Raport Badawczy
Research Report

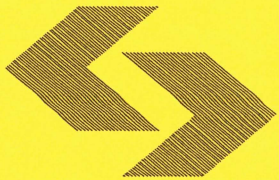
RB/31/2015

Content-based image retrieval

T. Jaworska

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences



POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 3810100

fax: (+48) (22) 3810105

Kierownik Zakładu zgłaszający pracę:
Prof. dr hab. inż. Janusz Kacprzyk

Warszawa 2015

Content-Based Image Retrieval CBIR

Tatiana Jaworska

Abstract—**This paper offers a survey of recent trends in the scientific approaches to image retrieval. We present, in a synthetic way, the richness of modern methods applied to content-based image retrieval (CBIR) systems. The number of systems requires their classification based on different criteria, such as: the level of automation of feature extraction and index generation, the level of adaptability to the needs of users and the query, etc. This paper attempts to provide a survey of variety of recent search engines concept depending on the user needs and applications, where we especially focus on the semantic approach to the search engine constructions. Finally, we discuss image databases use for evaluation of the retrieval algorithms.**

Index Terms—**content-based image retrieval, search engine, query by image, GUI, composed query, semantic retrieval.**

I. INTRODUCTION

IN order to discuss image retrieval, we have to answer some questions of which the first and foremost is how to define our goal: do we want to construct a new CBIR system from scratch or build it on our existing image collections, for example, art collections, medical images, scientific databases or generally, the World Wide Web. Our objective, in turn, predetermines the kind of queries we wish to put. Once we have answered these fundamental questions, we can start thinking about the construction of an effective system.

There are two most popular approaches to query formulation: typing some key words describing the image and setting an example image as a query [1], [2], [3]. The most infrequent strategy is the preparation of queries by drawing several objects with certain properties, like colour, texture, shape, size and location. In most cases, a rough sketch is sufficient. Query by drawing is not popular, perhaps because most users are rather poor at graphic design.

Some applications, for instance medical CBIR systems, allow selecting subregions of interest as part of a query [4]. The user chooses properties of these ROIs, such as shape or texture, to complete the query definition [5].

Chronologically speaking, the first systems used annotations [6], [7], which was an advantage, but this method did not take into account images described in an irrelevant way. Such systems are still used in news agency databases. Next, query by example [8], [9] appeared which allows the user to formulate a query by providing an example image. The system converts the example image into an internal representation of features. At present, most systems use this kind of query, but their/its drawback is the fact that the user first has to find an image which he wants to use as a query. In some systems, like police collections of mug shots or finger-prints, matching is obvious. Nevertheless, in some situations the most difficult task is to find this one proper image to give it to the system as a query by example.

However, there are some other ways, for instance, systems have appeared recently with a composed query introduced by GUI where the user can comprise his query from selected segments or patches [10].

II. GENERAL SYSTEMS STRUCTURE

CBIR systems developed by universities, government organizations, commercial companies or museums, generally use low-level visual contents of an image, such as colour, shape and texture. The middle-level contents, namely, objects and their spatial relationships, are more powerful on condition that the system can segment and recognize the objects. Different modifications

T. Jaworska is with the Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland (e-mail: Tatiana.Jaworska@ibspan.waw.pl).

are observed in particular systems, but their basic structure is presented in Fig.1. Here we can see the image collection from which visual features are extracted and saved as a feature bank and additionally, a set of text annotations (optional in some systems). When the user puts a query, the search engine searches the most similar group of images and sends them back to the user as retrieval results.

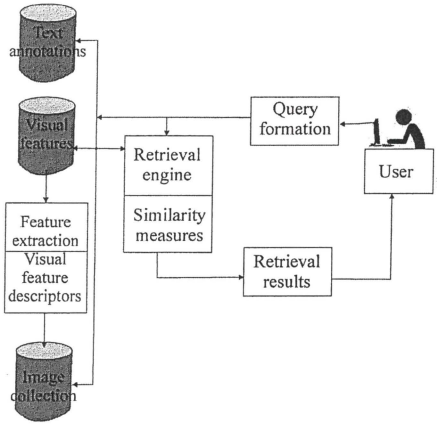


Fig. 1. General CBIR architecture.

Feature extraction can be calculated from the entire image (global features) or from specific regions, or segments (local). Proper feature descriptors should be resistant to accidental variance of features and invariant to scaling and rotation of image.

A. Criteria for a classification of CBIR systems

According to Chang et al. [11] classification of CBIR systems can be based on different criteria:

- **The level of automation of feature extraction and index generation**. At present, we can observe rapid progress in the automatic low-level feature extraction methods but much slower development of mid-level features, namely, segment extractions [12]. The most state-of-the-art method in this category is the scale-invariant feature transform (SIFT) [13], [14] (described in sec. III.G), which generates a large collection of feature vectors, each of which is invariant to image translation, scaling, and rotation, partially invariant to illumination changes and robust to local geometric distortion. Dominant orientations are assigned to localized keypoints. Index is generated based on storing SIFT keys and identifying matching keys from the new image. However, object classification remain still a complicated task.
- **The level of integration of multimedia modalities**. Media modalities cover images, video, films, graphics, multimodal satellite images, text and audio files. The most evident multimedia examples can now be found in medicine [15], where images acquired from different imaging techniques are combined for a 2D or 3D projection. The most dynamic development can be observed in video and film retrieval where particular images or objects need to be traced down or a scene has to be selected from the whole material. In this area the most rapid achievements are noticeable in news agencies, where there is a continual downpour of new information. Effective retrieval of the relevant fragment of a film is a very challenging task in any news agency's video database.
- **The level of adaptability to the needs of users and the query**. The most difficult issue in multimedia information retrieval is how to make a query describing the needs of the user. At present, the user requirements are reflected in the query asking methods, which can be generally [16] divided into:
  1. Interactive techniques based on feedback information from the user, commonly known as relevance feedback (RF) [17], [18];
  2. Automated techniques based on the global information derived from the entire collection;
  3. Automated (might also be interactive in some cases) techniques based on local information from the top retrieved results, commonly known as local feedback or collaborative image retrieval (CIR) [19] which is a powerful tool to narrow down the

semantic gap between a low- and high-level concept .

In a global analysis approach, all the documents in the collection are indexed. This structure is then used to select additional terms for the query expansion. In local analysis, the top retrieved documents for a query are examined (usually without any assistance from the user) at query time to determine the terms for the query expansion [20]. Large modern DBs actively employ user's interaction, namely, relevance feedback by labelling some top similar and dissimilar images as positive and negative samples. Images labelled in this way are incorporated into a training set. A more precisely labelled training set boosts algorithms to build a wider boundary between clusters [21], [22], [23].

- **The level of abstraction.** Among many content-based image retrieval (CBIR) systems available on the market, there exist three main approaches to CBIR. Firstly, images are determined by text annotations. Secondly, images are retrieved by matching an example based on low-level descriptors, such as colour, texture, shape, etc. [24]. Thirdly, some semantic information is selected from analyzing images to retrieve a similar scene [25], [26], [27].
- **The level of generality of the visual information domain**. Image information depends on the image domain. The most general images are found on the WWW. The medical images are the most varied because a vast array of diagnostic devices generates them, for example, magnetic resonance imaging (MRI), X-ray computer tomography (CT), positron emission tomography (PET) [15], medical ultrasonography, endoscopy, elastography, tactile imaging, thermography or medical photography. Another example can be satellite images, where a multispectral projection is most frequently used. Users of each domain require different information, which results in different methods of image processing.
- **The level of automation of the database collection**. Different image collections are acquired with diverse methods. Automatic acquisition is generally used in monitoring different processes from biological [28] through industrial (machine vision) [29] or microscopic scale to satellite. Personal images (portraits) are acquired manually unless they are photographs from a CCTV at airports or other monitored objects.
- **The level of information retrieval**. There are two measurements, *recall* and *precision,* to evaluate the performance of the retrieval system. For a query q, the database set of images relevant to the query q is denoted as $R(q)$, and the retrieval result of the query q is denoted as $Q(q)$. The precision (1) of the retrieval is defined as a fraction of the retrieved images that are indeed relevant to the query:

$$precision = \frac{|Q(q) \cap R(q)|}{|Q(q)|} \quad (1)$$

The recall (2) is the fraction of relevant images that is returned by the query:

$$recall = \frac{|Q(q) \cap R(q)|}{|R(q)|} \quad (2)$$

Usually, precision and recall are only rough descriptions of the performance of the retrieval system because recall tends to increase, while at the same time precision is likely to decrease. Image and video retrieval is based on how the contents of an image or a chain of images can be represented. The users of a CBIR system have a diversity of goals, in particular, *search by association*, *search for a specific image*, or *category search* [30].

The search by association often implies iterative refinement of the search, the similarity or the examples with which the search was started as the user has only vague aim of an image of interest. Systems in this category typically are highly interactive, where

the specification using sketches or example images. The search for a precise copy of the image in mind or for another image of the same object assumes that the target can be interactively specified as similar to a group of given examples.

For a given query, the system first retrieves a list of images ranked according to a predefined similarity metric. Gradually, the images start to be analysed as a "local concept" space which means that the perceptually and semantically distinguishable colour and texture patches from local image regions in individual images are examined. Then, a similarity can be expressed as a comparison of the query to the whole collection, or only as the local analysis for the correlations between the concepts based on the co-occurrence pattern [20]. Another approach takes into account multi-set data mining and object spatial relationship in a three stage search engine [31].

With the growing amount of images one of the latest developments the Peer-To-Peer (P2P) CBIR search engine. It has been designed to provide multi-instance query with multi-feature types to effectively reduce network traffic and maintain high retrieval accuracy. These systems have also been designed to provide scalable retrieval among the fully centralized and fully de-centralized database framework, which can adaptively control the query scope and progressively refine the accuracy of retrieved results.

At present, many commercial and academic search engines are offered (a list on the WWW page [32]). The recent progress in image retrieval has been made due to scene understanding [25], [27], [33].

- **The level of visual information compression.** Modern technologies enable us to use visual information, practically, in all user's front-end equipment. As visual information is very resource consuming, information distribution requires that data transmission is carried out only in a compressed form [34], [35]. Both image compression and transmission have forced the development of many new methods of sending data. Depending on compression rates we divide compression into lossy and lossless. The former, generally, provides higher compression rates, but it is more affected by impairments caused during data transmission on a wireless network, whereas, the latter compression algorithms fully regenerate the original information at the receiver. Some of the commonly used compression standards are JPEG (Joint Photographic Experts Group), GIF (Graphics Interchange Format), and PNG (Portable Network Graphics) [36]. The JPEG 2000 standard was developed using Discrete Wavelet Transformation (DWT) instead of Discrete Cosine Transformation (DCT), which is used for the JPEG codec. As a result, JPEG 2000 offers higher compression rates without introducing the blocky and blurry effects introduced by the original JPEG standard. Furthermore, JPEG 2000 allows progressive downloading of images with different resolution, quality, components, or spatial regions, eliminating the problem of decompression of the entire image before it can be displayed. This feature is particularly useful for Wireless Multimedia Sensor Networks (WMSN).

B. Classification of CBIR systems

According to a search engine mechanism we can divide CBIR systems into the following [37]:

- Retrieval based on low-level features
  - SIFT
- Retrieval based on annotations
- Hierarchical databases [38]
- Relevance feedback and learning
- Aim dedicated CBIR
  - medical
  - police
- Browsers of image collections
- Systems for image classification
- Semantic retrieval
- Scene understanding

These search engine mechanisms will be presented in details in sec. VII in terms of searching techniques.

III. Visual feature descriptors

There is no universal or exact definition of what constitutes a feature, and the exact definition often depends on the problem or the type of application. Feature detection is a low-level image processing operation. Below we present the most common used algorithms which are further useful in search engines construction.

A. Colour information

Colour is a commonly used feature because its layout in the image is the key information whereas the simpler systems extract only global features from the colour image. The more advanced ones use colour information about regions or separate segments [24]

Each pixel of the image can be represented as a point in a 3D colour space. Many colour spaces for image retrieval, including *RGB, Munsell, CIE L*a*b*, CIE L*u*v*, HSV* are used depending on the aims and the method of image acquisition.

Colour information gives the opportunity to construct such descriptors as:

- colour moments (mean, variance and skewness) [39], [40] help to describe colour distribution in the whole image, which is the basis for many CBIR retrieval processes. Nevertheless, they do not give the spatial information about pixels;
- the colour histogram is easy to compute and invariant in terms of scaling and rotation, however, it also fails to provide spatial information about pixels, so many images have similar histograms;
- the colour coherence vector (CCV) [41] is constructed based on the colour histogram. In this case, each histogram bin, a separate one for each colour, is partitioned into two parts: coherent if it belongs to a large uniformly-coloured region, and incoherent in the opposite case. It means that two pixels a and a' are coherent if they belong to region C such that $a,a' \in C$ and there exists a path in C between a and a'. For the image, the CCV is defined as the vector $[(\alpha_1,\beta_1), (\alpha_2,\beta_2),\ldots,(\alpha_N,\beta_N)]$, where α_i denotes the number of coherent pixels of the i^{th} colour bin, whereas β_N denotes the number of coherent pixels. The additional spatial information included in the CCV improves the results of retrieval in comparison to the simple colour histogram.
- the colour correlogram, also called a second-order histogram, describes the spatial correlation of pairs of colours. A colour correlogram is a table indexed by colour pairs, where the k^{th} entry for (i,j) specifies the probability of finding a pixel of colour j at a distance r from a pixel of colour i in the image. The colour correlograms for all sets of colours are rather large, therefore, a simplified version is an auto-correlogram which is a spatial relationship only between points of identical colour.

B. Texture Information

Texture is one of the most important visual cues to identify homogeneous regions [42]. The goal of texture classification is to identify each uniform texture region, whereas the goal of texture segmentation is to obtain the boundary map and further separate regions characterized by different textures. Another procedure is texture synthesis, often used for image compression and also important in computer graphics where the aim is to render object surfaces which are as realistic looking as possible.

For our purpose, the key operation is segmentation, in a nutshell, we can assume that image texture is an attribute representing the spatial arrangement of grey or colour levels of the pixels in a region [43]. Hence, the intensity variations in an image, characterizing texture, generally reflect physical variations in the real scene. To model these variations the following issues need to be addressed:

- pixel colour value in a spatial neighbourhood;
- spatial distributions of these values;
- their resolution or scale;
- the unrecognizability of separate primitive objects in a texture region.

Basically, texture representation methods can be classified into four categories: structural, statistical, fractal [44] and transformational [24] as it can be seen in Fig. 2. The first category of methods can be divided into morphological operators and adjacency graphs presenting texture as structural primitives and their placement rules. The primitive can be as simple as a single pixel that can take a grey value, but it is usually a collection of pixels. The placement rule is defined by a tree grammar. A texture is then viewed as a string in the language defined by the grammar whose terminal symbols are the texture primitives. An advantage of this method is that it can be used for texture generation, as well as texture analysis. The patterns generated by the tree grammars can also be regarded as ideal textures in Zucker's model [45]. They are more effective when we have a regular texture.

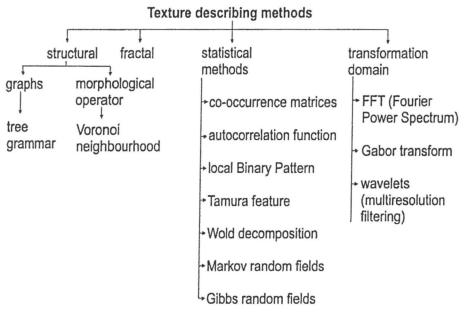


Fig. 2. The categories of texture describing methods.

Another example is Voronoi features [46], which were proposed because the local spatial distributions of tokens are reflected in the shapes of Voronoi polygons. Many of the perceptually significant characteristics of a token's environment are manifest in the geometric properties of Voronoi neighbourhoods. In order to apply geometrical methods to grey level images, we only need to first extract tokens from images.

The statistical methods describe texture by statistical distribution of image intensity. There are numerous statistical texture representations:

– co-occurrence matrices. Spatial grey level co-occurrence estimates image properties related to second-order statistics. Haralick [47] suggested the use of the $G \times G$ grey level co-occurrence matrix $P_{\mathbf{d}}$ for a displacement vector $\mathbf{d} = (dx, dy)$, defined as follows. The entry (i,j) of $P_{\mathbf{d}}$ is the number of occurrences of the pair of grey levels i and j which are a distance $\mathbf{d}$ apart. Formally, it is given as:

$$P_{\mathbf{d}}(i,j) = \\ = |\{((x,y),(t,v)) : I(x,y) = i, I(t,v) = j\}| \quad (3)$$

where $(x,y),(t,v) \in N \times N$, $(t,v) = (x+dx, y+dy)$, and $|.|$ is the cardinality of a set. Based on this matrix some useful texture features can be described, such as: energy, entropy, contrast, homogeneity or correlation.

– Autocorrelation function – can be used to assess the amount of regularity as well as the fineness – coarseness of the texture in the image. Formally, the autocorrelation function of an image $I(x,y)$ is defined as follows:

$$\sigma(x,y) = \\ = \frac{\sum_{u=0}^{N}\sum_{v=0}^{N} I(u,v)I(u+x,v+y)}{\sum_{u=0}^{N}\sum_{v=0}^{N} I^2(u,v)} \quad (4)$$

– Local Binary Pattern (LBP) operator [48] is, originally, based on a 3×3 pixel neighbourhood (see the *example* sub-table in TABLE I). Image pixels in each neighbourhood of a pixel (i,j) are exchanged into a binary threshold map where 1 is for a pixel larger than the central pixel and 0 where the values are less than the central one (see the *threshold* sub-table in TABLE I). The values of the pixels in the threshold map are multiplied by the weights given to the corresponding pixels. The weights are the power of 2 where the number of neighbourhood is the exponent (see the *weights* sub-table in TABLE I). Finally, the values of the eight weighted pixels are summed to obtain one factor (the *result* can be seen in TABLE I). The LBP histogram computed over a region is used as a texture description. Because of the LBP design, it is

invariant under any monotonic grey scale transformation and provides information about the spatial structure of the local image texture. Due to its 3×3 window operation, however, feature distributions may be sensitive to geometric distortion. This operator was extended later [49] for neighbourhoods of different sizes, for instance, circular neighbourhood and bilinear interpolation of non-integer pixel values.

TABLE I
COMPUTATION OF LOCAL BINARY PATTERN (LBP).

Example	Threshold	Weights	Result
8 5 2	1 0 0	1 2 4	1 0 0
7 6 1	1 x 0	8 x 16	8 x 0
9 13 7	1 1 1	32 64 128	32 64 128

LBP = 1+8+32+64+128=233

The original LBP operator was defined to only deal with the spatial information. Later, it was extended to a spatiotemporal representation for dynamic texture analysis. For this purpose, the so-called Volume Local Binary Pattern (VLBP) operator was proposed [50].

- Tamura feature; Tamura et al. [51] considered six basic textural features:
  - coarseness – relates to distances of notable spatial variations of grey levels, that is, implicitly, to the size of the primitive elements forming the texture.
  - contrast – measures how grey levels vary in the image and to what extent their distribution is biased to black or white. The second-order and normalised fourth-order central moments of the grey level histogram are used to define the contrast.
  - directionality – measured the distribution of oriented local edges against their directional angles using the Sobel edge detector.
  - line-likeness - is defined as an average coincidence of the edge directions.
  - regularity – is defined as the normalised sum of the standard deviations of the corresponding above-mentioned feature.
  - roughness – feature is given by simply summing the coarseness and contrast measures.
- Wold decomposition, [52], [43] provides three different components to describe texture: *harmonic, evanescent,* and *non-deterministic,* corresponding to *periodicity, directionality,* and *randomness* introduced by his predecesors. Periodic textures have a strong harmonic component, highly directional textures have a strong evanescent component, and less structured textures tend to have a stronger non-deterministic component. The deterministic periodicity of the image is analysed using the autocorrelation function. The corresponding Wold feature set consists of the frequencies and the magnitudes of harmonic spectral peaks (e.g. the largest peaks). The nondeterministic (random) components of the image are modelled with the multiresolution simultaneous autoregressive (MR-SAR) process. The retrieval uses matching of the harmonic peaks and the distances between the MR-SAR parameters. The similarity measure involves a weighted ordering based on the confidence level in the query pattern regularity.
- Markov random fields [53]. Random field models consider an image as a 2D array of random scalars (grey values) or vectors (colours). In other words, the signal at each pixel location is a random variable. Each type of texture is characterised by a joint probability distribution of signals that accounts for spatial inter-dependence, or interaction among the signals. The interacting pixel pairs are usually called neighbours, and a random field texture model is characterized by the geometric structure and quantitative strength of interactions among the neighbours. If pixel interactions are assumed translation invariant, the interaction structure is given by a set N of characteristic neighbours of each pixel. This results in the Markov random field model where the conditional probability of signals in each pixel (i,j) depends only on the signals in

the neighbourhood $\{(i+m, j+n): (m,n)$ from the set $N\}$.

- Gibbs random fields (GRF). This theory is borrowed from Gibbs principal ensembles of statistical thermodynamics. We move from particles to pixels and still analyse potential and energy functions. Hence, GRF assigns a probability mass function to the entire lattice:

$$P(\mathrm{X} = \mathrm{x}) = \frac{1}{Z}\exp\left[-\sum_{c_i \in C} E(c_i)\right], \quad \forall \mathbf{x} \in \Omega \quad (5)$$

where Z is a normalizing constant known as the partition function and $E(c_i)$ is the energy function.

For texture analysis, general generic Gibbs random field models with multiple pairwise pixel interactions allow to relate the desired neighbourhood to a set of most 'energetic' pairs of the neighbours. A Gibbs distribution is usually defined with respect to cliques, i.e. proper subgraphs of a neighbourhood graph on the lattice. A clique is a particular spatial configuration of pixels, in which all its members are statistically dependent on each other. Then the interaction structure itself and relative frequency distributions of signal co-occurrences in the selected pixel pairs can serve as texture features.

Many natural surfaces have a statistical quality of roughness and self-similarity at different scales. Fractals are very useful and have become popular in modelling these properties in image processing but scale variations can have a great impact on the imaged appearance of a texture. Self-similarity across scales in fractal geometry is a crucial concept. The fractal dimension gives a measure of the roughness of a surface.

