Automatic Fuzzy Segmentation of Textural Images using Adaptive Divergence Affinity Functions

José Silva Neto*, Waldson Leandro*, Matheus Gadelha‡, Tiago Santos§, Bruno M. Carvalho*, Edgar Garduño¶
*Federal University of Rio Grande do Norte, Natal, Brazil
†University of Massachusetts, Amherst, USA
‡Brazilian Federal Rodoviary Police, Guara, Brazil
§National Autonomous University of Mexico, Mexico City, Mexico
bruno@dimap.ufrn.br

***Abstract*—The goal of digital image segmentation is to assign different labels to different objects present in an digital image or volume. A wide variety of sources have been successfully segmented by the traditional fuzzy segmentation algorithm on several different applications. However, the traditional approach of the fuzzy segmentation algorithm usually does not work well when dealing with objects whose materials are characterized by complex textures that cannot be accurately represented by the simple fuzzy affinity functions of the traditional fuzzy segmentation algorithm. In this paper, we extend the fuzzy segmentation algorithm to use adaptive textural affinity functions for segmenting these objects in 2D images. These adaptive affinity functions define their optimal appropriate neighborhood size on execution time, by computing texture descriptors surrounding the seed spels, according to the characteristics of the texture being processed. The algorithm then segments the image with the appropriate neighborhood, that may be different for each object. We performed experiments on mosaic images that were composed using images from the Brodatz database, and compared our results with the ones produced by a recently published texture segmentation algorithm, showing the applicability of our method.**

***Index Terms*—Fuzzy segmentation, Texture segmentation, Adaptive fuzzy affinity functions, Skew divergence, Multi-dimensional scaling**

I. Introduction

The goal of digital image segmentation is to assign different labels to different objects present in an digital image or volume. It may seem like a simple process, since us, humans, and even animals can usually do so with ease. However, we perform this task by making use of several high-level information, such as parallax, relative size of objects, shadowing, occlusion and mainly our cumulative *a priori* knowledge about previously seen objects. Inserting this type of information into a computer program is not easy, and the initial problem itself may become even harder due to the presence of noise, uneven illumination, or when we have textural properties characterizing the material of an object.

Textures are described as the feel, shape, appearance or even consistency of a surface or substance. Since we are dealing here with digital images, we do not have to worry about the shape, feel or consistency, that are associated with physical textures. On the other hand, in the fields of Visual Computing, visual textures are defined by statistical distributions in arrangements that can be structural, with the repetition of a pattern, characterized by the kernel of the texture, or purely statistical (also called non-structural).

The field of texture analysis can involve the segmentation, classification or characterization of textures and it is a very challenging task, not only because of the nature of the textures, but also because of the results that changes in scale, illumination or noise may have on how we perceive them. In this work, we are concerned with the segmentation of images formed by a composition of natural statistical textures, by using a semi-automatic, interactive region-growing segmentation method, called *MOFS (Multi Object Fuzzy Segmentation)* [1]. Here, we extend the method through the usage of adaptive fuzzy affinity functions [2] that are tailored for characterizing and comparing textures by adopting scales (kernel sizes) that are associated to the statistical distributions of the textural patterns.

The interactive region-growing method used to segment the images is based on the concept of fuzzy connectedness, that has been successfully employed to segment images affected by noise and uneven illumination, as can be seen in [3]. The original fuzzy segmentation method used here, called MOFS, was first proposed in [1] and improved in [4]. This method is a semi-automatic method that belongs to the class of region-growing segmentation algorithms, where the user interaction with the method happens in the form of selected seed spels (short for spatial elements), such as pixels in the case of 2D images. However, we also propose in this article a method for automating the seed selection, thus transforming the method into a fully automatic texture segmentation method.

II. Related Work

There are many published methods for segmenting, classifying or characterizing textures, both for the periodic (structural) and statistical (non-structural) cases. Here, we are concerned with the task of unsupervised segmentation of images which may contain objects that exhibit statistical textures.

A classical approach for this segmentation task was proposed by [5], where they build a bank of Gabor filters that attempt to cover the spatial-frequency domain uniformly. The selection strategy used chooses the filters based on quality of the reconstruction of the input image produced from the filtered images. Then, the selected filtered images are used to compute measures of energy at different windows to generate feature

images. The feature images are then integrated in the final segmentation by means of a clustering step, that may incorporate spatial information, in order to increase the smoothness of the solution. A completely different classical approach to perform unsupervised texture segmentation that also employs Gabor filters is the one proposed by [6], that performs several statistical tests to measure the different degrees of homogeneity. The authors then reformulate the texture segmentation task as a clustering problem, like many other approaches do, and solve it by using multiscale Gabor filters to compute dissimilarities.

