong. Given that $NH(i)$ is the set of p-points in the neighborhood of point i, the 2-dimensional cluster-number histogram is given by:

$$h(l,m) = \sum_{i \in SP} \sum_{j \in NH(i)} \delta\left(CN(i) - l, CN(j) - m\right), \qquad l, m = 1, 2 \cdots k \quad (9)$$

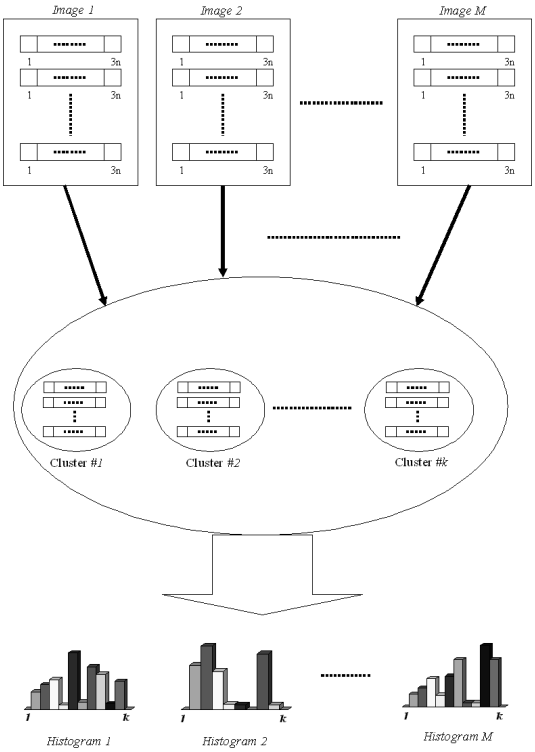


Figure 1: Building histograms from cluster numbers

4 Results

We conducted the tests on a database which consists of 15000 colored images. We use HSV color space rather than the RGB space as it was found that the former gives better results because it is perceptually more uniform [1]. The salient point extraction algorithm is applied to the V Channel of the images to determine the image points to be used. Both color and texture features are evaluated around these points.

The features for all images are extracted offline and saved with pointers to the images in a feature database. We use a query-by-example (QBE) methodology. A query image is submitted, its features are computed online and compared with the features of all other images in the database. To compare the histograms of the query image and the database images, we have used the χ^2 measure, which gives an indication of the difference (d) between two histograms:

$$\chi^2(h_q, h_d) = d = \sum_i \frac{(h_q(i) - h_d(i))^2}{h_q(i) + h_d(i)} \qquad (10)$$

Where h_q and h_d are the histograms of the query image and an image in the database respectively.

To increase the accuracy of the retrieval, one should integrate the results of both comparisons of texture and color features. Let d_c equal the difference between the query image and a database image based on color, and d_t equal to the difference based on texture. The difference based on both color and texture is given by:

$$d_{total} = \alpha d_c + \beta d_t, \qquad \alpha + \beta = 1 \qquad (11)$$

where α and β are weights assigned to the color-based difference and texture-based difference respectively.