Fractal-based texture analysis was introduced by Pentland in 1984 [54]. For images, we use two spatial dimensions and the third dimension is image intensity $I(x,y)$. To apply the fractal model to an image surface, we need to assume that: the random function $I(x)$ is a fractal Brownian function and the fractal dimension of a fractal Brownian function is invariant over transformations of scale[1]. In order to obtain the fractalness of an image, Pentland introduced the description of the image change $\Delta I = I(x + \Delta x) - I(x)$ with scale as follows (eq. (5) [54]):

$$E(|\Delta I_{\Delta x}|)\, \|\Delta x\|^{-H} = E(|\Delta I_{\Delta x = 1}|) \quad (6)$$

where: $E(|\Delta I_{\Delta x}|)$ is the expected value of the change in intensity ΔI over distance Δx, H is the Hurst exponent [55], [56]. Equation (6) is the mutual relation of the image intensities expressed in a statistical way.

We can assume that $\kappa = E(|\Delta I_{\Delta x = 1}|)$, hence we obtain in the above equation $E(|\Delta I|) = \kappa\, \|\Delta x\|^{H}$. By applying log to both sides we have

$$\log E(|\Delta I|) = \log \kappa + \log H \|\Delta x\|. \quad (7)$$

The Hurst exponent H can be obtained by using the least-squares linear regression to estimate the slope of the grey- level difference $GD(k)$ versus k in a log-log scale and k varies from 1 to the maximum value s, where:

$$GD(k) = \frac{\sum_{x=1}^{N}\sum_{y=1}^{N-k-1}|I(x,y) - I(x,y+k)|}{2N(N-k-1)} + \frac{\sum_{x=1}^{N-k-1}\sum_{y=1}^{N}|I(x,y) - I(x+k,y)|}{2N(N-k-1)} \quad (8)$$

The fractal dimension FD can be derived from the relation $FD=3-H$. The approximation error of the regression line fit should be determined to

[1] Following Mandelbrot [183] the increments of a random function $\{X(t,\omega);\ -\infty < t < \infty\}$ are said to be self-similar with parameter $H \geq 0$ if for any $h > 0$ and any moment t_0

$$\{X(t_0+\tau,\omega) - X(t_0,\omega)\} \triangleq \{h^{-H}[X(t_0+h\tau,\omega) - X(t_0,\omega)]\}.$$

If $X(t_0,\omega)$ has self-similarity and stationary increments and is mean square continuous, then $0 \leq H < 1$ there is a constant V such that

$$E\,[\,X(t_0+\tau,\omega) - X(t_0,\omega)]^2 = V\tau^{2H}.$$

For images, following Pentland [54], instead of time t we speak about spatial dimension x, so we have $E[I(x+\Delta x) - I(x)]^2 = V\,\Delta x^{2H}$.

prove that the analyzed texture is fractal, and thus be efficiently described using fractal measures. A small value of the fractal dimension *FD* implies that a large value of the Hurst exponent *H* represents fine texture, while a large *FD*, implying a smaller *H* value, corresponds to coarse texture [57].

Recently, texture descriptors have been based on transformational models. Let us recall the basic notions about the unitary transform. A general linear operation on the input image $I(x,y)$ results in an $M \times N$ output image $U(m,n)$ which is defined by:

$$U(m,n) = \sum_{x=0}^{K-1} \sum_{y=0}^{J-1} I(x,y) O(x,y;m,n) \qquad (9)$$

where: $O(x,y;m,n)$ is the operator kernel.

Based on this universal rule we can chronologically describe the most useful transformational methods:

- Fourier power spectra and Fast Fourier Transform (FFT) [58]. For image function $I(x,y)$ we compute its Fourier transform as:

$$F(u,v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} I(x,y) \exp\left\{\frac{-2\pi i}{N}(xu+yv)\right\} \qquad (10)$$

  where (u,v) are the spatial frequencies and the quantity $|F(u,v)|^2$ is defined as the power spectrum which, in fact, is the modulus of a complex number. In the image terms the energy distribution of the power spectrum reflects the periodical structure of a texture, whereas the directional nature of the texture is reflected in the direction distribution of energy in the power spectrum. Frankly speaking, the limitation at high frequencies is the image resolution.
- The Gabor transform [59], [60]. The Fourier transformation is an analysis of the global frequency content in the signal. Many applications require the analysis to be localized in the spatial domain. This is usually handled by introducing spatial dependency into the Fourier analysis. The classical way is using the window Fourier transform. Considering one dimension, the window Fourier transformation of a sinusoidal wave $f_{u_0}(x) = e^{iu_0 x}$ is defined as:
  $F(u) = 2\pi\delta(u - u_0)$, where δ is the Dirac function. Then its energy is spread over the frequency interval in the neighbourhood of a u_0:

$$\left[u_0 - \frac{\sigma_u}{2}, u_0 + \frac{\sigma_u}{2}\right].$$

When the window function $w(r)$ is Gaussian,

$$w(r) = \frac{1}{2\pi\sigma^2} e^{-\frac{r^2}{2\sigma^2}}; \qquad r^2 = x^2 + y^2 \qquad (11)$$

the transform becomes a Gabor transform [61], [62]:

$$G(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left\{-\pi\left[\frac{(x-x_0)^2}{\sigma_x^2} + \frac{(y-y_0)^2}{\sigma_y^2}\right]\right\} \cdot e^{i(u_0 x + v_0 y)} \qquad (12)$$

where (x_0,y_0) is the center of the receptive window in the spatial domain and (u_0,v_0) is the optimal spatial frequency of the filter in the frequency domain. σ_x and σ_y are the standard deviations of the elliptical Gaussian along x and y. The 2D Gabor function is thus a product of an elliptical Gaussian and a complex plane wave.

The 2D Gabor function consists of a sinusoidal plane wave of a certain frequency and orientation modulated by a Gaussian envelope given by:

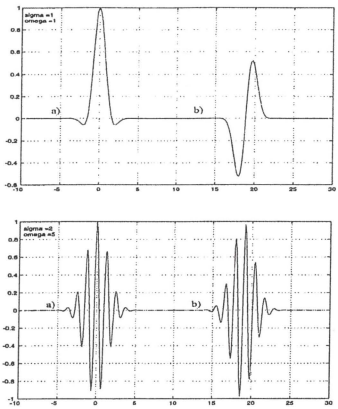


Fig. 3. 1D Gabor function, where a) the real part of the function and b) the imaginary part of the function **[63]**.

$$g(x,y) = \exp\left\{-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right]\right\} \cdot \cdot \cos(2\pi\omega_0(x\cos\theta + y\sin\theta)) \quad (13)$$

where ω_0 and θ are the frequency and phase of the sinusoidal wave, respectively. Then a set of Gabor filters can be obtained by appropriate dilations and rotations of $g(x,y)$ for angles $\theta = \frac{n\pi}{K}$, n= 0,1,..., K-1 (see Fig. 4). In this case, the Gabor transform of an image $I(x,y)$ is defined as:

$$W_n(x,y) = \int I(x,y)\,\overline{g_n}(x-x_1, y-y_1)dx_1dy_1 \quad (14)$$

for which: $\mu_n = \int |W_n(x,y)|\,dxdy$ (15)

$$\sigma_n = \sqrt{\int (|W_n(x,y)| - \mu_n)^2 dxdy} \quad (16)$$

where μ_n is the mean and σ_n is the standard deviation of the magnitude of $W_n(x,y)$ for a particular orientation.

The texture analyzers implemented here are based on 2D Gabor functions which offer a strong correlation with the actual human segmentation and respective visual field profiles are adequately modelled by Gabor filters.

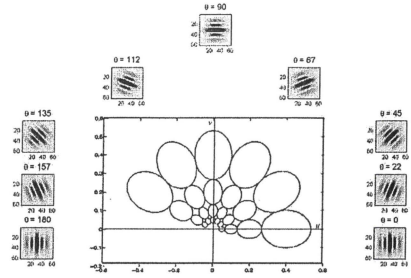


Fig. 4. The examples of 2D Gabor functions for particular angles $\theta = \frac{n\pi}{K}$, where K is the number of orientations. The central contours correspond to the half-peak magnitude of the filter responses in the set of Gabor filters with the upper centre frequency of interest, ω_h = 0.4, the lower centre frequency of interest, ω_l = 0.05, six orientations (K = 6), and four scales (S = 4), follows by **[64]**.

– the wavelet transformations are a big group of methods focused on the multiresolution analysis concept. Generally, the structures to be recognized have a very different size. Hence, it is impossible to define *a priori* an optimal resolution for image analysis. Burt [65] and Crowley [66] have each introduced pyramidal implementation to compute image details in different resolutions. A multiresolution analysis (MRA) yields a scale-invariant interpretation of the image. A multiresolution representation provides a simple hierarchical framework for interpreting the image information. In different resolutions, details of an image generally characterize different physical structures of the scene; in a coarse resolution, these details correspond to larger structures represented by 'big' image components.

The idea of wavelets is based on a basic function called a wavelet (17) with two parameters: one - s, characterizing the scale, the other one - u, indicating the position of the function, introduced instead of the sinusoidal basic function with one parameter ω.

$$\psi_{su}(x) = \frac{1}{\sqrt{s}}\psi\left(\frac{x-u}{s}\right) \tag{17}$$

Hence, the 1D continuous wavelet transform is the projection of an $f(x)$ signal, in the $L^2(R)$, onto the function family $\{\psi_{su}, s > 0,\ u \in R\}$ generated from the single function ψ by translation and dilation:

$$[W_\psi f](s,u) = <\psi_{su}, f> = \\ = s^{-\frac{1}{2}} \int_{-\infty}^{+\infty} \overline{\psi\left(\frac{x-u}{s}\right)}\, f(x)dx \tag{18}$$

The idea of this method is presented in Fig. 5.

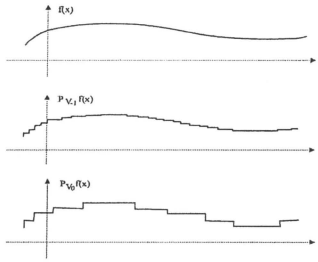


Fig. 5. A function $f(x)$ and its projection onto two consecutive levels V_{-1} and V_0 of the multiresolution analysis **[67]**.

The redundancy of the continuous wavelet transform (18) can be cleared by discretizing both the scale factor s and the translation u. Then, we need a dyadic scale space, $s=2^j$ and $u=k$ with $j,k \in Z$ where Z is an integer. The fragment of the orthogonal basis with levels from V_{-3} to V_{-7} for Symmlet wavelets can be seen in Fig. 6.

The theory of multiresolution signal decomposition was developed by Mallat [68], [69] and Meyer [70] and thus the paradigm for constructing wavelets was established. Polish mathematicians were also involved in this analysis [71].

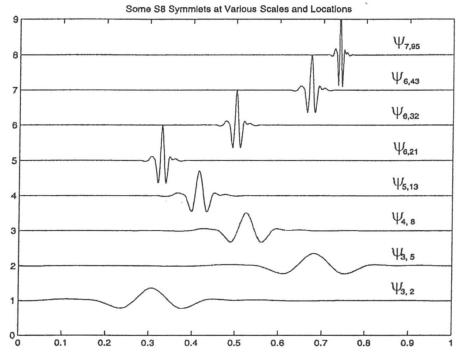


Fig. 6. An example of the dyadic Symmlet wavelet. A scale j and location k are presented for each wavelet $\psi_{j,k}$ on the right side **[63]**.

Many different wavelets have been introduced over the years, some of them are real and some others are complex. The Gabor wavelet (see Fig. 7) is an example of a function in the complex domain. Each ellipse from Fig. 4 represents the frequency support of a dyadic wavelet $\hat{\psi}_{2^j}^K$. This support size is proportional to 2^{-j} and its position rotates when K is modified.

The discrete Gabor wavelet transform (GWT), for example, is applied in texture recognition and segmentation. Sebe and Lew [64] prepared an efficient method based on GWT parameters, such as μ and σ to be used as the texture feature vector. Fig. 8 presents four different grey texture representatives (top left) and the organization of wavelet image coefficients $d_{j,k,l}^p = \langle I, \psi_{j,k,l}^p \rangle$ [72]. The dotted lines show the direction of details (top right). In the bottom left square we can see a wavelet transform for texture images, whereas in the bottom right square we can see the organization of GWT coefficients (cf. (15) and (16)) that constitute feature vector $f = \{\mu_n, \sigma_n\}$.

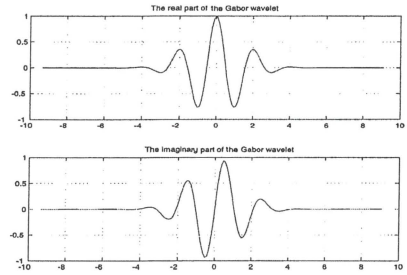


Fig. 7. The real and imaginary parts of the Gabor wavelet for σ =2 and ω=3 which are 'larger' than the example of the subset shown in Fig. 4 **[63]**.

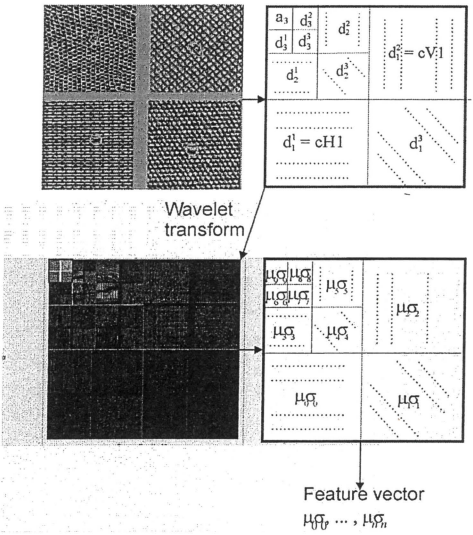


Fig. 8. A texture classifier based on the Gabor wavelet transformation (follows **[64]**, **[72]**).

At the same time, Faizal and Fausi [73] also used the DWT decomposition scheme and they noticed that the DWT of image $M \times M$ gave as a result $(3K+1) \times M^2$ coefficients. Based on this fact they transformed wavelet coefficients to the 3D domain where they looked for clusters whose centroids characterized a texture.

One of the reasons why so many kinds of wavelets were introduced is the fact that the 'shape' of the developed wavelets corresponds with the shape of an analysed image in order to describe its surface more precisely. The idea corresponds to the shape of wavelet applied by Jaworska [12], when the geometrical architectural texture according to its shape can be analysed by the simplest wavelets, namely, the Haar wavelets. Based on 2D FWT maps of object texture are prepared showing the change distributions in a horizontal and vertical direction of a texture.

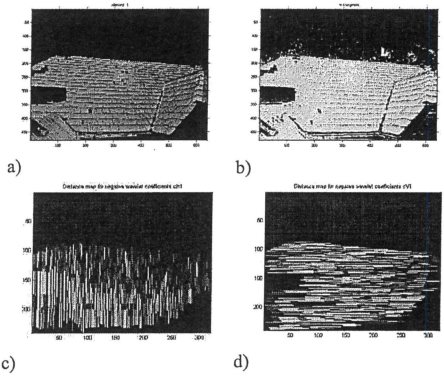


Fig. 9. Distance maps of texture calculated based on the 2D FWT with Haar wavelets. a) original image of a roof segment, b) the red component of the original image, c) distances for negative wavelet coefficients cH1, d) distances for negative wavelet coefficients cV1 [12].

C. Edge detection

The third visual feature which can be identified in an image is edges. In an ideal case, the result of applying an edge detector to an image may lead to a set of connected curves that indicate the boundaries of objects, the boundaries of surface markings, as well as curves that correspond to discontinuities in surface orientation.

Generally, edge detection can assume an image model in which discontinuities of image brightness are likely to correspond to:

- discontinuities in depth;
- discontinuities in surface orientation;
- changes in material properties;

- variations in scene illumination.

The edge or contour can be defined as a parametric curve, polygon, or B-spline, but then can cause problems with the description of non-uniformed topological objects.

The method presented below covers grey images, because edge detection for colour images is more complicated. If a pixel falls on the boundary of an object in an image, then its neighbourhood is a zone of grey-level transition. Edge detection operators examine each pixel neighbourhood, and quantify the slope, as well as the direction, of the grey level transition.

There are several methods to do this, for example:

- thresholding [74]
- watershed algorithm [75];
- gradient methods;
- active contours;
- Hough transform;
- fuzzy thresholding [76];

1) Gradient Methods

When we treat the slope and direction of a potential edge as the magnitude and direction of the gradient vector, respectively, we apply the second derivative of the intensity of a 2D image function $I(x,y)$, namely, the Laplacian:

$$\nabla^2 I(x,y)=\frac{\partial^2}{\partial x^2} I(x,y)+\frac{\partial^2}{\partial y^2} I(x,y) \tag{19}$$

The Laplacian is a linear, shift-invariant operator and its transfer function is equal to zero at the origin of a frequency space. Fig. 11 presents an example of edges and both derivatives.

At present, for the discrete image version, most methods are based on convolution with a set of directional derivative masks – filters:

$$H = I * M \tag{20}$$

$$H(m,n)=\sum_i \sum_j I(i,j)\, M(m-i,n-j) \tag{21}$$

where the exemplary masks **M** 3×3, for a discrete Laplacian $H(m,n)$ are shown in TABLE II.

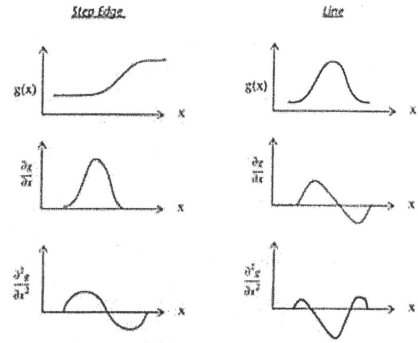


Fig. 10. The kind of edges (at the top), the first derivative of the edges (in the middle), the second derivative of the edges (at the bottom).

TABLE II
LAPLACIAN CONVOLUTION KERNELS.

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

The other well-known edge operators are suggested by: Sobel [77], Prewitt [78] and Kirsch [74]. In all of them each pixel in the image is convolved with both kernels. One kernel responds maximally to a generally vertical edge and the other to a horizontal edge. The maximum value of the two convolutions is taken as the output value for that pixel. The kernels for the Sobel edge operator are shown in TABLE III, whereas their results are presented in Fig. 11 c).

TABLE III
THE SOBEL CONVOLUTION KERNELS.

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

The kernels for the Prewitt edge operator are shown in TABLE IV.

TABLE IV
PREWITT CONVOLUTION KERNELS

-1	0	1
-1	0	1
-1	0	1

1	1	1
0	0	0
-1	-1	-1

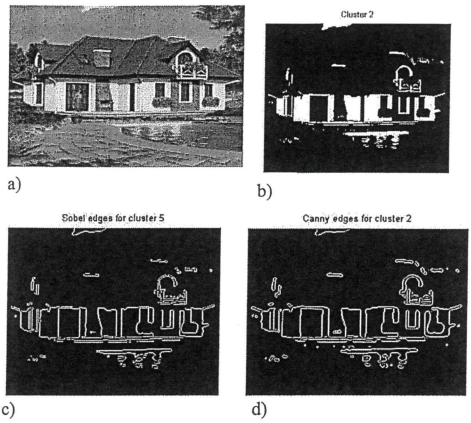


Fig. 11. An example of edge detection. a) the original image, b) a layer segmented by clustering, c) an example of the Sobel method for the layer from b), d) an example of the Canny method for the layer from b).

The Canny edge detector [79] is known to many as the optimal one for a number of reasons; the first and most obvious being its low error rate. It is important that edges presented in images should not be missing and that there are no responses to non-edges. The second reason is that the edge points are well localized, while the third that there is only one response to a single edge.

In the Canny algorithm the Gaussian filter, based on the 5×5 mask, is used to smooth the image because the larger the Gaussian mask, the lower the detector's rate of sensitivity to noise. Next, the Sobel operator is applied to estimate the gradient Gx in the x-direction and Gy in the y-direction. The magnitude, the so-called *edge strength* of the gradient is thus approximated, using the formula:

$$|G| = |Gx| + |Gy| \tag{22}$$

Then we find the edge direction

$$\theta = \arctan\left(\frac{Gy}{Gx}\right) \tag{23}$$

See the result of the Canny edge operator in Fig. 11 d).

2) Boundary tracking by Active Contours

In contrast to gradient-based representation, where the boundaries are detected based on the pixel intensity, the parametric model of active contours leads to the energy minimization problem.

Formally, let ρ be a metric (for instance the Euclidean metric) in $\mathbf{R}^2$ and $K(\mathbf{x}_0, \varepsilon) = \{\mathbf{x} \in \mathbf{R}^2 : \rho(\mathbf{x}_0, \mathbf{x}) < \varepsilon\}$ be a sphere with the centre $\mathbf{x} \in \mathbf{R}^2$ and radius $\varepsilon < 0$. A set $c \subseteq \mathbf{R}^2$ is a contour if and only if there exists a function f: $\mathbf{R}^2 \to \mathbf{R}$, such as:

$$c = \{x \in R^2 : \bigvee_{\varepsilon>0} \bigwedge_{x_1, x_2 \in K(x,\varepsilon)} f(x_1) \geq 0 \wedge f(x_2) < 0\}.$$

Active contours, or snakes, are computer-generated curves that move within images to find object boundaries [80]. A traditional snake is a parametric curve $C(p) = \begin{bmatrix} x(p) \\ y(p) \end{bmatrix}$, where $p \in [0,1]$, that moves through the spatial domain Ω of an image $I(x,y)$. A snake, which we call the gradient vector flow (GVF) snake, begins with the calculation of a field of forces, called the GVF forces, over the image domain.

$$J(C) = E_{int}(C) + E_{ext}(C) \tag{24}$$

The external energy function E_{ext} is derived from the image so it moves towards the image contour:

$$E_{ext} = \int_0^1 P(C(p))\, dp = -\nabla P(C(p)) \tag{25}$$

where $P(x,y)$ is a convolution of image $I(x,y)$ (seen as a line) with a 2D Gaussian function $G_\sigma(x,y)$ with a standard deviation σ, as follows:

$$P(x,y) = -|\nabla G_\sigma(x,y) * I(x,y)|^2 \quad (26)$$

The internal energy E_{int} controls the snake like a physical object resistant to both stretching and bending, towards the image boundaries:

$$E_{int} = \frac{1}{2}\int_0^1 \alpha |C'(p)|^2 + \beta |C''(p)|^2 \, dp \quad (27)$$

where the first derivative $C'(p)$ models stretching and elasticity, whereas the second derivative $C''(p)$ models bending and rigidity, where α and β are weight parameters.

