

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/228943913>

# Clustering visually similar images to improve image search engines

Article · January 2003

CITATIONS

20

READS

2,331

3 authors:



**Thomas Deselaers**

Google Inc.

117 PUBLICATIONS 7,667 CITATIONS

[SEE PROFILE](#)



**Daniel Keysers**

Google Inc.

143 PUBLICATIONS 6,064 CITATIONS

[SEE PROFILE](#)



**Hermann Ney**

RWTH Aachen University

1,025 PUBLICATIONS 41,092 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Lecture Translator [View project](#)



EU MATECAT [View project](#)

# Clustering visually similar images to improve image search engines

Thomas Deselaers, Daniel Keysers, and Hermann Ney  
deselaers@i6.informatik.rwth-aachen.de

Lehrstuhl für Informatik VI, Computer Science Department  
RWTH Aachen – University of Technology, D-52056 Aachen, Germany

Supervisor: Prof. Dr.-Ing H. Ney  
Type: Diploma thesis  
GI subjects: image understanding (1.0.4), machine learning (1.1.3)

## Abstract

At the moment Google image search is probably the only widely known way to search the world wide web for images. Google's search engine works based on text retrieval: The images are not indexed by their appearance but by text which can be found in the context of the image. To achieve enhancements for the user we propose to reorder the images using a combination of methods from computer vision and data mining. We use features invariant against translation and rotation to represent the image content and the  $k$ -means and LBG cluster algorithms to present the images in groups in a more convenient way to the user. To test this method we created a new database from Google image search results.

## 1 Introduction

The problem of searching a picture meeting certain requirements is a task which occurs in many applications. The idea that the world wide web with its vast and increasing amount of digitally available images should really help to find a picture meeting the requirements seems to be wrong. Instead most users are flooded with the amount of images available. To make image searching possible Google recently offered a way to search for images at <http://images.google.com>. This search is based upon textual information found in the context of the images on web-sites. This leads to reasonably good performance when searching for images, but there is also a major drawback. Many words are ambiguous and searching for them results in very different types of images. E.g. the search for "cookie" results in more or less three different types of images: images of edible cookies, screen-shots of programs dealing with cookies in the context of the Internet, and images not concerned with cookies at all. And even for words with less ambiguity nearly always two groups of images are returned: One group of images which meet the requirements and one group of images not suitable. Here we present an approach to help the user reaching his search goals faster and more comfortably. This is done using image processing and image retrieval methods.

The remainder of this document is structured as follows: Section 2 introduces invariant features and invariant feature histograms as a way to determine visual similarities between images, section 3 illustrates two well known clustering techniques used to regroup the images, section 4 describes the database we use to test the approach described and gives some first results, and finally we conclude this work in section 5 and propose further research for the future.

## 2 Invariant Features

In [Siggelkow 02, Siggelkow & Schael<sup>+</sup> 01] Siggelkow et al. propose features invariant against translation and rotation to retrieve images from a general image database to describe the content of the images. The features do not directly model objects but instead the global appearance of the images is modeled. Invariant means that the feature remains unchanged if the modeled transformations are applied to the image. The features are based on the integral approach to create invariant features. A feature  $F(X)$  is constructed from an image  $X$  by integration over a transformation group  $G$ :

$$F(X) := \frac{1}{|G|} \int_{g \in G} f(gX) dg$$

where  $gX$  is the image transformed by the transformation  $g \in G$  and  $F(X)$  is an arbitrary function depending on the pixel values of  $X$ . Applied to the group of translations and rotations  $G_{rt}$  this results in

$$F(X) = \frac{1}{2\pi N_0 N_1} \int_{t_0=0}^{N_0} \int_{t_1=0}^{N_1} \int_{\phi=0}^{2\pi} f(g_{t_0, t_1, \phi} X) d\phi dt_1 dt_0$$

and choosing for example  $f(X(i, j)) = X(1, 0) \cdot X(0, 2)$  this results in:

$$F(X) = \frac{1}{2\pi N_0 N_1} \int_{t_0=0}^{N_0} \int_{t_1=0}^{N_1} \int_{\phi=0}^{2\pi} X(\sin \phi + t_0, \cos \phi + t_1) \cdot X(2 \cos \phi + t_0, -2 \sin \phi + t_1) d\phi dt_1 dt_0.$$

