Unsupervised image categorization
Gunther Heidemann*
Neuroinformatics Group, Bielefeld University, P.O. Box 100131, D-33501 Bielefeld, Germany
Received 29 January 2004; received in revised form 5 May 2005; accepted 5 May 2005

Abstract
Large image collections require efficient organization and visualization. This paper describes an approach to establish image categories automatically by unsupervised learning. The method works free of context and previous knowledge: in a first stage, features are formed automatically, then images are clustered to form categories. The human database designer has to decide only whether a category is useful or too inhomogeneous from a high level point of view. To collect images that cannot be categorized automatically, an additional 'miscellaneous' category exists. Categories are visualized by displaying the most typical image(s) of the categories as thumbnails. The main benefit of the approach is that it deals with color and shape in a unified way on a local scale, combined with the advantages of histogram techniques on the global scale. To judge results, an evaluation scheme which is adequate for the task of categorization is proposed.

Keywords: Image categorization; Image retrieval; Image indexing; Salient points; Interest points; Object recognition; Unsupervised learning; Vector quantization; Color features

1. Introduction
The upcoming of large photo collections—both commercial and on the Internet—has caused great research efforts to handle such quantities of images automatically. Most effort is directed towards content based image retrieval (CBIR), with the aim to find similar images to given query images (for a survey see e.g. [56,68,67]). The problem of such systems, however, is the need to supply suitable query images—which requires again to search a database. Therefore, the much less investigated field of efficient image browsing becomes increasingly important, i.e. the development of tools that allow to get an overview of large photo collections.

Up to now, image collections like Corel [9] or ArtExplosion [43] are mostly sorted into heuristic categories and are visualized in browsers by thumbnails. Attempts have been made to improve image browsing, e.g. investigation of zoomable interfaces [8,5]. Browsing systems, however, rely on a one-to-one correspondence of images to thumbnails and are therefore bound to hit the limits of the users perceptual capacity for large databases. Grouping images by similarity seems to improve visual accessibility [48] and thus exploits human capabilities better. But the problem remains that an ‘exhaustive’ visualization of the complete image set is infeasible for large collections. The only way out is visualization of entire categories by one or a few representative images.

Image categorization is still mostly carried out by human database designers, a proceeding which requires not only large effort but becomes impossible for steadily growing databases like the Internet. Moreover, humans have difficulties in consistently applying a single system of categorization criteria, especially, when several categorization methods appear equally plausible. So automatic categorization is desirable, allowing a fast sorting based on comprehensible mechanisms.

1.1. Aim of this contribution
This contribution describes an approach for automatic image categorization, where ‘automatic’ includes establishing the categories, i.e. categories are not pre-defined but found by unsupervised learning. This procedure requires a measure of image similarity. In principle, possible measures range from low level features like the color distribution up
to semantic contents. However, up to date a semantic understanding of image contents is far beyond the range of technique. Therefore, the only hope is to define low level similarities such that the resulting categories ‘mimic’ semantical structure to a certain extent.

This paper proposes a similarity measure based on interest points (IPs), i.e. not an entire image is evaluated but only regions with an increased probability to yield relevant information. From such regions sparse feature vectors are formed, which allow to judge image similarity. In this feature extraction, only the mechanism of IP-detection is hard coded, whereas the features are formed by unsupervised learning from sample images.

Based on this similarity measure, categories are formed by clustering the feature vectors of the entire database. The only interaction with a human ‘knowledge designer’ is required after clustering, when inhomogeneous or unsatisfying categories are sorted out. So, application of the system is restricted to categories where feature similarity correlates with semantic similarity—the philosophy being to do the possible instead of trying the impossible.

### 1.2. Organization of the paper

The following section gives an overview of the approach for scene categorization and its motivation. Then the system is described in more detail in Sections 3–5; Section 3 motivates the use of edge-and corner detection and gives an overview of suitable algorithms. Section 4 introduces the feature formation approach, which is based on vector quantization of image patches centered at the detected IPs (Section 4.1). To reduce visual complexity, in a previous processing step the image patches are rotated to achieve a common alignment (Section 4.2). The second adaptation stage which forms the actual categories from image features is described in Section 5. The database for testing the system is outlined in Section 6, where results are discussed qualitatively. A quantitative evaluation with tests for correctness and coverage follows in Section 7, before the concluding discussion and outlook (Section 8).

### 2. System overview

This section gives a system overview and discusses the relations to previous research.

#### 2.1. Scene categorization using adaptive features

The aim of the approach is to sort a database into categories about which no previous knowledge is available. Each category is represented by one (or optionally some) images and comprises a set of related images. In addition, a ‘miscellaneous’ category is introduced to summarize all images which cannot be sorted into the others.

Figs. 1 and 2 show the process of category formation: from a gray-level version interest points (IPs) are generated on each image. An IP indicates saliency of its near surroundings. Therefore, IP-centered windows of a fixed size are extracted. All windows are then analysed for their intensity-distribution using steerable filters, so they can be rotated to a common alignment (Fig. 2). The reason for this procedure becomes clear in Section 4.2. Fig. 1 illustrates the further processing: in the feature formation stage (above), the rotationally aligned windows are clustered by the vector quantizer described in Section 4.1 in pixel space. To form categories (below), the obtained \( N_f \) different reference vectors serve to construct \( N_f \)-dimensional feature vectors as a robust image representation: each rotated window gives a contribution to the one component of the feature vector which corresponds to the best match reference vector. The details of this procedure are given in Section 4. Finally, the feature vectors are clustered to categories. The best match feature vector of a category cluster becomes the representative of the category.