A snake that minimizes $J(C)$ must satisfy the Euler equation:

$$\alpha C''(p) - \beta C''''(p) - \nabla P(C(p)) = 0 \quad (28)$$

that can be viewed as a force balance equation

$$F_{int} + F_{ext}^{(p)} = 0 \quad (29)$$

where: $F_{int} = \alpha C''(p) - \beta C''''(p)$

and $F_{ext}^{(p)} = -\nabla E_{ext}$.

The GVF forces create the gradient of an image edge map (see Fig. 12).

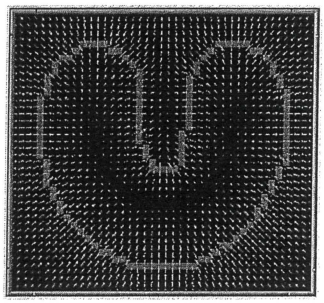


Fig. 12. A gradient vector flow (GVF) field for a U-shaped object. These vectors will pull an active contour towards the object boundary. (follows: *Active Contours, Deformable Models, and Gradient Vector Flow* Chenyang Xu and Jerry L. Prince web page: http://www.iacl.ece.jhu.edu/static/gvf/)

In comparison with the classical edge detection techniques, snakes have multiple advantages:

- They produce closed and smooth object boundaries.
- They autonomously and adaptively search for the minimum state.
- External image forces act upon the snake in an intuitive manner.
- Incorporating Gaussian smoothing in the image energy function introduces scale sensitivity.

But they also have some key drawbacks:

- They are sensitive to local minima states,
- Minute features are often ignored during energy minimization over the entire contour.
- Their accuracy depends on the convergence policy

3) Hough Transform

The classical Hough transform was concerned with the identification of lines in the image, but later the Hough transform was extended to identify the positions of arbitrary shapes, most commonly circles or ellipses.

The simplest variant of the Hough transform is used to detect of straight lines. In general, the straight line $y = mx + b$ can be represented as a point (b,m) in the parameter space. However, vertical lines pose a problem. Instead, Duda and Hart [81] propose the polar coordinate representation of a line

$$\rho = x\cos\theta + y\sin\theta, \quad (30)$$

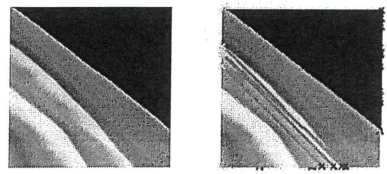


Fig. 13. Left: The original image, Right: Lines found by the Hough transform.

where ρ is the distance from the origin to the closest point on the straight line, and θ is the angle

between the x axis and the line connecting the origin with that closest point.

Each point in image generates the sinusoid in Hough space, and each point along this sinusoid corresponds to the ρ-θ values for a single line passing through the original point.

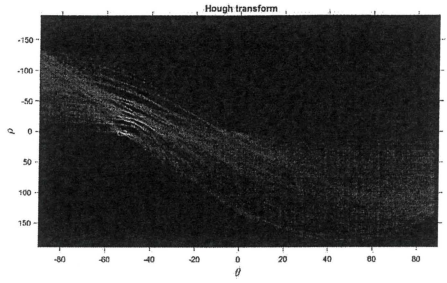


Fig. 14. The Hough transform space. White sinusoids represents lines visible in Fig. 13.

The Hough space is an accumulator space which means that it sums up the votes of many pixels in the image, and points in Hough space that have a large total vote are then interpreted as indicating the corresponding alignment on the real space image. To construct the Hough transform, every point present in the real-space image casts its votes into the Hough space for each of the lines that can possibly pass through it.

D. Shape Information

Shape extraction is a non-trivial operation, but shape-based methods are particularly challenging due to the intrinsic difficulties in dealing with shape location and recognition. Nevertheless, there is no doubt that shape is one of the basic features describing image content.

Hence, we define the key properties of shape feature:

- identifiability: shapes which are found perceptually similar by humans have the same features that are different from the others;
- translation, rotation and scale invariance;
- the location, the rotation and the scaling changes of the shape must not affect the extracted features;
- affine invariance: the affine transform performs a linear mapping from one to another coordinates system that preserves the "straightness" and parallelism" of lines. Affine transform can be constructed using sequences of translations, scales, flips, rotations and shears. The extracted features must be as invariant as possible with affine transforms noise resistance: features must be as robust as possible against noise, i.e., they must be the same irrespective of the strength of the noise in a given range that affects the pattern.
- occultation invariance: when some parts of a shape are occulted by other objects, the feature of the remaining part must not change compared to the original shape.
- statistically independent: two features must be statistically independent. This represents compactness of the representation.
- reliability: as long as one deals with the same pattern, the extracted features must remain the same

Shape description can be generally divided into two kinds of methods: contour-based and region-based. Under each kind, the methods are further divided into a structural and global approach based on whether the shape is represented as a whole or by segments (primitives). The whole breakdown is shown in Fig. 15.

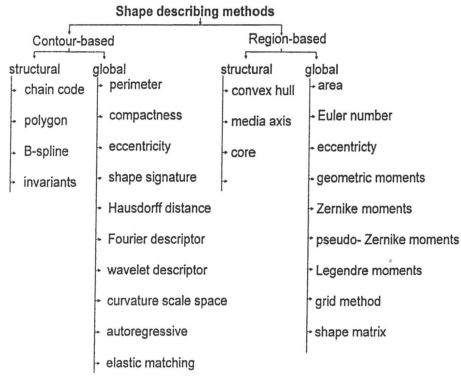


Fig. 15. Shape describing methods **[82]**.

Contour shape techniques only exploit shape boundary information. Zhu et al. [83] use salient contours, extracted from bottom-up contour grouping, as tokens for image-model shape matching. Shape matching with contours instead of isolated edges has several advantages. Long salient contours are more distinctive, which leads to efficiency of the search as well as the accuracy of shape matching. Furthermore, by requiring the entire contour to match objects as a whole, we remove accidental alignment causing false positive detections.

Contour shape techniques only exploit shape boundary information. Zhu et al. [83] use salient contours, extracted from bottom-up contour grouping, as tokens for image-model shape matching. Shape matching with contours instead of isolated edges has several advantages. Long salient contours are more distinctive, which leads to efficiency of the search as well as the accuracy of shape matching. Furthermore, by requiring the entire contour to match objects as a whole, we remove accidental alignment causing false positive detections.

Boundary-based methods such as [84] represent shapes by the locations of the maxima of its curvature scale space (CSS) image. Shapes are smoothed by selecting the appropriate scale and then matched by shifting the CSS contours so that the major maxima of one image overlaps that of the other. The shape boundaries are approximated using polygonal curves and are progressively simplified through discrete curve evolution based on a novel relevance measure. The weakness of the boundary-based approach is that it does not represent the interior of the shape [85] and is, therefore, very sensitive to spatial reconfigurations of parts and local boundary perturbations.

In region-based techniques, all the pixels within a shape region are taken into account to obtain the shape representation. Common region based methods use moment descriptors to describe shapes [86]. Other region based methods include grid method, shape matrix, convex hull and media axis.

The most efficient as shape descriptors are moments of inertia:

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q I(x,y), \quad p,q = 0,1,2 \qquad (31)$$

and Zernike moments [87]. Zernike moments are a set of complex polynomials $\{V_{pq}(x,y)\}$ which form a complete orthogonal set over the unit disk of $x^2 + y^2 \le 1$. Hence, the definition of 2D Zernike moments with p^{th} order with repetition q for intensity image $I(x,y)$ of the image is described as:

$$Z_{pq} = \frac{p+1}{\pi} \iint_{x^2+y^2 \le 1} V^*_{pq}(x,y) I(x,y)\, dxdy \qquad (32)$$

where:

$$V^*_{pq}(x,y) = V_{p,-q}(x,y) . \qquad (33)$$

Generally, the first 10 Zernike moments i.e. those from Z_{00} to Z_{33}, are sufficient as a shape feature. The scale invariance is obtained by normalizing Z_{00} by the total number of image pixels.

Characteristic features of Zernike moments are: (i) invariance to rotation only; (ii) the translation invariance is achieved by the location of the original image centroid in the centre of the coordinates.

Although the Zernike moment descriptor has a robust performance, it has several shortcomings. First, the kernel of Zernike moments is complex to compute, and the shape has

to be normalized into a unit disk before deriving the moment features. Second, the radial features and circular features captured by Zernike moments are not consistent, one is in the spatial domain and the other is in the spectral domain. It does not allow multi-resolution analysis of a shape in radial direction. Third, the circular spectral features are not captured evenly at each order, this can result in a loss of signilcant features which are useful for shape description. To overcome these shortcomings, a generic Fourier descriptor (GFD) has been proposed by Zhang and Lu [82]. The GFD is acquired by applying a 2D Fourier transform on a polar-raster *PF:*

$$PF = = \sum_r \sum_i I(r,\theta_i) \exp\left[j2\pi\left(\frac{r}{R}\rho + \frac{2\pi i}{T}\varphi\right)\right] \quad (34)$$

where: $0 \le r < R$ and $\theta_i = i\,(2\pi/T)$ $(0 \le i < T)$; $0 \le \rho < R$, $0 \le \varphi < T$. R and T are the radial frequency resolution and angular frequency resolution, respectively. The normalized coefficients are the GFD. The similarity between two shapes is measured by the city block distance between their GFDs.

It has been found that methods operating within the spatial domain suffer from two main drawbacks: noise sensitivity and a high dimension of the feature vector. The problems can be solved in four ways: histogram, moments, scale space and spectral transforms.

E. Local Feature Descriptors

1) Scale-Invariant Feature Transform (SIFT)

The scale-invariant feature transform (SIFT) was introduced by Lowe [14], [13] to identify objects in two images, even if these objects were cluttered or under partial occlusion, because the SIFT feature descriptor is invariant to uniform scaling and orientation. In like manner, it is partially invariant to affine distortion and illumination changes.

The algorithm starts from key-points detection in order to identify locations and scales that can be repeatably assigned under differing views of the same object. Key-point locations are defined as maxima and minima of the difference of the Gaussians $G(x,y,\sigma)$ applied in a scale-space to a series of smoothed and resampled images.

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \quad (35)$$

where $L(x,y,\sigma)$ is the product of a convolution.

There is a separation of a multiplicative factor k for two nearby scales:

$$D(x,y,\sigma) = L(x,y,k\sigma) - L(x,y,\sigma) \quad (36)$$

The scale-space extrema detection produces too many key-point candidates, so at first, for each candidate key-point, interpolation of nearby data is used to accurately determine its position. The interpolation is done using the quadratic Taylor expansion of the difference-of-Gaussian scale-space function, D (cf. (36)) with the candidate key-point as the origin. This Taylor expansion is given by:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \quad (37)$$

for $\mathbf{x} = (x,y,\sigma)^T$.

Low contrast candidate points and edge response points along an edge are discarded. Dominant orientations are assigned to localized key-points. These steps ensure that the key-points are more stable for matching and recognition. The SIFT descriptors robust to local affine distortion are then obtained by considering pixels around a radius of the key location, blurring and resampling of local image orientation planes.

The next step is the orientation assignment when each key-point is assigned one or more orientations based on local image gradient directions. First, the Gaussian-smoothed image $L(x,y,\sigma)$ at the key-point's scale σ is taken so that all computations are performed in a scale-invariant manner. For an image sample $L(x,y)$ at scale σ, the gradient magnitude $m(x,y)$ and orientation $\theta(x,y)$ are precomputed using pixel differences:

$$m(x,y) = \sqrt{\left(L(x+1,y) - L(x-1,y)\right)^2 + \left(L(x,y+1) - L(x,y-1)\right)^2}$$

$$\theta(x,y) = \arctan\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right) \quad (38)$$

assuming that $\forall(x,y)$ in the neighbourhood (x_0,y_0).

The SIFT key samples generated at a larger scale are given twice the weight of those at a smaller scale. This means that the larger scale is in effect able to filter the most likely neighbours at the smaller scale. This also improves recognition performance by giving more weight to the least-noisy scale. To avoid the problem of boundary

effects in bin assignment, each key-point match votes for the 2 closest bins in each dimension, giving a total of 16 entries for each hypothesis and further broadening the pose range.

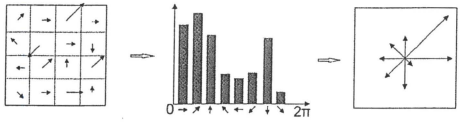


Fig. 16. The gradient magnitude and orientation at each point of a 4x4 set of samples (on the left) which are accumulated into orientation histograms with 8 bins each (in the middle). The key-points descriptor summarizes the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. Peaks in the orientation histogram correspond to dominant directions of local gradients.

Hough transform (cf. (30)) is used to cluster reliable model hypotheses to search for keys that agree upon a particular model pose. Hough transform identifies clusters of features with a consistent interpretation by using each feature to vote for all object poses that are consistent with the feature. When clusters of features are found to vote for the same pose of an object, the probability of the interpretation being correct is much higher than for any single feature. An entry in a hash table is created predicting the model location, orientation, and scale from the match hypothesis. The hash table is searched to identify all clusters of at least 3 entries in a bin, and the bins are sorted into decreasing order of size.

Each identified cluster is then subject to a verification procedure in which a linear least squares solution is performed for the parameters of the affine transformation relating the model to the image. The affine transformation of a model point $[x\ y]^T$ to an image point $[u\ v]^T$ can be written as below:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \tag{39}$$

where the model translation is $[t_x\ t_y]^T$ and the affine rotation, scale, and stretch are represented by the parameters m_1, m_2, m_3 and m_4. To find the transformation parameters the equation above can be rewritten to gather the unknowns into a column vector.

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \cdots & & & \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ . \\ . \end{bmatrix} \tag{40}$$

This equation shows a single match, but any number of further matches can be added, with each match contributing two more rows to the first and last matrix. At least 3 matches are needed to provide a solution. We can write this linear system as

$$A\mathbf{x} \approx \mathbf{b} \tag{41}$$

where A is a known n-by-p matrix (usually with $n > p$), $\mathbf{x}$ is an unknown p-dimensional parameter vector, and $\mathbf{b}$ is a known n-dimensional measurement vector.

Therefore, the minimizing vector $\mathbf{x}$ is a solution of the normal equation.

$$A^T A\mathbf{x} = A^T\mathbf{b} \tag{42}$$

hence, we obtain:

$$\mathbf{x} = (A^T A)^{-1} A^T \mathbf{b} \tag{43}$$

which minimizes the sum of the squares of the distances from the projected model locations to the corresponding image locations.

2) RootSIFT

SIFT was originally, by Lowe [13], designed to be used with the Euclidean distance, but since there is a histogram comparison, Arandjelović and Zisserman [88] introduced alternative histogram distance measures, namely the Hellinger kernel.

The Hellinger kernel for two L_1 normalized histograms, x and y (i.e. $\sum_i^n x_i = 1$ and $x_i \geq 0$), is defined as follows:

$$H(x,y) = \sum_{i=1}^{n} \sqrt{x_i y_i} \quad (44)$$

where n is a number of vector with unit Euclidean norm such as: $\|\mathbf{x}\|_2 = 1$.

The RootSIFT application increases the average precision of retrieval.

3) Rotation-Invariant Generalization of SIFT (RIFT)

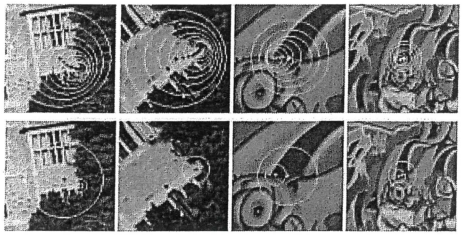


Fig. 17. Scale invariant interest point detection: (Top) Initial multi-scale Harris points (selected manually) corresponding to one local structure **[89]**.

RIFT is a rotation-invariant variant of SIFT dedicated to texture images where the notion of orientation is difficult to define. The RIFT descriptor is constructed using circular normalized patches divided into concentric rings of equal width and within each ring a gradient orientation histogram is computed [89], [90]. To maintain rotation invariance, the orientation is measured at each point relative to the direction pointing outward from the center.

When the size of the LoG kernel matches with the size of a blob-like structure, the response attains an extremum.

$$|\mathrm{LoG}(\mathbf{x}, \sigma_n)| = \sigma_n^2 (L_{xx}(\mathbf{x}, \sigma_n) + L_{yy}(\mathbf{x}, \sigma_n)| \quad (45)$$

The LoG kernel can therefore be interpreted as a matching filter.

4) Fisher Vector (FV)

The Fisher kernel has been proposed, in the context of measuring the amount of information that an observable random variable X carries about an unknown parameter θ of a distribution that models X. Formally, it is the variance of the score, or the expected value of the observed information. From the theory of information we know that the entropy of a random value V is:

$$H(V) = -\sum p(v) \log(p(v)) \quad (46)$$

Further, this approach has been moved to an image classification, where the Fisher kernel method is to derive a function that measures the similarity between two sets of data X and Y, such as the sets of local descriptors extracted from two images. The idea is to characterize a signal with a gradient vector derived from a probability density function (pdf) which models the generation process of the signal [91].

This representation can then be used as input to a discriminative classier. For the problem of image categorization the input signals are images. Perronnin and Dance proposed to use, as a generative model the Gaussian Mixture Models (GMM) which approximates the distribution of low-level features in images, i.e. a visual vocabulary.

For sample $X = \{x_t,\ t = 1,\ldots,T\}$ of observations, which is a set of T local descriptors extracted from an image and $p(X|\lambda)$ represents pdf with λ parameters, the gradient of the log-likelihood describes the contribution of the parameters to the generation process [92]:

$$G_\lambda^X = \frac{1}{T} \nabla_\lambda \log p(X|\lambda) \quad (47)$$

The dimensionality of this vector depends only on the number of parameters λ, not on the number of patches T.

In the context of image retrieval the FV are usually ℓ^2-normalized since, as proved it Perronnin et al. [92] this is a way to eliminate the fact that distinct images contain different amounts of image-specific information. The Fisher Vector is applied to non-binary local features, using the Gaussian Mixture Model to represent the average distribution $p(X|\lambda)$.

5) Vectors of Locally Aggregated Descriptors (VLAD)

The VLAD method analyzes the local descriptors contained in an image to create statistical summaries that still preserve the effectiveness of local descriptors and allow treating them as global descriptors [93]. This image descriptor was designed to be very low dimensional (e.g. 16 bytes per image).

This method encodes a set of local feature descriptors $F = (x_1, \ldots, x_k)$, extracted from an image treated as a codebook with k visual words, using a dictionary based on a clustering method, such as GMM or k-means clustering. Each local descriptor x_j is then associated with its nearest centroid $NN(x_j) = \mu_i$.

$$v_{i,j} = \sum_{x \text{ such that } NN(x)=\mu_i} (x_j - \mu_{i,j}) \tag{48}$$

where i=1,…, k is the index of visual word (k – number of centroids) and j=1,…,d is the local descriptor component. Hence, the whole image representation dimension is $D = k \times d$.

For each cluster, the residual vectors (i.e. the difference between the centroid and the associated descriptors) are accumulated and the sum of the residual is concatenated into a single vector $V = [v_1^T \ldots v_k^T]$. Next, vector v is normalised by $v{:=}v/\|v\|_2$ and the Euclidean distance is sufficient to compare two VLADs.

6) Features from accelerated segment test (FAST)

FAST is a corner detection method, introduced by Rosten and Drummond in 2006 [94], which could be used to extract feature points and later used to track and map objects [95]. A FAST corner detector uses a circle of 16 pixels (a Bresenham circle of r =3) to classify whether a candidate point p is actually a corner. Each pixel in the circle is labeled from integer number 1 to 16 clockwise. For a set of N contiguous pixels, if the pixels in the circle are all brighter than the intensity of candidate pixel p (denoted by I_p), plus a threshold value t, or all darker than the intensity of candidate pixel p, minus threshold value t, then p is classified as a corner. The conditions can be written as:

i. A set of N contiguous pixels S, $\forall\, x \in S$, the intensity of x denoted by (I_x) can be $I_x > I_p + t$;
ii. A set of N contiguous pixels S, $\forall\, x \in S$, $I_x < I_p - t$;

So when either of the two conditions is met, candidate p can be classified as a corner. There is a tradeoff between selecting N, the number of contiguous pixels and the threshold value t. Then, N is usually selected as 12. A high-speed test method could be applied to exclude non-corner points.

Generally, the FAST detector is employed to find objects in video frames because of its effectiveness.

7) Oriented FAST and Rotated BRIEF (ORB)

ORB is basically a fusion of the FAST key-point detector and a BRIEF descriptor with many modifications to enhance the performance introduced by Rublee et al. [96] in 2011. First, it uses FAST to find key-points, then apply the Harris [97] corner measure to find top N points among them. It also uses a pyramid to produce multiscale-features.

In order to compute orientation, they found moments of order p and q, such as:

$$m_{pq} = \sum_x \sum_y x^p y^q I(x,y), \quad p,q = 0,1,2 \tag{49}$$

and intensity centroids: $C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right)$.

Then, a vector $\overrightarrow{OC}$ from the corner's center, O, to the centroid, can be constructed. The orientation of the patch then simply is:

$$\theta = \text{atan2}\,(m_{01}, m_{10}), \tag{50}$$

where atan2 is the arctangent function with two arguments. In order to improve the rotation invariance of this measure authors made sure that moments are computed with x and y remaining within a circular region of radius r. They

empirically selected r to be the patch size, so that that x and y run from $[-r, r]$.

