Finding Salient Regions in Images:

Non-Parametric Clustering for Image Segmentation and Grouping

E.J. Pauwels* and G. Frederix

ESAT-PSI, Dept. of Electrical Eng., K.U.Leuven
K. Mercierlaan 94, B-3001 Leuven, Belgium

Email: {Eric.Pauwels, Greet.Frederix}@esat.kuleuven.ac.be
Phone: + 32 - 16 - 321706   Fax: + 32 - 16 - 321986

21 August 1998

Keywords: Clustering, segmentation, grouping, perceptual organisation, intermediate level processing, content-based image retrieval, connectivity, nearest neighbours, Turing-test.

TO APPEAR IN:

Journal of Computer Vision and Image Understanding

(Special issue, May 1999)

[*] Post-Doctoral Research Fellow, Fund for Scientific Research (F.W.O. Vlaanderen), Belgium. Corresponding author.

Abstract

A major problem in Content-Based Image Retrieval (CBIR) is the unsupervised identification of perceptually salient regions in images. We contend that this problem can be tackled by mapping the pixels into various feature-spaces, whereupon they are subjected to a grouping-algorithm. In this paper we develop a robust and versatile non-parametric clustering algorithm that is able to handle the unbalanced and highly irregular clusters encountered in such CBIR-applications. The strength of our approach lies not so much in the clustering itself, but rather in the definition and use of two cluster-validity indices that are independent of the cluster-topology. By combining them, an optimal clustering can be identified, and experiments confirm that the associated clusters do indeed correspond to perceptually salient image-regions.

Contents

1 Intermediate-level processing for Content-Based Image Retrieval

Recent years have witnessed a marked resurgence in the interest for *intermediate level processing* in computer vision. One of the driving forces behind this renaissance is the rapidly growing interest in *content-based image access and retrieval* (CBIR) for multi-media libraries. Here the aim is to retrieve images that are similar to a query-image. Extensive experimentation over the last few years has shown that matching natural images solely on that basis of global similarities is often too crude an approach to produce satisfactory results. What is required is some form of perceptually relevant segmentation that allows one to identify a (small) number of salient and semantically meaningful image-regions which can then serve as the basis for more discerning region-based matching. E.g. it stands to reason that when it comes to perceptual matching, the foreground in an image is far more important than the background.

By its very definition, low-level vision, although often an essential preprocessing step, can only take things so far where this sort of perceptual organisation or automatic interpretation is concerned. On the other hand, high-level reasoning about images is still far too rudimentary and fragile to produce reliable results.

Faced with these problems, an increasing number of researchers are now exploring the middle ground between high- and low-level processing, shifting the focus of attention away from the purely local and pixel-based indicators to more global measures that seem to provide a stepping stone towards a more robust high-level processing. The idea is that intermediate-level processing can be used to perform a sort of triage, identifying in the image perceptually salient regions-of-interest that should be assigned higher weights in similarity computations (e.g. foreground versus background), or deserve closer scrutiny from dedicated modules (e.g. shape-descriptors, systems for face-recognition or object-identification). Furthermore, once a number of salient regions have been identified, it becomes possible to quantify their location and spatial organisation with respect to each other, often a crucial factor in the interpretation-process: e.g. a blue extended patch at the top of an image probably represents a clear sky (see also [2, 18]). Well-known examples of such region-based intermediate level processing are *figure-ground separation, blob-detection,* and *contour completion* (for examples and applications see e.g. [1, 3, 9, 13]).

Clearly, as a first step towards tackling this problem, we need to agree what sort of image-regions are interesting or perceptually salient. For the problems at hand *saliency* is defined in terms of *features* that capture essential visual or perceptual qualities such as colour, texture, shape-characteristics such as linearity or circularity etc.... Put differently, this means that when an image is mapped into the appropriate feature-space, salient regions (by their very definition) will stand out from the rest of the data and can more easily be identified. By the same token, pixels from disconnected parts in the image that have perceptually similar characteristics, will be mapped onto the same region in the appropriate feature-space and as a consequence can be grouped together.

Therefore, from an abstract point of view, segmentation and perceptual organization can be interpreted as a problem of *selecting appropriate features,* followed by *cluster-detection in feature-space.* In fact, we can tighten up this argument even further since both steps are but two aspects of the same problem, as a particular feature-space is deemed appropriate whenever it shows pronounced clusters. Indeed, if mapping the pixels into the feature-space lumps them all together, this particular set of features is obviously of little use in defining perceptual saliency.

Furthermore, as pointed out by several authors (e.g. [14, 13]) similarities based on clustering are in many cases far more natural than similarities based on distances alone. This point is schematically illustrated in Fig. 1: Although point a is closer (in distance) to b as it is to c, it will often be the case that, *perceptually,* a is more similar to c than it is to b. Indeed, being in the same cluster means — by definition — that there is a dense data-cloud bridging the relatively wide gap between a and c, allowing us to visually “morph” the two points, thus establishing a similarity. In contrast, the lack of such a morphing sequence between a and b highlights the dissimilarity.

This approach is certainly not new and goes back at least to Coleman and Andrews [4], but

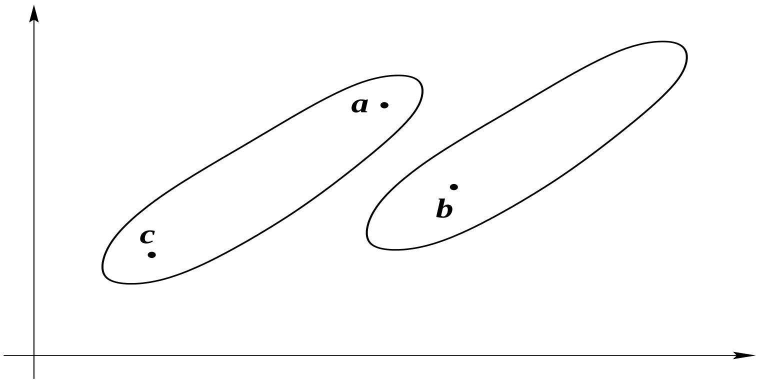

Figure 1: *The advantage of clustering over distance-measurement. Although point a is closer (in distance) to b as it is to c, it will often be the case that perceptually a is more similar to c than it is to b. Being in the same cluster means that there is a dense data-cloud bridging the relatively wide gap between a and c, allowing us to visually "morph" the two points, thus establishing a similarity.*

a viable implementation of such a strategy has been severely hampered by the lack of reliable clustering-algorithms able to meet the challenges set by the highly unbalanced and convoluted clusters that are rife in image-processing applications. Indeed, clustering problems commonly encountered in intermediate level processing are particularly challenging as mapping the images to feature-spaces often produces very irregular data-clouds, a far cry from the Gaussian-like clusters seen in most textbook applications. Furthermore, given the fact that segmentation and region-extraction should proceed automatically, we cannot assume that prior knowledge about the number of clusters or their shape is available. This open-ended problem-formulation strongly suggests to adopt *non-parametric clustering methods.* For this reason we will in the rest of this paper extend earlier work published in [15, 16], and detail a versatile *non-parametric clustering algorithm* and show how this can be used to *extract saliency* and produce *semantically meaningful segmentation.*

The organisation of this paper is as follows: In section 2 we give a concise overview of the main "classical" clustering-techniques, indicating briefly why their performance fall short of expectations for the problems we are facing in these applications. In section 3.1 we outline a clustering-method based on non-parametric density-estimation and define two cluster-validity indices in section 3.2. Their definition and the way in which they are combined into a single performance-indicator, as detailed in section 3.3, are key to the success of our clustering-algorithm. In section 4 we report on experiments and discuss ways in which this new clustering-methodology can be brought to bear on CBIR-related problems. To conclude, we mention that a number of recent papers (see e.g. [3, 17]) used clustering for segmentation-purposes, with very encouraging results indeed. Although the underlying rationale is similar to ours, the clustering-methodology itself is very different from the one expounded in this paper.