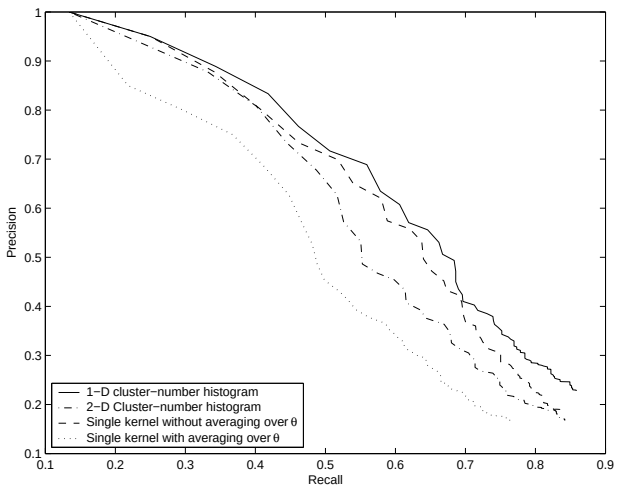


Figure 2: Average precision-recall for the 30 queries

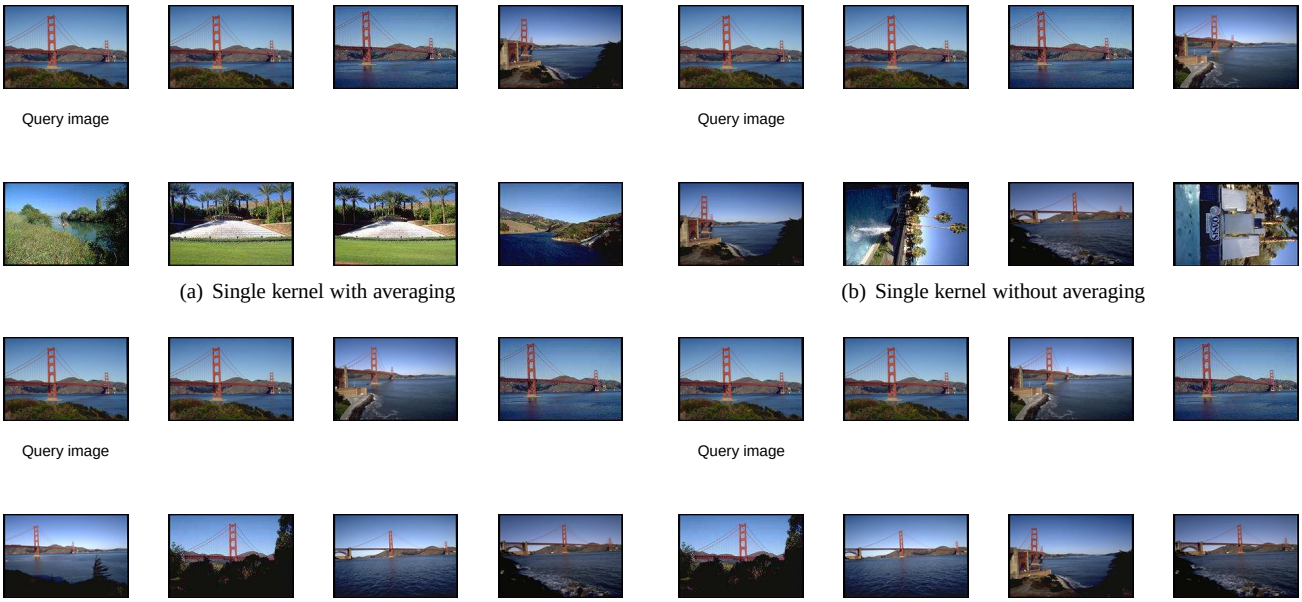
We have chosen a set of 30 representative images from the database to serve as queries for our experiments.

Setting equal weights for color and texture ($\alpha = \beta = 0.5$), we have tested the performance of the different histogram construction methods. For the case of using a single kernel function, the relational kernel, $f(\mathbf{I}) = rel(\mathbf{I}(3,0) - \mathbf{I}(0,6))$, with $\epsilon = 0.098$ in Equation 7 (image pixel values $\in [0,1]$) and the monomial kernel, $f(\mathbf{I}) = (\mathbf{I}(3,0).\mathbf{I}(0,6))^{\frac{1}{2}}$, were chosen to construct the texture and color features respectively. All three channels of the color space were taken into consideration when calculating the features. Two $6 \times 6 \times 6$ Histograms (color Histogram and texture Histogram) were constructed.

Alternatively, two sets of 12 monomial kernels and 12 relational kernels were used to construct two 36-dimensional feature vectors (color and texture respectively) around each point. Based on the resulting feature vectors, and after applying the k-means clustering algorithm, two 256-bin and two 16×16-bin cluster-number histograms were constructed.

Fig. 2 shows the average precision-recall of the 30 queries. It can be observed that the histogram with averaging over rotation gives the worst performance, while the one-dimensional cluster-number histogram gives the best retrieval results. This behavior can be interpreted by the fact that the process of averaging over rotation still causes loss of local details. Applying only one kernel function ends up with relatively little image-structure information retained. Therefore, applying a set of kernel functions gives a better representation of the local structure of the image which leads at the end to better results.