Next, the Binary Robust Independent Elementary Features (BRIEF) descriptor is used for a simple binary test τ between pixels in a smoothed image patch **p**, as follows:

$$\tau(\mathrm{p}; \mathrm{x}, \mathrm{y}) := \begin{cases} 1: & \mathrm{p}(\mathrm{x}) < \mathrm{p}(\mathrm{y}) \\ 0: & \mathrm{p}(\mathrm{x}) \geq \mathrm{p}(\mathrm{y}) \end{cases} \tag{51}$$

where **p**(x) is the intensity of **p** at a point x. The feature is defined as a vector of n binary tests:

$$f_n(\mathrm{p}) := \sum_{1 \leq i \leq n} 2^{i-1} \tau(\mathrm{p}; \mathrm{x}_i, \mathrm{y}_i) \tag{52}$$

The test pairs of **x** and **y** are selected by Gaussian distribution around the centre of the patch or PCA for good discrimination.

As tests for typical frames of size 640x480 proved, the ORB descriptor gives significant time decreasing.

IV. Object Segmentation

Based on the above-described features the objects from an image can be extracted. There are many different methods of image segmentation. One approach extracts a central object from a mono-chromatic background, for example based on morphological operations by Chen et al. [98], based on the curvature shape by Abbassi et al. [84], or by Sze [99]. The second approach segments multi-object images which is most challenging.

A. Based on colour

Let us begin from the colour segmentation. Some general algorithms such as k-means or colour histogram do not offer the optimal solution. Hence, we present here a very fast algorithm based on the principal colour from the (R,G,B) triple.

With the aim of labelling a pixel, the biggest value from the triple (R,G,B) is selected and a cluster colour is defined. In this way, three segments – red, green and blue are obtained. Additionally, points with equal values of RGB are labelled as grey. For better results, each colour is divided into three shades, according to their brightness of colour. These shades are shown as three regions (I, II, III) which determine point brightness. The idea of the segmentation is illustrated in Fig. 18. The radius r of the dividing sphere was counted in the Euclidean measure, namely

$$r = \frac{\sqrt{R_{\max}^2 + G_{\max}^2 + B_{\max}^2}}{3}. \tag{53}$$

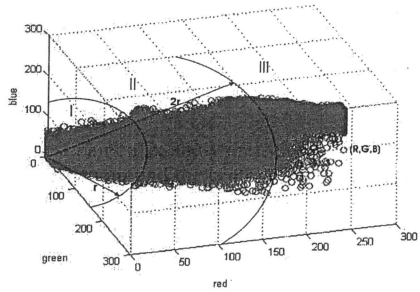


Fig. 18. The way of labelling the set of pixels. Regions I, II, III show pixel brightness and the biggest value of triple (R,G,B) determines the segment colour.

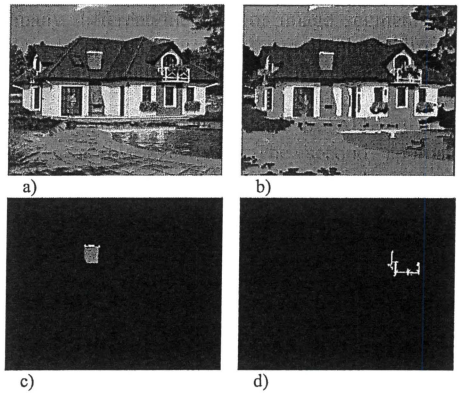


Fig. 19. Colour segmentation results: a) the original RGB image, b) segmented objects presented in their average colours, c), and d) extracted objects.

Generally, $R_{\max} = G_{\max} = B_{\max} \leq 255$ because full saturation of colours is rare. Moreover, we added

three segments: black, grey and white for pixels where R = G = B or was not exactly equal according to their region (I, II, III).

We assumed that 'not exactly equal' meant that $|R-G| < \sigma$ and $|R-B| < \sigma$, where $10 < \sigma < 15$. Having done this, we obtained images segmented into 12 clusters. For visualization, the objects are presented in their average colours (see Fig. 19 b)). The examples of separated objects are seen in Fig. 19 c) - roof and Fig. 19 d) – balcony railing.

B. Based on texture

In the case of textured objects the LBP operator, introduced by Ojala et al. in 2002 [49], can be applied to object segmentation (see subsec. III. B). Fig. 20 presents a texture mosaic composed of five textures from outdoor scenes, such as those frequently encountered in satellite images.

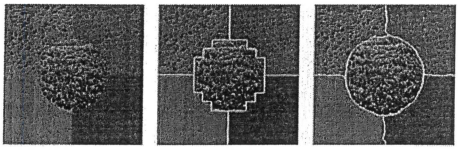


Fig. 20. Texture mosaic segmentation based on LBP **[49]**.

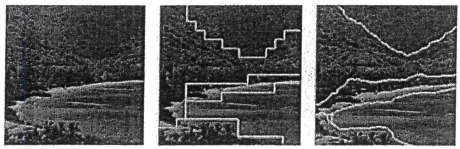


Fig. 21. Natural scene segmentation based on the texture according to the LBP **[49]**.

Acharyya and Kundu [100] applied an orthogonal and linear phase *M*-band wavelet transform to decompose an image into *M*x*M* channels. Various combinations of these bandpass sections were taken to obtain different scales and orientations in the frequency plane. Texture features are obtained by subjecting each bandpass section to a nonlinear transformation and computing the measure of energy in a window around each pixel of the filtered texture images. Unsupervised texture segmentation was obtained by a simple *K*-means clustering.

Another attempt at automatic texture segmentation, i.e. without any a priori knowledge of either the type of textures or the number of textures in the image was taken by Faizal et al. [73]. As it has been mentioned in subsec. III. B the method used a modified discrete wavelet frame (DWF) decomposition to extract important features from an image before a mean shift algorithm is used together with a fuzzy C-means (FCM) clustering to cluster or segment the image into different texture regions. The proposed algorithm has the advantage of high accuracy while low computational costs.

C. Based on shape

The idea behind the Curvature Scale Space (CSS) representation [84] is that the contour can be represented by set of points where the contour curvature changes, as well as curvature fragments between these points. For each point in the contour, it is possible to compute the curvature of the contour at that point, based on the neighboring points; a point whose two closest neighbours have different curvature values is considered a curvature change. In fact, not all curvature changes are needed to compute the CSS representation, but only those where the curvature goes from a positive to a negative value or vice-versa. When it happens, the curvature values have to go necessarily through zero and therefore these changes are called zero-crossings of the curvature, as illustrated in Fig. 22. As for the average curvature between two of these zero-crossings, it basically corresponds to the angle difference between the tangents to the contour at these two points divided by the arc length joining these two points.

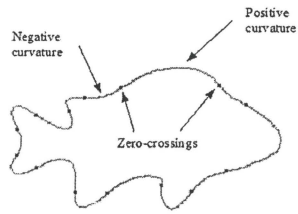


Fig. 22. Zero-crossings of the curvature.

A different approach was represented by Latecki and Lakämper [101] who described shape as line segments for silhouette presentation. To reduce the influence of digitization noise, as well as segmentation errors, the shapes are simplified to a set of segments and their length function normalized with respect to the total length of the whole contour.

Goh and Chan presented a part-based shape descriptor that incorporates both the description of the general shape form of each subpart, as well as the local boundary perturbation (boundary texture) [102]. A shape is decomposed into subparts along segmented sections of the extracted shape axes and each part is described by two 1-D histograms derived from the local gradient vector field. The shape part descriptor, associated with each subpart of an object, is a saliency measure which weighs its visual significance based on the proportion of the overall shape region to the subpart.

The shape description is used in most applications, for examples, archeological ones. In this case, decoration patterns are described by Xu and Liu [103] as a closed contour set. The contour of a 2D object is a simple closed curve and the area enclosed by the curve is topologically homeomorphic to a disk. It has also been assumed that the centroid of the curve has been moved to the origin of the 2D coordinate system. A contour L is described by the function $z(t) = (x(t), y(t))$, $0 \le t < 1$ which is the arc-length parameterization.

D. Based on local features

Mutch and Lowe [104] modified the model of Serre, Wolf, and Poggio [105] by applying Gabor filters at all positions and scales; then feature complexity and position/scale invariance were built up by alternating template matching and max pooling operations. Images were reduced to feature vectors, which were then classified by an SVM. Features are computed hierarchically in five layers: an initial image layer and four subsequent layers, each built from the previous by alternating template matching and max pooling operations (see Fig. 23).

Image layer - the image is converted to grayscale and the shorter edge is scaled down to 140 pixels while maintaining the aspect ratio. Next an image pyramid of 10 scales is created, each a factor of $2^{1/4}$ smaller than the last (using bicubic interpolation).

Gabor filter (S1) layer - is computed from the image layer by centering 2D Gabor filters with 4 orientations at each possible position and scale (compare Fig. 4).

Local invariance (C1) layer - pools nearby S1 units (of the same orientation) to create position and scale invariance over larger local regions, and as a result can also subsample S1 to reduce the number of 10x10 units across in position and 2 units deep in scale.

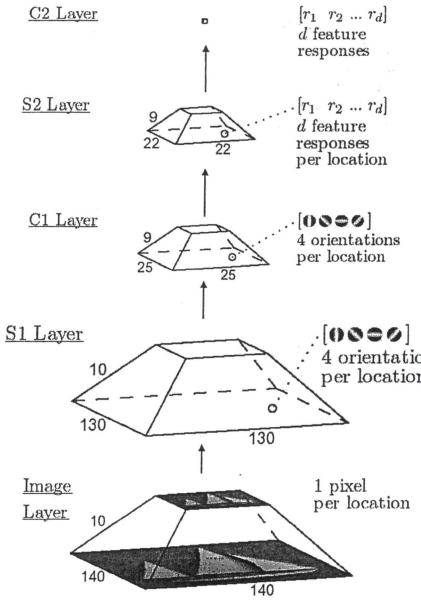


Fig. 23. Feature computation in the base model. Each layer has units covering three spatial dimensions (*x*/*y*/scale), and at each 3D location, an additional dimension of feature type. The image layer has only one type of pixels, layers S1 and C1 have 4 types, and the upper layers have *d* (many) types per location. Each layer is computed from the previous one by applying template matching or max pooling filters [104].

Intermediate feature (S2) layer. At every position and scale in the C1 layer, authors performed template matches between the patch of C1 units cen-

tred at that position/scale and each of d prototype patches. These prototype patches represent the intermediate-level features of the model.

Global invariance (C2) layer. Finally a *d*-dimensional vector was created, each element of which is the maximum response (anywhere in the image) to one of the model's *d* prototype patches. At this point, all position and scale information has been removed, i.e. we have a "bag of features".

V. Object Recognition

The goal of object recognition is to detect objects in images using different models and identify these selected objects by classifying them. Additionally, of interest is how these objects are mutually located to each other in the image.

As we have mentioned above, we are interested in image segmentation and object recognition as the first stage of a CBIR construction. At present, there are tendencies to use different methods to separate foreground objects from the monolithic background, beginning from separate colour and texture regions as it was presented by Li and Shapiro in [106], through using wavelets [107] or morphological operations [98], whereas there is a need to recognize the foreground objects, sometimes overlapped by the others, which are found against complicated, multi-object background as we can see in Fig. 19. Surely, such images are more challenging and the recognition process forces us to use different methods to obtain the proper classification.

A. Object Similarity/Dissimilarity

The simplest approach to object similarity/dissimilarity is the comparison of feature vectors of two objects. In the context of object recognition, we are more interested in object classification than in plain object comparison. Then, the simplest solution is the comparison of an object feature vector $\mathbf{x}$ to the previously prepared patterns P_k for each class used, for instance, Euclidean (54) or Minkowski's (56) metrics. The designed classes should attribute objects in accordance with human perception into *M* semantic classes.

Many measures exist for quantitative variables, mostly constructed in an additive way after counting the differences for each variable separately. The basic metrics are presented in Table V:

Table V
Dissimilarity Metrics for Quantitative Data in $\Re^M$.

Metric name	Dissimilarity $d(\mathbf{x},\mathbf{y})$	No.
Euclidean	$\sqrt{(\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})}$	(54)
Weighted Euclidean	$\sqrt{(\mathbf{x}-\mathbf{y})^T \mathrm{diag}(w_i^2)(\mathbf{x}-\mathbf{y})}$	(55)
Minkowski's	$\sqrt[p]{\sum_{i=1}^{m} \lvert x_i - y_i \rvert^p}, \; p \geq 1, \; p \neq 2$	(56)
Mahalanobis	$\sqrt{(\mathbf{x}-\mathbf{y})^T C^{-1}(\mathbf{x}-\mathbf{y})}$ C is covariance matrix	(57)
City block	$\sum_{i=1}^{m} \lvert x_i - y_i \rvert$	(58)
Max norm	$\max_i \lvert x_i - y_i \rvert$	(59)

For texture features, when the feature vector represents relative frequency distribution (e.g., a normalised grey level co-occurrence histogram), the dissimilarity can also be measured by the relative entropy, or Kullback-Leibler (K-L) divergence. Let $D(\mathbf{g},\mathbf{q})$ denote the divergence between two distributions, $\mathbf{f_g} = (f_{g,t}: t = 1, ..., T)$ and $\mathbf{f_q} = (f_{q,t}: t = 1, ..., T)$. Then:

$$D(\mathbf{g},\mathbf{q}) = \sum_{t=1}^{T} f_{g,t} \log \frac{f_{\mathbf{g},t}}{f_{\mathbf{q},t}} \tag{60}$$

This dissimilarity measure is asymmetric and does not represent a distance because the triangle inequality is not satisfied. The symmetric distance is obtained by averaging $D(\mathbf{g},\mathbf{q})$ and $D(\mathbf{q},\mathbf{g})$ [108].

B. Object Classification

The semantic approach to images, or even object recognition, requires image/object classification. The methods finding a similarity/dissimilarity among images or objects are insufficient because an assignment to a particular class is necessary.

Generally, the classical classification algorithms have been adapted to image recognition.

1) Decision Trees

In the construction of decision trees [109] a measure of discrimination is used in order to rank attributes and select the best one. The construction of a decision tree is equivalent to a restriction of the whole set of attributes which describes the data to a set of pertinent attributes. Each vertex of a binary tree is associated with an attribute [110].

Inductive learning regarding a given domain is based on a set of examples. Each example is a case already solved or completely known. It is associated with a pair [description, class] where the description is a set of pairs [attribute, value] which is the available knowledge. The class of the example is the decision (or category, or solution...) associated with the given description. Such a set of examples is called a training set. Samples considered as examples can be taken from a database with their attributes and classes as descriptors of each case. The aim of inductive learning is to find general rules enabling us to classify a case only known by means of the attributes, or to answer a query regarding the class.

The decision tree construction methods are based on the hypothesis that the value for the class is equally distributed. Thus, we have to balance the number of objects of each class by randomly selecting a subset of the whole development dataset because the process of tree construction is very sensitive to a lack of representation of certain important attributes of the minority class or imbalanced classes.

There exist many works on the construction and utilization of many kinds of decision trees (DT) [111], [112]: symbolic (DT), binary (DT) [113], fuzzy (DT) [114], etc.

2) Naïve Bayes classifier (NB)

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all NB classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. An NB classifier considers each of these features to contribute independently to the probability, regardless of any possible correlations between them.

An advantage of the NB is that it only requires a small amount of training data to estimate the parameters necessary for classification. For some types of probability models, NB classifiers can be trained very efficiently in a supervised learning setting [115].

In many practical applications, including image processing, parameter estimation for the NB models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

Generally, the NB is a conditional probability model: given a problem instance to be classified, represented by a vector $\mathbf{x} = (x_1, \ldots, x_n)$ representing some n features (independent variables), it assigns to this instance probabilities:

$$p(C_m \mid x_1, \ldots, x_n), \tag{61}$$

for each of M possible classes. Using Bayes' theorem, the conditional probability can be decomposed as:

$$p(C_m \mid \mathbf{x}) = \frac{p(C_m)p(\mathbf{x}|C_m)}{p(\mathbf{x})} \tag{62}$$

In practice, there is interest only in the numerator of that fraction, because the denominator does not depend on C and the values of the features are given, so that the denominator is effectively constant. From the definition of conditional probability we know that:

$$p(C_m \mid x_1, \ldots, x_n) = p(C_m)p(x_1, \ldots, x_n \mid C_m) \tag{63}$$

Assuming conditional independence of each feature

$$p(C_m \mid x_1, \ldots, x_n) \propto p(C_m)\, p(x_1 \mid C_m)\, p(x_2 \mid C_m)\, p(x_3 \mid C_m) \ldots \tag{64}$$

$$\propto p(C_m)\prod_{i=1}^{n} p(x_i|\, C_m)$$

Based on this assumption, a classification can be constructed where the function that assigns a class label $\hat{y} = C_k$ for some m looks as follows:

$$\hat{y} = \underset{m\in\{1,\ldots,M\}}{\arg\max}\, p\,(C_m)\prod_{i=1}^{n} p(x_i|\, C_m) \qquad (65)$$

Despite the fact that the far-reaching independence assumptions are often inaccurate, the naive Bayes classifier has several properties that make it surprisingly useful in practice [116]. In particular, the decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one-dimensional distribution. This helps alleviate problems stemming from the *curse of dimensionality*, such as the need for data sets that scale exponentially with the number of features.

3) Support Vector Machine (SVM)

Support Vector Machine (SVM) is a non-probabilistic binary linear classifier introduced by Cortes and Vapnik [117] in 1995. An SVM model is a representation of the samples as points in a space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible [118]. New examples are then mapped into the same space and predicted to belong to a category based on whichever side of the gap they fall on.

The SVM constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

For easy visualization, the case of a 2D input space can be considered. Data are linearly separable and there are many different hyperplanes that can perform separately (Fig. 24). Actually, for $\mathbf{x} \in \mathbf{R}^2$, the separation is performed by 'planes' $w_1 x_1 + w_2 x_2 + b = 0$, which is the decision boundary.

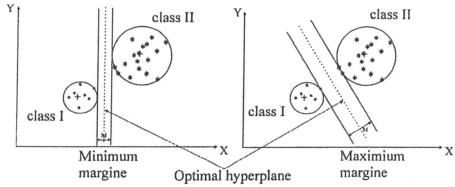


Fig. 24. Optimal hyperplane and margins for an SVM trained with samples from two classes. The samples on the margin are called the support vectors.

There are many functions that can find the optimal separating function without knowing the underlying probability distribution. In the case of a classification of linearly separable data, the idea is the following: among all the hyperplanes that minimize the training error (i.e, empirical risk) find the one with the largest margin.

By using given training examples, during the learning stage, the SVM finds parameters $\mathbf{w} = [w_1\ w_2\ \ldots\ w_n]^T$ and b of a discriminant or decision function $d\,(\mathbf{x},\mathbf{w}, b)$:

$$d(\mathbf{x},\mathbf{w},b) = \mathbf{w}^T\mathbf{x} + b = \sum_{i=1}^{n} w_i x_i + b\,, \qquad (66)$$

where: $\mathbf{x}, \mathbf{w} \in \mathbf{R}^n$, and the scalar b is called a bias. The dashed separation lines in Fig. 24 represent the line that follows from $d(\mathbf{x},\mathbf{w}, b) = 0$.

We can notice that the hyperplane is in the canonical form with respect to training data $\mathbf{x} \in \mathbf{X}$. If

$$\min_{x_j\in X} |\mathbf{w}^T\mathbf{x} + b| = 1 \qquad (67)$$

and if the canonical hyperplane has a maximum margin M then this hyperplane is located in the middle of M. From the geometric properties the margin can be described as $M = 2/\|\mathbf{w}\|$ where: $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T\mathbf{w}} = \sqrt{\sum_i w_i^2}$. If $\|\mathbf{w}\|$ is minimal, M is maximum.

SVMs belong to a family of generalized linear classifiers. Their special property is that they simultaneously minimize the empirical classification error and maximize the geometric margin; hence they are also known as maximum margin classifiers.

4) Fuzzy rule-based classifier (FRBC)

FRBC uses fuzzy sets for reasoning and has been introduced by Ishibuchi [119]. Let consider an M-class classification problem in an n-dimensional normalized hyper-cube $[0,1]^n$. For this issue, fuzzy rules of the following type are used:

Rule R_q :
If x_1 is A_{q1} and ... and x_n is A_{qn} then Class L_q with CF_q , (68)

where R_q is the label of the q^{th} fuzzy rule, $\mathbf{x} = (x_1, ..., x_n)$ is an n-dimensional feature vector, A_{qi} is an antecedent fuzzy set ($i = 1,...,n$), L_q is a class label, CF_q is a real number in the unit interval [0,1] which represents a rule weight. The rule weight can be specified in a heuristic manner or it can be adjusted, e.g. by a learning algorithm introduced by Ishibuchi et al. [120], [121].

We use the n-dimensional vector $A_q = (A_{q1}, ..., A_{qn})$ to represent the antecedent part of the fuzzy rule R_q in (68) in a concise manner.

A set of fuzzy rules S of the type shown in (68) forms a fuzzy rule-based classifier. When an n-dimensional vector $\mathbf{x}_p = (x_{p1},..., x_{pn})$ is presented to S, first the *compatibility grade* of $\mathbf{x}_p$ with the antecedent part A_q of each fuzzy rule R_q in S is calculated as the product operator

$$\mu_{A_q}(\mathbf{x}_p) = \mu_{A_{q1}}(\mathbf{x}_{p1}) \times ... \times \mu_{A_{qn}}(\mathbf{x}_{pn}) \quad \text{for } R_q \in S \quad (69)$$

where $\mu_{A_{qi}}(.)$ is the membership function of A_{qi}. Then a single winner rule $R_{w(\mathbf{x}_p)}$ is identified for $\mathbf{x}_p$ as follows:

$$w(\mathbf{x}_p) = \arg\max_q \{CF_q \times \mu_{A_q}(\mathbf{x}_p) \mid R_q \in S\}, \quad (70)$$

where $w(\mathbf{x}_p)$ denotes the rule index of the winner rule for $\mathbf{x}_p$.

The vector $\mathbf{x}_p$ is classified by the single winner rule $R_{w(\mathbf{x}_p)}$ belonging to the respective class. If there is no fuzzy rule with a positive *compatibility grade* of $\mathbf{x}_p$ (i.e., if $\mathbf{x}_p$ is not covered by any fuzzy rules in S), the classification of $\mathbf{x}_p$ is rejected. The classification of $\mathbf{x}_p$ is also rejected if multiple fuzzy rules with different consequent classes have the same maximum value on the right-hand side of (70). In this case, $\mathbf{x}_p$ is on the classification boundary between different classes. We use the single winner-based fuzzy reasoning method in (70) for pattern classification.