Usually the integrals are replaced by sums to achieve discretization. The feature  $F(X)$  is invariant against rotation and translation, but only results in one value per image. This is not discriminative enough, that is this one value does not contain enough information to distinguish between different images. To avoid this problem we replace one (or more) of the sums by histogramization. Here we replace the sums accounting for translations by histogramization. This yields the histogram

$$H_F(X) = \text{hist}_{t_0=1}^{N_0} \text{hist}_{t_1=1}^{N_1} \frac{1}{R} \sum_{r=1}^R f(g_{t_0, t_1, \frac{2\pi r}{R}} X)$$

Here histogramization is denoted by the operator *hist* and rotation is carried out in  $R$  steps. A histogram is an estimation of the distribution of a variable. For this the feature space  $\mathcal{S}$  is divided into  $M$  regions  $\mathcal{S}^m$ . Usually these region form a regularly spaced grid, e.g. the regions  $\mathcal{S}^m$  are hypercubes of the same size, but this is not a requirement. Formally:

$$\begin{aligned} \mathcal{S}^m \subset \mathcal{S} \quad \text{with} \quad \bigcup_{m=1}^M \mathcal{S}^m &= \mathcal{S} \\ \text{and} \quad \mathcal{S}^m \cap \mathcal{S}^{m'} &= \emptyset \quad \forall m \neq m' \end{aligned}$$

The probability for data points falling into one of these regions is determined by counting. Let  $K^m$  be the number of data points falling into region  $\mathcal{S}^m$ , then the probability for any data point falling into this region is given by  $P(m) = P(x \in \mathcal{S}^m) = \frac{K^m}{N}$ .

Because this features account mainly for color and in computer vision texture is very important we also use texture features proposed by Tamura in [Tamura & Mori<sup>+</sup> 78]. We calculate coarseness, contrast and directionality for every pixel and create a histogram of these values. This histogram is then combined with the invariant feature histogram.

### 3 Clustering Algorithms

Clustering is the unsupervised classification of patterns into groups (clusters). The clustering problem has been addressed in many contexts and has shown to be useful in many applications. However, clustering is a combinatorially difficult problem. To cluster images into groups of visually similar images we propose to use the feature histograms as proposed above to represent the images and two well known clustering methods:  $k$ -means [McQueen 67] and LBG clustering [Dempster & Laird<sup>+</sup> 77, Linde & Buzo<sup>+</sup> 80]. Both are explained only briefly here.

The  $k$ -means is a simple algorithm and uses a squared error criterion. It starts with a random initial partition and keeps reassigning the patterns to cluster centers based on the similarity between the pattern and the cluster centers until a convergence criterion is fulfilled. A problem with this algorithm is that it is sensitive to the selection of the initial partition and that it might converge to a local minimum. Also the user has to specify the number of clusters. The result can be interpreted as a mixture of Gaussians (normal distributions).

The LBG clustering algorithm is an expansion of the  $k$ -means algorithm to overcome the problem of choosing an initial partition. This is important here, because when searching for images the user is not able to foresee how many clusters are needed. Initially the algorithm sees the data as one large Gaussian which is iteratively split and reestimated to yield a mixture of Gaussians.

