Cluster-Based Segmentation of Natural Scenes

Eric J. Pauwels
ESAT-PSI
K.U.Leuven
K. Mercierlaan 94
B-3001 Leuven, BELGIUM

Greet Frederix
Dept. of Mathematics
K.U.Leuven
Celestijnenlaan 200 B
B-3001 Leuven, BELGIUM

Abstract

In cluster-based segmentation pixels are mapped 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 irregular clusters encountered in such segmentation-applications. The strength of our approach lies in the definition and use of two cluster-validity indices that are independent of the cluster-topology. By combining them, an excellent clustering can be identified, and experiments confirm that the associated clusters do indeed correspond to perceptually salient image-regions.

Keywords: Segmentation, clustering, non-parametric, content-based image retrieval, grouping, density-estimation, regions.

1 Introduction

Segmentation has always played a central role in computer vision and recent years have witnessed marked resurgence in interest, due in part to the growing importance of 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 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. Once a number of such regions have been identified, it becomes possible to quantify their visual characteristics, location and spatial organisation with respect to each other, all crucial factors in the interpretation-process.

Clearly, as a first step towards tackling this problem, we need to agree what sort of image-regions are interesting or perceptually salient. Saliency is here 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 readily 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.

This approach goes back at least to Coleman and Andrews [2], but 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 low and 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 au-
 systematically, 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, and we will base our clustering-algorithm on non-parametric density-estimation.

2 Proposed non-parametric clustering-algorithm

2.1 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 \( \{ x_i \in \mathbb{R}^n; i = 1 \ldots N \} \), a density \( f(x) \) is obtained by convolving the dataset with a unimodal density-kernel \( K_\sigma(x) \) (typically a Gaussian):

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

where \( \sigma \) is the size-parameter for the kernel, measuring its width. 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 \( \sigma \) of the convolution kernel \( K_\sigma \) or the number of neighbours \( k \)) are preset within a fairly narrow range, this procedure will result in either too many (if \( \sigma \) is chosen too small) or too few clusters (if \( \sigma \) is set too large). A major part of the work on density-estimation concerns itself with this problem of choosing an “optimal” value for \( \sigma \), but 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 \( \sigma \) which is small (with respect to the range of the dataset, see sect. 3) and, as before, proceed to identify candidate clusters by locating local maxima of the density \( f \). As mentioned above, this choice of \( \sigma \) 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 hierarchically ordered family of clusterings. (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 on the development of indices of cluster-validity that directly assign a performance-score to every proposed clustering of the data.

2.2 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 (see e.g. [3]). 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 ellipsoidally 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.

Although an exact definition is difficult to come by, 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 \( v_k(x) \) 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. 1). 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:

\[
\text{NN-norm: } \Lambda^{(k)} = \frac{1}{N} \sum_x \nu_k(x).
\]

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.

![Figure 1: Schematic representation 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.](image)

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 erroneously 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. 2). 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 \( b \), 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):

\[
\text{C-norm: } C = \frac{1}{K} \sum_{i=1}^{K} f(t_i)
\]

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 anchor-points makes the C-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.

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.

2.3 **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. 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 $\xi_i$ in a sample $\xi_1, \ldots, \xi_t$ with mean $\bar{\xi}$ and standard deviation $\sigma_\xi$, is defined to be the standardised deviation:

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

In fact, because the jumps cause the index-values for different clusterings to be rather irregularly spaced, additional sensitivity can be obtained by using $Z(\xi_i) = (\xi_i - \text{median}(\xi))/\text{MAD}(\xi)$, the robust version of the Z-score (MAD stands for median absolute deviation).

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 $C$ 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 $L_p(x)$, (in most cases, $p$ is equal to the total number of clusters in the clustering). Let $\mathcal{N}_p$ and $C_p$ be the corresponding NN-norm and C-norm respectively, as defined by eqs.(2) and (3). The (robust) Z-score for the $p^{th}$ clustering is then defined to be

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

and among the possible clusterings listed in the tree, we pick the one which maximises this robust Z-score.

### 3 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 and texture).

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 [4], Zhu et al. [6], Carson et al. [1], and Shi et al. [5].

#### 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 \( k = 10 \). The second is the width \( \sigma \) of the convolution kernel \( K_\sigma \) that is used to generate the nonparametric density \( f \). This is taken proportional to the average radius of the \( (\text{smallest}) \) 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 \( \sigma \) 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 datapoints \( N \). 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. Our algorithm successfully partitioned a number of artificial datasets with complicated structures (such as banana- or ring-shaped clouds) that tend to trip up the more conventional clustering-algorithms (we refer the reader to our website\(^1\)).

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. Some of the results are shown in Fig 3. The images in the first four rows are segmented by clustering in colour-space (in casu, opponent colours defined by \( Y_1 = R - G \), \( Y_2 = 2B - R - G \), and \( W = R + G + B \)). In the last row we show segmentation results based on texture-features. More precisely, in a window about each point we measured a small number of simple variability-coefficients, such as local variance and correlation-length. Clustering points with respect to these measures and mapping the identified clusters back into the image produced very acceptable segmentations.

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 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!

Conclusion

We have developed a generic and robust clustering-algorithm that determines both the shape and number of clusters. Segmentation based on the application of this algorithm in different feature-spaces, is able to extract perceptually salient regions from the image. Compared to classical clustering-algorithms such as K-means or Gaussian Mixture Models, our approach has the distinct advantage that (i) it determines the number of clusters automatically, and (ii) that it recognizes irregular (non-Gaussian) and unbalanced (large differences in cluster-size) clusters. Furthermore, being density-based it has none of the chaining-effects that vex hierarchical methods (which are distance-based).

We have illustrated the potential of the proposed algorithm on both artificial and real datasets. The latter were generated by mapping natural scenes in colour- or texture-spaces, but since the clustering-methodology is generic nothing prevents us from looking at other feature-spaces such as disparity, optical flow or edge-characteristics.

References


\(^1\)http://www.esat.kuleuven.ac.be/~frederix/segmentation.html
Figure 3: Segmentation based on clustering in colour- (top 4 rows) and texture-space (bottom row). Scanning from left to right, top to bottom, the number of regions (for colour-segmentation in mean or false colours) identified by clustering was: stork and flamingo, 2 regions each; monkey-scene, 3; butterfly, 3 (notice the small yellow region); trees, 4; deer-scenes, both 3; antelope, 2; and 3 regions in each of the textural scenes.