An ideal theoretical example of a simple three-class, two-dimensional pattern classification problem with 20 patterns from each class is considered by Ishibuchi and Nojima [119]. There three linguistic values (*small*, *medium* and *large*) are used as antecedent fuzzy sets for each of the two attributes, and 3×3 fuzzy rules are generated.

S: fuzzy rule-based classifier with nine fuzzy rules [119]
R_1: If x_1 is *small* and x_2 is *small* then Class2 with 1.0,
R_2: If x_1 is *small* and x_2 is *medium* then Class2 with 1.0,
R_3: If x_1 is *small* and x_2 is *large* then Class1 with 1.0,
R_4: If x_1 is *medium* and x_2 is *small* then Class2 with 1.0,
R_5: If x_1 is *medium* and x_2 is *medium* then Class2 with 1.0,
R_6: If x_1 is *medium* and x_2 is *large* then Class1 with 1.0,
R_7: If x_1 is *large* and x_2 is *small* then Class3 with 1.0,
R_8: If x_1 is *large* and x_2 is *medium* then Class3 with 1.0,
R_9: If x_1 is *large* and x_2 is *large* then Class3 with 1.0.

The theoretical method presented by Ishibuchi does not answer the question how to construct membership functions, especially those corresponding to linguistic values. Hamilton and Stashuk [122] gave a suggestion for the construction of membership functions based on the standardized residual analysis but they applied it to continuous data.

For the discrete data, this problem can be solved by calculating the mean value $\bar{x}$ and standard deviation σ for the elements of each of the three classes. The membership function of each class, as suggested Jaworska in [123], is constructed as a symmetrical trapezoidal function in respect to the mean value $\bar{x}$ where the smaller basis has the σ length and the longer one - 2σ. Then, the ranges of

features x_1 and x_2 are divided into three equal intervals. Next, the mean value of a particular class is assigned to correspondent intervals which represent the proper linguistic values. The effect is visible in Fig. 25 for the horizontal and vertical axes in the side subplots, respectively.

In each case, the fuzzy rule-based classifier is constructed automatically by matching the membership function related to the proper linguistic value, resulting in the right class for each rule.

In multi-class systems a FRBC can be used as a second level classifier which has a decisive role in the ambiguity situation of classification at the first level. It means that when an object has not been classified unequivocally to the same class by decision tree, NB and SVM classifiers at the first stage, the FRBC is applied and it decides definitely about the object class.

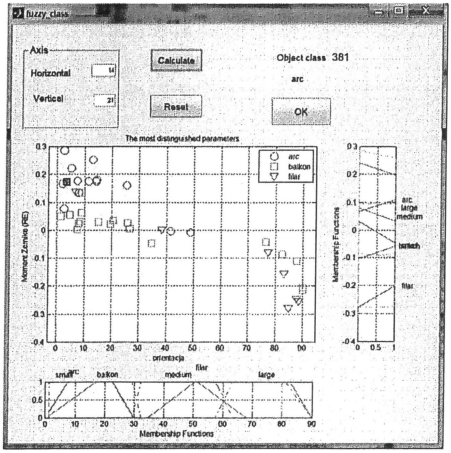


Fig. 25. Classification example **[123]**. The new element marked by the full green square is recognized as an arc among classes: arc, pillar and balcony. Membership functions are represented by solid colour lines and linguistic intervals are drawn in dashed lines. In this case, x_1 is orientation and x_2 the real part of Zernike's moment.

C. Spatial Relationship of Graphical Objects

It is easy for the user to recognize visually spatial object location but the system supports full automatic identification based on rules for location of graphical elements which is a challenging task.

Let us assume that we analyse a house image. Then, for instance, an object which is categorized as a window cannot be located over an object which is categorized as a chimney. For this example, rules of location mean that all architectural objects must be inside the bounding box of a house. For the image of a Caribbean beach, an object which is categorized as a palm cannot grow in the middle of the sea, and so on.

In the system designed by Jaworska [31], spatial object location in an image is used as the global feature. For this purpose, the mutual position of all objects is checked. Moreover, object location reduces the differences between high-level semantic concepts perceived by humans and low-level features interpreted by computers.

An image I_i is interpreted as a set of n objects o_{ij} composing it:

$$I_i = \{o_{i1}, o_{i2}, \ldots, o_{in}\} \tag{71}$$

where n is a number of objects in the image I_i.

Each object o_{ij} is characterized by a unique identifier and a set of features discussed earlier. This set of features includes a centroid $C_{ij} = (x_{ij}, y_{ij})$ and a label L_{ij} indicating the class of an object o_{ij} (such as window, door, etc.), identified in the process described in [124]. Let us assume that there are, in total, M classes of the objects recognized in the database. For convenience, the classes of the objects are numbered and thus L_k's are just IDs of classes.

Formally, let I be an image consisting of n objects and k be the number of different classes of these objects, $k \leq M$, because usually there are some objects of the same type in the image, for example, there can be four windows in a house.

Then, by the signature of an image I_i (71) we mean the following vector:

$$\text{Signature}(I_i) = [\text{nobc}_{i1}, \text{nobc}_{i2}, \ldots, \text{nobc}_{iM}] \tag{72}$$

where: nobc_{ik} denotes the number of objects of class L_k present in the representation of an image I_i, i.e. such objects o_{ij}.

Now, let C_p and C_q be two object centroids with $L_p<L_q$, located at the maximum distance from each other in the image, i.e.,

$$\text{dist}(C_p, C_q) = \max\{\text{dist}(C_i, C_j)\ \forall i,j \in \{1,2,\ldots,k\} \text{ and } L_i \neq L_j\} \quad (73)$$

where: dist(•) is the Euclidean distance between two centroids (see Fig. 26 middle subplot). The line joining the most distant centroids is the line of reference and its direction from centroid C_p to C_q is the direction of reference for computed angles θ_{ij} between other centroids. This way of computing angles makes the method invariant to image rotation.

Thus, the mutual location of two objects in the image is described in relation to the line of reference by triples (L_i, L_j, θ_{ij}) (see Fig. 26 middle subplot). So/Hence, there are $T = m\,(m\text{-}1)/2$ numbers of triples, generated to logically represent an image consisting of m objects. Let S be a set of all triples, then we apply the concept of principal component analysis (PCA) proposed by Chang and Wu [125] and later modified by Guru and Punitha [126] to determine the first principal component vectors (PCVs).

First, a matrix of observations $X_{3\times N}$ where each triple is one observation is constructed based on a set of observations S. Next, the mean value u of each variable is calculated, and the deviations from the mean vector **u** is subtracted in order to generate matrix **B=X-u1**, where **1** - vector of all 1s. In the next step, the covariance matrix $\mathbf{C}_{3\times3}$ is found from the outer product of matrix **B** by itself as:

$$\mathbf{C} = \mathbb{E}[\mathbf{B}\otimes\mathbf{B}] = \mathbb{E}[\mathbf{B}\,\mathbf{B}^*] = 1/N\,[\mathbf{B}\,\mathbf{B}^*]. \quad (74)$$

where: $\mathbb{E}$ is the expected value operator, $\otimes$ is the outer product operator, and * is the conjugate transpose operator. Eventually, eigenvectors, which diagonalises the covariance matrix **C** are found as follows:

$$V^{-1}\,C\,V=D \quad (75)$$

where: **D** is the diagonal matrix of the eigenvalues of **C**. Vectors V are our three principal components.

For further analysis we use the first nine coefficients of the PCV which are the "spatial components" of the representation of an image I_i, and are denoted PCV_i.

Fig. 26 presents the most important stages in the determination of the spatial object location: from the presentation of the original image (top), through the object centroid locations (colours indicate particular classes) (middle subplot), to the 3D subplot of the principal components (bottom).

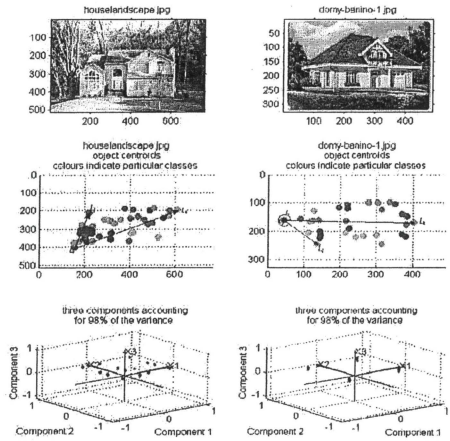


Fig. 26. The main stages of the PCV applied to determine the unique object spatial location in an image.

VI. Graphical User Interface

A. Query Concept Overview

Each CBIR system needs to be asked a query by the user. In traditional, alpha-numeric DBs the system of queries has been highly developed, whereas for the image retrieval a content query has not come up to users' expectations. It stems from the fact that content-based searches have important distinctions compared to traditional searches. The limitations of these systems include both the image representations they use and their

methods of accessing those representations to find images. Systems based on keyword querying are often unintuitive and offer little help in understanding why certain images were returned and how to refine the query.

Independently of a diversity of methods focused on the retrieval, in particular, search by association, search for a specific image, or category search [127] we can generally divide query methods into:

- Query by keywords [128]
- Query by example [8], [9]
- Query by canvas [129], [40]
- Query by sketches [130]
- Query by spatial icons [131]
- Query by image region [131]
- Designed query for semantic retrieval [132], [133]

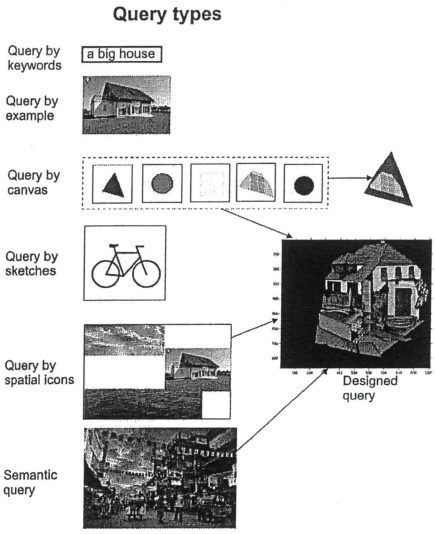


Fig. 27. Query types.

Query by keywords: The first method for asking the database a question was using keywords as an analog to queries in alpha-numeric DBs. This method requires manually added text annotation for images collected in DB. It is still used in WWW image search engines but it is notorious for being incomplete, inconsistent, context sensitive and the ambiguity of meaning of the keywords. In this case [128], the keyword ambiguity is expanded to the selected reference classes most relevant to the query keyword. For example, the keyword 'apple' can mean: 'apple fruit', 'apple computer', 'apple logo', or 'apple tree' from which reference classes are selected. These attempts try to fill in the semantic gap that exists between the description of an image and the image itself.

Query by example: At present, most systems use query by example (QBE) whose major advantage is the capability to determine a set of attributes or features that describe the contents of the user's desired image [129], [9]. In a nutshell, QBE provides a clue regarding the search area for the search engine, but its drawback is the fact that the user first has to find an image which he wants to use as a query. In some situations, the most difficult task is to find this one proper image which the user keeps in mind to feed it to the system as a query by example. An evident example is shown in [130] where the face sketches are needed for face recognition.

Query by canvas allows the user to compose a visual query using geometrical shapes, colours, and textures. This approach inherently tends to specify objects of interest in an indirect way using primitive features [40]. Moreover, the similarity matching between query and images relies on effective pre-segmentation of regions in the images, which is generally complex and difficult [129].

Query by sketches enables the user to draw the shape of an object as query but is not popular, perhaps because most users are rather poor at graphic design [9]. For this reason applications have used a query by sketches in a limited form only to images of dominant objects in a uniform background.

Query by spatial icons is represented by the visual icons with spatial constraints. It specifies a query using a higher-level visual semantic representation by a query formulae chains specifies a region which are designated via logical

operators. explicitly in a Boolean expression [131]. In the case of implicit query expression, specifying pool water, trees, or a group of people is unnatural, if not impossible.

Query by image regions [131] is an enhancement to QBE. The query by multiple regions approach [134] allows for the composition of a query from multiple regions from example images with or without spatial layout. It is useful when the user is looking for abstract visual concepts. The visual query term specifies the region where a semantic support region (SSR) should appear, and a query forming chains of these terms using logical operators. the concept of SSRs, which possess the following properties:

- They are extracted directly from images without segmentation and possess semantic power. They can be used to avoid/bypass the *semantic extraction problem.*
- Spatial information is retained in the index based on SSRs.
- SSRs are learned and detected from multiscale tessellated image blocks which are generally numerous and statistical significant.

Semantic query is difficult for a semantic interpretation because computer systems extract only image low-level features when the user expects many different objects in both the foreground and the background of such an image [128]. Hence, the multiple semantic interpretation results in a retrieval problem because it needs to take into account simultaneously: low-level features, object layout and a big number of involved objects. For this reason, recently systems have appeared offering designed queries to the user [132], [135], [133] or sampled independently images. Later/Then they used different procedures to estimate a density distribution, for example, as a mixture of Gaussianas [136].

B. User Designed Query Concept

At present, most systems use query by example (QBE), but its drawback is the fact that the user first has to find an image which he wants to use as a query. In some situations the most difficult task is to find this one proper image which the user keeps in mind to feed it to the system as a query by example. An evident example is shown in [130] where face sketches are needed for face recognition.

We propose a graphical editor which enables the user to compose the image he/she has in mind from the previously segmented objects (see Fig. 28). It is a bitmap editor which allows for a selection of linear prompts in the form of contour sketches generated from images existing in the DB. The contours are computed as edges based on the Canny algorithm and as a vector model set to the DB during the pre-processing stage. Next, from the list of object classes the user can select elements to prepare a rough sketch of an imaginary landscape. There are many editing tools available, for instance:

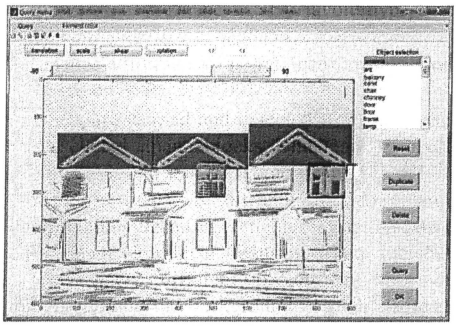


Fig. 28. The main GUI window. An early stage of a terraced house query construction.

- creating masks to cut off the redundant fragments of a bitmap (see Fig. 29 a) and b));
- changing a bitmap colour (Fig. 29 c) and d));
- basic geometrical transformation, such as: translation, scale, rotation and shear;
- duplication of repeating fragments;
- reordering bitmaps forward or backward.

This GUI is a prototype, so it is not as well-developed as commercial programs, e.g. CorelDraw, nevertheless, the user can design an image consisting of as many elements as they need. The only constraint at the moment is the number of classes introduced to the DB, which now stands at 40 but is set to increase. Once the

image has been drafted, the UDQ is sent to the search engine and is matched according to the rules described in sec. VII pt. J.

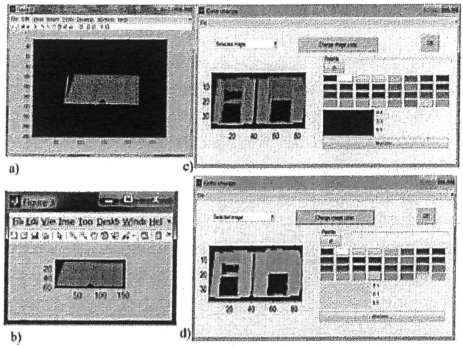


Fig. 29. Main components of the GUI. We can draw a contour of the bitmap (see a) and b)) and change the colour of an element (see c) and d)).

However, in case of the absence of UDQ, the search engine can work with a query consisting of a full image downloaded, for example, from the Internet.

VII. Search Engine – Retrieval Techniques

A. Introduction

Below, we analyse the most common search engines beginning from the simplest one based on low-level features, through engines including annotations and ending with engines attempting to use semantic matching. In each case we describe the search method, we emphasize goals to which the engine is dedicated and we conclude with a presentation of pros and cons.

Search engines are constructed to fulfil particular criteria which we have described in sec. II part A. The discussion of these issues will determine which matching mechanism listed below is recommended as more efficient than the others. For instance, if the user wants to find one object in many pictures, e.g. a face in an airport video, they will need a different mechanism than the user who only orders their collection of holidays photos, etc.

Hence, the currently predominant engine categories listed below are based on [37]:

- low-level features;
- search by metadata;
- using object ontology to define high-level concepts,
- bag-of-visual-words (BoW), stemming from text analysis,
- object retrieval using SIFT and its modification methods,
- relevance feedback (RF) into a retrieval loop for continuous learning about users' intentions,
- generating a semantic template (ST) to support high-level image retrieval,
- making use of both the visual content of images and the textual information obtained from the Web for WWW (the Web) image retrieval,
- combining visual properties of selected objects (or a set of relevant visual features), spatial or temporal relationships of graphical objects [137], [138], with semantic properties [139], [37].

B. Visualization and Browsing of Image Databases

Image browsing systems [140] attempt to provide the user with an intuitive interface, displaying at once many images as thumbnails in order to harness the cognitive power of the human mind to recognize and comprehend an image in a second. Interaction with a traditional QBE system can often lead to confusion and frustration on the part of the users, which was confirmed in the study by Rodden and Wood [141].

Browsing systems give a useful alternative to QBE, providing an overview of the database to the user, which allows for intuitive navigation throughout the system. This is particularly the case when images are arranged according to mutual similarity, as has been shown in [142], where a random arrangement of images was compared with a visualisation which positioned images according to their visual similarities, i.e. where images that are visually similar to each other are located close to one another in the visualisation space [143]. The user can then focus on regions of

the visualisations that they are attracted to or believe will harbour a particular concept they have in mind. Browsing such visualisations can increase the rate of retrieval.

Fig. 30. DB browsing based on visual similarity **[144]**.

For image database browsing, the mapping-based visualization is a typical mechanism which shows the potential relationships within the DB. In order to visualize these high-dimensional features, we have to map them down to 2D on a computer screen.

A variety of methods have been devised in order to visualise images:

- Principal Component Analysis (PCA) which is the simplest dimensionality reduction approach, working in a linear manner (cf. sec. IV part C).
- Multi-Dimensional Scaling (MDS) in turn preserves the original distances in a high dimensional space, calculating a similarity matrix which describes all pair-wise distances between objects in the original space and next it projects them to the low-dimensional space. Based on the similarity matrix, the 'stress' measure can be formulated as follows [143]:

$$\mathrm{st} = \frac{\sum_{i,j}\left(\hat{\delta}_{ij} - \delta_{ij}\right)^2}{\sum_{i,j}\delta_{ij}^2} \qquad (76)$$

where δ_{ij} is the original distance between objects i and j, and $\hat{\delta}_{ij}$ is the distance in the low-dimensional space. Rubner et al. [9] who employed MDS based on colour signatures of images and the earth mover's distance (EMD) was able to create a representation of the high-dimensional feature space using MDS, placing image thumbnails at the co-ordinates derived by the algorithm, see Fig. 30.

- Fast Map is an alternative dimensionality reduction technique devised by Faloutsos and Lin [145]. Fast Map reduces high-dimensional spaces down to a linear 2D or 3D space. The algorithm, having a linear complexity $O(kn)$, selects two pivot objects, an arbitrary image and its furthest possible neighbour. All points are mapped to the line that connects the two pivots.
- Clustering-based visualization. Content-based cluster-ing uses extracted feature vectors in order to group perceptually similar images together. The advantage of this approach is that no metadata or prior annotation is required in order to arrange images in this manner, although image features or similarity measures which do not model human perception well, can create groupings that may potentially make it difficult for a user to intuitively browse an image database.

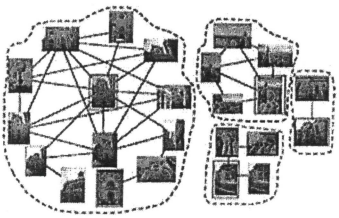


Fig. 31. A content-based image clustering method for public image repositories **[146]**.

- Graph-based visualization utilizes links between images to construct a graph where the nodes of the graph are the images, and the edges form the links between similar images.

Links can be established through a variety of means including visual similarity between images, or shared keyword annotations, for instance the Pathfinder network [147], see Fig. **32**. The graph-based visualization appears to be less common because it is typically quadratic in complexity, and therefore can only be computed off-line in order to allow for real-time browsing.

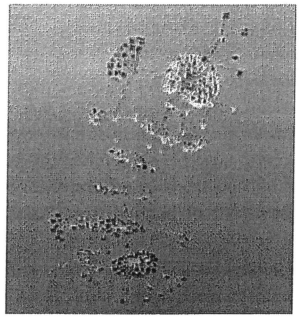


Fig. 32. Pathfinder networks of images organized by color histogram **[147]**.

C. *Information Retrieval Based on Low-level Features*

Image I can be modelled as a function O of the raw image file D, its features F, and representations R. The image model is described below and is also shown in Fig. 33:

$$I = O(D,F,R) \tag{77}$$

where D is the raw image data, for instance, an image file, $F = \{f_j\}$, $j = 1, \ldots, J$ is a set of low-level image features, such as colour, shape, texture, etc, $R_j = \{r_{jk}\}$, $k = 1, \ldots, K_j$ is a set of representations for a given feature f_j, e.g. the colour histogram and colour moments are representations of the colour feature. Each representation r_{jk} is a vector consisting of multiple components, i.e.:

$$r_{jk} = [r_{jk1},\ldots,r_{jkl},\ldots, r_{jkL_{jk}}] \tag{78}$$

where L_{jk} is the length of the vector r_{jk}.

This image model has three abstract information levels (data, feature, representation), increasing in the informative granularity. Furthermore, different weights (U at the data level, V_j at the feature level and W_{jk} at the representation level) exist to reflect a particular entity's importance of its level.

In order to compare the distance between two images, we need to define the retrieval model. The image model $O(D,F,R)$ together with a set of distance measures specify the retrieval model. Hence, we measure the distance at the three levels: image – query $\Phi()$, features $\Theta()$ and representations $\Psi()$. Let r_{mjk} be the jk^{th} representation vector for the m^{th} image in the database, where $m = 1,\ldots, M$ and M is the total number of images in the DB. Let $q_{jk}, j = 1,\ldots, J$, $k = 1, \ldots, K_j$ be the query vector for the jk^{th} representation. The retrieval process is illustrated in Fig. 33 and can be described as follows.