Another approach that performs unsupervised texture segmentation is the one proposed by [7] that uses a framework to learn textural pattern features. It does so by using a cost function that matches the piecewise constant Mumford-Shah model. This forces the features to provide an approximately piecewise constant feature image with a small training set.

There are also several classical model-based approaches [8], [9] that tackle the texture segmentation problem by modeling the intensity field of textures as a Gauss-Markov or as a Markov random field to represent the local spatial dependencies between pixel intensities. However, these techniques are usually very computationally intensive because they require a large number of iterations to converge. On a recently published approach, [10] defined a Laplace mixture model coupled with a wavelet-domain hidden Markov tree (LMM-HMT) to perform texture segmentation. There, they train their LMM-HMT with homogeneous texture samples using the tree-structure Expectation-Maximization (EM) algorithm, and estimate the likelihood of each subtree at the different scales, before performing a multiscale fusion and pixel-level segmentation.

III. Fuzzy Segmentation

The MOFS [1] is a segmentation method that employs the concept of fuzzy connectedness to segment multiple objects simultaneously. Since it defines a very general segmentation setting, it can be applied to segment data organized in arbitrary digital spaces [11]. In order to match this very general setting, in its theory, we refer to the elements of the image (represented by the set V) as *spels*, which is the short for *spatial elements*. Thus, the spels can be a variety of things, such as the traditional pixels of an image [4], feature vectors [12] (in the case of multispectral data), dots in the plane or voxels on a volume [13], [14].

The idea behind this approach is to partition the set V, and, in doing so, assign a fuzzy grade of membership, i.e., a number between 0 and 1, for all spels in V to one or more objects. We do that by assigning, to every ordered pair (c, d) of spels, a *fuzzy connectedness* of c to d, which is also a number between 0 and 1. In [4], the original theory of the method was maintained but a new, more efficient algorithm was proposed. The theory of the method defines *chains* of spels as sequences of ordered pairs of spels (*links*). Each link has associated *ψ-strengths* that are computed by the *fuzzy spel affinity* functions $\psi : V^2 \rightarrow [0, 1]$, while a chain also has a ψ-strength, that is defined as the value of the ψ-strength of its weakest link.

Then, a *fuzzy connectedness* function $\mu_\psi : V^2 \rightarrow [0, 1]$ is used to compute the fuzzy connectedness from c to d. The function μ_ψ returns the ψ-strength value of the strongest chain connecting c and d. The picture formed by the fuzzy connectedness from a spel o to all spels $c \in V$ is called the *ψ-connectedness map* f, where $(f(c) = \mu_\psi(o, c))$, for all $c \in V$. Then, extending this idea to multi-object segmentation, we have M objects with their associated ψ_m fuzzy affinity functions. Finally, an *M-segmentation* of V maps each $c \in V$ into an $(M+1)$-dimensional vector $\sigma^c = (\sigma_0^c, \sigma_1^c, \cdots, \sigma_M^c)$, where σ_0 denotes the maximal fuzzy connectedness computed for the M objects which are stored in the σ_m for each m $(1 \leq m \leq M)$.

For a more detailed description of this segmentation method, we recommend the reader to [4], where the Fast-MOFS algorithm, a more efficient implementation of the same method, was proposed. In this article, we use the Fast-MOFS as a base for the proposed changes to the segmentation method.

A. Defining Fuzzy Affinity Functions

As mentioned before, the MOFS was developed as a semi-automatic algorithm, and thus, the fuzzy spel affinity functions ψ_m and the seed spel sets V_m $(1 \leq m \leq M)$ were determined by the user using the following procedure. The user clicked on spels, identifying them as belonging to the object m (inserting them into V_m). Then, the fuzzy spel affinity functions ψ_m were defined by computing the mean (g_m) and the standard deviation (h_m) of the average intensity for all adjacent pairs of spels in V_m and their neighborhoods, and the mean (a_m) and the standard deviation (b_m) of the absolute differences of brightness for the same neighborhoods. Then, the affinity functions were defined as:

$$\psi(c, d) = \begin{cases} \frac{\rho_{g_m, h_m}(g) + \rho_{a_m, b_m}(a)}{2}, & \text{if } c \text{ and } d \text{ are edge-adjacent,} \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where g is the mean and a is the absolute difference of the intensities of c and d and $\rho_{r,s}(x)$ is the probability density function of a Gaussian distribution with mean r and standard deviation s.