#### 2.2. Relations to previous research

For categorization and retrieval, images must be represented such that feature distances coincide with perceived distances. In CBIR, global features which attempt to represent the entire image are one of the standard methods. An example is color indexing, which goes back to the work in object recognition of Swain and Ballard [60], an approach that uses a global color histogram for image characterization and can be made quite robust against illumination variation [16]. Related methods compute the global color moments [59]. However, while global features yield good characterizations of isolated, segmented objects (as originally intended in [60]), they are inappropriate for the heterogeneity of natural scenes: the number of pixels covered by a scene constituent (e.g. sky) may strongly differ among photos showing essentially the same scene.

As a consequence, local descriptors have become popular, the idea being that areas in an image are not equally relevant but attention should rather be given to the most interesting or ‘salient’ ones. Schmid and Mohr [51] propose to extract local grayvalue invariants at interest points. Tian et al. [61] use a multi-resolution interest point detector based on Haar-wavelets [34] in the context of CBIR.

So far, most works in CBIR treat color and shape as separate features: while shape information is discarded in color histograms [60], color sets [57] or color moments [59], only gray values are exploited to obtain shape features. Well known methods are, e.g. the use of gray level edges [71], elastic deformation matching [12] or gray value invariants [51,65]. For an overview on shape for CBIR see [38].

Attempts to combine shape and color in a common, unified feature are rare and mostly concentrate on color texture. A CBIR system that relies on spatio-chromatic
features is proposed by Tieu and Viola [62]. Images are convolved with small filter kernels forming center-surround filters and bar-and-edge filters in a hierarchical scheme to obtain a large number of highly specific feature maps. Convolutions are performed on the RGB-channels. The system for categorization proposed here and the CBIR system of [62] have the common property of using spatio-chromaticity, but differ in that [62] uses pre-defined, not learned features, and that the entire image is evaluated without the use of methods for focus-of-attention.

Qiu et al. [44] study several types of color histogram descriptors in the context of CBIR, some of which also incorporate spatial color features. The results indicate that the combination of histograms with spatio-chromatic features is more successful than mere histograms. E.g. layered color indexing uses separate color histograms for different spatial frequency—i.e. there are different histograms for different ‘surface granularities’ [45]. Though this method still discards a great deal of the spatio-chromatic pattern, it clearly outperforms a conventional color histogram. Best results are achieved by the spatial color features of MPEG-7 [35].

For video mining, Sivic and Zisserman [55] use the spatial configurations of several local descriptors (including color). This approach is complementary to the one proposed here: in [55], the relative position of descriptors is a major source of information, whereas the present paper discards descriptor locations and sums up how often a certain type of descriptor appears.

A system that relies on color shapes of several scales is presented in [20], which uses local principal component analysis of image patches centered at interest points for feature extraction. However, the approach is not aimed at a categorization of scenes, rather, it concentrates on isolated objects to become independent of the particular scene. In other words, an approach rooted in view based object recognition [42,39] was applied for CBIR.

Fig. 1. Feature formation and subsequent category formation. Above: From the training images, IP-centered windows are extracted. To reduce visual complexity, the windows are rotated to a predefined oriented direction. Vector quantization in pixel space for the entire window set leads to ‘prototypic windows’ (Figs. 5 and 6) represented by the reference vectors. Below: Feature vectors are assigned to images in the way that each window contributes to one component of the feature vector, which corresponds to the best match reference vector (in the way described in Section 4). Image categories are formed by a second vector quantization, now in feature space.

Fig. 2. Visual complexity reduction by window rotation. For each of the extracted windows, the overlap with a steerable filter pair (first derivatives of a Gaussian) is computed on the intensity image. From the filter responses, the local gradient and its oriented direction $\alpha \in [-\pi, \pi]$ is computed after Eq. (4). The window is then back-rotated by $-\alpha$ to obtain a unified alignment. See Section 4.2.
An interesting approach to overcome the so far heuristic hunt for low level image distance measures can be found in [40]. Here, distance measurement in the field of color texture is derived from psychophysical experiments and applied to CBIR. However, the method is neither foveated, nor does it treat color and shape in a unified representation. A work that experimentally investigates the helpfulness to users for similarity based image groupings is presented in [48], the underlying metric based on the spatial arrangement of color regions is described in [47].

The approach proposed in this contribution tries to combine the benefits of local descriptors, color, and shape. It differs from CBIR systems in the way that image features are not used to find images similar in terms of features to a given query image, instead, first features themselves are formed from all images of the database, then categories are formed. So not queries will be answered, but a set of category representatives is produced. In other words, instead of allowing the user to make arbitrary queries—which might be unanswerable—the system shows the set of ‘query’ images (the representatives) for which adequate results (the category members) can be found.

Closest in thought to this approach is the PicSOM system proposed by Laaksonen et al. [28], which uses tree-structured self-organizing maps [27] to organize a database by similarity criteria. From an initial collection of thumbnails, query images can be chosen, then the system searches the tree for related images. From the resulting set, again a query can be defined. The structure of the system proposed here is simpler and intended to be more intuitive as it does not confront the user with the necessity to formulate queries by designing image sets, but presents categories in a browser-like fashion as folders which contain all images of the category—a solution most users are familiar with. Further refinement into a tree of sub-folders is possible. The main advantage of this simpler solution is that the user knows where he or she is currently searching in the database.

Other attempts to form categories automatically based on image features are rare. The RAGMD algorithm [37] uses unsupervised learning for category formation, however, the degree of self-organization is less than in the system proposed here, because the features (Gabor filters) are predetermined and adapted to the task. This is possible because categorization is not intended for unconstrained images but for the special case of texture (VisTex texture database [69]). In contrast, the present work forms both features and categories automatically.