2 Brief overview of classical clustering-methodology

Given the importance of *clustering* (or *unsupervised learning* as it tends to be called in the neural network literature), it is most unfortunate that there is no crisp definition of the concept itself. However, most researchers would contend that the two main characteristics of a cluster are that it is both *compactly connected* and *isolated* with respect to the other datapoints in the ambient space. Most existing work in clustering focuses on the different clustering-strategies themselves, but in this paper the emphasis will be on quantifying *cluster-validity* (see sections 3.2 and 3.3) so that we are able to select an "optimal" (or at least acceptable) clustering from among a number of possibilities.

This sounds reasonable enough, but our use of the word "optimal" forces us to explain what exactly we mean by a "correct clustering"? Rather than agonizing endlessly about an acceptable mathematical formulation of this rather intuitive geometric interpretation we take a very pragmatic stance and insist that any clustering-algorithm worth its name, should come up with the right answer in those cases where it is obvious to a human observer what the "correct" answer is. Some researchers might take exception at our loose use of the term "correct", arguing that in most cases there is no groundtruth. Be this as it may, we contend that our interpretation of clustering is operational in the sense of a *Turing-test:* we consider a clustering algorithm to be successful if on a wide variety of datasets, it produces results that are judged acceptable by a human. Put differently, when presented with two sets of clustered data, one clustered by a human and one by the computer-algorithm, it should be impossible to tell which is which.

Before we proceed by explaining in more detail our approach to the clustering problem, we give a short overview of "classical" approaches to clustering, highlighting in particular the shortcomings these methods have in the face of the problems typically encountered in image-processing. For more details we refer the interested reader to a number of excellent books on clustering such as [5, 12, 7, 10, 11].

- **K-means:** This is probably the best-known and most widely-used clustering-algorithm because it is straightforward and extremely fast. Unfortunately, it has a number of drawbacks. For starters, the number of resulting clusters need to be specified in advance, which in applications such as unsupervised segmentation is out of the question. Secondly, and probably even more importantly, K-means is unable to handle unbalanced clusters where one cluster has significantly more points than a neighbouring cluster. Elongated clusters are another such problem. In such cases, the K-means algorithm will erroneously split the larger cluster into artificial subclusters. Finally, since the algorithm produces a Voronoi-tessellation of the data-space, the resulting clusters are convex by construction.

- **Hierarchical clustering:** Starting with many clusters (typically as many clusters as datapoints) clusters are successively merged if they are sufficiently close to each other. There are various variants of this algorithm, depending on how the distance between clusters is computed (minimal, mean or maximal). The result however is a partially ordered tree in which the number of clusters decreased monotonically, rather than a single cluster. Furthermore, although, unlike for K-means, clusters can be of arbitrary shape, this method suffers from a most unwelcome "chaining"-effect: nearby but distinct clusters are often lumped together whenever there is a chain of data-points bridging the gap.

- **Parametric density estimation:** This approach hinges on the basic assumption that the underlying data-density is a mixture of g Gaussian densities. If this assumption holds then it sufficies to estimate the g means and covariances of these Gaussians, in addition to the partitioning of the data among them, to arrive at a satisfactory result (the well-known EM-algorithm belongs to this class). However, on the down-side we have that the number g of clusters has to be determined prior to the initialisation, and performance is poor for the sort of irregular clusters encountered in many of the CBIR-applications.

- **Non-parametric density estimation:** Again the aim is to estimate the underlying density of the data-points, but this time we no longer assume that it is a mixture of Gaussian distributions. Instead, the density is determined by convolving the datapoints with a density-kernel of limited support (for more details see below). Clusters are then identified by locating local density-maxima. Notice that there are no a priori restrictions on the shape of the resulting density and that there is no need to specify in advance the number of clusters (local maxima).

As we pointed out earlier, the clustering problems encountered in intermediate-level processing are particularly challenging as the expected number of clusters isn't known in advance, and the shape of the clusters is often markedly non-Gaussian. Although some recent papers (e.g. [6]) have made some interesting contributions towards more shape-indepedent clustering, we have opted for methodology based on non-parametric density estimation as the core of the clustering-algorithm, an approach taken up in the next section.

3 Non-parametric clustering for segmentation and feature-extraction

3.1 Clustering based on non-parametric density-estimation

Clustering based on non-parametric density-estimation starts from the construction of a data-density obtained by convolving the dataset by a density-kernel. More precisely, given an n-dimensional dataset $\{x_i \in I\!R^n; i = 1 \dots N\}$ a density $f(x)$ is obtained by convolving the dataset with a unimodal density-kernel $K_\sigma(x)$:

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} K_\sigma(x - x_i), \tag{1}$$

where σ is the size-parameter for the kernel, measuring its spread. Although almost any unimodal density will do, one typically takes K_σ to be a (rotation-invariant) Gaussian density with σ^2 specifying its variance:

$$K_\sigma(x) = \left(\frac{1}{2\pi\sigma^2}\right)^{n/2} \; e^{-\|x\|^2/2\sigma^2}$$

After convolution we identify candidate-clusters by using *gradient ascent* (hill-climbing) to pinpoint local maxima of the density f. Specifically, the k nearest neighbours of every point are determined, whereupon each point is linked to the point of highest density among these neighbours (possibly itself). Upon iteration, this procedure ends up assigning each point to a nearby density-maximum, thus carving up the data-set in compact and dense clumps.

However, it is obvious that unless the clustering parameters (such as the width σ of the convolution kernel K_σ or the number of neighbours k) are pre-set within a fairly narrow range, this procedure will result in either too many (if σ is chosen too small) or too few clusters (if σ is set too large). Although a huge bulk of the work on density-estimation concerns itself with this problem of choosing an "optimal" value for σ (e.g. see the book by Thompson and Tapia [19]), it is fair to say that it remains extremely tricky to try and estimate optimal (or even acceptable) clustering parameters.

For this reason we have taken a different route. We pick a value for σ which is small (with respect to the range of the dataset) and, as before, proceed to identify candidate clusters by locating local maxima of the density f. As mentioned above, this choice of σ will result in an over-estimation of the number of clusters, carving up the dataset in a collection of relatively small "clumps" centered around local maxima. Next, we construct a hierarchical family of derived clusterings by using the data-density to systematically merge neighbouring clumps. More precisely, we establish an order of merging by comparing the density-values at neighbouring maxima with respect to the density at the saddlepoint in-between which is defined as being the point of maximal density among the boundary points (i.e. points having neighbours in both clusters). Working systematically through this list of mergers produces the hierachically ordered tree of clusterings. (We refer to Fig. 4 for an illustration on a simple artificial dataset.)

Notice how this is very similar to the tree constructed in the case of hierarchical clustering, but with the crucial difference that the merging is based on the *density*, rather than on the *distance*, thus eliminating the unwelcome chaining-effect that vexes hierarchical clustering. Now, in order to pick out the most satisfactory clustering we will concentrate (in the next section) on the development of *indices of cluster-validity* that directly assign a performance-score to every proposed clustering of the data.

3.2 Non-parametric measures for cluster-validity

A cursory glance at the clustering-literature reveals that there is no shortage of indices that measure some sort of grouping-quality. Some of the most successful are the silhouette coefficient (introduced by Kaufman and Rousseeuw), the (modified) Hubert-coefficient, the intra- over inter-variation quotient and the BD-index, introduced by Bailey and Dubes (for more information we refer to Kaufmann and Rousseeuw [11], or Jain and Dubes [10]). However, all of these coefficients are basically variations on the same theme in that they compare inter- versus intra-cluster variability and tend to favour configurations with ball-shaped well-separated clusters. Irregularly shaped clusters are problematic. It is for this reason that we have opted to restrict our attention to non-parametric indices which don't suffer the above-mentioned drawbacks.