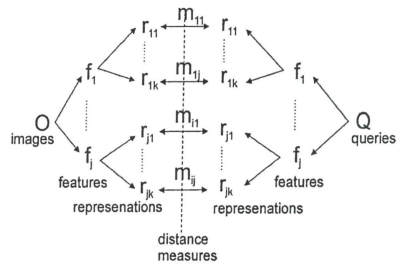


Fig. 33. Retrieval process based on feature object representation **[148]**.

First, we initialize the values of the weights U, V_j and W_{jk}. The distance between image and a query in terms of the jk^{th} representation is:

$$d_m(r_{jk}) = \Psi_{jk}(r_{mjk}, q_{jk}, W_{jk}), \quad m = 1,\ldots, M, j = 1,\ldots, J, k = 1, \ldots, K_j \tag{79}$$

where $d_m(r_{jk})$ denotes the distance between the m^{th} image and the query in terms of representation jk. Then, the distance between the image and the query in terms of feature j is:

$$d_m(f_j) = \quad (80)$$
$$= \Theta_j(d_m(r_{jk}), V_j) = \Theta_j(\Psi_{jk}(r_{mjk}, q_{jk}, W_{jk}), V_j)$$

Then, overall distance is:

$$d_m = \Phi(d_m(f_j), U) =$$
$$= \Phi(\Theta_j(\Psi_{jk}(r_{mjk}, q_{jk}, W_{jk}), V_j), U) \quad (81)$$

The images in DB are ordered by their overall distances to the query (d_m). The N most similar ones are returned to the user, where N is the number of images the user wants to retrieve.

According to the user's preferences the system dynamically updates the weights U, V_j and W_{jk}. For the Euclidean distance among the feature vector Y. Rui and Th. Hhuang [148] suggested computed weight as $w_{jk} = \frac{1}{\sigma_{jk}}$ means one over standard deviation.

1) Scale- Invariant Feature Transform SIFT

The detection and description of local image features can help in object recognition. The SIFT features are local and based on the appearance of the object at particular interest points, and are invariant to image scale and rotation. They are also robust to changes in illumination, noise, and minor changes in viewpoint. In addition to these properties, they are highly distinctive, relatively easy to extract and allow for correct object identification with low probability of mismatch. They are relatively easy to match against a (large) database of local features but however the high dimensionality can be an issue, and generally probabilistic algorithms such as *k-d* trees with best bin first search are used. Object description by set of SIFT features is also robust to partial occlusion; as few as 3 SIFT features from an object are enough to compute its location and pose.

An object is recognized in a new image by individually comparing each feature from the query to an image from a database and finding candidate matching features based on the Euclidean distance of their feature vectors. From the full set of matches, subsets of key points that agree on the object and its location, scale, and orientation in a query are identified to filter out good matches. Consistent clusters are determined by using an efficient hash table implementation of the generalized Hough transform. Each cluster of 3 or more features that agree on an object and its pose is then subject to further detailed model verification and subsequently outliers are discarded. Finally, the probability that a particular set of features indicates the presence of an object is computed through the Bayesian probability analysis, given the accuracy of the fit and number of probable false matches. Object matches that pass all these tests can be identified as correct with high confidence.

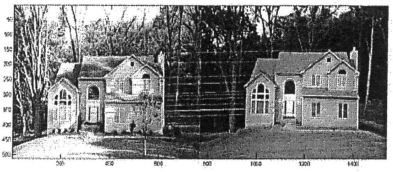


Fig. 34. Point-to-point correspondence found by the SIFT descriptors.

This property suggested that this method retrieves all images containing a specific object, even in a large scale image dataset, when that object is given as a query by example (QBE).

Hence, SIFT needs the query-by-example, but in some situations it may be difficult to provide, for instance, when we have an image in our mind but it is difficult to find it as QBE and additionally, we do not need the whole collection of similar images.

SIFT's additional advantage is the fact that it solved the problem of searching for disparity, independently of the issue of epipolar lines in stereovision. The example of point-to-point correspondence is presented in Fig. 34.

D. Object ontology to define high-level concepts

Generally speaking, ontologies define the concepts and relationships used to describe and represent an area of knowledge. Ontology gives the ability to model the semantics contained in images, such as objects or events. It provides, in a formal way, mutual understanding in a specific domain between humans and computers. Hence,

ontology represents knowledge in a hierarchical structure which is used to describe and organize an image collection and it also shows the relation between these images.

In the early approaches high-level concepts were described using the intermediate-level descriptors of the object's ontology. These descriptors were automatically mapped from the low-level features calculated for each region in the database, thus allowing the association of high-level concepts and potentially relevant image regions [149]. Later, ontology was employed to spatial relationships in images such as connectivity, disjoint, meet, adjacency, overlap, cover, or inside. But the image was divided into 3x3, 5x5 or 9x9 windows instead of separate objects [150].

For ontological DBs the Web Ontology Languages (OWL), as a family of knowledge representation languages, have been constructed for authoring ontologies characterized by formal semantics.

An example of a search engine for multimedia has been proposed by Doulaverakis [151] and the system architecture is illustrated in FIG. 35. Here the user initiates a query by providing a QBE. This is depicted as case A in FIG. 35 and comprises three steps. In the first step (1A) the content-based search is completed by analysing the provided multimedia content (i.e. performing the segmentation, extracting the low-level MPEG-7 descriptors and evaluating the distance between the prototype and the other figures stored in the multimedia database). The second step (2A) takes into account the metadata (which are mapped to the relevant ontologies) of the highest ranked results. For instance, the system may detect the highest ranked results in terms of visual similarity. Based on this information, an ontology-based query is formulated internally in the search engine, which links the knowledge base and enriches the result set with multimedia content that is close semantically to the initial content-based results (3A).

Eventually, the response returned to the user covers a wider range of items of interest, thus facilitating the browsing through the collection and shifting the burden of composing queries to the system instead of the user. The reverse process is equally interesting (case B in FIG. 35). Here, the initial query is a combination of terms defined in the ontology, e.g. 'Artefacts from the 1st century BC'. The knowledge base storing the ontology returns the items that fall into that category, as the first step (1B). The second step (2B) involves the extraction and clustering of the low-level multimedia features of this initial set, which is followed by multimedia retrieval, leading to the final step (3B).

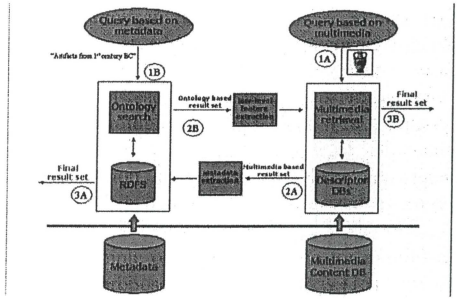


Fig. 35. A hybrid ontology and content-based search engine architecture follows **[151]**.

Ontology is also a method for organizing extra large-scale image collections, like the ImageNet dataset, created at Stanford University [152].

There are some advantages of ontology:

- its application bridges the semantic gap;
- there is a special language for the user to ask a question;
- ontology-based algorithms are easy to design and are suitable for applications with simple semantic features.

The disadvantage is the necessity of preparing a special DB and annotating the introduction.

E. Bag of Visual Words

A simple approach to classifying images is to treat them as a collection of regions, describing only their appearance and ignoring their spatial structure which is very important in image representation. Similar models have been successfully used in the text community to analyse documents and are known as "bag-of-words"

models, since each document is represented by a distribution over fixed vocabulary. Using such a representation, methods such as the probabilistic latent semantic analysis (pLSA) [153] and the latent Dirichlet allocation (LDA) [154] are able to extract coherent topics within document collections in an unsupervised manner.

Some time ago, Fei-Fei and Perona [155] and Sivic et al. [156] applied such methods to the visual domain using [153] and [154] in their algorithm.

They model an image as a collection of local patches which are detected by a sliding grid and random sampling of scales. Each patch is represented by a code-word from a large vocabulary of code-words which are sorted in descending order according to the size of their membership and represent simple orientations and illumination patterns. By learning they achieved a model that best represents the distribution of these code-words in each category of scenes. In the recognition process they identified all the code-words in the unknown image. The training and testing process is presented in Fig. 36. in a symbolic way.

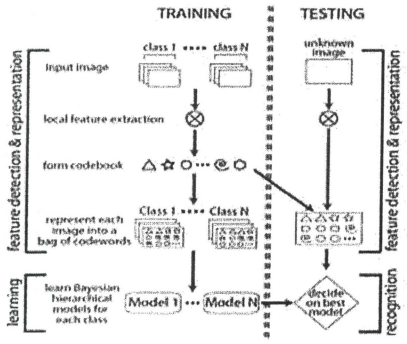


Fig. 36. Flow chart of the algorithm follows **[155]**.

They found the category model that matched best the distribution of the code-words of the particular image. Their model is based on a principled probabilistic approach to learn automatically the distribution of code-words and the intermediate-level themes treated as texture descriptions.

An advantage of the BoW model that it is applicable in case of complex indoor and outdoor images. One of the notorious disadvantages of BoW is that it ignores the spatial relationships among the patches, which are very important in image representation. Additionally, the system needs the preparation of code-words, classes and Bayesian hierarchical models for each class.

F. Relevance Feedback (RF)

Relevance feedback [17], [19], [22] is an interactive technique based on feedback information between a user and a search engine by requiring the user to label semantically similar or dissimilar images with the query image, which are treat as positive and negative samples, respectively. During the last decade, various RF techniques have been proposed to involve the user in the loop to enhance the performance of CBIR [19], [22], [21].

Large modern DBs actively employ user's interaction for relevance feedback (RF). This is an interactive technique based on feedback information between the user and a search engine in which the user labels semantically similar or dissimilar images with a query image, which is treated as positive and negative samples, respectively. Images labelled in this way are incorporated into a training set. The general architecture of such systems is presented in Fig. 37.

A more precisely labelled training set boosts algorithms to build a wider boundary between cluster features. For this purpose either Support Vector Machine (SVM) is applied to estimate the density of positive feedbacks or regarding the RF as a strict two-class on-line classification problem or discriminant analysis is used to find a low dimensional subspace of the feature space, so that positive feedbacks and negative feedbacks (which we can see in a relevance feedback in Fig. 37) are well separated after projecting onto subspace.

During the last decade, various RF techniques have been proposed to involve the user in the loop to enhance the performance of CBIR [157], [158].

For example, Rui and Huang [148] suggest that for each of the retrieved images, the user provides a degree-of-relevance score, according to the user's feedback, such that the adjusted query q_{jk} and the weights U, V_j and W_{jk} (cf. (81)) better match the user's information needs. The user may use a special scroll bars to interactively introduce values of weights which is a more effective mechanism that only binary distinction (as it is illustrated in Fig. 37).

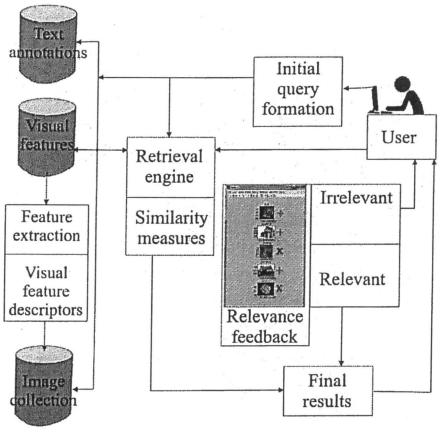


Fig. 37. CBIR architecture with the relevance feedback (RF) mechanism.

Whether a retrieval model can update its weights it can better distinguishe the *interactive approach* from the *isolated approach* in which all the weights are fixed. Because of the fixed parameters, this approach models the user's information needs and perception subjectivity less effectively. For the interactive approach weights and query vectors are dynamically updated via relevance feedback which improves the efficiency of the system.

Whereas, for instance, L. Zhang et al [19] propose a framework of subspace learning when the training images are associated with only similar and dissimilar pairwise constraints, i.e., Conjunctive Patches Subspace Learning (CPSL) with side information, to explicitly exploit the user's historical feedback log data. It means that they minimize the distances between samples with similar pairwise constraints and to maximize the distances between samples with dissimilar pairwise constraints simultaneously. Samples are whole images for which neighbourhood is calculated as locally linear embedding (LLE) [159].

An option of RF is the adaptive technique based on the ostensive model of developing information needs proposed by J. Urban [160].

Generally, an advantage of RF approach is the fact that the system can start with a limited number of samples because the user will provide next labelled samples. RF has been proved to be effective in boosting image retrieval accuracy. The disadvantage is that most current systems requires about several iterations before it converges to a stable performance level, but users are usually impatient and may give up after two or three tries.

G. Semantic Template

In [161] Chang et al. introduced the idea of the semantic visual template (SVT) to link low-level image features to high-level concepts for video retrieval. A visual template is a set of icons or example scenes or objects denoting a personalized view of concepts such as meetings, sunsets, etc. The feature vectors of these example scenes or objects are extracted for the query process. To generate SVTs, the user first defines the template for a specific concept by specifying the objects and their spatial and temporal constraints, the weights assigned to each feature of each object. This initial query scenario is put to the system. Through the interaction with users, the system finally converges to a small set of exemplar queries that 'best' match (maximize the recall) the concept in the user's mind.

Firstly, the user submits a query image with a concept representing the image. After several iterations, the system returns some relevant images to the user. The feature centroids of these images are calculated and used as the representation of the query concept. Then the ST is defined as $ST = \{C, F, W\}$ with C the query concept, F the centroid feature obtained, and W being the weight applied to feature vectors. During the retrieval

process, once the user submits a query concept, the system can find a corresponding ST, and use the corresponding F and W to find similar images.

A disadvantage of this system is the necessity of possessing a big lexical database [162].

H. WWW Image Retrieval

Image search is based on comparison of metadata associated with the image as keywords, text, etc. and it is obtained a set of images sorted by relevance. The metadata associated with each image can reference the title of the image, format, color, etc. and can be generated manually or automatically. This metadata generation process is called audiovisual indexing.

WWW search engines exploit the evidence from both the HTML text and visual features of images and develop two independent classifiers based on text and visual image features, respectively. The URL of an image file often has a clear hierarchical structure, including some information about the image, such as image category. In addition, the HTML document also contains some useful information in the image title, ALT-tag, the descriptive text surrounding the image, hyperlinks, etc.

However, the disadvantage is the fact that the retrieval precision is poor and as a result the user has to go through the entire list to find the desired images. This is a time-consuming process which always contains multiple topics which are mixed together. To improve the Web image retrieval performance, researchers are making an effort to fuse the evidence from textual information and visual image contents.

For example, Rasiwasia at al. proposed a combination of a query-by-visual-example (QBVE) with a query-by-semantic-example (QBSE) based on the probability of existance of a visual level represented as a set of feature vectors and the probability of a semantic concept by which an image is annotated. By using the Bayes rule and a similarity function based on methods measuring the distance between two probability distributions (such as the Kullback-Leibler Divergence, Jensen-Shannon Divergence, correlation, etc), they retrieve images most similar to the semantic signature [136].

On the other hand Wang et al. combine the visual features of images with the signatures received from the visual semantic space. For each relevant keyword, a semantic signature of the image is extracted by computing the visual similarities between the image and the reference classes of the keyword using the earlier trained classifiers. The reference classes form the basis of the semantic space of the keyword. If an image has N relevant keywords, then it has N semantic signatures to be computed and stored offline [163].

An advantage of the Web image retrieval is that some additional information on the Web is available to facilitate semantic-based image retrieval.

I. Hybrid semantic strategy

Now, we will describe how the similarity between two images is determined and used to answer a query. Let the query be an image I_q, such as $I_q = \{o_{q1}, o_{q2}, \ldots, o_{qn}\}$, where o_{ij} are objects. An image in the database is denoted as I_b, $I_b = \{o_{b1}, o_{b2}, \ldots, o_{bm}\}$. Let us assume that there are, in total, $M = 40$ classes of the objects recognized in the database, denoted as labels $L_1, L_2, \ldots, L_M$. Then, by the image signature I_i we mean the following vector:

$$\text{Signature}(I_i) = [\text{nobc}_{i1}, \text{nobc}_{i2}, \ldots, \text{nobc}_{iM}] \quad (82)$$

where: nobc_{ik} denotes the number of objects of class L_k present in the representation of an image I_i, i.e. such objects o_{ij}.

In order to answer the query I_q, we compare it with each image I_b from the database in the following way. A query image is obtained from the GUI, where the user constructs their own image from selected DB objects. First of all, we determine a similarity measure sim_{sgn} between the signatures of query I_q and image I_b:

$$\text{sim}_{\text{sgn}}(I_q, I_b) = \sum_i (\text{nob}_{qi} - \text{nob}_{bi}) \quad (83)$$

computing it as an analogy with the Hamming distance between two vectors of their signatures (cf. (72)), such that $\mathrm{sim}_{\mathrm{sgn}} \geq 0$ and $\max_i(\mathrm{nob}_{qi} - \mathrm{nob}_{bi}) \leq \mathrm{tr}$, tr is the limit of the number of elements of a particular class by which I_q and I_b can differ. It means that we prefer images with the same classes as the query. Similarity (83) is non-symmetric because if some classes in the query are missing from the compared image the components of (83) can be negative.

If the maximum component of (83) is bigger than a given threshold (a parameter of the search engine), then image I_b is rejected, i.e., not considered further in the process of answering query I_q. Otherwise, we proceed to the next step and we find the spatial similarity $\mathrm{sim}_{\mathrm{PCV}}$ (84) of images I_q and I_b, based on the Euclidean, City block or Mahalanobis distance between their PCVs as:

$$\mathrm{sim}_{\mathrm{PCV}}(I_q, I_b) = 1 - \sqrt{\sum_{i=1}^{3}(PCV_{bi} - PCV_{qi})^2} \qquad (84)$$

If the similarity (84) is smaller than the threshold (a parameter of the query), then image I_b is rejected. The order of steps (83) and (84) can be reversed because they are the global parameters and hence can be selected by the user.

Next, we proceed to the final step, namely, we compare the similarity of the objects representing both images I_q and I_b. For each object o_{qi} present in the representation of the query I_q, we find the most similar object o_{bj} of the same class, i.e. $L_{qi} = L_{bj}$. If there is no object o_{bj} of the class L_{qi}, then $\mathrm{sim}_{\mathrm{ob}}(o_{qi}, o_b) = 0$. Otherwise, similarity $\mathrm{sim}_{ob}(o_{qi}, o_b)$ between objects of the same class is computed as follows:

$$\mathrm{sim}_{\mathrm{ob}}(o_{qi}, o_{bj}) = 1 - \sqrt{\sum_{l}(Fo_{qil} - Fo_{bjl})^2} \qquad (85)$$

where l is the index of feature vectors F_O used to represent an object. Thus, we obtain the vector of similarities between query I_q and image I_b.

$$\mathrm{sim}(I_q, I_b) = \begin{bmatrix} \mathrm{sim}_{\mathrm{ob}}(o_{q1}, o_{b1}) \\ \vdots \\ \mathrm{sim}_{\mathrm{ob}}(o_{qn}, o_{bn}) \end{bmatrix} \qquad (86)$$

where n is the number of objects present in the representation of I_q. In order to compare images I_b with the query I_q, we compute the sum of $\mathrm{sim}_{ob}(o_{qi}, o_b)$ and then use the natural order of the numbers. Therefore, the image I_b is listed as the first in the answer to the query I_q, for which the sum of similarities is the highest.

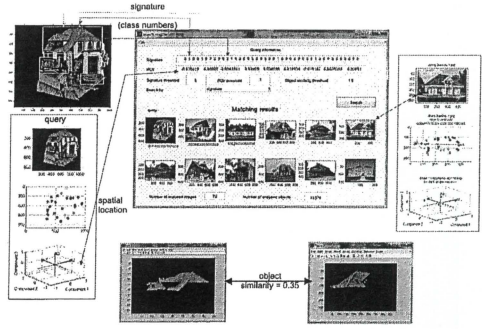


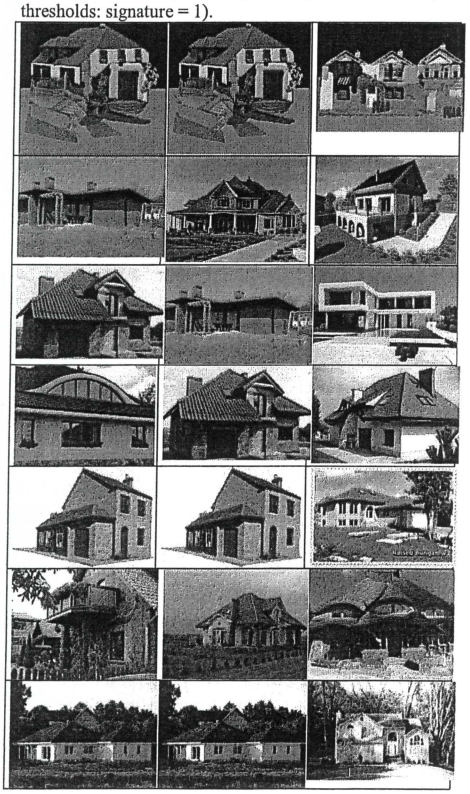
Fig. 38. A main concept of the search engine.

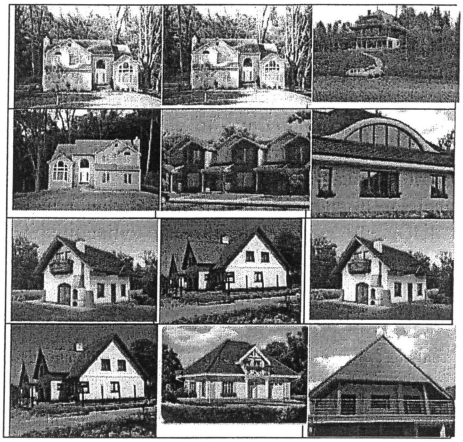
Fig. 38 presents the main elements of the search engine interface with reference images which are present in the CBIR system. The main (middle) window displays the query signature and PCV, and below it the user is able to set threshold values for the signature, PCV and object similarity. At this stage of system verification it is useful to have these thresholds and metrics at hand. In the final internet version these parameters will be invisible to the user, or limited to the best ranges. The lower half of the window is dedicated to matching results. In the top left of the figure we can see a user designed query comprising elements whose numbers are listed in the signature line. Below the query there is a box with a query miniature, a graph showing the centroids of query components and, further below, there is a graph with PCV components. In the bottom centre windows there are two elements of the same class (e.g. a roof)

and we calculate their similarity. On the right side there is a box which is an example of PCA for an image from the DB. The user introduces thresholds to calculate each kind of similarity.