### 4 Database & Experimental Results

To test the approach we created a database of Google image search results by querying Google image search with 100 English words and saving the first 120 thumbnails the search returned. This yielded a database of 12 000 images from 100 classes. From all of these images we extracted the invariant feature histograms as described above and applied the clustering methods to each of the classes to observe how the images are rearranged. The results are visually promising, but at the moment it is not possible to present quantifiable

















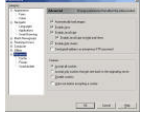

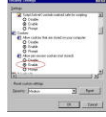

cluster 1					
cluster 2					
cluster 3					
cluster 4					

Table 1: Results from clustering images resulting from a google image search for “cookie” with the EM-algorithm. 4 clusters are found, 5 images from each cluster are shown. Cluster 1 contains mainly images dealing with cookies and people, cluster 2 contains mainly images of edible cookies, cluster 3 contains mainly light grey images, and cluster 4 contains mainly screenshots from applications dealing with Internet-cookies.

results for this database, since we do not know which images belong to the same cluster. Some example results are shown in tables 1 and 2. Though it is not possible to show complete clusters, it can be seen, that the clusters mainly consist of visually similar images. Also it can be seen that the similarity is mainly based on the color distribution of the images.

Using a database of 1000 images from 10 classes (available at <http://wang.ist.psu.edu>) we are able to create quantifiable results. The database is a manually selected subset of the Corel database which is well known in image retrieval applications. We use two measures for a given partition: cluster purity  $S$  and class completeness  $R$ . Let  $C$  be the number of clusters,  $K$  the number of classes,  $K_i$  the set of images from class  $i$ , and  $C_j$  the set of images from cluster  $j$ .

$$S := \frac{1}{C} \sum_{i=1}^C \max_{j=1}^K \frac{|K_j \cap C_i|}{|C_i|} \quad R := \frac{1}{K} \sum_{j=1}^K \max_{i=1}^C \frac{|K_j \cap C_i|}{|K_j|}$$

If only one cluster is created then  $R$  is always 1 and if there is one cluster per observation  $S$  is always 1. So it is important to find a good tradeoff between this two measures. Results obtained on the WANG database are shown in table 3.

## 5 Conclusion & Perspective

We presented a method to improve text based searching in image databases using methods from computer vision and data mining. The results are a good starting point, since the clusters contain mainly visually similar images. Also the results obtained on the WANG database show that the method produces good clusters and that the user will be presented with an easier browseable set of images.

To give more precise results we plan to use other measures to compare cluster results. Using the proposed features in other applications like image retrieval and classification we hope to gain further information on how to improve the results here.





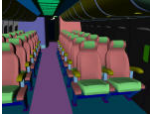











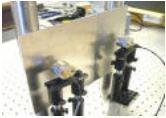








cluster 1					
cluster 2					
cluster 3					
cluster 4					
cluster 5					

Table 2: Results from clustering images for “aircraft” with the LBG cluster algorithm. 5 clusters are found. Cluster 1 contains mainly artificial images of aircrafts, cluster 2 contains images of aircrafts with a lot of sky in the background, cluster 3 contains different types of images, cluster 4 contains images of aircrafts with a lot of the color gray and cluster 5 contains many images of black and white drawings.

number of clusters	S	R
4	0.64	0.83
16	0.73	0.52

Table 3: Results on the WANG database using LBG clustering and different numbers of splits.

## References

- [Dempster & Laird<sup>+</sup> 77] A. Dempster, N. Laird, D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. Royal Statistical Society Series B*, Vol. 39, pp. 1–38, 1977.
- [Linde & Buzo<sup>+</sup> 80] Y. Linde, A. Buzo, R. Gray. An algorithm for vector quantization design. *Proc. IEEE Transactions on Communications*, Vol. 28, pp. 84–95, January 1980.
- [McQueen 67] J. McQueen. Some methods for classification and analysis of multivariate observations. *Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.
- [Siggelkow & Schael<sup>+</sup> 01] S. Siggelkow, M. Schael, H. Burkhardt. SIMBA — Search IMages By Appearance. *Proc. of the 23rd DAGM Symposium*, Vol. 2191, pp. 9–16, september 2001.
- [Siggelkow 02] S. Siggelkow. *Feature Histograms for Content-based Image Retrieval*. Ph.D. thesis, Universität Freiburg, 2002.
- [Tamura & Mori<sup>+</sup> 78] H. Tamura, S. Mori, T. Yamawaki. Textural Features Corresponding to Visual Perception. *IEEE Transaction on Systems, Man, and Cybernetics*, Vol. SMC-8, No. 6, pp. 460–472, June 1978.