As we pointed out in the introduction, for want of a better definition, a "cluster" is understood to be a relatively *well-connected* region of high data-density that is *isolated*, in the sense that it is separated from other clusters by regions of low data-density (voids). We therefore introduce two non-parametric measures that quantify these qualitative descriptions for a given clustering of the dataset.

1. **Isolation** is measured in our algorithms by the *k-nearest neighbour norm* (NN-norm). More precisely, for fixed k (the precise value of which is not very critical), the k-nearest neighbour norm $\nu_k(x)$ of a data-point x is defined to be the fraction of the k nearest neighbours of x that have the same cluster-label as x. Obviously, if we have a satisfactory clustering and x is taken well within a cluster, then it is completely surrounded by points with identical labels and therefore $\nu_k(x) \approx 1$. However, even nearby the boundary of a well-defined cluster we can still expect $\nu_k(x) \approx 1$, since most of the nearest neighbours will be located well within the interior of the cluster (see Fig. 2). Only when a bad clustering has artificially broken a densely populated region into two or more parts, we'll see that for points along the "faultline" $\nu_k(x)$ is significantly smaller than 1 (e.g. $\nu_k(x) \approx 0.5$). We get an measure of the homogeneity of the total clustering by averaging over all N points in the dataset:

   $$\underline{\text{NN-norm:}} \qquad \mathcal{N}^{(k)} = \frac{1}{N} \sum_x \nu_k(x). \tag{2}$$

   In many regards, this is an extremely attractive quality-measure for clustering as it captures the fact that a cluster should be *isolated* with respect to the rest of the data. Furthermore, unlike most of the other criteria discussed above, it does not favour a particular cluster-structure, and is therefore very robust with respect to variations in the geometry of the cluster. This is most welcome, as most other criteria are biased towards compact sphere-like clusters.

   However, the major drawback of this index is that it doesn't notice whenever two clusters are merged, even if they are well-separated. In fact, lumping all points together in one big cluster, will result in an optimal score for this criterion. For this reason we need an additional criterion that measures *connectivity,* i.e. that penalizes clusterings that erroneoulsy lump together widely separated clusters.

2. **Connectivity** relates to the fact that for any two points in the same cluster, there always is a path connecting both along which the data-density remains relatively high. In our algorithm we quantify this by choosing at random two points (say a and b) in the same cluster and connecting them by a straight line (see Fig. 3). We then pick a testpoint t halfway along this connecting line and subject it to gradient ascent to seek out its local density maximum. However, the constraint is that during its evolution the distance of this testpoint to either of the two "anchor-points" should remain roughly equal (to avoid that the testpoint converges to one of the anchor-points). In case the cluster has a curved shape, this allows the testpoint to position itself along the high-density crescent connecting the anchor-points. If the cluster-label at the repositioned testpoint coincides with the clusterlabels at the anchor-points a and

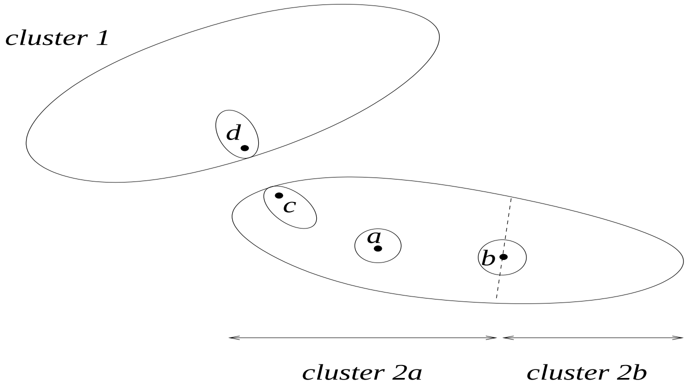

Figure 2: *Schematic represenation of the so-called "nearest neighbour norm"; the neighbours of the point a obviously all have the same label. This is also the case for "boundary-points" such as c and d, since, unless k is very large, their neighbourhood of k nearest neighbours will be asymmetric and biased towards the high-density regions of their respective clusters. However, at the point b where the cluster is erroneously split in two different parts, only about half of the points have the same label as b.*

b, the data-density $f(t)$ at this final position(averaged over a number of random choices for the anchor-points) can be used as a connectivity-indicator C (the so-called C-norm):

$$\underline{\text{C-norm:}} \qquad C = \frac{1}{K}\sum_{i=1}^{K} f(t_i) \tag{3}$$

where t_i is the testpoint for K randomly chosen pairs of anchor-points (a_i, b_i). Notice how this dependence on a randomly chosen testset of anchorpoints makes theC-norm a *stochastic* measure. This has the advantage that we can easily quantify the confidence in the measure by generating several randomly chosen sets of anchorpoints. We refer the reader to Fig. 5 (top right panel) where the C-norm is plotted with errorbars representing its standard error based on 10 resamplings.

Clearly, if the proposed clustering lumps together two well-separated clusters, many of these testpoints will get stuck in the void between the high-density regions, thus significantly lowering the value of this non-parametric connectivity-index.

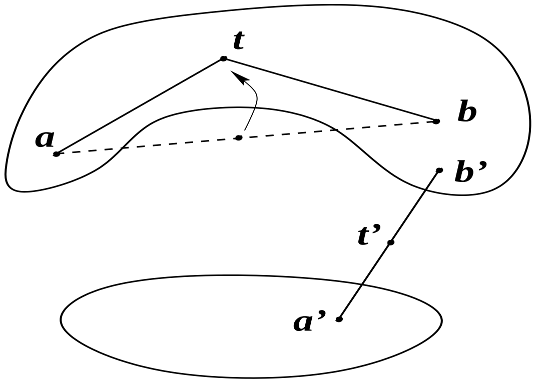

Figure 3: *Schematic representation of the (stochastic) connectivity index: the testpoint t is chosen halfway between the anchor-points a and b, and migrates towards the high density region without violating its "midpoint"-constraint. After convergence, the density at t is a measure of connectivity.*

3.3 Combining cluster-validity indices to select a clustering

From the previous considerations it transpires that in order to get a satisfactory clustering-result one has to try and maximise both indices simultaneously. However, as they are inversely correlated, any choice will involve a trade-off between connectivity and isolation. The problem is further compounded by the fact that the relevant information is captured primarily by the way these indices change, rather than by their specific values. Typically, the NN-norm will decrease as the number of clusters grows, while the connectivity-index tends to increase, but both trends will usually exhibit a sudden transition whereafter they more or less level off (see Fig. 5). Localising such jump-events is the key to identifying important qualitative changes in the clustering.

However, as it is tricky to reliably identify such a "knee" in a graph, we go about it in a slightly different way. First of all, in order to make the indices directly comparable, we compute their Z-scores. Recall that the Z-score of an observation ξ_i in a sample $\xi_1, \ldots, \xi_\ell$ with mean $\overline{\xi}$ and standard deviation σ_ξ, is defined to be the standardised deviation:

$$Z(\xi_i) = \frac{\xi_i - \overline{\xi}}{\sigma_\xi}.$$

In fact, because the jumps cause the index-values for different clusterings to be rather irregularly spaced, additional sensitivity can be obtained by using a robust version of the Z-score, defined by

$$Z(\xi_i) = \frac{\xi_i - \text{median}(\xi)}{\text{MAD}(\xi)},$$

where $\xi = \{\xi_1, \ldots, \xi_\ell\}$ represents the whole sample and MAD stands for *median absolute deviation*:

$$\text{MAD}(\xi) = \text{median } \{|\xi_i - \text{median}(\xi)| \ : \ i = 1, \ldots, \ell\}$$

We invite the reader to have a look at Figs. 4 and 5 for a simple illustration. Fig. 4 shows the family of possible clusterings obtained by merging basins of attraction of local density-maxima (see section 3.1). The corresponding graphs for the NN- and C-norm are shown in Fig.5.

Since the NN-norm decreases as the number of clusters increases, large values for the Z-score will typically occur *before* the major downward jump of this graph, favouring well-isolated clusters. By the same token, high Z-scores for the connectivity-index will be associated with clusterings following the major up-jump in this graph, thus drawing attention to clusterings for which the individual clusters are well-connected.