An ambitious line in the field of CBIR tries to bridge the ‘semantic gap’ between the signal level and the semantic level by the introduction of symbolic image descriptions. In [64], an ‘ontological’ language is introduced for query formulation in natural terms as an alternative to queries by sample images. Queries can be made for combinations of pre-defined scene constituents obeying geometrical constraints, e.g. ‘tarmac in bottom half’. A system that learns the connection between words and segmented image regions is proposed in [4], allowing automatic annotation of both complete images and regions. For category formation, semantic descriptions of this kind would be very valuable. However, both approaches still depend on human effort and world knowledge as they rely on labeled sample regions, moreover, only such scene constituents can be annotated which can be found on the low level by segmentation techniques. For these reasons, category formation on the semantic level was not considered for the present paper.

3. Window extraction

As the image features are formed from a set of extracted windows, a key module of the system is the detection of interest points (IPs). The term ‘interest point’ should not be misunderstood in the way that a single pixel is of particular interest, rather, an IP represents the saliency of the surrounding area from which it was computed. The major advantage of IPs is that they are generated free of context and can be used to start the processing flow. Applications range from active vision [6,11,3] over object recognition [46] to CBIR [65,61]. For the purpose of image categorization, corners and edges appear particularly well suited.

The idea of using feature vectors from corner-and edge-centered windows for image similarity measurement is that such windows yield most information: the transition from one (approximately) uniform region to another carries information about both. While the shape and color of homogeneous regions vary with viewpoint and illumination conditions, the corner and edge points are much more stable since the transition as such remains. Another important property of corner-and edge-centered windows is illustrated in Fig. 3: though the scale is subject to a large change, a (small) corner-or edge-centered window will comprise the same structure. Even more, as described in Section 4, it is possible to construct a robust feature extraction to which shape details like the precise region partitioning around a corner do not matter, so natural patterns as in Fig. 3 become manageable. A related idea which exploits the robustness of features in the neighborhood of edges was proposed by Gevers and Smeulders [17], who search for color edges and evaluate the hue values on opposite sides.

Originally proposed by Moravec [41], many corner- and edge detection algorithms rely on discrete approximations of the auto-correlation function of the signal [19,14,63]. Alternatives are morphological operations [29], evaluation of contours [2,25,53], or wavelet-based methods [32].

In this contribution, the Plessey detector will be used, which is also known as Harris detector (Fig. 4). It was introduced by Harris and Stephens to improve the earlier version of [41] and can be considered the ‘canonical’ solution. The Plessey detector could be shown to perform...
better than other, related approaches \cite{10,14,24,25} by Schmid et al. \cite{52}. Though it is still not optimally stable \cite{72}, there are three reasons to use the Plessey detector instead of a more complicated method: (i) it is structurally simple and intuitive, widely used and thus easy to reproduce; (ii) it fulfills the key criteria for the task, i.e. stability against variation of viewpoint, illumination direction and -spectrum, and scale \cite{52,21}; (iii) in the same works it could be shown that regions indicated by the Plessey detector are indeed highly salient in the sense that they differ significantly sets of randomly chosen image patches. In addition, it could be shown in \cite{22} that the saliency of patches found by the Plessey detector is not limited to the immediate neighborhood of the actually evaluated group of pixels but outreaches this area significantly. Therefore, the Plessey detector is able to detect large salient structures by evaluating a small ‘aperture’—a key property in IP-detection. The Plessey detector is briefly described in Appendix A.1.

An equally good candidate for corner- and edge detection is the SUSAN detector \cite{58}, which does not rely on the computation of derivatives. However, though part of the IPs found by SUSAN do differ from the Plessey-IPs, the overall representation of the image is comparably good. For SUSAN, the tests described in Section 7 did not yield significantly different results. Therefore, SUSAN is not considered in the following.

4. Feature formation

To each image $I$, a feature vector $\vec{f}(I) \in \mathbb{R}^{N_i}$ is assigned. $\vec{f}(I)$ is composed from contributions of all windows extracted from $I$. Section 4.1 describes the feature formation first without window rotation, Section 4.2 then motivates...

---

**Fig. 3.** Corners and edges as such are unaffected by scale or viewpoint.

---

**Fig. 4.** Examples for interest point detection using the Plessey detector. From the original, a continuous valued saliency map $M_p(x, y)$ is computed. Interest points are detected as the $N_{wp}$ highest maxima of this map.
this additional step. Section 4.3 explains how the clustered windows are used to obtain feature vectors.

4.1. Window clustering

From each image $I_i$ of the database $D = \{I_1, \ldots, I_{N_l}\}$ the $N_W$ windows centered at the $N_W$ highest maxima of $M_0$ are extracted. The window size is fixed to $21 \times 21$ pixels, while typical image sizes of the sample database are about $600 \times 400$. The RGB-values of each window form a vector $\tilde{w} \in \mathbb{R}^d$, where the dimension is $d = 3 \times 21^2 = 1323$. To obtain prototypes of typical windows which can serve as features, for the whole dataset $\mathcal{Q}$ of windows (i.e. all windows of all images), vector quantization (VQ) is carried out as a standard method for clustering and data mining [7,26].

VQ represents a given $d$-dimensional data set $\mathcal{Q}$ by a set $A$ of $N_l = |A|$ different reference vectors $A = \{\tilde{r}_1, \ldots, \tilde{r}_{N_l}\}$, $\tilde{r}_i \in \mathbb{R}^d$, which are of the same dimensionality $d$ as the data. Various algorithms exist to obtain $A$ from $\mathcal{Q}$ such as K-means [33] or the Neural-Gas [36]. After computation of $A$, the reference vectors can be used as a compressed representation of $\mathcal{Q}$, provided $N_l < |\mathcal{Q}|$. Compression can be realized by representing a data vector $\tilde{w} \in \mathcal{Q}$, by the best match reference vector $\tilde{r}_k$ which can be found in $A$:

$$\tilde{r}_k(\tilde{w}) = \arg \min_{\tilde{r}_i, i=1,\ldots,N_l} \| \tilde{w} - \tilde{r}_i \|, \quad (1)$$

where $\| \cdot \|$ denotes a distance measure (here the Euclidian). The effect of compression is achieved since several data vectors are represented by one reference vector. In principle, the step of compression is carried out independently of finding $A$, but since compression is one of the major applications of VQ, most VQ algorithms are aimed at minimizing the mean square reconstruction error over the given dataset.