We can see in Equation (1) that the first term measures the degree of homogeneity in a local way, by computing the average of the two intensities and passing it to the fuzzy affinity function, while the second term computes the absolute difference between the two intensity values, capturing some inhomogeneity information, that can be looked at as a very basic textural information.

B. Textural Fuzzy Affinity Functions

While Equation (1) has been successfully used in several types of textures [2], images [3], [4], and even grids [15], we detected that it cannot capture enough information about more complex textures, specially the ones whose textural characteristics can only be described at larger scales. By analyzing the results, we detected the need to encode more information about the intensity distribution into the affinity functions. The idea

is to compare intensity distributions, thus providing a more powerful and robust way to describe the visual characteristics of textures.

In the fields of Statistics and Information Theory, the well-known Kullback-Leibler (KL) divergence [16], also known as relative entropy, is a non-symmetric measure of the distance or difference between two probability distributions q and r. In Information Theory, the KL divergence of r from q, denoted by $KLD(q||r)$, is a measure of the information lost when r is used to approximate q, and is given by Equation 2:

$$KLD(q||r) = \sum_{y} q(y)(\log q(y) - \log r(y)). \tag{2}$$

The KL divergence is a tool to measure how far away two distributions are. However, the KL divergence is not defined when $r(y) = 0$, and that happens quite often when we are dealing with distributions that are characterized by histograms with few counts. For example, if we are dealing with monochromatic images with 256 gray levels and neighborhoods of size 11×11, we are characterizing a distribution by using a histogram with 256 bins and whose total count is only 121. In order to deal with this problem, Lee [17] proposed an approximation to the KL divergence, that is given by Equation 3:

$$SD_{\alpha}(q, r) = D(r||\alpha q + (1 - \alpha)r), \tag{3}$$

for $0 \leq \alpha \leq 1$, where α can be seen as a degree of confidence in the distribution q and $(1 - \alpha)$ as the amount of smoothing of the distribution q over r. This allows us to work nicely with sparse data problems, such as the low count histograms of small neighborhoods.

C. Adaptive Fuzzy Affinity Functions

If one is dealing with structural textures, there are several methods for determining the window size of its kernel. However, the kernel is not defined in statistical textures, where the textures can have distributions with very different characteristics at different scales.

In this work, as it was done in [2], we employ adaptive fuzzy affinity functions, that select the neighborhoods sizes by observing and comparing the intensity distributions surrounding the selected seed spels at adjacent scales. Thus, these neighborhood sizes are related to the scales of the textures being segmented. In this approach, we use the adaptive neighborhoods, but differently to what was done in [2], the computation of a fuzzy spel affinity uses the distributions observed over these areas.

We find the appropriate scale for each object by computing statistics for a neighborhood of size 3×3 surrounding the seed spels, and increasing this neighborhood until the differences of the statistics collected on the neighborhoods of two consecutive sizes are smaller than predefined thresholds or we reach a pre-determined maximum size. We do that by computing a measure of the distance between the regions surrounding two seed spels. In both cases, when using the Kullback-Liebler or skew divergences, we set the minimum value of 0.8 for the divergence measure as a threshold. These thresholds were

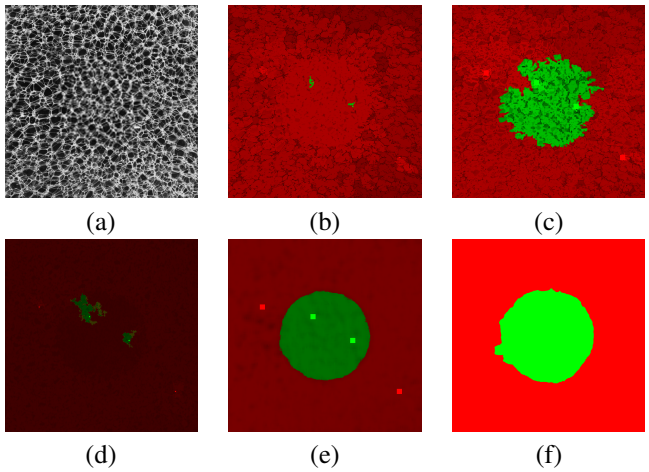

Fig. 1. Original Mosaic produced by combining the same texture at two different scales (a) and correspondent connectedness maps produced using the Gaussian affinity function with a 3×3 (b) and adaptive (c) neighborhoods, and the Skew Divergence affinity function with a 3×3 (d) and adaptive (e) neighborhoods, and the segmentation produced by the method of Yuan et al. [18].

selected empirically for detecting the point when the collected statistics stabilize. The function used to compute the statistics is generic in the sense that it can work with square, circular (digital ball), hexagonal or any other neighborhood shape, following the general definitions given in [11].