To bring this to bear on the problem at hand, let L_p be the labeling for the p^{th} clustering in the above-defined hierarchical tree, i.e. L_p maps each datapoint x to its corresponding cluster-label $L_p(x)$, and let $\mathcal{N}_p$ and C_p be the corresponding NN-norm and C-norm respectively, as defined by eqs.(2) and (3). (For instance, p ranges from 1 through 6 in Fig. 4) The (robust) Z-score for the p^{th} clustering is then defined to be

$$Z_p = Z(\mathcal{N}_p) + Z(C_p) \qquad (4)$$

and among the possible clusterings listed in the tree, we pick the one which maximises this robust Z-score (again, see Fig. 5).

4 Applications and discussion

Our original reason for embarking on this project was strongly motivated by the difficulties encountered in automatic segmentation of images for content-based image retrieval (CBIR). We therefore report in this paper on experimental results related to applications of the above-expounded clustering-algorithm to segmentation problems (mainly based on colour). Although the underlying methodology is basically very simple and versatile, we consider it encouraging to find that the segmentations obtained are comparable to the state-of-the-art results, as can be found in recently published papers by, among others, Ma [13], Zhu *et.al.* [20], Carson *et.al.* [3], and Shi *et.al.* [17].

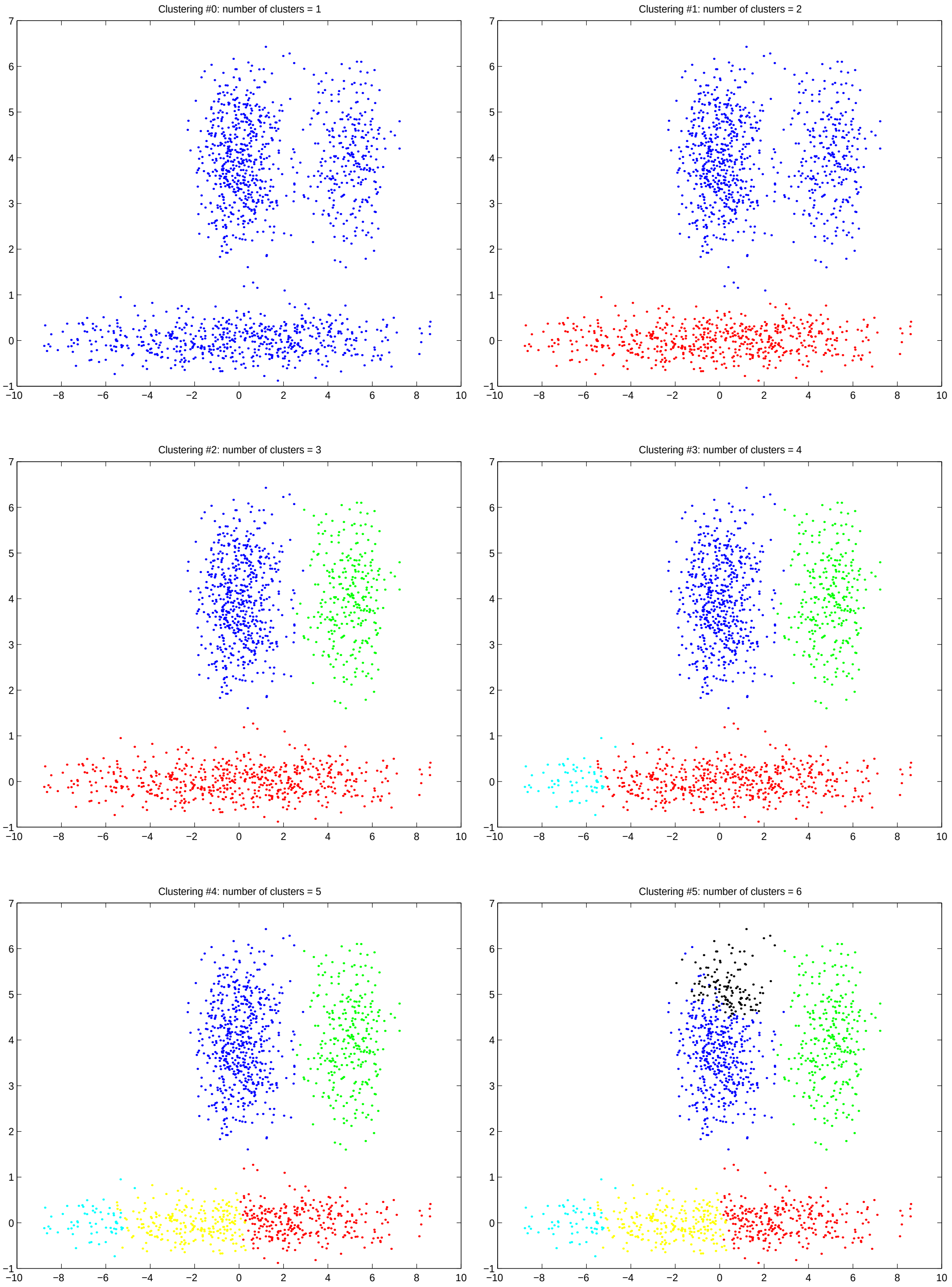

Figure 4: *Overview of the proposed clusterings for an artifical dataset. The number of clusterings ranges from one (upper left) to six (bottom right), where the latter was derived by identifying the local maxima in the density f, while the others were obtained by successively merging neighbouring clusters.*

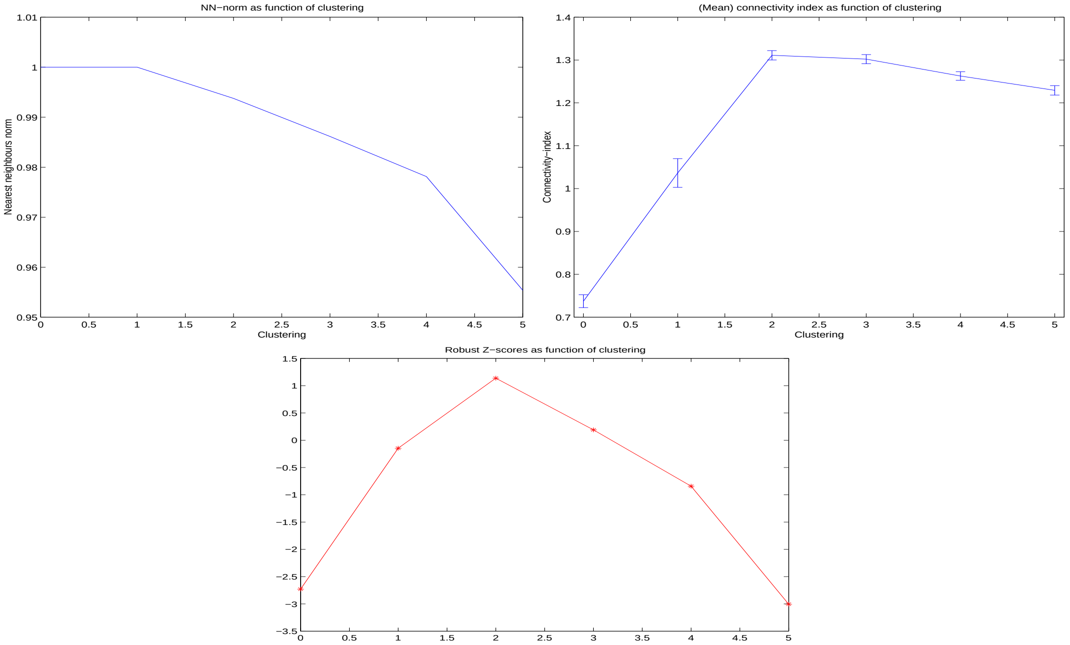

Figure 5: *NN-norm (top left), C-norm (top right) and robust Z-score (bottom) for the clusterings depicted in Fig. 4. The correct (third) clustering is identified by the Z-score. (Recall from eq.(3) that the C-norm is averaged over a random-sample. The error-bars on its graph represent the standard error on each value, providing us with a canonical measure to gauge the difference in successive values.)*

Pre-processing To speed up computations, we didn't cluster all pixels in the image, but drew a random subsample (typically of size 2000) which we use as input for the algorithm. The remaining pixels were classified by computing the Mahalanobis-distance to the identified clusters. An additional bonus of subsampling is the fact that it is easy to draw a new and independent sample offering the possibility to check the stability of the first clustering.