For the current task, the Activity Equalization VQ (AEV) algorithm [23] is applied. In short, AEV is based on the winner takes all rule with an additional supervising component which explicitly counts codeword access frequencies. Under-utilized reference vectors are re-positioned by this supervising component such that approximately equal access frequencies are achieved for all reference vectors. By this means the problem of codeword under-utilization [18] can be avoided, a related idea is realized in [1]. For details see the original work [23].

The result of VQ over the database described in Section 6 is shown in Fig. 5 for $N_l = 144$: The reference vectors are prototypes of edges and corners selected by the Plessey detector. Most show clearly either edges or corners under different angles. For corners, also different aperture angles can be observed. In addition, the dominant color transitions are represented.

4.2. Visual complexity reduction

As visible in Fig. 5, several reference vectors represent essentially the same shape and color but differ by rotation. This is understandable since there is no dominant orientation of edges or corners in the original images. So the available reference vectors have to cover the entire angular range. In principle, increasing $N_l$ would provide a better angular representation. However, the representation would still remain discontinuous, an additional problem is the huge computational effort. A better solution is an rotational alignment of the extracted windows before VQ is carried out. Alignment means that all windows are rotated in the image plane to an arbitrary but fixed angle, e.g. the (oriented) horizon. So a feature is needed which can detect the deviation of a given window from the defined common oriented direction in general.

From the full color information, calculation of oriented direction is not straightforward since for the three-component color vector a spatial gradient does not exist. The color tensor does allow only computation of direction $\alpha \in [-\pi/2, \ldots, \pi/2]$ but not oriented direction $\alpha \in [-\pi, \ldots, \pi]$ [70]. For a compact survey on the color tensor see also [66]. Therefore, only gray value information can be exploited, which is, however, sufficient in the present case because most of the IP-centered windows exhibit a dark and a bright part.

To obtain oriented direction $\alpha \in [-\pi, \ldots, \pi]$ from the intensity gradient measured on the scale of a window, a natural and computationally simple method is the use of steerable filters. The first derivatives of a Gaussian $G$ with width $\sigma$

$$G(x, y) = \exp \left( -\frac{x^2 + y^2}{2\sigma^2} \right), \quad (2)$$

which are denoted by $G'$

$$G'(\phi = 0) = \frac{\delta G}{\delta x} = -\frac{x}{\sigma^2} G, \quad \quad G'(\phi = \frac{\pi}{2}) = \frac{\delta G}{\delta y} = -\frac{y}{\sigma^2} G \quad (3)$$

are a suitable filter set\(^1\) because they are prototypic ‘edge filters’, as visible in Fig. 2. E.g. $G'(\phi = 0)$ responds most when it is centered on an IP which is located on a vertical edge of intensity. The responses of $G'(\phi = 0)$ and $G'(\phi = \pi/2)$ together are sufficient to find out the intensity gradient on an arbitrary point because they are the basis of a rotationally steerable filter [15].

Rotational steerability means that the derivative $G'(\phi)$ in an arbitrary direction $\phi$ can be obtained as a linear combination of $G'(\phi = 0)$ and $G'(\phi = \pi/2)$, which serve as

\(^1\) The normalization factor can be omitted in Eq. (3).
basis functions. Similarly, when $G'(\phi = 0)$ and $G'(\phi = \pi/2)$ are applied to an image patch, from the corresponding filter responses (overlaps) $A(\phi = 0)$ and $A(\phi = \pi/2)$ the response $A(\phi)$ for a filter in arbitrary direction $\phi$ can be calculated. In particular, the direction $\alpha$ for which the maximum filter response $\hat{A}$ is reached is given by

$$\alpha = \text{atan2}(A(\phi = \pi/2), A(\phi = 0)), \quad (4)$$

where $\alpha \in [-\pi, \ldots, \pi]$, and $\text{atan2}(y, x)$ is the C-style notation for computation of the principal value of the arctan of $y/x$, using the signs of both arguments to determine the quadrant of the return value. So, $\alpha$ is the oriented direction in which the window is most similar to a Gaussian derivative and $\hat{A}$ is a measure for this similarity.

The process of rotation detection and ‘back-rotation’ is depicted in Fig. 2: each window is rotated by $-\alpha$ to achieve a common rotational alignment. Fig. 6 shows results of VQ for rotated windows: all reference vectors represent now edges or corners with the brighter side to the right. The shapes and colors are similar to Fig. 5, however, as the same number $N_f$ of reference vectors is not forced to deal with the rotational degree of freedom any more, a better representation of shape and color variation is possible. So e.g. a greater variety of aperture angles for corners can be observed in Fig. 6.

