On Adapting Pixel-Based Classification to Unsupervised Texture Segmentation

Jaime Melendez\(^1\), Domenec Puig\(^1\), Miguel Angel Garcia\(^2\)
\(^1\) Intelligent Robotics and Computer Vision Group
Rovira i Virgili University, Dept. of Computer Science and Mathematics,
Av. Països Catalans 26, 43007 Tarragona, Spain
\(^2\) Autonomous University of Madrid, Dept. of Informatics Engineering
Francisco Tomas y Valiente 11, 28049 Madrid, Spain.

Abstract

An inherent problem of unsupervised texture segmentation is the absence of previous knowledge regarding the texture patterns present in the images to be segmented. A new efficient methodology for unsupervised image segmentation based on texture is proposed. It takes advantage of a supervised pixel-based texture classifier trained with feature vectors associated with a set of texture patterns initially extracted through a clustering algorithm. Therefore, the final segmentation is achieved by classifying each image pixel into one of the patterns obtained after the previous clustering process. Multi-sized evaluation windows following a top-down approach are applied during pixel classification in order to improve accuracy. The proposed technique has been experimentally validated on MeasTex, VisTex and Brodatz compositions, as well as on complex ground and aerial outdoor images. Comparisons with state-of-the-art unsupervised texture segmenters are also provided.

1. Introduction

Image segmentation aims at clustering image pixels into salient homogeneous regions. This is a difficult task, since it requires the detection of uniform features within each region, as well as the location of the boundaries among regions.

Texture segmentation can be supervised or unsupervised depending on whether prior knowledge about the image or its texture classes is available or not. Supervised texture segmentation identifies and separates regions that match texture properties previously learnt in training samples. In turn, unsupervised texture segmentation has to discriminate the texture classes of the image before separating them into regions. Compared to supervised segmentation, the unsupervised case is more flexible, although more challenging.

Despite the success of many of the unsupervised texture segmentation algorithms proposed in the literature, their supervised counterparts usually perform better in terms of segmentation accuracy as demonstrated in previous work (e.g., [3]). The reason is that, by definition, unsupervised texture segmentation algorithms do not take advantage of the prior knowledge corresponding to the texture patterns to be discriminated, since such information is not available. On the contrary, supervised algorithms are specifically trained to identify a group of patterns, thus being more likely to succeed.

This paper proposes a two-stage unsupervised texture segmentation technique that aims at incorporating the aforementioned specificity by stacking a supervised classifier on top of an unsupervised segmentation algorithm. During the initial stage, a clustering algorithm is used to discover the texture patterns of a given image. Since the objective of this stage is only to find a group of suitable patterns and not to accurately define the image’s regions, this process is carried out using a small number of feature samples, thus significantly reducing the computational cost. In the second stage, a number of non-boundary pixels corresponding to each texture pattern is selected and the feature vectors associated with them are used to train a supervised classifier. This classifier yields the final segmented image after performing pixel-based classification considering the complete set of feature vectors. Multi-sized evaluation windows following a top-down approach are utilized in order to improve accuracy within regions of homogeneous texture, as well as near boundaries. Experiments show that the proposed technique is effective in terms of both seg-
mentation quality and computation time.

The organization of this paper is as follows. The pattern discovery stage is described in Section 2. Section 3 details the classification methodology applied in the second stage. Section 4 shows and discusses experimental results. Finally, conclusions and further improvements are given in Section 5.

2. Pattern Discovery Through Graph Clustering

A clustering algorithm is first applied to a subset of feature vectors in order to discover the texture classes present in the processed image.

These vectors have been obtained through a Gabor filter bank with four scales, six orientations and a range of frequencies between 0.05 and 0.4 as in [7]. The texture features that characterize a pixel are the mean and standard deviation of the module of the resulting coefficients evaluated over that pixel and its surrounding neighborhood (evaluation window).

Clustering is performed by GRACLUS [2], a graph clustering technique that avoids the eigenvector computation originally proposed in [10] for optimizing the normalized cut criterion. In this context, each node of the graph is a feature vector and the weight of the edge between every pair of nodes is the Euclidean distance between them in the feature space.

In order to apply GRACLUS to the pattern discovery stage of the proposed technique, the problem of specifying the number of clusters must be addressed. Usually, the alternative is to first bipartition the whole graph and then bipartition the already segmented parts if the normalized cut is below a specified value [10]. In this work, a slightly different approach, which leads to higher accuracy, has been applied. It first partitions the whole graph into two clusters and then verifies the value of the normalized cut. If this value does not reach a specified threshold, the algorithm increments the number of clusters by one and repartitions the whole graph using this new setting. This process is repeated until the desired threshold is satisfied.

3. Supervised Pixel-Based Classification

The texture patterns found by the previous pattern discovery stage are used as texture models for a supervised pixel-based classifier, thus effectively transforming the original unsupervised problem into a supervised one.

A pixel-based classifier aims at determining the class to which every pixel of an input image belongs based on several measures computed by applying a set of texture feature extraction methods, such as the Gabor filters described in the previous section, which leads to the segmentation of the image as a collateral effect.