Since the clustering-algorithm does not depend on the dimensionality of the space, it is easy to incorporate spatial information by including the pixel-coordinates as features (as has been suggested by a number of authors). This however turns out to be not a good idea as it often destroys cluster-information by spreading data-points over a extended region in feature-space. A far better way to include spatial information is by slightly averaging pixel-values (using convolution with a small mask) *before* mapping them into feature-space.

Choice of parameters We recall that there are basically two free parameters in our algorithm that needs to be preset. The first is k, the number of nearest neighbours that is used when computing the NN-norm (2). We fixed k to be *one percent* of N, the total number of datapoints, but with a minimum of $k = 10$:

$$k = \max\{0.01N, 10\}.$$

The second is the width σ of the convolution kernel K_σ that is used to generate the nonparametric density f. This is taken proportional to the average radius of the (smallest) ball that encloses the k nearest neighbours of a point. Hence the parameters are completely determined by the data and scale with the size and range of the dataset. Moreover, it is worth stressing the following two points:

- First, their specific value is not critical at all; all we need to make sure is that σ is "small" with respect to the range of the data (otherwise, the density will be over-smoothed), and that k is "small" compared to the total number of data-points N.

- Secondly, we fixed the recipes for the determination of the parameter-values at the beginning of our experiments and then used *identical parameter-settings* for *all* datasets (both real and artificial)! So, this clustering proceeds truly *unsupervised*.

Experimental results We tested our algorithm on both artificial and real data. The artificial datasets included a number of complicated structures (such as banana- or ring-shaped clouds) that tend to trip up the more conventional clustering-algorithms. Our algorithm, succeeded in finding the "correct" results in all cases as is shown in Fig. 6, bolstering our claim that it passed its "Turing-test".

To generate real data we took a number of challenging natural images and mapped them into different colour- and texture-spaces whereupon we proceeded to cluster them. The results are shown in Figs. 7 through 10. We stress that *in all cases the number of regions in which the images were segmented was determined by the algorithm itself, on the basis of the number of clusters that were reliably identified.* The fact that these clusters obviously correspond to perceptually meaningful regions underscores our contention that saliency is the result of datapoints clustering in appropriate feature-spaces!

The first two figures (7 and 8) represent segmentations based on clusterings in colour-space. We tried a number of colour-spaces but in most cases the final results are very similar (we therefore used *Lab*-space throughout).

Figure 9 shows some results of segmentations based on clustering in various texture-spaces. The top-row is a typical Brodatz-composition which was segmented in a feature-space based on Gabor-filters. The texture in the two natural images (rock-formation and predator) was characterized by quantifying the autocorrelation of the images in small windows around randomly sampled pixels. It is also interesting to report that in a number of examples, the algorithm returned the trivial clustering (i.e. a single cluster) as its selected answer. This was for instance the case when we mapped the lion-image (Fig. 9, bottom, left) into the colour-space. Sure enough, detailed inspection of the data revealed that the datapoints adhered to a roughly unimodal distribution, indicating that the bad perceptual result on the output image was due not to the failure of the cluster-algorithm, but rather to the fact that we were concentrating on unsuitable features. This, of course, is completely in line with our suggestion to use validity-indicators for clusters to select for each individual image those features that best capture the salient regions in that image.

Finally, to illustrate the advantages of our methodology with respect to the more classical clustering-algorithms such as K-means, we included an image (Figs. 10) where the non-parametric approach clearly outperforms the conventional ones. Although a simple algorithm such as K-means performs remarkably well on many images, results are less satisfactory whenever the algorithm is fooled by the complicated or irregular structure of the data-set (as is the case in this image). The middle row shows the "optimal" result obtained by clustering using K-means, for k (the number of clusters) ranging from 2 to 10. The "optimal" value for k was picked by both the DB- and silhouette-coefficients for cluster-validity (see [10] or [11] for more details on these coefficients). Compare this to the results obtained by the non-parametric algorithm on the bottom row.

Computational burden At the moment, all the routines are implemented in MATLAB which means that processing is rather slow: segmenting an image based on a subsample of 2000 pixels typically takes about 4 to 5 minutes on an average PC. However, we expect that a C-implementation will speed up things by an order of magnitude. Moreover, by its very nature, this algoritm is highly parallellizable, such that further dramatic speed-ups can be achieved by an appropriate architecture!

Discussion and conclusions In this paper we have developed a robust and versatile non-parametric clustering algorithm that is able to handle unbalanced and highly irregular clusters. In fact, we believe that the strength of our approach lies not so much in the clustering itself (which boils down to non-parametric density estimation), but rather in the definition and use of two cluster-validity indices that allow us to select a — in some sense — optimal clustering from among the nodes of a hierarchically ordered tree. Of course, it goes without saying that these validity-criteria are completely independent of the density-based clustering to which we apply them, and could just as well be used in conjunction with “classical” hierarchical clusterings which also produce a tree as output. (For K-means and mixture-based clustering it is unlikely that our indices will outperform existing validity-coefficients as listed in section 3.2, as in those cases the resulting clusters are more or less elllipsoidal and therefore have little to gain from a non-parametric approach.)

It is our opinion that this cluster-based approach to intermediate-level processing could be extremely useful in the context of CBIR-applications for the following reasons:

1. By using this clustering on data-clouds obtained by mapping images into different feature-spaces, it becomes possible to extract image-regions that are salient and/or semantically meaningful. In fact, we envisage a system that is comparable to the organisation in *visual streams* encountered in neurophysiology, where it has been shown that input to the visual cortex is sent to different specialized modules simultaneous for parallel processing. Likewise, we intend to build a system where the input image is mapped into different feature-spaces where data are clustered. Since, in contrast to most other validity measures, our clustering selection procedure is able to select the trivial clustering (all data in one single cluster) as the optimal clustering, it becomes possible to identify and reject non-interesting feature-spaces (this strategy resulted in the rejection of colour as an interesting segmentation-feature for the third image in Fig 9).

   Characterizing these segmented regions in terms of the corresponding features (colour, texture, location, shape, etc.) provides us with a powerful and natural tool to describe images and the perceptual similarities among them.

2. There is no reason to restrict the use of this algorithm to the clustering of pixels in a single image. The methodology works just as well on numerical data that are associated with whole images. So, once images in a database have been tagged with numerical attributes, the algorithm can be used to organise the database by grouping images with similar attributes.

3. Finally, it is our opinion that segmentation and perceptual organisation based on clustering is also very important at the query-stage, as it allows one to formulate *fuzzy* queries which are more akin to the way people think about such items. For instance, if we are looking for images with patches of fine-grained texture, it would be natural to formulate a query: “Find images with regions of *high* variation”. In most current systems one has to specify a threshold value that will quantify what is meant by “high”. However, in many cases we have no clue as to what the appropriate value should be, not to mention that it might be different from image to image).

   To understand how clustering might furnish us with a natural solution for this conundrum, think of the simplest possible case where an image-attribute is measured by a single real-valued feature (e.g. grey-value). If the histogram of the grey-values in an image has a pronounced bi-modal structure then it’s obvious that one mode will represent the (relatively) bright patches in the images. So, without having to specify any threshold whatsoever, clustering makes it possible to give a meaningful answer to the fuzzy query *“Show me the bright regions”*. Clustering allows us to extend this strategy to higher dimensional feature spaces so that it becomes feasible to equip the interface or search-engine with the capabilities to interpret and handle this sort of fuzzy, but for humans very natural, interaction.