### 4.3. Building feature vectors

The feature vector $f$ of an image $I$ is built in the following way:

1. Extract windows $\hat{w}_1, \ldots, \hat{w}_{N_f}$ from $I$.
2. Set all components of $f$ to 0: $f_k = 0 \ \forall \ k = 1, \ldots, N_f$.
3. For each $\hat{w}_i$ do
   - Rotate $\hat{w}_i$ to obtain $\hat{w}_i'$.
   - Find best match reference vector:
     $$\hat{r}_k = \arg \min_{\hat{r}_k} ||\hat{w}_i - \hat{r}_k||.$$
   - Compute distance $D$ between $\hat{w}_i'$ and $\hat{r}_k$:
     $$D = ||\hat{w}_i' - \hat{r}_k||.$$
   - Add up the contribution of $\hat{w}_i'$ to the component of $f$ corresponding to $\hat{r}_k$:
     $$f_k \leftarrow f_k + I(D) \quad (\text{see below for } I(\cdot)). \quad (5)$$
   - end
The idea is that \( f_i \) should reflect to what extent a feature is present in the image. To be more precise, \( f_i \) has to indicate (a) how often reference vector \( r_i \) was best match and (b) whether these best matches were ‘close hits’ or not. Therefore, the weighting function \( G(.) \) for the distance \( D \) is chosen to be a mirrored and scaled Fermi function:

\[
G(D) = \frac{2}{1 + \exp(\gamma \cdot D)}, \quad \gamma > 0. \tag{6}
\]

Fig. 7(a) shows \( G(D) \) for \( \gamma = 1 \). The contribution of a window is maximal (\( \Gamma = 1 \)) for an exact match (\( D = 0 \)), and tends to zero for large \( D \): \( \Gamma \to 0 \) for \( D \to \infty \). So both the influence of very small and very large \( D \) is attenuated.

The parameter \( \gamma \) must be chosen according to the \( D \)-values that actually occur. Fig. 7(b) shows a typical histogram. A small value of \( \gamma \) will allow even large \( D \)-values to give a contribution, whereas a large \( \gamma \) excludes all but the close hits. In practice, a good choice is adapting \( \Gamma(D) \) such that the medium value of \( \Gamma = 0.5 \) is exceeded only by the top (i.e. smallest) 20% of the \( D \)-values. I.e. \( D_0 \) is defined as the threshold distance for which lower values are achieved only by 20% of all IPs. Then \( \gamma \) follows from the condition \( \Gamma(D_0) = 0.5 \) as:

\[
\gamma = \frac{\ln(3)}{D_0}. \tag{7}
\]

\( D_0 \) is computed numerically from the distance histogram (Fig. 7(b)). For this choice, most feature vectors are ‘sparse representations’—the term should, however, not be confused with the field of sparse coding [54]. Fig. 7(c and d) shows two typical feature vectors, which exhibit substantial values only in a few components and are highly specific.

4.4. Discussion of feature vectors

As discussed in Section 2.2, to date the combination of color and form features is rare: though color is extensively applied in CBIR, it is seldom treated together with form in a unified approach. The benefit of the method proposed here is the combination of color and shape on a local scale but discarding spatial information on the global scale. So, typical local properties are captured both in color and shape, but the unforeseeable distribution in natural, cluttered scenes has no influence, because on the large scale, color and shape features are treated in a histogram like fashion. Therefore, the method is best suited for the representation of images that show one or few features repeatedly, e.g. vegetation with typical corners or edges of green leaves. As
such evidence for a certain structure is summed up in a single component of $\hat{f}$, redundancy and stability is gained.

To test the influence of the color space, the calculations presented in Sections 6, 7 were also carried out for the perceptually uniform CIE—$L^*a^*b^*$ space. This representation appears particularly well-suited since its metric corresponds better than RGB to color differences perceived by humans, and keeps the same Euclidian form (e.g. [13]). But though the resulting categories are slightly different, no advantage could be made out for either representation, so the RGB-space was kept.

In principle, other color spaces could yield a representation which is more robust against variations of lighting spectrum. So, e.g. a beach in daylight and at sunset could be sorted into the same category. This was, however, not attempted for two reasons: (i) pixel-based color transformations can usually not account for real changes of illumination in natural images. To achieve color constancy, elaborate algorithms exploiting the color distribution of the complete image are necessary (e.g. [31,30,49]). Such methods could in principle be applied for preprocessing. However, (ii) it has to be questioned whether it is adequate to do image categorization in an object recognition like fashion: is the beach really the object of interest, that should be recognized under all circumstances, or is it the situation ‘daylight’ versus ‘sunset’? So as long as it is not clear on the semantic level for which aspects discrimination or invariance, respectively, are required, tuning of low level features does not appear sensible.

5. Category formation

The feature vectors $\hat{f}(I)$ of all images are clustered in a second VQ step using $N_C$ reference vectors (Fig. 1, below, right, and Fig. 9). So a maximum of $N_C$ categories can be formed. However, the number of actually usable, ‘labeled categories’ $N_L$ is usually smaller. The reference vectors of the second VQ have to be sorted by a human expert such that they represent either labeled categories $L_i$, $i = 1, \ldots, N_L \leq N_C$ or that they become part of the ‘miscellaneous category’ $M$.

A reference vector represents a labeled category $L_i$ if the images within its Voronoi tessellation cell are mostly ‘coherent’ from a semantical point of view. In this case, the reference vector gets a class label, the best match image becomes the visual representative. In contrast, images for
which no common topic can be identified fall in the miscellaneous category \( M \), such reference vectors have \( M \) as a common class label. So to each \( L_i \), belongs only one reference vector, whereas \( M \) may be represented by several reference vectors. By splitting the set \( \left\{ \vec{r}_i \right\} \) into \( \{L_i\} \) and \( M \), the automatic category formation is restricted artificially to those categories which express themselves in terms of feature clusters—the only possible solution so far. \( M \) may be further sorted by other mechanisms or manually.

The question arises whether the splitting procedure can be done automatically. This can be realized by defining a priori a minimum distance \( R \) in feature space: it can be checked whether feature vectors \( \vec{f} \) are within a hypersphere with radius \( R \) around the corresponding reference vectors. If a certain minimum fraction of the \( \vec{f} \) within a Voronoi tessellation cell is also within the sphere, then the category can be considered ‘coherent’ and is accepted. By this means, large Voronoi tessellation cells with many images close to the borders can be sorted out (\( M \)).

Though this method would further automatize categorization by ‘self-restriction’ of the system to homogeneous categories, it is not applied here, because it is (again) based only on feature similarity. Human knowledge is more useful and easy to apply at this processing stage. Therefore, this method was not pursued.