D. Automatic Segmentation

Since our method performs semi-automatic segmentation,the quality of the final result is strongly correlated to the initial seed positions, which are selected manually. We overcome this issue by selecting the seeds through a clustering method in a embedded space induced by the skew-divergence metric. By using this approach, the whole segmentation process can be done without relying on human selection of the seeds, just the number of classes. Our procedure is defined as follows.

We start by dividing the input image in $N \times N$ patches, where π_i is the ith patch. Then, we compute the pixel intensity histogram H_i for each patch. Using this histogram, we compute a pairwise distance matrix D using the following equation:

$$D_{ij} = \frac{1}{2}\Big(KL(H_i||H_j) + KL(H_j||H_i)\Big) + \lambda \left|\mu(\pi_i) - \mu(\pi_j)\right|_2, \tag{4}$$

where λ is a user-defined weight and $\mu(\pi)$ is an operator that returns the mean coordinate of pixels in the patch π. After preliminary experiments, we decided to set $\lambda = 0.5$ in all of our following experiments. The reason for adding the second term is to increase the distance between patches that are far from each other in the original image. This is particularly useful in our region-growing based method: distant pixels have a smaller probability of belonging to the same class of the original seeds. By varying the value of λ, we can give more or less importance on how close can be different connected components from each other when belonging to the same object, with $\lambda = 0$ allowing for two connected components to be labelled as part of the

same object, regardless of the spatial distance separating them. The matrix D is used to create an embedded space through Multi-Dimensional Scaling (MDS) [19].

In other words, MDS will output a set $\mathcal{P}$ in a 3-dimensional space that preserves the distances in D. The set $\mathcal{P}$ and the number of classes K will serve as an input to the K-means clustering [12] procedure that will label each point $p \in \mathcal{P}$ as belonging to one class. From the MDS computation and the K-means, we can associate each patch π to a class c. Using this information, we define K disjoint sets of patches $\Pi_1, \Pi_2, ..., \Pi_k$. The centroid $\bar{\mu_c}$ of a region assigned to a class c is computed as follows. The optimal number of classes K can be deduced as the result of an algorithm such as the G-means, proposed by [20], which runs the K-means with increasing values of K in a hierarchical manner until all the tests that the data assigned to each cluster are Gaussian are true.

$$\overline{\mu_c} = \frac{1}{|\Pi_c|} \sum_{\pi \in \Pi_c} \mu(\pi) \tag{5}$$

One could simply use μ_c of Equation (5) as a seed position, but we noticed through experimentation that sampling multiple seeds for each class is more effective than using a single seed. Thus, we select the seeds of a class $c \in [1, k]$ by sampling from a multivariate normal distribution $\mathcal{N}(\bar{\mu_c}, I)$, where I is a 2×2 identity covariance matrix. We used 3 samples per class in all automatic Fast-MOFS experiments shown in this paper.

IV. Results and Discussion

In order to show the applicability of our approach, we first selected 10 textures from the Brodatz textures data set [21] that exhibit features on several different scales. We selected other Brodatz textures that were not used in the segmentation experiments shown below to determine empirically values for the thresholds mentioned in Section III-C, that were set to 0.06 and 0.04 for the mean and standard deviation, respectively.