We are currently exploring some of these issues and hope to report on them in a forthcoming paper.

Acknowledgments: E.P. gratefully acknowledges partial support by the Belgian Fund for Scientific Research (F.W.O. Vlaanderen), under grant G.0366.98, and by the Int. Assoc. for promotion of cooperation with scientists from NIS, INTAS-96-785.

References

[1] A. Amir and M. Lindenbaum: *Grouping-based hypothesis verification in object recognition.* Proc. IEEE Workshop on Perceptual Organization in Computer Vision. Santa Barbara, June 1998.

[2] A. Del Bimbo and E. Vicario: Using Weighted Spatial Relationships in Retrieval by Visual Content. Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries; June 21, 1998, Santa Barbara, Cal. pp. 35-39.

[3] C. Carson, S. Belongie, H. Greenspan, and J. Malik: *Region-Based Image Querying.* Proc. of CVPR'97 Workshop on Content-Based Access of Image and Video Libraries.

[4] G. Coleman and H.C. Andrews: *Image segmentation by clustering.* Proc. IEEE 67, 1979, pp. 773-785.

[5] R.O. Duda and P.E. Hart: *Pattern Classification and Scene Analysis.* Wiley 1973.

[6] H. Frigui and R. Krishnapuram: *Clustering by Competitive Agglomeration.* Pattern Recognition, Vol. 30, No. 7, ppe. 1109-1119, 1997.

[7] K. Fukunaga: *Introduction to Statistical Pattern Recognition.* Academic Press, 1990.

[8] I.J. Good and R.A. Gaskins: *Nonparametric roughness penalties for probability densities.* Biometrika 58, 255-77, 1971.

[9] N. R. Howe: Percentile Blobs for Image Similarity. Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries; June 21, 1998, Santa Barbara, Cal. pp. 78-83.

[10] A.K. Jain and R.C. Dubes: *Algorithms for Clustering Data.* Prentice Hall, 1988.

[11] Leonard Kaufman and Peter J. Rousseeuw: *Finding Groups in Data: An Introduction to Cluster Analysis.* J. Wiley and Sons, 1990.

[12] Brian S. Everitt: *Cluster Analysis.* Edward Arnold, 1993.

[13] W-Y Ma: *Netra: A Toolbox for Navigating Large Image Databases.* Ph.D.-thesis, Dept. of Electrical and Computer Engineering, University of California at Santa Barbara, June 1997.

[14] T.P. Minka and R.W. Picard: *Interactive Learning using a Society of Models.* Pattern Recognition, Vol. 30, 1997. pp. 565.

[15] E.J. Pauwels, P. Fiddelaers and F. Mindru: *Fully Unsupervised Clustering using Center-Surround Receptive Fields with Applications to Colour-Segmentation.* Proc. of the 7th. International Conference on Computer Analysis of Images and Patterns. Kiel, Germany, Sept 10-12, 1997.

[16] E.J. Pauwels and G. Frederix: *Non-parametric Clustering for Segmentation and Grouping.* Proc. of International Workshop on Very Low Bitrate Video Coding VLBV'98, Urbana, IL, Oct. 1998. pp. 133-136.

[17] J. Shi and J. Malik: *Normalized Cuts and Image Segmentation.* Proc. IEEE Conf. oon Comp. Vision and Pattern Recognition, San Juan, Puerto Rico, June 1997.

[18] John R. Smith and Chung-Sheng Li: *Decoding Image Semantics using Composite Region Templates.* Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries; June 21, 1998, Santa Barbara, Cal. pp. 9-13.

[19] J.R. Thompson and R.A. Tapia: *Nonparametric Function Estimation, Modeling and Simulation.* SIAM, 1990.

[20] S.C. Zhu and A. Yuille: *Region Competition: Unifying Snake/balloon, Region Growing, and Bayes/MDL/Energy for multi-band Image Segmentation.* IEEE Trans. on PAMI, Vol. 18, No. 9, Sept. 1996.

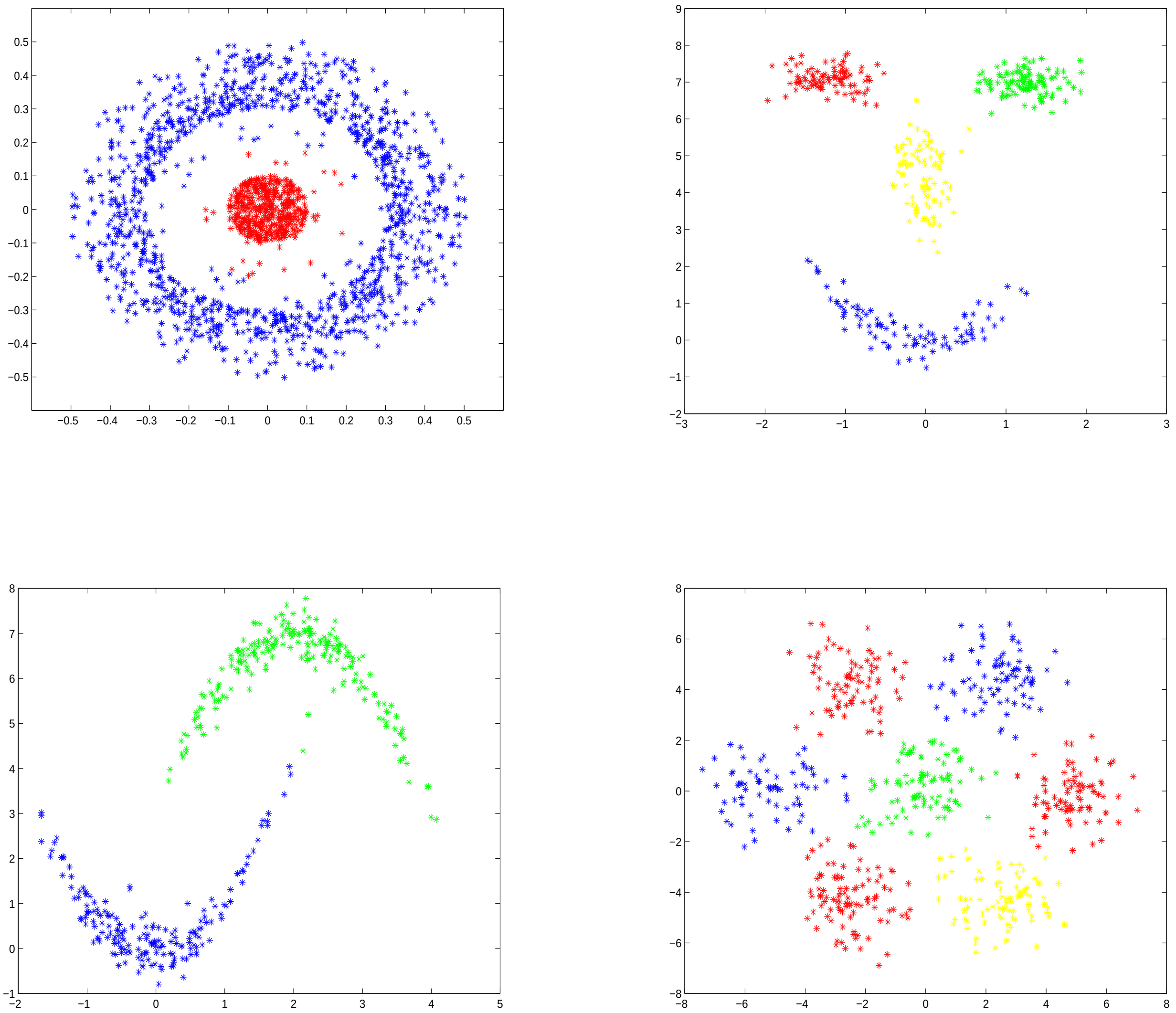

Figure 6: *Artificial datasets and the corresponding clustering results which show that the algorithm passed its "Turing test".*