The above interpretation is further supported by the results obtained when the averaging process is eliminated, which is also shown in Fig. 2. Getting rid of the averaging over the angle of rotation leads to keeping more details about the local evaluations which in turn causes the histogram to retain more information about the image structure. It can be seen from the figure that this leads to significantly better results than with averaging although only one kernel function is used. The performance of the his-



(a) Single kernel with averaging

(b) Single kernel without averaging

(c) One-dimensional cluster-number histogram

(d) Two-dimensional cluster-number histogram

Figure 3: Results of a sample query

togram without averaging over rotation comes in the second place after that of the one-dimensional cluster-number histogram.

Surprisingly, the results of the two-dimensional cluster-number histogram are not satisfactory. One possible explanation is that this type of histograms is too restrictive in the case of full-image query. A retrieval example for the different methods is shown in Fig. 3.

5 Conclusion

In this paper we have compared the performance of different methods of histogram construction using invariant integral features computed around salient points. It was found that histograms based on using a a single kernel with averaging over rotation have comparatively the worst performance because the averaging causes loss of important local information. Getting rid of the averaging process or using a set of kernel functions improves the performance significantly. The problem of high dimensionality faced when using a set of kernel functions was solved by mapping the feature vectors into numbers through clustering of these vectors and then considering the cluster numbers in the process of histogram construction rather than the features themselves.

Acknowledgement

Alaa Halawani would like to thank the German Academic Exchange Service (DAAD) for granting him a scholarship for his PhD studies at the University of Freiburg in Germany.

References

[1] A. Halawani and H. Burkhardt. Image Retrieval by Local Evaluation of Nonlinear Kernel Functions around Salient Points. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 2, pages 955–960, Cambridge, United Kingdom, August 2004.

[2] E. Loupias and N. Sebe. Wavelet-based Salient Points for Image Retrieval. Technical Report TR. 99.11, Laboratoire Reconnaissance de Formes et Vision, INSA Lyon, 1999.

[3] T. Ojala, M. Pietikäinen, and T. Mäenpää. Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns. In *Proc. Sixth European Conference on Computer Vision*, pages 404–420, Dublin, Ireland, 2000.

[4] O. Ronneberger, H. Burkhardt, and E. Schultz. General-purpose Object Recognition in 3D Volume Data Sets using Gray-Scale Invariants - Classification of Airborne Pollen-Grains Recorded with a Confocal Laser Scanning Microscope. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 2, Quebec, Canada, 2002.

[5] M. Schael. Invariant Grey Scale Features for Texture Analysis Based on Group Averaging with Relational Kernel Functions. Technical Report 1/01, Albert-Ludwigs-Universität, Freiburg, Institut für Informatik, January 2001.

[6] H. Schulz-Mirbach. Invariant Features for Gray Scale Images. In G. Sagerer, S. Posch, and F. Kummert, editors, *17th DAGM - Symposium "Mustererkennung"*, pages 1–14, Bielefeld, 1995. Springer.

[7] N. Sebe and M. S. Lew. Comparing salient point detectors. *Pattern Recognition Letters*, 24(1-3):89–96, January 2003.

[8] S. Siggelkow and H. Burkhardt. Improvement of Histogram-Based Image Retrieval and Classification. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 3, pages 367–370, Quebec, Canada, September 2002.

[9] S. Siggelkow and M. Schael. Fast Estimation of Invariant Features. In W. Förstner, J. M. Buhmann, A. Faber, and P. Faber, editors, *Mustererkennung, DAGM 1999, Informatik aktuell*, pages 181–188, Bonn, September 1999.

[10] S. Siggelkow, M. Schael, and H. Burkhardt. SIMBA - Search IMages By Appearance. In B. Radig and S. Florczyk, editors, *Proceedings of 23rd DAGM Symposium, number 2191 in LNCS Pattern Recognition*, pages 9–16. Springer, September 2001.