A. Experiments

All experiments were run on a notebook equipped with an Intel i5 CPU running at 1.8 GHz and 6GBs of RAM. Figure 1 shows the segmentations of one 640×640 image that was composed by putting together two objects made out of the same texture from the Brodatz data set but on different scales in order to show the importance of the scale computation to properly characterize a texture. In this figure, we display the connectedness maps with the color indicating the object to which the spel belongs to, while the intensity is proportional to the grade of membership of that spel to the object in question, i.e., spels that appear very dark have a low grade of membership to the object to which they belong to. In this experiment, two seeds were selected for each texture, and the scale for each object was determined according to the procedure described on Section III-C. We used square neighborhoods and the largest neighborhood used was 15×15. It is clear from the results that our approach had no problem in segmenting these textures when using adaptive fuzzy affinity functions. One can clearly see on Figures 1(b) and 1(d) that the neighborhood size of 3×3 was not sufficiently large to characterize the texture pattern of the objects. Not only it led to a very poor segmentation, but also to very low connectedness values (as we can see from the overall low intensity values of the map), or very low confidence on the result. It is important to point out that the colors shown on Figure 1(e) do not change in intensity within the same object because the method used for producing it is crisp, i.e., is not fuzzy in nature, and thus, does not compute a grade of membership for each spel.

Mosaic	MOFS Skew		Approach of [18]			
			60×60		120×120	
	Dice	Time(μs)	Dice	Time(μs)	Dice	Time(μs)
M2-1	**0.994**	32.22	0.986	6.91	0.980	17.68
M5-1	**0.987**	30.84	0.670	9.90	0.955	28.81
M5-2	**0.988**	31.72	0.600	10.92	0.615	26.86
M5-3	0.966	31.35	0.690	12.88	**0.974**	24.88

TABLE I
Accuracies of fuzzy segmentations produced using the Gaussian affinities and Skew affinities with a pre-defined affinity neighborhood and the automatic adaptive neighborhood.

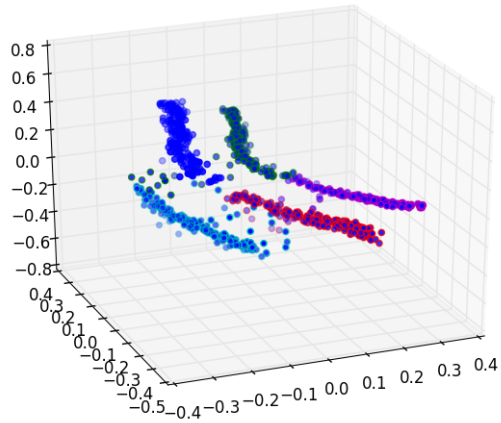

Fig. 2. Clustering into 5 objects of the 64×64 patch centers for Mosaic M5-1 (top left image of Figure 3) where can see the clear separation of the patches belonging to the different textures. (The three axes are related to the normalized coordinates of the color space)

We tested the automatic seed selection algorithm on three mosaics of size 1280×1280 composed of five textures obtained from the Brodatz Album [21]. To illustrate the process, we show on Figure 2 the clustering results for the mosaic shown on Figure 3(a). The K-means clustering results show 5 distinct sets of points, which means that selecting seeds near the centroid of each set may give us good seed candidates. After we choose the seeds for each class, we run the MOFS algorithm to segment the mosaic. In our experiments, the mosaics were divided in 64×64 patches with 20×20 pixels each. The statistics are then collected over the neighborhood whose size was computed as mentioned before.

Several papers have emerged in the area of image segmentation, especially using approaches based on training and deep learning [22], [23]. However, these works are proposed to

segment images of generic characteristics, without any specialization for textural cases. Therefore we decided to compare the results obtained by our algorithm with the one proposed in [18] because, like our method, it proposes a method for segmenting images with textural properties requiring little to no training. The work of [18] presents a texture segmentation method based on a matrix factorization that is obtained through the use of pre-defined filters which act as texture descriptors for the image. Thus, the choice of these filters and the size of the integration scale (analogous to the size of neighborhood used in our approach) are determinant to the accuracy of the segmentation. In our experiments, we used the same filter bank proposed by Yuan et al. [18], but varied the integration scale between 60×60 (value used in the paper) and 120×120, for which we obtained the best segmentation results.

In Figure 3, we compare the results obtained using such configurations to the ones achieved by the automated version of MOFS using the two previously defined affinity functions and a maximum neighborhood window size of 15×15. In this figure, we show segmentation maps that do not code the fuzzy membership, for comparisons reasons. We computed Dice coefficients [24] for each object and weighted them by the object sizes on the ground truth image, as suggested by [25], producing a single score for each segmentation, which are shown on Table I. The execution times shown in the table are expressed in microseconds and reported on a per pixel basis, for ease of comparison.