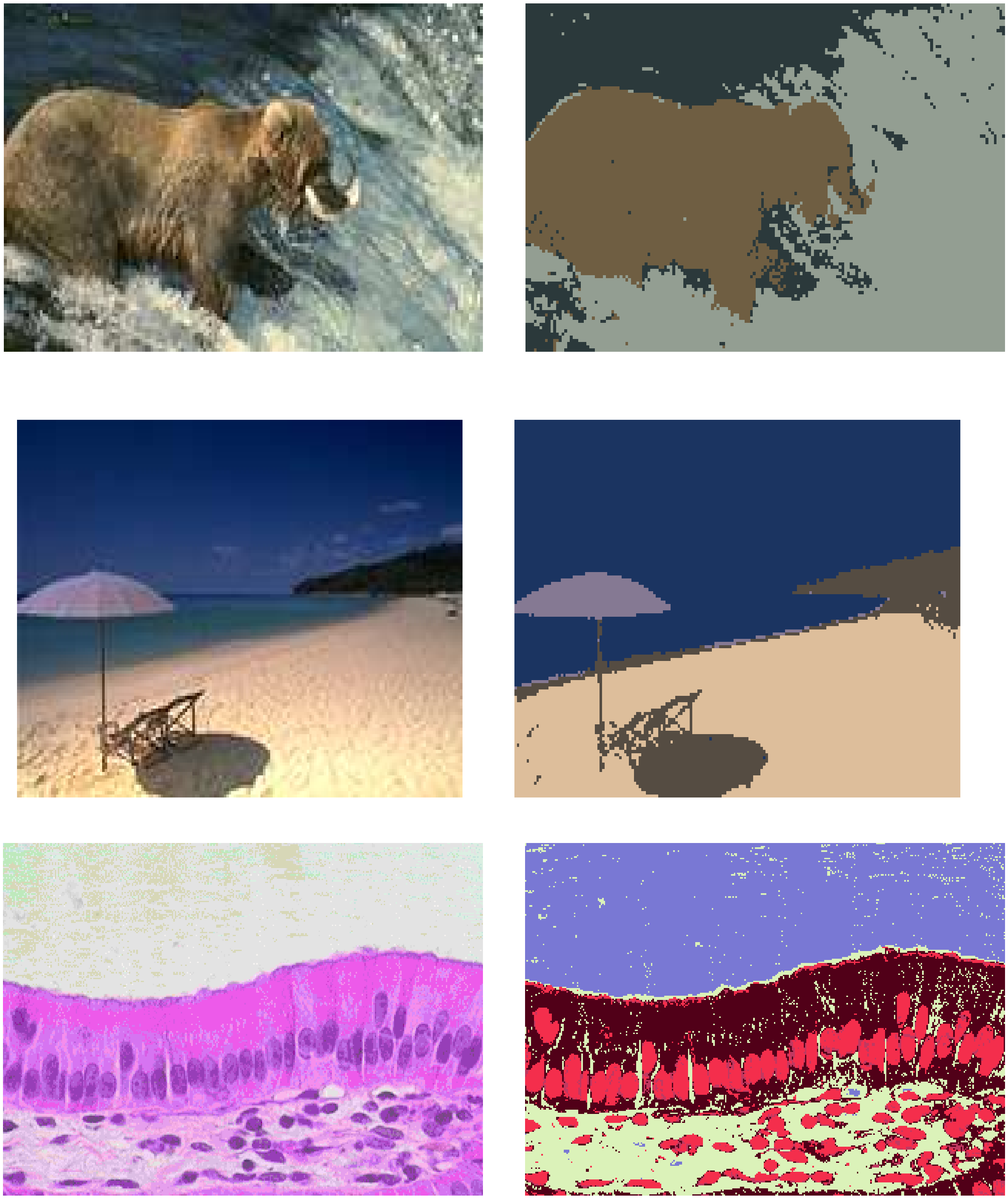

Figure 7: *Application of clustering algorithm to segmentation of colour-images.* Top: *Original input image and its segmentation in three regions (represented in mean colours) based on the clusters found in the 3D opponent colour space.* Middle: *Two different projection of representation of the bear-image in the 3D opponent colour space which clearly shows the non-Gaussian structure of the clusters. The three identified clusters are indicated.* Bottom: *Original histology image and its segmentation represented in false colours to show more clearly the boundaries of segmentation.*

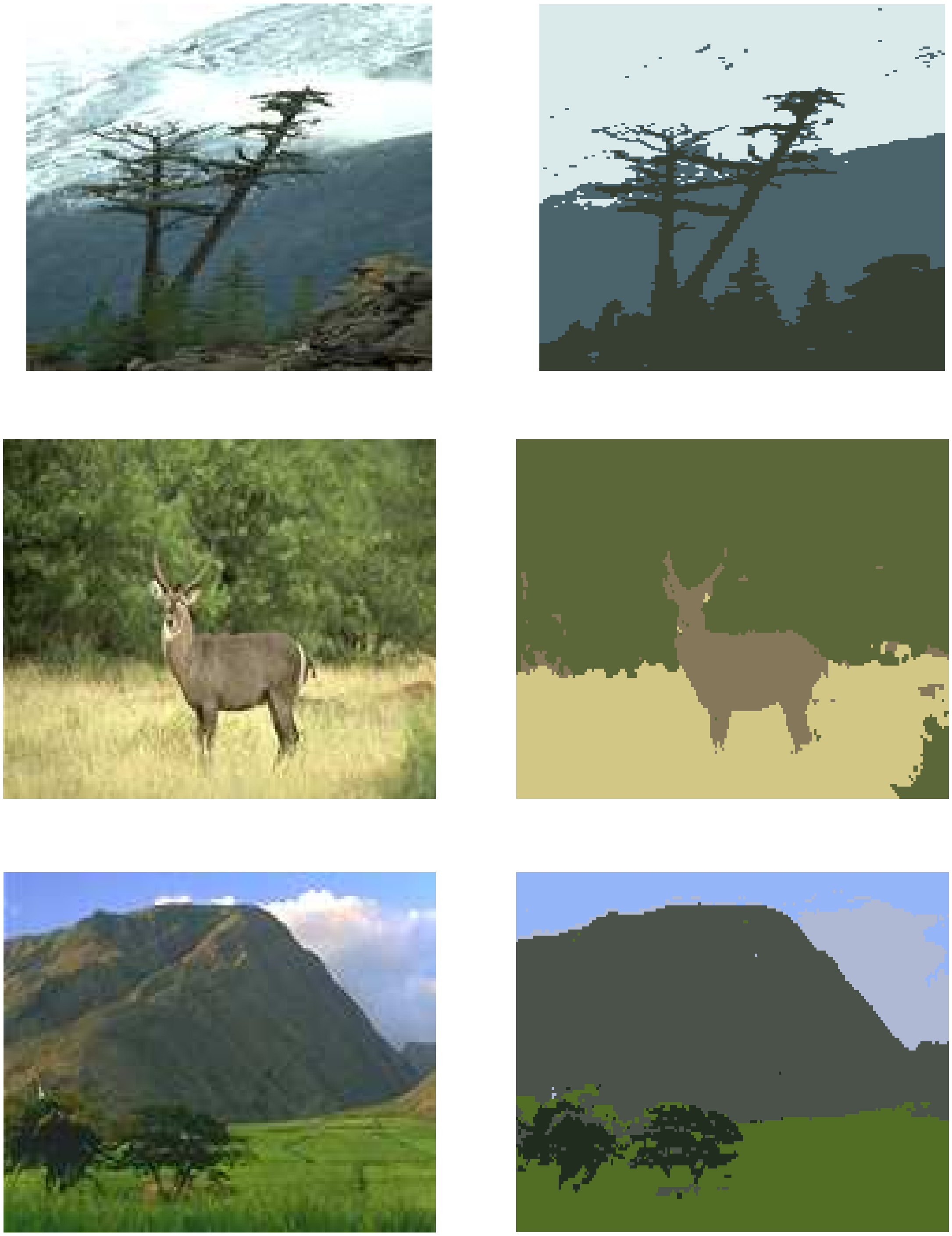

Figure 8: *Further applications of clustering algorithm to segmentation of colour-images. Left: Original input image. Right: Segmented image in mean colours.*