6. Application and qualitative results

6.1. Image material

The method outlined in the last sections was tested for \( N_f = 10000 \) images from the ‘photos’ section of the ArtExplosion® 600000 photo gallery [43]. The only restriction in the selection of images was to leave out highly specialized topics: e.g. ‘fireworks’, a category of the textual indexation, was omitted since it is both too special and too easy to detect. From each image, \( N_w = 40 \) windows of size \( 21 \times 21 \) were extracted and clustered using \( N_l = 144 \) reference vectors. The corresponding 144-dimensional feature vectors were then clustered in the second VQ with \( N_c = 100 \) reference vectors to obtain categories.

6.2. Discussion of categories

Fig. 8 shows examples of the resulting categories: the image in the red frame is the best match, the following four images are ‘good’ category members, i.e. images for which a sensible connection to the best match exists on the semantical level. This row is indicated as ‘True Labels’ (TL). The second row of each category shows ‘False Labels’ (FL), i.e. images which make no sense in connection with the best match.

The images illustrate both the benefits and problems of the approach. The first category (‘buildings/city’) is well suited for the edge and corner based approach in connection with rotational alignment and color: since there are lots of sharp transitions from concrete color to either dark (mostly windows) or sky color, this type of primary reference vector \( \vec{r} \) is the most activated in this category. In Fig. 6 these \( \vec{r} \) are e.g. (3,4), (3,6), (5,4), (9,6) (11,8) (row, column). This category is one of the most homogeneous, there are few outliers. The problem can be seen, however, in the second row: while the mountain is clearly similar only in features (stone/sky), the window, power pole and interior are also related semantically, which makes separation into ‘good’ and ‘bad’ category members difficult.

In the second category, IPs were mainly centered on transitions from bright yellow to dark. The category makes sense (‘cities at night’), however, the same problem as above occurs - the line between True and False Labels is difficult to draw, because a broader semantic category like ‘lights in the night’ would include also the lower row.

The third category illustrates another property of the approach: since the detection of IPs is not ‘foveated’ towards a particular object, the image of the child is sorted into the category ‘leaves’ because of the background. It is difficult to judge whether this is correct.

The last category shows part of an \( M \)-reference vector: though the category makes sense from a low level point of view (blue-dark transitions), it would be hardly useful and was sorted out.

The discussion of the features determining categories (concrete to sky, yellow on dark, etc.) was restricted intentionally to simple, intuitive patterns and should not mislead to the impression that the relation between features and categories is always simple and one-to-one. E.g. the categories ‘water’ and ‘animals with fur’ (both above the \( M \)-category) are based on compositions of many highly complex features.

7. Evaluation

For evaluation of a categorization system, the established methods of CBIR are not applicable. In CBIR, the results of queries can be characterized by three quantities: the set of returned images, the set of images that should have been returned, and the rest. The success of sample queries can be expressed in terms of precision-recall diagrams [50,56]. In contrast, for categorization evaluation more effort is needed. Since categories are not a priori defined but emerge, categories have to be checked for coverage of the database and correctness. Section 7.1 introduces quantities to characterize a given categorization, which are applied in Section 7.2 for evaluation.

7.1. Characterization of categorization

To quantify the success of categorization, all images were manually sorted into five classes (see also Fig. 9):
† TL: True Label, image assigned correctly to a labeled category ($L_i \rightarrow L_i$)
† FL: False Label, image that belongs to a labeled category but was assigned to the wrong labeled category ($L_i \rightarrow L_j, i \neq j$)
† ML: M-Label, image that belongs to M but was assigned to a labeled category ($M \rightarrow L_i$)
† TM: True Miscellaneous, image assigned correctly to miscellaneous category ($M \rightarrow M$)
† FM: False Miscellaneous, image that belongs to a labeled category but was assigned to miscellaneous category ($L_i \rightarrow M$)

Note this system cannot be replaced by the established true/false positive/negative (TP, FP, TN, FN) classification:

Fig. 8. (Previous two pp.): Examples of the image categorization results. For each category $L_i$, the representative image is shown first (framed red), followed by other ‘True Labels’ (TL). To show the limitations of the system, the second row shows ‘False Labels’ (FL). The distinction between TLs and FLs is drawn based on the semantic interpretation, not the features. The last two rows belong to the ‘miscellaneous’ class $M$, divided into ‘True/False Miscellaneous’ (TM/FM). For an interpretation of the images see Section 6, the detailed results are given in Table 1.
trying to identify TL = TP, TM = TN, FM = FN leads to the problem that FL and ML would be two sorts of false positives (FP).

Based on this class system, the following quantities for characterization of a category system can be defined, where $N_I$ denotes the total number of images:

- **Fraction of categorization:**
  \[
  \text{CAT} = \frac{\#TL + \#FL + \#ML}{N_I} \quad \text{categorization},
  \]

- **REJ = \#TM + \#FM \quad \text{rejections.} \]

Fig. 8 (continued)
Quality of categorization:

\[
\text{COV} = \frac{\#\text{TL} + \#\text{FL} + \#\text{FM}}{N_f} \quad \text{topical coverage},
\]

(10)

\[
\text{HIT} = \frac{\#\text{TL}}{\#\text{TL} + \#\text{FL} + \#\text{FM}} \quad \text{hits},
\]

(11)

\[
\text{MIS} = \frac{\#\text{FL} + \#\text{FM}}{\#\text{TL} + \#\text{FL} + \#\text{FM}} \quad \text{misses}.
\]

(12)

User relevant quantities:

\[
\text{COR} = \frac{\#\text{TL}}{\#\text{TL} + \#\text{FL} + \#\text{ML}} \quad \text{correct},
\]

(13)

\[
\text{WRO} = \frac{\#\text{FL} + \#\text{ML}}{\#\text{TL} + \#\text{FL} + \#\text{ML}} \quad \text{wrong}.
\]

(14)

Note that

\[
N_f = \#\text{TL} + \#\text{FL} + \#\text{ML} + \#\text{TM} + \#\text{FM},
\]

(15)

\[
\text{CAT} = \frac{1}{N_f} \sum_{i=1}^{N_c} ||L_i||.
\]

(16)

\[
\text{REJ} = \frac{1}{N_f} ||M||.
\]

(17)

The quantities \(\text{CAT}\) and \(\text{REJ} = 1 - \text{CAT}\) are the overall fractions of images sorted into labeled categories \(L_i\) and miscellaneous \(M\), respectively. \(\text{CAT}\) is merely the part of images which are now available under labeled categories, without taking into account whether the sorting makes sense.