In order to take advantage of the output of those methods, texture features have been computed for different evaluation window sizes and processed during classification by following the top-down approach described below:

1. Select the largest available evaluation window and classify the test image pixels labeled as unknown (initially, all pixels are labeled as unknown).
2. Locate the pixels of the classified image that constitute the frontiers between regions of different texture and mark them as unknown, as well as their neighborhoods. The size of the neighborhood corresponds to the size of the evaluation window used to classify the image.
3. Discard the current evaluation window.
4. Repeat steps 1 to 3 until the smallest evaluation window has been utilized.

In this way, large windows are applied inside regions of homogeneous texture in order to avoid noisy classified pixels and small windows are applied near the frontiers between those regions in order to refine them. Furthermore, the above strategy renders classifying every image pixel with all the available windows unnecessary. Hence, it leads to low computation times.

Pixel classification in the previous scheme is performed by means of SVMs. Since an SVM is a binary classifier, an extension is needed in order to solve multiclass problems. One-against-one SVMs [5], which separate every class from any other, have been applied, since they are the best alternative both in terms of classification accuracy and computation time, as only “small”, two-class problems need to be solved.

4. Experimentation

The proposed methodology has been evaluated on composite images of well-known Brodatz [1], Vistex [8] and MeasTex [11] textures, and on real outdoor images taken both at ground level and by aerial devices (e.g., Fig. 1), and compared with other unsupervised texture segmentation techniques in terms of segmentation quality and computation time. These alternative techniques are: EdgeFlow [6], locally-sensitive hashing-based adaptive mean shift (LSHAMS) [4], the spatial/spectral segmenter (SSS) introduced in [9] and
Figure 1. Examples of synthetic compositions and outdoor scenes.

the compression-based texture merging (CTM) algorithm [12]. Segmentation results obtained by GRACLUS [2] have also been included as a reference benchmark.

All of the evaluated techniques have been optimally configured for each test image. For those that are not designed to work with multiple evaluation window sizes, the single size that yields the best score along the entire image database or the one suggested by the authors has been set.

The quality of the segmentation maps produced by the evaluated approaches has been measured by the following segmentation quality factor:

\[ Q = \frac{1}{A_I} \sum_{r=1}^{R} A^g_r \cap A^s_r, \]

where \( A^g_r \) is the area of the \( r \)-th region in the groundtruth, \( A^s_r \) is the area of its corresponding region in the evaluated segmentation map, \( A^g_r \cap A^s_r \) is the area of the overlapping portion between both regions and \( A_I \) is the area of the whole image. The intersection between \( A^g_r \) and \( A^s_r \) is an indicator of how good the segmentation of that particular region is. The area of a region is defined as its number of pixels. The above quality factor is normalized between zero and one, although the \([0, 100]\) interval has been preferred.

Average results obtained after experimentation are summarized in Table 1. All experiments have been run on a Pentium IV at 3.2 GHz. Since the CTM algorithm works with the \( L^*a^*b^* \) color space, only those images that have a color version have been considered when evaluating its performance. Moreover, the publicly available CTM implementation is written in C (mex files) and Matlab, hence the computation time corresponding to the latter part, which is around 50% of the total, is expected to decrease if it were also written in C. Finally, due to the lack of memory to process a complete dissimilarity matrix, only 1/4 of the feature vectors have been fed to GRACLUS.

Results in Table 1 show that the proposed technique obtains higher segmentation quality scores than the remaining approaches. For instance, it is superior to the closest approach, which is GRACLUS for the synthetic compositions and CTM for the outdoor scenes, by around five points. Fig. 2 and Fig. 3 show some segmentation maps produced by these relevant approaches.

The reason for that higher performance is that the multiwindow approach followed by the classification stage performs both pattern refinement in the feature space and boundary refinement in the image domain. Hence, appropriate discrimination between patterns is possible.

In terms of computation time, the proposed technique is among the fastest. Compared to GRACLUS and CTM, it is more than four and ten times faster respectively. Hence, the higher segmentation quality discussed above comes along with a considerable saving in computation time due to: a) The efficient use of multiple evaluation window sizes following a top-down approach. In general, the smaller the window, the more expensive the classification, as features become less discriminating and yield a higher number of support vectors. Since only frontier pixels are classified each time a smaller window size is used, significant computation is saved. b) The reduced number of feature vectors processed by the pattern discovery stage, as only a rough approximation of the image’s texture patterns is needed for obtaining a good final segmentation.

### 5. Conclusions

This paper has presented a new unsupervised texture segmentation technique based on a supervised pixel-based classifier that achieves good segmentation quality with low computational time. A pattern discovery stage is applied in order to identify the texture patterns of a
given image by means of a clustering algorithm, thus effectively transforming the initial unsupervised problem into a supervised one, which allows the proposed technique to benefit from the advantages of supervised classifiers.

The proposed approach has been compared with the baseline clustering algorithm applied during the pattern discovery stage and with alternative segmentation techniques. Results in terms of segmentation quality and computation time have always been favorable.

Future work will consist of investigating different multi-sized evaluation window schemes in order to improve both the accuracy and the computation time of the pixel-based classification stage of the current algorithm. In addition, it is possible to extend the proposed methodology in order to integrate a broader variety of feature extraction methods in a coherent way.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Education and Science under project DPI2007-66556-C03-03.

References