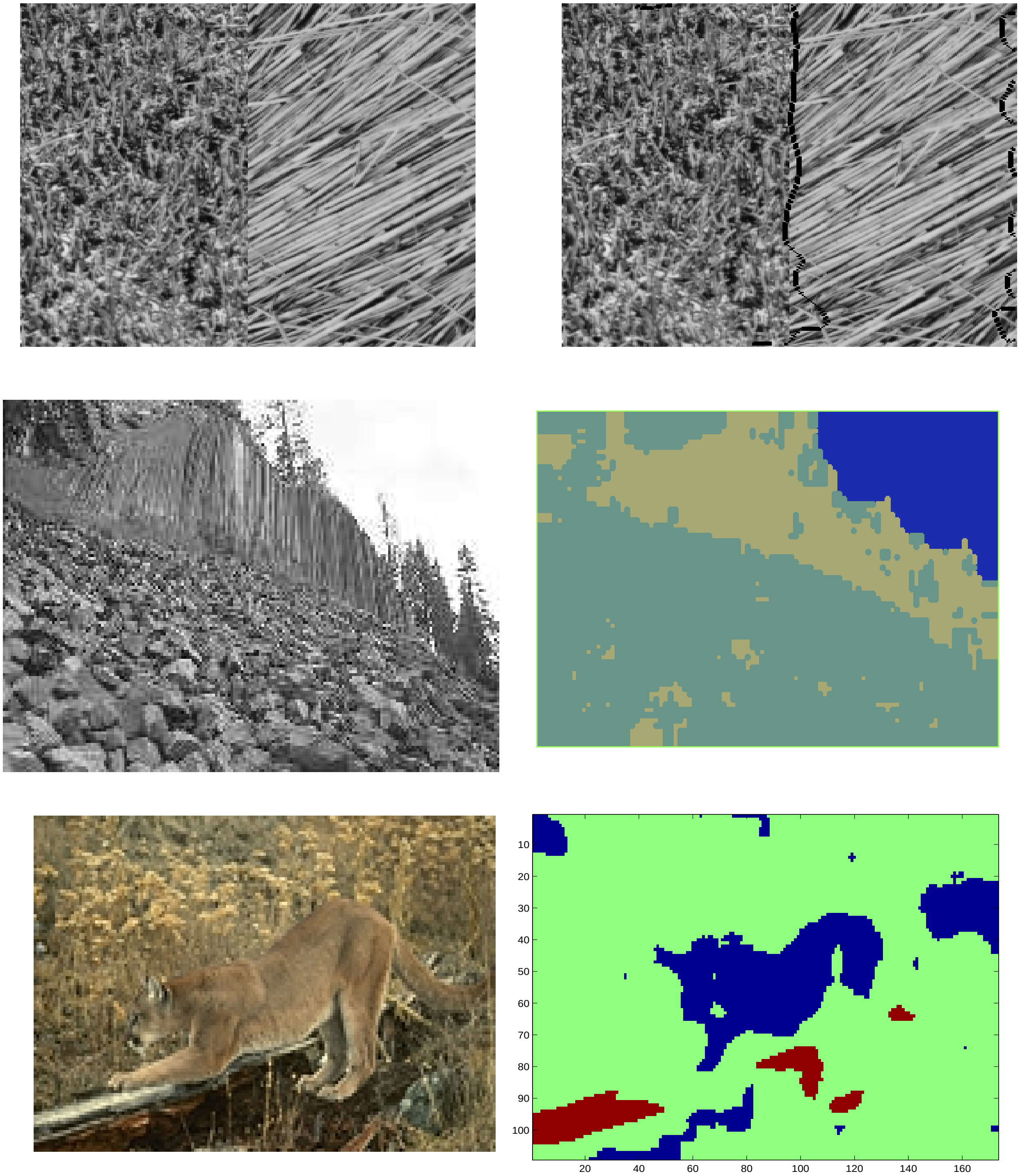

Figure 9: *Application of clustering algorithm to segmentation of a textured image.* Top: *Brodatz-image and segmentation based on clustering in feature-space created by Gabor-filters.* Middle and bottom: *Natural textured images segmented in texture space based on autocorrelation measures for small windows around each pixel. It is interesting that for the lion-image clustering based on colour returned a single cluster, proving that the lion's camouflage is effective.*

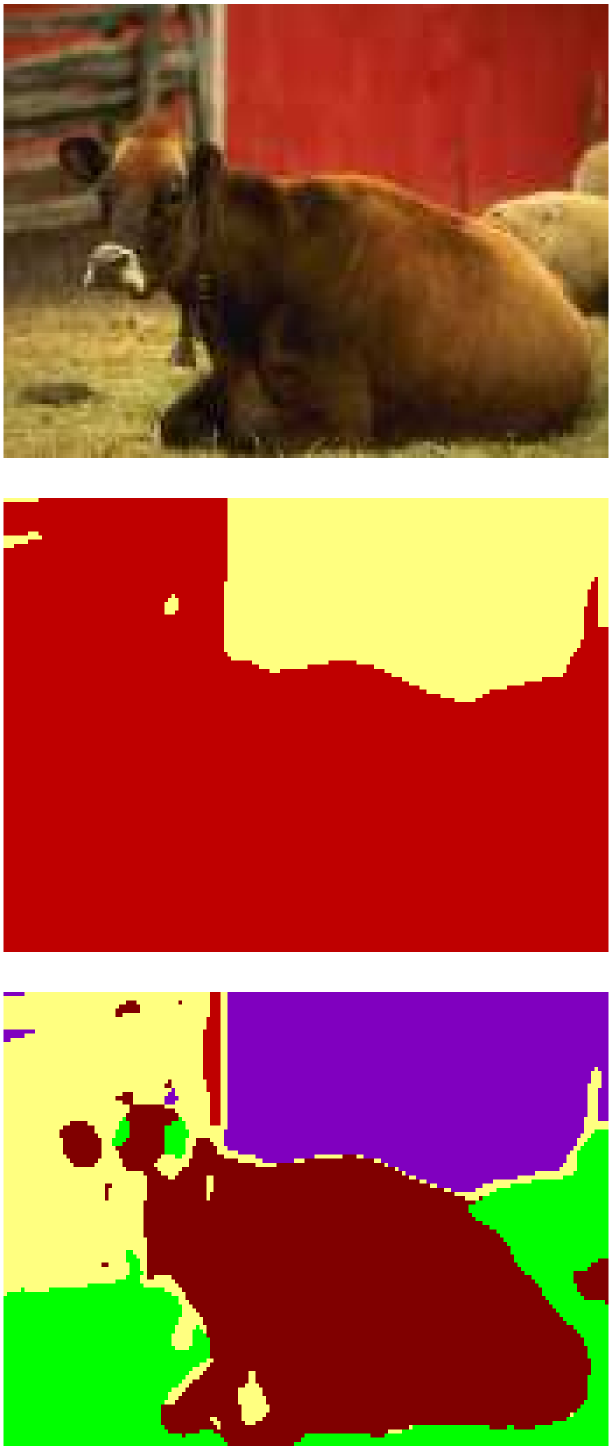

Figure 10: *Comparison of the non-parametric algorithm and K-means.* Top: *Input colour image.* Middle: *Segmentation obtained by K-means clustering, where the optimal value for k is determined using both the DB- and silhouette-coefficients.* Bottom: *Segmentation based on nonparameteric clustering of same data.*