The strong side of our system is its semantic context which limits the semantic gap by taking into account middle-level features, such as objects, their numbers and spatial locations in an image. Additionally, we offer the user the GUI to compose their query by which we eliminate the necessity of looking for a QBE.

TABLE VI.
The matching results for queries (in the first row) and the universal image similarity index for these matches when PCV similarity is calculated based on: (column 1) the Euclidean distance, (column 2) the City block distance (for thresholds: signature = 1).





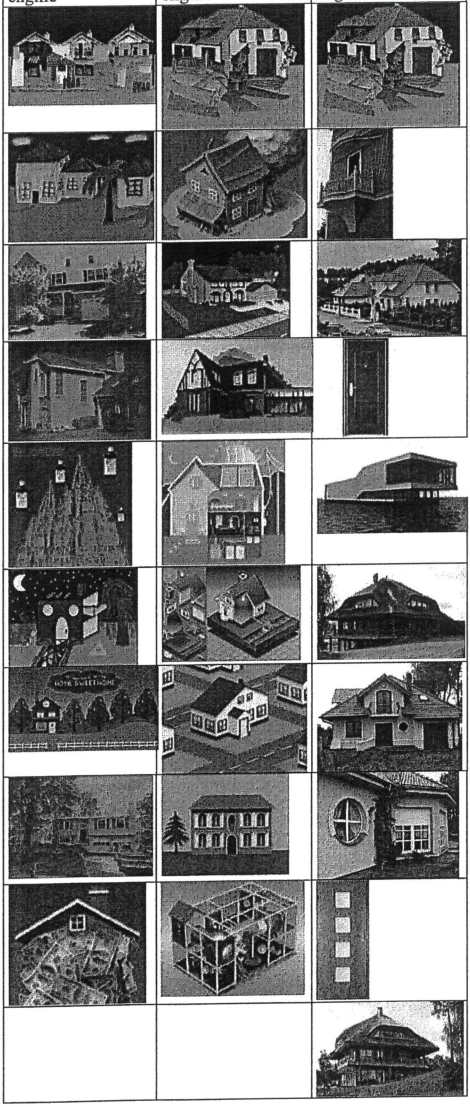
On the other hand, our system requires the preparation of DB containing objects, patterns, and classes.

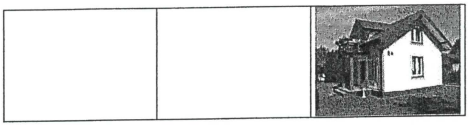
As we have mentioned (see sec. VI part B), a query is generated with the UDQ interface and its size, number of elements (patches) and complication depends on the user. The search engine displays a maximum of 11 best matched images from our DB. Although the user designed a few details, the search results are quite acceptable (see TABLE **VI**).

We also decided to compare our results with the Google image search engine. The results are presented in TABLE VII. We also compare our search engine with the SIFT method and TABLE VII column 3 presents the matching results for a query designed in our system. As it can be seen the best selected matches are those images whose elements can be found in the designed query.

We have opted for this comparison because these systems match images without annotations, which has been the most important condition. Systems using annotations belong to quite a different category while our focus is on pure image matching.

TABLE VII.
MATCHES FOR THE GOOGLE AND SIFT IMAGE SEARCH ENGINE
(Queries in the second row.)

The Google search engine	The Google search engine	The SIFT search engine
		



The default comparison of search engines should be carried out based on the standard DB benchmarks. However, the user needs are more specific and we shall prepare dedicated search engines for these requirements, for instance, recognition of licence plate locations [164].

As we can see in TABLE VII. the Google engine treats the sketch houses as drawings, not as real photographs, whereas the SIFT one found the images from which the designed query consists, which is proper for this method, but has not been the user's intention who wants to receive house images most similar to their query in general and in detail.

VIII. IMAGE COLLECTIONS

All the above-described search engines had to be tested as a result of which many classified, indexed or annotated reference image databases were developed.

Some new systems use a subset of the *Corel image dataset* [165], others use either self-collected images or other image sets such as LA resource pictures [166]. The Corel image database contains 10,800 images from the Corel Photo Gallery divided into 80 concept groups, ranging from animals and outdoor sports to natural sceneries. These images are professionally pre-classified into different categories. Each group includes more than 100 homogeneous images.

The *Kodak database* of consumer images [167] and Brodatz textures [168], [169] are widely used in perceptual texture feature studies. Images collected from the Internet serve as data source, especially for systems targeting Web image retrieval [136] [163].

The *Pascal Visual Object Classes* (VOC) consist of a publicly available dataset of images together with ground truth annotation and standardised evaluation software [170], [171]. The data is divided into two main subsets: training or validation data, and test data. There is complete

annotation for twenty classes: i.e. all images are annotated with bounding boxes for every instance of the twenty classes for the classification and detection.

The *ImageNet* is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by 14,197,122 labeled high-resolution images organized by 21841 indexes and belonging to roughly 22, 000 categories. Currently, we have an average of over five hundred images per node [38], [172].

The *Caltech-256 Image Set* is an image database released in 2006 consisting of 257 categories of images, created based on the Caltech-101 image set. It contains 30608 pictures in total, with 80 to 824 homogeneous pictures per category [173], [174].

The *Oxford Buildings Dataset* consists of 5062 high resolution (1024×768) images collected from "Flickr" by searching for particular Oxford landmarks [175], [176]. The collection has been manually annotated to generate a comprehensive ground truth for 11 different landmarks, each represented by 5 possible queries. The images are taken in different scales, viewpoints and lighting conditions.

The *Paris Dataset* consists of 6412 images collected from Flickr by searching for particular Paris landmarks [177], [178] and is analogous to the previous *Oxford Dataset*.

The PubFig83 and LFW datasets form a new benchmark dataset for open-universe face identification [179], [180]. Based on the realistic scenarios of an automatic search for people in web photos, or tagging friends and family in personal photo albums, the purpose of the dataset is to allow algorithms to find and identify some individuals while ignoring all the others as background, or distractor faces. This mimics many real-world applications where face recognition needs to ignore many background faces that appear in photos, but are not relevant to the user. PubFig83+LFW has 13,002 faces representing 83 individuals from PubFig83, divided into 2/3 training (8720 faces) and 1/3 testing set (4,282 faces). From LFW, 12,066 faces representing over 5,000 images are used as a distractor set.

IX. Conclusions

The future of content-based image retrieval (CBIR) systems seems to lead towards semantic recognition. This paper provides useful insights into how to obtain salient low-level features to facilitate 'semantic gap' reduction, focusing on the differences between CBIR with high-level semantics and traditional systems with low-level features. In addition, current search engines are described.

Test dataset and performance evaluation of CBIR systems are also discussed. We believe that establishing standard benchmark sets and an evaluation model are necessary for objective performance comparison of search algorithms. Moreover, the branch image sets need to be developed in order to receive specialized algorithms.

Based on the ongoing technologies available and the demand for practical applications, many open issues are identified from the system point of view, including a query-language or query-image design, a high-dimensional image feature indexing, semantic similarity, etc.

In order to implement a fully-fledged image retrieval system with high-level semantics we have to integrate salient, low-level feature extraction, effective learning of high-level semantics, friendly user interface, and efficient indexing and classification tools. A CBIR framework providing a more balanced view of all the constituent components is needed.

References

[1] S. Nandagopalan, B. S. Adiga and N. Deepak, "A Universal Model for Content-Based Image Retrieval," *World Academy of Science, Engineering and Technology,* vol. 46, pp. 644-647, 2008.

[2] M. Yasmin, S. Mohsin, I. Irum and M. Sharif, "Content Based Image Retrieval by Shape, Color and Relevance Feedback," *Life Science Journal,* vol. 10, no. 4s, pp. 593-598, 2013.

[3] M. Rehman, M. Iqbal, M. Sharif and M. Raza, "Content Based Image Retrieval: Survey," *World Applied Sciences Journal,* vol. 19, no. 3, pp. 404-412,

2012.

[4] T. M. Lehmann, M. O. Güld, C. Thies, B. Fischer, D. Keysers, K. Spitzer, H. Ney, M. Kohnen, H. Schubert and B. B. Wein, "Content-Based Image Retrieval in Medical Applications," *Methods on Imformatic in Medicine,* vol. 43, pp. 354-361, 2004.

[5] S. Antani, J. Cheng, J. Long, R. L. Long and G. R. Thoma, "Medical Validation and CBIR of Spine X-ray Images over the Internet," in *Proceedings of IS&T/SPIE Electronic Imaging. Internet Imaging VII*, San Jose, C, 2006.

[6] R. K. Srihari, "Automatic Indexing and Content-Based Retrieval of Captioned Images," *IEEE Computer,* vol. 28, no. 9, pp. 49-56, September 1995.

[7] V. Khanaa, M. Rajani, K. Ashok and A. Raj, "Efficient Use of Semantic Annotation in Content Based Image Retrieval (CBIR)," *International Journal of Computer Science Issues,* vol. 9, no. 2, pp. 273-279, March 2012.

[8] C. Carson, S. Belongie, H. Greenspan and J. Malik, "Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 24, no. 8, pp. 1026-1038, Aug 2002.

[9] Y. Rubner, C. Tomasi and L. J. Guibas , "The Earth Mover's Distance as a Metric for Image Retrieval," *International Journal of Computer Vision,* vol. 40, no. 2, pp. 99-121, 2000.

[10] B. Xiao , X. Gao, D. Tao i X. Li, „Recognition of Sketches in Photos," w *Multimedia Analysis, Processing and Communications*, tom 346, W. Lin, D. Tao, J. Kacprzyk , Z. Li, E. Izquierdo i H. Wang , Redaktorzy, Berlin, Springer-Verlag, 2011, pp. 239-262.

[11] G. Chang, M. J. Healey , J. A. M. McHugh i J. T. L. Wang, Mining the World Wide Web: An Information Search Approach., Norwell: Kluwer Academic, 2001.

[12] T. Jaworska, „Object extraction as a basic process for content-based image retrieval (CBIR) system.," *Opto-Electronics Review,* tom 15, nr 4, pp. 184-195, December 2007.

[13] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Internationa Journal of Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[14] D. G. Lowe, "Object Recognition from local scale-invariant features," in *International Conferences on Computer Vision*, Corfu, Greece, 1999.

[15] . C. Leininger , „Fusion d'images : des outils au service des neurochirurgiens," June 2006. [Online]. Available: https://interstices.info/jcms/c_16870/fusion-d-images-des-outils-au-service-des-neurochirurgiens.

[16] M. R. Azimi-Sadjadi, J. Salazar and S. Srinivasan, "An Adaptable Image Retrieval System With Relevance Feedback Using Kernel Machines and Selective Sampling," *IEEE Transactions on Image Processing,* vol. 18, no. 7, p. 1645 1659, 2009.

[17] J. Urban, J. M. Jose and C. J. van Rijsbergen, "An adaptive technique for content-based image retrieval," *Multimedial Tools Applied,* no. 31, pp. 1-28, July 2006.

[18] X. S. Zhou and T. S. Huang, "Relevance Feedback in Image Retrieval: A Comprehensive Review," *ACM Multimedia Systems,* vol. 8, no. 6, pp. 536-544, 2003.

[19] L. Zhang, L. Wang and W. Lin, "Conjunctive patches subspace learning with side information for collaborative image retrieval," *IEEE Transactions on Image Processing,* vol. 21, no. 8, pp. 3707-3720, 2012.

[20] M. M. Rahman, S. K. Antani and G. R. Thoma, "A query expansion framework in image retrieval domain based on local and global analysis," *Information Processing and Management,* vol. 47, pp. 676-691, 2011.

[21] L. Zhang, L. Wang and W. Lin, "Generalized biased discriminant analysis for content-based image retrieval," *IEEE Transactions on System, Man, Cybernetics, Part B - Cybernetics,* vol. 42, no. 1, pp. 282-290, 2012.

[22] L. Zhang, L. Wang and W. Lin, "Semi-supervised biased maximum margin analysis for interactive image retrieval," *IEEE Transactions on Image Processing,* vol. 21, no. 4, pp. 2294-2308, 2012.

[23] L. Wang, W. Lin and L. Zhang, "Geometric Optimum Experimental Design for Collaborative Image Retrieval," *IEEE Transactions on Circuits and System for Video Technology,* vol. 24, pp. 346-359, 2014.

[24] F. Long, H. Zhang and D. D. Feng, "Fundamentals of content-based image retrieval," in *Multimedia Information Retrieval and Management Technological Fundamentals and Applications.*, New York, Sprainger-Verlag, 2003, pp. 1-26.

[25] S. Gould and X. He, "Scene Understanding by labellilng Pixels," *Communications of the ACM,* vol. 57, no. 11, pp. 68-77, November 2014.

[26] J. Yao, S. Fidler and R. Urtasun, "Describing the Scene as a Whole: Joint Object Detection, Scene Classification and Semantic Segmentation," in *The 26th IEEE Conference on Computer Vision and Pattern Recognition*, Providence, Rhode Island, 2012.

[27] L.-J. Li, H. Su, . E. P. Xing and L. Fei-Fei, "Object Bank: A High-Level Image Representation for Scene Classification and Semantic Feature Sparsification," in *24th Annual Conference on Neural Information Processing Systems* , Vancouver, Canada, 2010.

[28] D. M. Wells, A. P. French, A. Naeem, O. Ishaq and R. Traini, "Recovering the dynamics of root growth and development using novel image acquisition and

analysis methods," *Phisiological Transactions of The Royal Society B,* no. 367, p. 1517–1524, 2012.

[29] C. Steger, M. Ulrich and C. Wiedemann, Machine Vision Algorithms and Applications, Weinheim: Wiley-VCH, 2008.

[30] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta and R. Jain, "Content-Based Image Retrieval at the End of the Early Years," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGEN,* vol. 22, no. 12, pp. 1349 - 1380, Dec 2000.

[31] T. Jaworska, „A Search-Engine Concept Based on Multi-Feature Vectors and Spatial Relationship," w *Flexible Query Answering Systems*, tom 7022, H. Christiansen, G. De Tré, A. Yazici, S. Zadrożny i H. L. Larsen, Redaktorzy, Ghent, Springer, 2011, pp. 137-148.

[32] "List of CBIR engines," 2015. [Online]. Available: http://en.wikipedia.org/wiki/List_of_CBIR_engines.

[33] L.-J. Li, C. Wang, Y. Lim, D. M. Blei and L. Fei-Fei, "Building and Using a Semantivisual Image Hierarchy," in *IEEE Conference on Computer Vision and Pattern Recognition* , June, 2010.

[34] F. Wu, Advances in Visual Data Compression and Communication: Meeting the Requirements of New Applications, CRC Press, 2014, p. 513.

[35] J. G. Kolo, K. P. Seng, L.-M. Ang and S. R. S. Prabaharan, "Data Compression Algorithms for Visual Information," in *Informatics Engineering and Information Science*, vol. 253, A. A. Manaf, . S. Sahibuddin, . R. Ahmad , . S. M. Daud and . E. El-Qawasmeh , Eds., Berlin, Springer-Verlag, 2011, pp. 484-497.

[36] N. Sharda, "Multimedia Transmission ober Wireless Sensor Networks," in *Visual Information Processing in Wireless Sensor Networks: Technology, Trends and Applications*, L. Ang, Ed., 2011.

[37] Y. Liu, D. Zhang, G. Lu and W.-Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognition,* vol. 40, pp. 262-282, 2007.

[38] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *IEEE Conference on Computer Vision and Pattern Recognition*, Miami, USA, June, 2009.

[39] W. Niblack, M. Flickner, D. Petkovic, P. Yanker, R. Barber, W. Equitz, E. Glasman, C. Faloutsos and G. Taubin, "The QBIC Project: Querying Images by Content Using Colour, Texture and Shape," *SPIE,* vol. 1908, pp. 173-187, 1993.

[40] V. E. Ogle and M. Stonebraker, "CHABOT: Retrieval from a Relational Database of Images," *IEEE Computer,* vol. 28, no. 9, pp. 40-48, September 1995.

[41] G. Pass and R. Zabith, "Histogram refinement for content-based image retrieval," *IEEE Workshop on Applications of Computer Vision,* pp. 96-102, 1996.

[42] M. Pietikäinen, Ed., Texture Analysis in Machine Vision, vol. 40, World Scientific, 2000.

[43] N. Sebe and M. S. Lew, "Texture Features for Content-Based Retrieval," in *Principles of Visual Information Retrieval*, M. S. Lew, Ed., Springer Science & Business Media, 2013, pp. 50-81.

[44] M. Tuceryan and A. K. Jain, "Texture Analysis," in *The Handbook of Pattern Recognition and Computer Vision*, 2 ed., C. H. Chen, L. F. Pau and P. S. P. Wang, Eds., World Scientific Publishing Co., 1998, pp. 207-248.

[45] S. W. Zucker, "Toward a Model of Texture," *Computer Graphics and Image Processing,* vol. 5, pp. 190-202, 1976.

[46] N. Ahuja, "Dot Pattern Processing Using Voronoi Neighborhoods," *IEEE Transaction on Pattern Analysis and Machine Intelligence,* no. 4, pp. 336-343, May 1982.

[47] R. M. Haralick, "Statistical and Structural Approaches to Texture," *Proceedings of the IEEE,* vol. 67, pp. 786-804, 1979.

[48] . M. Pietikäinen, T. Ojala and D. Harwood, "A Comparative Study of Texture Measures with Classification Based on Feature Distributions.," *Pattern Recognition,* vol. 29, no. 1, pp. 51-59, January 1996.

[49] T. Ojala, M. Pietikäinen and T. Mäenpää, "Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns.," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 7, pp. 971-987, 2002.

[50] M. Pietikäinen, A. Hadid, G. Zhao and T. Ahonen, Computer Vision Using Local Binary Patterns, vol. 40 in Computational Imaging and Vision, Springer Science & Business Media, 2007.

[51] H. Tamura, S. Mori i T. Yamawaki, „Texture features corresponding to visual perception," *IEEE Transactions On Systems, Man and Cybernetics,* tom 8, pp. 460-473, 1978.

[52] R. Sriram , J. M. Francos and W. A. Pearlman, "Texture coding using a Wold decomposition model.," *IEEE Transactions of Image Processing* , vol. 5, no. 9, pp. 1382-1386, 1996.

[53] G. L. Gimel'farb and A. K. Jain, "On retrieving textured images from an image data base.," *Pattern Recognition,* vol. 29, no. 9, pp. 146l-1483, 1996.

[54] A. P. Pentland, "Fractal-based description of natural scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 6, no. 6, pp. 661-674., June 1984.

[55] B. B. Mandelbrot, Fractal Geometry of Nature, New York: Freeman, 1982.

[56] H. E. Hurst, „Long-term storage capacity of reservoirs," *Transactions of the American Society of Civil Engineers,* pp. 770-808, 1951.

[57] S. Ezekiel and J. A. Cross, "Fractal-based Texture Analysis," in *APCC/OECC'99, Joint Conference of 5th Asia-Pacific Conference on Communications (APCC) and 4th Opto-Electronics and Communications Conference (OECC)*, 1999.

[58] J. Millard, P. Augat, T. M. Link, M. Kothari, D. C. Newitt, H. K. Genant, and S. Majumdar, "Power Spectral Analysis of Vertebral Trabecular Bone Structure from Radiographs: Orientation Dependence and Correlation with Bone Mineral Density and Mechanical Properties," *Calcified Tissue International,* vol. 63, pp. 482-489, 1998.

[59] S. Selvarajah and S. R. Kodituwakku, "Analysis and Comparison of Texture Features for Content Based Image Retrieval," *International Journal of Latest Trends in Computing,* vol. 2, no. 1, pp. 108-113, March 2011.

[60] G. M. Haley and B. S. Manjunath, "Rotation-Invariant Texture Classification Using a Complete Space-Frequency Model," *IEEE Transactions on Image Processing,* vol. 8, no. 2, Feb. 1999.

[61] D. Gabor, "Theory of communication," *Journal of the Institution of Electrical Engineers* , pp. 445 - 457, 1946.

[62] T. S. Lee, "Image Representation Using 2D Gabor Wavelets," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 18, no. 10, October 1996.

[63] T. Jaworska, "Point-to-point correspondence into stereo pair of images," Silesian University of Technology, Gliwice, Poland, 2001.

[64] N. Sebe and M. S. Lew, "Wavelet Based Texture Classification," in *Proceedings. 15th International Conference on Pattern Recognition*, 2000.

[65] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE TRANSACTIONS ON COMMUNICATIONS,* Vols. COM-31, no. 4, pp. 532-540, April 1983.

[66] J. L. Crowley, "A representation for visual information," 1987.

[67] I. Daubechies, Ten lectures on wavelets, Philadephia: Society for Industrial and Applied Mathematics, 1992.

[68] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 11, no. 7, pp. 674-693, 1989.

[69] S. Mallat, "Multiresolution Approximation and Wavelet Orthonormal Bases of L2(R)," *Transactions American Mathematical Society,* vol. 315, no. 1, pp. 69-87, 1989.

[70] Y. Meyer, Les ondelettes. Algorithmes et applications, Paris: Armand Colin, 1992.

[71] P. Wojtaszczyk, Wavelet Theory (in Polish), Warsaw: PWN, 2000.

[72] S. Mallat, A wavelet tour of signal processing, Academic Press, 1998.

[73] M. Faizal, A. Fauzi and P. H. Lewis, "Automatic texture segmentation for content-based image retrieval application," *Pattern Analysis and Applications,* vol. 9, p. 307–323, 2006.

[74] R. A. Kirsch , "Computer determination of the constituent structure of biological images," *Computers and Biomedical Research,* vol. 4, no. 3, p. 315–328, July 1971.

[75] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 13, no. 6, p. 583–598, 1991.

