



Universiteit
Leiden
The Netherlands

Content-based retrieval of visual information

Oerlemans, A.A.J.

Citation

Oerlemans, A. A. J. (2011, December 22). *Content-based retrieval of visual information*. Retrieved from <https://hdl.handle.net/1887/18269>

Version: Corrected Publisher's Version

License: [Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden](#)

Downloaded from: <https://hdl.handle.net/1887/18269>

Note: To cite this publication please use the final published version (if applicable).

Chapter 2

Features

This chapter describes the low level features we have used in the content-analysis of images. First, a short introduction is given to explain what a low level feature is and then the features that were used in this thesis are explained in detail. Also, we describe a few measures for calculating the similarity of low level features.

2.1 Introduction

The contents of images need to be described in a form that the search system understands. This can be done in various ways and usually one starts with the extraction of low level features. A low level feature can be extracted from an image by calculating a mathematical formula or by running a simple algorithm on the image data. The result is a number or a set of numbers that represents the feature and this set of numbers is called the feature vector. These vectors are almost always normalized to unit length. A low level feature generally focuses on aspects such as color, texture or shape.

Low level features can be combined to form more complex descriptions of image contents and these are often called high level features or high level semantics. Examples are 'grass', 'building' or 'flag'. The higher level features are difficult to measure directly from the image contents and often need to be trained with examples to be usable as a feature.

As mentioned in the previous chapter, the bridge between these two representations is called the semantic gap and there is still no clear solution on how to define a high level feature in terms of low level features. One might question if there will ever be an unambiguous way of representing high level concepts with low level features.

2.2 Color features

2.2.1 Color histogram

A color histogram represents the distribution of colors in the image. For example, if we take an image with RGB pixel values in the range $[0, 255]$, a histogram of the distribution of these RGB values can be created with 64 bins by quantizing the color information for each channel into 4 ranges: $[0 \dots 63]$, $[64 \dots 127]$, $[128 \dots 191]$, $[192 \dots 255]$. In other words, this is the same as reducing the bits per channel to 2 and then using the combined 6 bit RGB value as an index in the histogram.

The color space used for the histogram is arbitrary, although it has been shown that using the YUV space has better retrieval performance than the RGB space [74]. Also, the number of bits per channel can be of influence to the performance of the feature.

2.2.2 Color moments

Color moments are also based on the distribution of color values in the image, but this feature tries to capture the distribution in just a few parameters. In statistics, the n -th central moment μ_n of a random variable X or a probability density function $f(x)$ with mean μ is:

$$\mu_n = E[(X - E[X])^n] = \int_{-\infty}^{\infty} (x - \mu)^n f(x) dx \quad (2.1)$$

The first central moment μ_1 is defined as zero. The second central moment is equal to the variance of the distribution and the third central moment is termed the skewness, a measure of symmetry for the distribution.

The fourth central moment is the kurtosis of the distribution, a value representing the type of measurements that resulted in the given variance. Higher kurtosis means that the variance is the result of a small number of more extreme measurements, instead of a larger number of measurements with lower variance.

In this research we have used the second, third and fourth central moment of the distribution of color values as a low level feature, which again can be applied to each of the individual color channels of the color space that is used. Note that these moments can also be used on grayscale values, which then results in a feature that is on the boundary of color (intensity) and texture.

2.3 Texture features

2.3.1 Local binary patterns

The Local Binary Patterns (LBP) texture feature is a feature that was introduced by Harwood [24] and Ojala [60] and that is invariant to monotonic changes in gray scale. The basis of the feature is the distribution of grayscale differences in regions of 3x3 pixels. The center pixel in the 3x3 region is used as a threshold for the other 8 pixels and each of these pixels is then converted to a binary value and then multiplied by a fixed value based on its location in the region, after which they are summed to get the LBP value for the 3x3 region.

$$LBP = \sum_i 2^i |I_i > threshold| \quad (2.2)$$

Where i ranges over the 8 locations mentioned before. In Figure 2.1, an example is shown of how the LBP value for a 3x3 region is calculated. In this case, the final LBP value is 25.

9	3	4	1	0	0	1	2	4	1	0	0
7	5	6	1		1	8		16	8		16
2	4	4	0	0	0	32	64	128	0	0	0
a			b			c			d		

Figure 2.1: a) an example of a 3x3 region with grayscale values, b) the thresholded and converted values of the region, c) the fixed values for each pixel location, d) the values that are taken into account for the overall LBP value

The distribution of LBP values over an image results in a histogram with 256 bins and this histogram can be used for similarity comparisons.

2.3.2 Symmetric covariance

The symmetric covariance texture measure was published in 1993 by Harwood [24]. It forms a histogram of values that are calculated for 3x3 regions of pixels, very much like the LBP texture measure, only this feature focuses on the pair-wise differences of two pixels in the neighborhood, instead of looking at the entire 3x3 region.

Given a 3x3 neighborhood of a pixel as seen in figure 2.2, the SCOV value is defined as:

g_2	g_3	g_4
g_1		g_1'
g_4'	g_3'	g_2'

Figure 2.2: A 3x3 region around a pixel, with each surrounding pixel labeled

$$SCOV = \frac{1}{4} \sum_{i=1}^4 (g_i - \mu)(g_i' - \mu) \quad (2.3)$$

where μ is the mean grayscale value of the 3x3 region.

2.3.3 Gray level differences

Ojala et al. [60] describe four different texture features, based on the absolute gray level differences of neighboring pixels. The simplest two, DIFFX and DIFFY, create a histogram of the absolute differences in horizontal and vertical directions. The DIFF2 feature creates one histogram for both the horizontal and vertical directions and DIFF4 also includes the diagonal directions. DIFF4 is therefore rotational invariant (with respect to 45 degree angles).

2.4 Feature vector similarity

When comparing image features, there are several methods for calculating the similarity. Examples of commonly used similarity measures are:

- L_P , where P can be $1, 2, \dots, \infty$.

$$d_{L^P}(X, Y) = \left(\sum_{i=1}^n |x_i - y_i|^P \right)^{\frac{1}{P}} \quad (2.4)$$

For $P = 1$, this results in the sum of absolute differences and for $P = 2$, it is the Euclidean distance, or the commonly used distance between two vectors in geometry.

- EMD, or earth-movers-distance. The EMD computes the difference between two distributions in terms of the amount of work it takes to redistribute the values in one distribution to end up with the values of the second distribution. It is defined as

$$EMD(P, Q) = \frac{\sum_{i=1}^m \sum_{j=1}^n f_{ij} d_{ij}}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}} \quad (2.5)$$

where d_{ij} is the distance between two elements of the distribution and f_{ij} is taken from the flow $F = [f_{ij}]$, that is the result of minimizing:

$$WORK(P, Q, F) = \sum_{i=1}^m \sum_{j=1}^n f_{ij} d_{ij} \quad (2.6)$$

However, although these methods do result in a ranking of search results, there is not much intuition behind using these formulas if one does not take a very close look at what feature values really represent. For example, when a YUV feature with three elements is used, does a difference of 0.1 in Y represent the same visual difference as a difference of 0.1 in U? The L_1 distance assumes this, but it is clearly not intuitive.