In contrast, \(\text{COV}\) denotes the fraction of images that belong from a high level point of view to one of the established labeled categories. Thus, \(\text{COV}\) quantifies the usefulness of the found categories with respect to topical coverage of the database. \(\text{COV}\) is independent from the actual sorting of images into the categories. Which part of the images that should be sorted into the labeled categories really goes to the correct one is quantified by \(\text{HIT} = 1 - \text{MIS}\).

While \(\text{COV}\), \(\text{HIT}\) and \(\text{MIS}\) are the ‘bird’s eye view’ on categories as such, \(\text{COR}\) and \(\text{WRO} = 1 - \text{COR}\) reflect which accuracy a user can finally expect. \(\text{COR}\) is the probability that a sample image drawn at random from a labeled category actually belongs there.

7.2. Categorization results

While category selection does not require large effort for \(N_f = 10,000\) images and \(N_c = 100\) reference vectors (the subsets can be easily browsed), counting TL, FL, etc. is extremely time-consuming. Therefore, only 3000 randomly selected images out of 10000 were manually evaluated. Table 1 shows the results. About half of the database (52.2%) is categorized, the topical coverage is slightly less (50.8%). In other words, from a semantical point of view 50.8% of the database belong to the established categories, but the system put there 52.2%. The probability that an image from a category \(L_i\) is correctly labeled is 72.9%, whereas the probability that an image of the database that belongs to a certain labeled category actually can be found there is 74.9%.

At first glance, these numbers may seem unsatisfactory, but it must be kept in mind that (a) the task is difficult and (b) the system gives valuable assistance both to a database designer and the user: as for the difficulty, bridging the ‘semantic gap’ is in category formation no less difficult than in CBIR, for a discussion see e.g. \([56,68,67]\). As for usability, a database designer saves about half the work in sorting a database, and a user gets to each prototypic image lots of similar ones, intermixed with only 27.1% others. From the point of view of CBIR, 72.9% correct results in answer to a query would be exceedingly good. This result is achieved here by the restriction of the categorization to the 52.2% of the material that can be well captured by low level features.

8. Conclusion

An approach was presented for the unsupervised formation of categories for a large image database and the
semi-automatic sorting into these categories. The work of
the database designer is reduced to checking categories for
(relative) consistency and to decide whether a category
should be rejected. The system is thus not able to process the
entire collection but is restricted to the part where semantics
can be detected from low level accessible features—a
sensible restriction considering today’s development status
in the processing of unconstrained natural images.

The system achieves good results despite this necessary
restriction: on a database of 10,000 images, more than half
are automatically categorized and about three out of four
are sorted correctly. This is achieved by a unified
representation of color and shape on a local scale, and
combining these single features in a way such that the
distribution of features on the global scale is represented in a
histogram-like fashion, discarding the large scale spatial
distribution.

The approach makes obvious both the capabilities and
limitations of today’s technique: tasks like categorization
and CBIR can be solved as long as low level features correlate with semantic expectation, but otherwise,
understanding of unconstrained images does not exist.
Since the latter task will not be solved in the foreseeable
future, the question is what direction further research can take. One possibility is to make use of the—obviously
existing correlation between low level features and high
level semantics in a more skillful way. So far, research
was primarily directed to finding low level features
which can establish a correspondence of image distances and
semantic similarity in general. But as illustrated by the ‘beach’ example of Section 4.4 (scene differs in time
of day but the object is the same), different low level
features may establish ‘axes’ that uncover different
semantic category systems. While in CBIR the application
of variable feature sets comes into use to answer
queries, things are less straightforward in image
categorization. Here, not a particular query has to be
answered using particular features, but a general
categorization scheme must be found. A possible solution is to establish several categorization systems in parallel
which are organized by different features.

To further develop the categorization system, the integration of a larger variety of features will be
necessary. In particular, only local features are
employed so far. Global features such as the global
color distribution or global geometrical scene proper-
ties are promising candidates for a supplementing
overall scene characterization. A problem that still
has to be solved in this context is the weighting of
different types of features such as local vs. global
ones. First experiments in combining the proposed
local features with global color histograms have shown that heuristically chosen fixed weightings lead
to better categorization of some types of scenes while
making worse others. So, a key question is how
information on the success of categorization can be
obtained to allow an adaptive weighting of features. This will be a topic of future research.

Acknowledgements

I would like to thank Helge Ritter, head of the
Neuroinformatics Group at Bielefeld University, for the
long term support of my research. This work was funded by
the German Research Foundation (DFG) within project SFB 360.