[76] O. Basir, H. Zhu and F. Karray, "Fuzzy Based Image Segmentation," in *Fuzzy Filters foe Image processing*, vol. 122, Berlin, Springer, 2003, pp. 101-128.

[77] H. M. Sobel, Multivariate Observations, Wiley, 1984.

[78] J. M. S. Prewitt, "Object Enhancement and Extraction," in *Picture Processing and Psychopictorics,*, B. S. B. S. Lipkin and A. Rosenfeld, Eds., NY, Academic Press, 1970.

[79] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vols. PAMI-8, no. 6, pp. 679-698, 1986.

[80] C. Xu and J. L. Prince, "Snakes, Shapes, and Gradient Vector Flow," *IEEE TRANSACTIONS ON IMAGE PROCESSING,* vol. 7, no. 3, pp. 359-369, March 1998.

[81] R. O. Duda and P. E. Hart, "Use of the HOUGH Transformation to Detect Lines and Curves in Pictires," 1971.

[82] D. Zhang and G. Lu, "Review of shape representation andd escription techniques," *Pattern Recognition,* vol. 37, p. 1 – 19, 2004.

[83] Q. Zhu, . L. Wang, Y. Wu and J. Shi,, "Contour Context Selection for Object Detection: A Set-to-Set Contour Matching Approach,," in *The 10th European Conference on Computer Vision (ECCV)*, Marseille, France, 2008.

[84] S. Abbasi, F. Mokhtarian and J. Kittler, "Curvature scale space image in shape similarity retrieval," *Multimedia Systems,* no. 7, p. 467–476, 1999.

[85] T. B. Sebasian and B. B. Kimia, "Curves vs Skeltons in Object Recognition," in *Proceedings of International Conference on Image Processing*, Thessaloniki, 7-10 Oct. 2001 .

[86] L. Kotoulas i I. Andreadis, „Image analysis using moments," w *Proceedings of 5th International*

Conference on Technology and Automation, Thessaloniki, Greece, 2005.

[87] M. R. Teague, "Image analysis via the general theory of moments," *Journal of the Optical Society of America,* vol. 70, no. 8, pp. 920-930, 1980.

[88] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, 2012.

[89] K. Mikolajczyk and C. Schmid, "Scale & Affine Invariant Interest Point Detectors," *International Journal of Computer Vision,* pp. 63-86, 2004.

[90] T. Tuytelaars and K. Mikolajczyk, "Local Invariant Feature Detectors: A Survey," *Computer Graphics and Vision,* vol. 3, no. 3, p. 177–280, 2007.

[91] F. Perronnin and C. Dance, "Fisher Kernels on Visual Vocabularies for Image Categorization," in *Proceeding Computer Vision and Pattern Recognition*, 2007.

[92] F. Perronnin, J. Sanchez and T. Mensink, "Improving the Fisher Kernel for Large-Scale Image Classification," in *European Conference on Computer Vision, Lecture Notes in Computer Science*, Heraclion, Greece, Sep, 2010.

[93] H. Jegou, M. Douze, C. Schmid and P. Perez, "Aggregating local descriptors into a compact image representation," in *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, 13-18 June, 2010.

[94] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *IEEE International Conference on Computer Vision*, 2005.

[95] E. Rosten i T. Drummond, „Machine learning for high-speed corner detection," w *European Conference on Computer Vision*, 2006.

[96] E. Rublee, V. Rabaud , K. Konolige and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spane, 6-12, Nov, 2011.

[97] M. Brown, R. Szeliski i S. Winder, „Multi-image matching using Multi-Scale Oriented Patches," *Computer Vision and Pattern Recognition,* nr 2, pp. 510-517, 2005.

[98] J.-J. Chen, C.-R. Su, W. E. L. Grimson, J.-L. Liu and D.-H. Shiue, "Object Segmentation of Database Images by Dual Multiscale Morphological Reconstructions and Retrieval Applications," *IEEE Transactions on Image Processing,* vol. 21, no. 2, pp. 828-843, Feb. 2012.

[99] C.-J. Sze, H.-R. Tyan, H.-Y. M. Liao, C.-S. Lu and S.-K. Huang, "Shape-based Retrieval on a Fish Database of Taiwan," *Tamkang Journal of Science and Engineering,* vol. 2, no. 3, pp. 63-173 , 1999.

[100] M. Acharyya and M. K. Kundu, "An adaptive approach to unsupervised texture segmentation using M-Band wavelet transform," *Signal Processing,* no. 81, pp. 1337-1356, 2001.

[101] L. J. Latecki and R. Lakamper, "Application of planar shape comparison to object retrieval in image databases," *Pattern Recognition,* no. 35, pp. 15-29, 2002.

[102] W.-B. Goh and K.-Y. Chan, "A Shape Descriptor for Shapes with Boundary Noise and Texture," in *British Machine Vision Conference*, Norwich, 24 June, 2003.

[103] C. Xu and J. Liu, "2D Shape Matching by Contour Flexibility," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 31, no. 1, Jan. 2009.

[104] J. Mutch and D. G. Lowe, "Object class recognition and localization using sparse features with limited receptive fields," *International Journal of Computer Vision (IJCV),* vol. 80, no. 1, pp. 45-57, Oct 2008.

[105] T. Serre, L. Wolf and T. Poggio, "Object Recognition with Features Inspired by Visual Cortex," in *Proceedings on Computer Vision and Pattern Recognition*, Los Alamos, 2005.

[106] Y. Li and L. G. Shapiro, "Object Recognition for Content-Based Image Retrieval," Dagstuhl Seminar, Leibniz, Austria, 2002.

[107] G. Quellec, M. Lamard, G. Cazuguel, B. Cochener and C. Roux, "Fast Wavelet-Based Image Characterization for Highly Adaptive Image Retrieval," *IEEE Transactions on Image Processing,* vol. 21, no. 4, pp. 1613-1623, April 2012.

[108] V. Castelli i L. D. Bergman, Redaktorzy, Image Databases: Search and Retrieval of Digital Imagery, New York: Wiley, 2002.

[109] U. M. Fayyad and K. B. Irani, "The attribute selection problem in decision tree generation," in *the 10th National Conference on Artificial Intelligence, AAAI*, 1992.

[110] L. Breiman , J. Friedman , C. J. Stone and R. A. Olshen, Classification and Regression Trees, New York: Chapman and Hall, 1984, p. 368.

[111] J. R. Quinlan, "Induction of Decision Trees," *Machine Learning,* vol. 1, pp. 81-106, 1986.

[112] J. R. Quinlan, C4.5: Programs for Machine Learning, San Mateo: Morgan Kaufmann Publishers, 1993.

[113] J. Ylioinas, J. Kannala, A. Hadid and . M. Pietikainen, "Learning Local Image Descriptors Using Binary Decision Trees," in *Proceedings of IEEE Winter Conference on Applications of Computer Vision (WACV 2014),*, Steamboat Springs, CO, USA,, 2014.

[114] B. Bouchon-Meunier and C. Marsala, "Fuzzy decision tree and databases," in *Flexible Query Answering Systems*, T. Andreasen, H. Christiansen and H. L. Larsen, Eds., Kluwer Academic Publisher, 1997, pp. 277-288.

[115] I. Rish, "An empirical study of the naive Bayes

classifier," in *IJCAI-2001 workshop on Empirical Methods in AI*, 2001.

[116] J. D. M. Rennie, L. Shih, J. Teevan and D. R. Karge, "Tackling the Poor Assumptions of Naive Bayes Text Classifiers," in *Proceedings of the 20th International Conference on Machine Learning*, Washington, DC, USA, 2003.

[117] C. Cortes and V. Vapnik , "Support-Vector Networks," *Machine Learning,* vol. 20, p. 273–297, 1995.

[118] L. Wang, Ed., Support Vector Machines: Theory and Applications, Berlin: Springer, 2005, p. 450.

[119] H. Ishibuchi and Y. Nojima, "Toward Quantitative Definition of Explanation Ability of Fuzzy Rule-Based Classifiers," in *IEEE International Conference on Fuzzy Systems*, Taipei, Taiwan, June 27-39, 2011.

[120] H. Ishibuchi and T. Yamamoto, "Rule weight specification in fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems,* vol. 13, no. 4, pp. 428-435, 2005.

[121] K. Nozaki, H. Ishibuchi and H. Tanaka , "Adaptive fuzzy rule-based classification systems," *IEEE Transactions on Fuzzy Systems,* vol. 13, no. 4, pp. 238-250, 1996.

[122] A. Hamilton-Wright and D. W. Stashuk, "Constructing a Fuzzy Rule Based Classification System Using Pattern Discovery," in *Annual Meeting of the North American Fuzzy Information Processing Society*, 2005.

[123] T. Jaworska, "Application of Fuzzy Rule-Based Classifier to CBIR in comparison with other classifiers," in *11th International Conference on Fuzzy Systems and Knowledge Discovery*, Xiamen, China, 19-21.08.2014.

[124] T. Jaworska, "Database as a Crucial Element for CBIR Systems," in *Proceedings of the 2nd International Symposium on Test Automation and Instrumentation*, Beijing, China, 16-20 Nov., 2008.

[125] C.-C. Chang and T.-C. Wu, "An exact match retrieval scheme based upon principal component analysis," *Pattern Recognition Letters,* vol. 16, pp. 465-470, 1995.

[126] D. S. Guru and P. Punitha, "An invariant scheme for exact match retrieval of symbolic images based upon principal component analysis," *Pattern Recognition Letters* , vol. 25, p. 73–86, 2004.

[127] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta and R. Jain, "Content-Based Image Retrieval at the End of the Early Years," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGEN,* vol. 22, no. 12, pp. 1349 - 1380, Dec 2000.

[128] X. Wang, K. Liu and X. Tang, "Query-Specific Visual Semantic Spaces forWeb Image Re-ranking.," in *Computer Vision and Patern Recognition Paper*, 2011.

[129] W. Niblack, M. Flickner, D. Petkovic, P. Yanker, R. Barber, W. Equitz, E. Glasman, C. Faloutsos and G. Taubin, "The QBIC Project: Querying Images by Content Using Colour, Texture and Shape," *SPIE,* vol. 1908, pp. 173-187, 1993.

[130] B. Xiao , X. Gao, D. Tao and X. Li, "Recognition of Sketches in Photos," in *Multimedia Analysis, Processing and Communications*, vol. 346, W. Lin, D. Tao, J. Kacprzyk , Z. Li, E. Izquierdo and H. Wang , Eds., Berlin, Springer-Verlag, 2011, pp. 239-262.

[131] J.-H. Lim and J. S. Jin, "A structured learning framework for content-based image indexing and visual query," *Multimedia Systems,* vol. 10, p. 317–331, 2005.

[132] J. Fauqueur and N. Boujemaa, "Mental image search by boolean composition of region categories," *Multimed Tools and Applications,* vol. 31, p. 95–117, 2006.

[133] T. Jaworska, "Multi-criteria object indexing and graphical user query as an aspect of content-based image retrieval system.," in *Information Systems Architecture and Technology*, L. Borzemski, A. Grzech, J. Świątek and Z. Wilimowska, Eds., Wrocław, Wrocław Technical University Publisher, 2009, pp. 103-112.

[134] . B. Moghaddam, H. Biermann and D. Marg, "Regions-of-Interest and Spatial Layout for Content-Based Image Retrieval," *Multimedia Tools and Applications,* vol. 14, no. 2, pp. 201-210, June 2001.

[135] J. Fauqueur, "Instantaneous mental image search with range queries on multiple region descriptors," Cambridge, UK, Jan, 2005.

[136] N. Rasiwasia, P. J. Moreno and N. Vasconcelos, "Bridging the Gap: Query by Semantic Example," *IEEE TRANSACTIONS ON MULTIMEDIA,* vol. 9, no. 5, pp. 923-938, Aug 2007.

[137] S. K. Candan and W.-S. Li, "On Similarity Measures for Multimedia Database Applications," *Knowledge and Information Systems,* vol. 3, pp. 30-51, 2001.

[138] J. C. Cubero, N. Marín, J. M. Medina, E. Pons and A. M. Vila, "Fuzzy Object Management in an Object-Relational Framework," in *Proceedings of the 10th International Conference IPMU*, Perugia, Italy, 4-9 July, 2004.

[139] F. Berzal, J. C. Cubero, J. Kacprzyk, N. Marín, A. M. Vila and S. Zadrożny, "A General Framework for Computing with Words in Object-Oriented Programming.," in *International Journal of Uncertainty. Fuzziness and Knowledge-Based Systems.*, vol. 15 (Suppl), Singapore, World Scientific Publishing Company, 2007, pp. 111 131,.

[140] W. Plant and G. Schaefer, "Visualization and Browsing of Image Databases," in *Multimedia Analysis, Processing and Communications*, vol. 346,

W. Lin, D. Tao, J. Kacprzyk , Z. Li, E. Izquierdo and H. Wang, Eds., Berlin, Springer, 2011, pp. 3-57.

[141] K. Rodden and K. R. Wood, "How Do People Manage Their Digital Photographs?," in *SIGCHI Conference on Human Factors in Computing Systems*, Ft. Lauderdale, Florida, USA., April 5–10, 2003.

[142] K. Rodden, K. R. Wood, W. Basalaj and D. Sinclair, "Evaluating a Visualisation of Image Similarity as a Tool for Image Browsing," in *IEEE Symposium on Information Visualisation*, 1999.

[143] W. Basalaj, "Proximity visualisation of abstract data," University of Cambridge, Cambridge, 2001.

[144] K. Rodden, „Evaluating similarity-based visualisations as interfaces for image browsing," University of Cambridge, Cambridge, 2002.

[145] C. Faloutsos and K. Lin, "Fast Map: A Fast Algorithms for Indexing, Data-Mining and Visualization of Traditional and Multimedia Datasets," in *ACM SIGMOD international conference on Management of data* , New York, USA, May, 1995.

[146] A. Bursuc and T. Zaharia, "ARTEMIS@ MediaEval 2013: A Content-Based Image Clustering Method for Public Image Repositories," *ACM Multimedia,* pp. 18-19, Oct. 2013.

[147] C. Chen, G. Gagaudakis and P. Rosin, "Similarity-Based Image Browsing," in *Proceedings of the 16th IFIP World Computer Congress, International Conference on Intelligent Information Processing*, Beijing, China, 2000.

[148] Y. Rui and T. S. Huang, "Relevance Feedback Techniques in Image Retrieval," in *Principal of Visual Information Retrieval*, M. S. Lew, Ed., London, Springer, 2001, pp. 219-258.

[149] V. Mezaris, I. Kompatsiaris and M. G. Strintzis, "An ontology approach to object-based image retrieval," in *Proceedings of International Conference on Image Processing ICIP 2003.*, 2003.

[150] A. D. Gudewar and L. R. Ragha, "Ontology to Improve CBIR System," *International Journal of Computer Applications,* vol. 52, no. 21, pp. 23-30, 2012.

[151] C. Doulaverakis, E. Nidelkou, A. Gounaris and Y. Kompatsiaris, "A Hybrid Ontology and Content-Based Search Engine For Multimedia Retrieval," in *Workshop Proceedings in Advances in Databases and Information Systems ADBIS '2006*, Thessaloniki, 2006.

[152] O. Russakovsky and L. Fei-Fei, "Attribute Learning in Large-scale Datasets," in *Proceedings of the 12th European Conference of Computer Vision (ECCV), 1st International Workshop on Parts and Attributes.*, Crete, Greece, 2010.

[153] T. Hofmann, "Probabilistic latent semantic analysis," in *Proceedings of the15th Conference on Uncertainty in Artificial Intelligence*, Stockholm, 1999.

[154] D. M. Blei, A. Y. Ng and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research,* vol. 3, pp. 993-1022, 2003.

[155] L. Fei-Fei and P. Perona, "A Bayesian Heirarcical Model for Learning Natural Scene Categories," in *Computer Vision & Pattern Recognition CVPR*, 2005.

[156] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman and W. T. Freeman, "Discovering objects and their location in images," in *Proceedings of Internationa Conference of Computer Vision*, Beijing, 2005.

[157] L. Zhang, L. Wang and W. Lin, "Generalized biased discriminant analysis for content-based image retrieval," *IEEE Transactions on System, Man, Cybernetics, Part B - Cybernetics,* vol. 42, no. 1, pp. 282-290, 2012.

[158] L. Zhang, L. Wang and W. Lin, "Semi-supervised biased maximum margin analysis for interactive image retrieval," *IEEE Transactions on Image Processing,* vol. 21, no. 4, pp. 2294-2308, 2012.

[159] S. T. Roweis and L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science,* vol. 290, no. 5500, pp. 2323-2326, Dec. 2000.

[160] J. Urban, J. M. Jose and C. J. van Rijsbergen, "An adaptive technique for content-based image retrieval," *Multimedial Tools Applied,* no. 31, pp. 1-28, July 2006.

[161] S.-F. Chang, W. Chen and H. Sundaram, "Semantic Visual Templates: Linking Visual Features to Semantics," in *International Conference on Image Processing, 1998. ICIP 98.*, Chicago, 1998.

[162] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross and K. Miller, "Introduction to WordNet: An On-line Lexical Database," *Communications of the ACM,* vol. 38, no. 11, pp. 39-41 , Nov. 1995.

[163] X. Wang, S. Qiu, K. Liu i X. Tang, „Web Image Re-Ranking Using Query-Specific Semantic Signatures," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* tom 36, nr 4, pp. 810-823, April 2014.

[164] M. Koyuncu and B. Cetinkaya, "A Component-Based Object Detection Method Extended with a Fuzzy Inference Engine," in *Proceedings of the International Conference on Fuzzy Systems Fuzz-IEEE2015*, Istambul, 2015.

[165] Corel comp., "The COREL Database for Content based Image Retrieval".

[166] Z. Yang and C.-C. Jay Kuo, "Learning image similarities and categories from content analysys and relebance feedback," in *Proceedings of the ACM Multimedia Workshops. Multimedia00'*, Los Angeles, CA, USA , Oct 30 - Nov 03, 2000.

[167] the Eastman Kodak Company, [Online]. Available: http://r0k.us/graphics/kodak/.

[168] D.-C. He and A. Safia, "Multiband Texture Database," 2015. [Online]. Available: http://multibandtexture.recherche.usherbrooke.ca/.

[169] D.-C. He and A. Safia, "New Brodatz-based Image Databases for Grayscale Color and Multiband Texture Analysis," *ISRN Machine Vision,* vol. Article ID 876386, pp. 1-14, 2013.

[170] M. Everingham, A. S. Eslami, L. Van Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The PASCAL Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision,* no. 111, p. 98–136, 2015.

[171] M. Everingham, L. Van Gool, C. K. I. Williams, A. Zisserman, J. Winn, A. S. Eslami and Y. Aytar , "The PASCAL Visual Object Classes Homepage," 2015. [Online]. Available: http://host.robots.ox.ac.uk/pascal/VOC/index.html.

[172] L. Fei-Fei, K. Li, O. Russakovsky, J. Krause, J. Deng and A. Berg, "ImageNet," Stanford Vision Lab, Stanford University, Princeton University, 2014. [Online]. Available: http://www.image-net.org/.

[173] G. Griffin, A. D. Holub and P. Perona, "The Caltech 256," California Institute of Technology, Los Angeles, 2006.

[174] G. Griffin, "Caltech256," 2006. [Online]. Available: http://www.vision.caltech.edu/Image_Datasets/Caltech256/.

[175] J. Philbin, O. Chum and M. a. S. J. a. Z. A. Isard, "Object Retrieval with Large Vocabularies and Fast Spatial Matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[176] J. Philbin, R. Arandjelović and A. Zisserman, "The Oxford Buildings Dataset," Department of Engineering Science, University of Oxford, Nov 2012. [Online]. Available: http://www.robots.ox.ac.uk/~vgg/data/oxbuildings/.

[177] J. Philbin, O. Chum and M. a. S. J. a. Z. A. Isard, "Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases," in *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, USA, 23-28 June, 2008.

[178] J. Philbin i A. Zisserman, „The Paris Dataset," Visual Geometry Group, Department of Engineering Science, University of Oxford , 2008. [Online]. Available: http://www.robots.ox.ac.uk/~vgg/data/parisbuildings/.

[179] B. C. Becker, "PubFig83 + LFW Dataset," 2015. [Online]. Available: http://www.briancbecker.com/blog/research/pubfig83-lfw-dataset/.

[180] B. C. Becker and E. G. Ortiz, "Evaluating Open-Universe Face Identification on the Web," in *CVPR 2013, Analysis and Modeling of Faces and Gestures Workshop.*, Portland, Oregon, USA, 23-28 June, 2013.

[181] The Moving Picture Experts Group, "MPEG," [Online]. Available: http://mpeg.chiariglione.org/. [Accessed 2015].

[182] M. Flickner, H. Sawhney, W. Niblack , J. Ashley, Q. Huang, B. Dom, M. Gorkani , J. Hafner, D. Lee, D. Petkovic, D. Steele and P. Yanker , "Query by Image and Video Content: The QBIC System," *IEEE Computer,* vol. 28, no. 9, pp. 23-32, September 1995.

[183] B. B. Mandelbrot and J. W. Van Ness, "Fractional Brownian Motions, Fractional Noises and Applications," *SIAM Review,* vol. 10, no. 4, pp. 422-437, October 1968.

[184] A. Kundu and J.-L. Chen, "Texture classification using QMF bank-based subband decomposition," *CVGIP: Graphical Models and Image Processing,* vol. 54, no. 5, p. 369–384, 1992.

[185] R. Datta, J. Li and J. Z. Wang, "Content-Based Image Retrieval - Approaches and Trends of the New Age," in *Multimedia Information Retrieval (MIR '05)*, Singapour, 2005.

[186] "Fast Wavelet-Based Image Characterization for Highly Adaptive Image Retrieval," *IEEE Transactions on Image Processing,* 2012.

[187] D. Eads, D. Helmbold and E. Rosten, "Boosting in Location Space," Santa Cruz, 2013.

[188] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack and D. Petkovic, "Efficient and Effective Querying by Image Content.," *Journal of Intelligent Information Systems,* vol. 3, pp. 231-262, 1994.

[189] J. Philbin, O. Chum and M. a. S. J. a. Z. A. Isard, "Object Retrieval with Large Vocabularies and Fast Spatial Matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.