Appendix A

A.1 The Plessey detector

The Plessey (or HARRIS) detector proposed in [19] is
aimed at finding edges and corners. Let the gray values of an
image be denoted by \( I(x, y) \) and the spatial derivatives by \( I_x \),
\( I_y \):

\[
I_x(x, y) = \frac{\partial I(x, y)}{\partial x}, \quad I_y(x, y) = \frac{\partial I(x, y)}{\partial y}.
\]

(18)

In the present context, \( I_x, I_y \) are computed by convolution
with 5\times5-Sobel operators. Then the weighted products \( \langle I_x \rangle_{W(x,y)}^2 \), \( \langle I_y \rangle_{W(x,y)}^2 \) and \( \langle I_x I_y \rangle_{W(x,y)} \) are computed, where \( \langle \cdot \rangle_{W(x,y)} \) denotes a Gaussian weighting function, applied
within a window \( W(x, y) \) centered at \((x, y)\). To be more
precise, e.g. each pixel of the ‘x-gradient strength map’ \( I_x(x, y) \) is first squared to obtain an \( I_x^2 \)-map, then the \( I_x^2 \)-map is
convolved with a Gaussian (width is \( \sigma=2 \)) to obtain
\( \langle I_x^2 \rangle_{W(x,y)} \).

In the next step, a matrix \( C \) is calculated:

\[
C(x,y) = \begin{pmatrix}
\langle I_x^2 \rangle_{W(x,y)} & \langle I_x I_y \rangle_{W(x,y)} \\
\langle I_x I_y \rangle_{W(x,y)} & \langle I_y^2 \rangle_{W(x,y)}
\end{pmatrix}.
\]

(A.1)

\( C \) is an approximation of the auto-correlation function
of the image matrix, for details see, e.g. [52]. A corner
point is indicated where both eigenvalues \( \lambda_1, \lambda_2 \) of \( C \) are
large. So for each pixel a saliency value \( M_P \) is calculated as:

\[
M_P(x, y) = \lambda_1 \lambda_2 - \beta(\lambda_1 + \lambda_2)^2.
\]

(A.2)

For \( \beta \), a value of 0.06 is used according to [52]. The
explicit eigenvalue computation can be easily avoided
since the right hand side of Eq. (A.2) equals \( \det(C) - \beta(\text{Tr}(C))^2 \).

IPs are detected as the \( N_W \) highest maxima of \( M_P \). Fig. 4
shows examples for the resulting saliency maps (middle)
and IPs (right).
A.2. Image material

All images in this paper are part of the 'photos' section of the ArtExplosion® [43] collection.

References


[43] Nova Development Corporation, 23801 Calabasas Road, Suite 2005
Calabasas, California 91302-1547, USA. Art Explosion®
Photo Gallery.

[44] G. Qiu, X. Feng, J. Fang, Compressing histogram representations for
automatic colour photo categorization, Pattern Recognition 37 (2004)
2177–2193.

[45] G. Qiu, K.M. Lam, Spectrally layered color indexing in: M.S. Lew,
N. Sebe, J.P. Eakins (Eds.), Image and Video Retrieval CIVR 2002,


visualisation of image similarity as a tool for image browsing, in:

similarity assist image browsing? In: Proceedings of the SIGCHI’01,

[49] J. Rubner, K. Schulten, Regularization and color constancy,

[50] G. Salton, Automatic Text Processing, Addison-Wesley, Reading,

[51] C. Schmid, R. Mohr, Local grayvalue invariants for image retrieval,
IEEE Transactions on Pattern Analysis and Machine Intelligence 19

[52] C. Schmid, R. Mohr, C. Bauckhage, Evaluation of interest point
151–172.

[53] E. Shihat, M. Werman, Y. Gdalyahu. Ridge’s corner detection and
contour is in presence, in: Proceedings of the Conference on Computer
976–981.

[54] E.P. Simoncelli, B.A. Olshausen, Natural image statistics and neural
representation, Annual Review of Neuroscience 24 (2001) 1193–
1216.

[55] J. Sivic, A. Zisserman. Video data mining using configurations of
viewpoint invariant regions, in: Proceedings of the IEEE Conference on

[56] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain,
Content-based image retrieval at the end of the early years, IEEE
Transactions on Pattern Analysis and Machine Intelligence 22 (12)

[57] Smith, J.R., Chang, S.-F. Tools and techniques for color image
retrieval, in: Storage and Retrieval for Image and Video Databases

[58] S. Smith, J. Brady, SUSAN—a new approach to low level image
45–78.

of the SPIE Storage and Retrieval for Image and Video Databases,

[60] M. Swain, D. Ballard, Color indexing, International Journal of

[61] Q. Tian, N. Sebe, M.S. Lew, E. Lopias, T.S. Huang, Image retrieval
using wavelet-based salient points, Journal of Electronic Imaging 10

[62] K. Tieri, P. Viola, Boosting image retrieval, International Journal of

[63] C. Tomasi, T. Kanade, Detection and tracking of point features,
Technical Report CMU-CS-91-132, Carnegie Mellon University,

[64] C. Town, D. Sinclair, Ontological query language for content based
image retrieval, in: Proceedings of the IEEE Workshop on Content-

[65] T. Tuytelaars, L. van Gool. Content-based image retrieval based on
local affinity invariant regions, in: Third International Conference on
Visual Information Systems, Visual’99, Amsterdam, The Nether-

images, in: Proceedings of the 12th Color Imaging Conference: Color
Science and Engineering Systems, Technologies, Applications CIC

survey, Technical Report, Department of Computing Science, Utrecht


[69] Vision Texture Database (VisTex) by R. Picard, C. Graczyk, S. Mann,
J. Wachman, L. Picard, L. Campbell, 1995. Copyright 1995 by the
Massachusetts Institute of Technology. See http://www-white.media.
mit.edu/vismod/imagery/VisionTexture/vistex.html.

[70] S. Di Zenzo, A note on the gradient of a multi-image,
116–125.

for image feature extraction based on edge maps, in: Proceedings of
the IEEE International Conference on Image Processing, Japan, 1999.

[72] B. Zitová, J. Kautsky, G. Peters, J. Flusser, Robust detection of
significant points in multi-frame images, Pattern Recognition Letters