We can notice that the increase of the integration scale (bottom row compared with fourth row of images on Figure 3) tends to smooth the boundaries obtained by the segmentation, as well as to improve its accuracy. However, increasing it too much may lead to distorted segments that do not resemble the original texture's shape. Unlike our approach, for which we automatically chose an optimum window size to be used in affinity functions calculations, the method proposed by [18] was shown to be quite sensitive to the choice of the integration scale. In that sense, we ran their method as a semi-automatic one, by trying different scales and selecting the best one.

In addition to the fact that our method achieved better segmentation accuracy on three out of four cases, it also obtained boundaries that are closer to the original mosaics, an important requirement for various texture segmentation applications [26]. In the case where our method did not achieve the best segmentation accuracy, obtaining a Dice coefficient of 0.966 compared to a Dice coefficient of 0.974 of the [18] method, the region growing of the bottom left object was blocked by a white streak on that object. On a semi-automatic setting, the user would add another seed spel and execute the algorithm again. That is a flexibility that allows the method to be used on a general application setting.

Both algorithms have similar execution times - in the same order of magnitude, with our method being between 2.4 and 4.7 times slower than the approach of [18]. Since we are now working on a parallelization approach of the MOFS algorithm and considering that there is no requirement for executing this task in real time, such a time difference becomes less important.

V. Conclusion

In this article, we propose an extension to the fuzzy segmentation approach advocated by [4], that allows the method to perform a more robust segmentation of objects that contain texture properties. We used two different methods to automate two steps of the process, the definition of the appropriate scale for each fuzzy spel affinity and the selection of the seed spels for the M objects. The fuzzy affinity functions can take several formulas, such as the Kullback-Leibler or the skew divergences, to compute the affinity between the histograms of neighbor windows. We then applied our method to the segmentation of mosaics made up of real Brodatz textures. The results produced by our method showed its effectiveness and can be compared favorably to a recently published method [18], producing more accurate boundaries.

As mentioned before, we are currently working on defining the best approach to parallelize the Fast-MOFS algorithm. The idea is to use it for segmenting large 2D images or 3D volumes that have objects with textural properties in them and also to perform dynamic texture segmentation. Image modalities that produce images wit low SNR appear as likely candidates for the proposed method, including optical microscopy of petrographic data, ultrasound images, low-dose computerized tomography or some types of radar imagery.

Acknowledgment

The authors would like to thank the financial support provided by the Coordenação de Aperfeioamento de Pessoal de Nvel Superior Brasil (CAPES) Finance Code 001, during the development of this work.

References

[1] G.T. Herman and B.M. Carvalho. Multiseeded segmentation using fuzzy connectedness. *IEEE Trans. Pat. Anal. and Mach. Intel.*, 23(5):460–474, 2001.

[2] B. M. Carvalho, T. S. Souza, and E. Garduño. Texture fuzzy segmentation using adaptive affinity functions. In *ACM SAC*, pages 51–53, 2012.

[3] B M Carvalho, L M Oliveira, and E Garduno. Semi-automatic single particle segmentation on electron micrographs. In *International Symposium on Biomedical Imaging:From Nano to Macro*, pages 1024–1027, Washington, DC, USA, April 2006. IEEE.

[4] B. M. Carvalho, G. T. Herman, and T. Y. Kong. Simultaneous fuzzy segmentation of multiple objects. *Discr. Appl. Math.*, 151(1-3):55–77, October 2005.

[5] Anil K. Jain and Farshid Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recogn.*, 24(12):1167–1186, December 1991.

[6] Thomas Hofmann, Jan Puzicha, and Joachim M. Buhmann. Unsupervised texture segmentation in a deterministic annealing framework. *IEEE Trans. Pat. Anal. Mach. Intel.*, 20(8):803–818, 1998.

[7] M. Kiechle, M. Storath, A. Weinmann, and M. Kleinsteuber. Model-based learning of local image features for unsupervised texture segmentation. *IEEE Transactions on Image Processing*, 27(4):1994–2007, April 2018.

[8] B. S. Manjunath and R. Chellappa. Unsupervised texture segmentation using markov random field models. *IEEE Trans. Pat. Anal. Mach. Intel.*, 13:478–482, May 1991.

[9] R. Robertson, E. Villebrun, and P. Hoeher. A comparison of optimal and suboptimal map decoding algorithms operating in the log domain. In *IEEE Int. Conf. on Commun.*, volume 2, pages 1009–1013, 1995.

[10] Yulong Qiao and Ganchao Zhao. Texture segmentation using Laplace distribution-based wavelet-domain hidden Markov tree models. *Entropy*, 18(11), 2016.

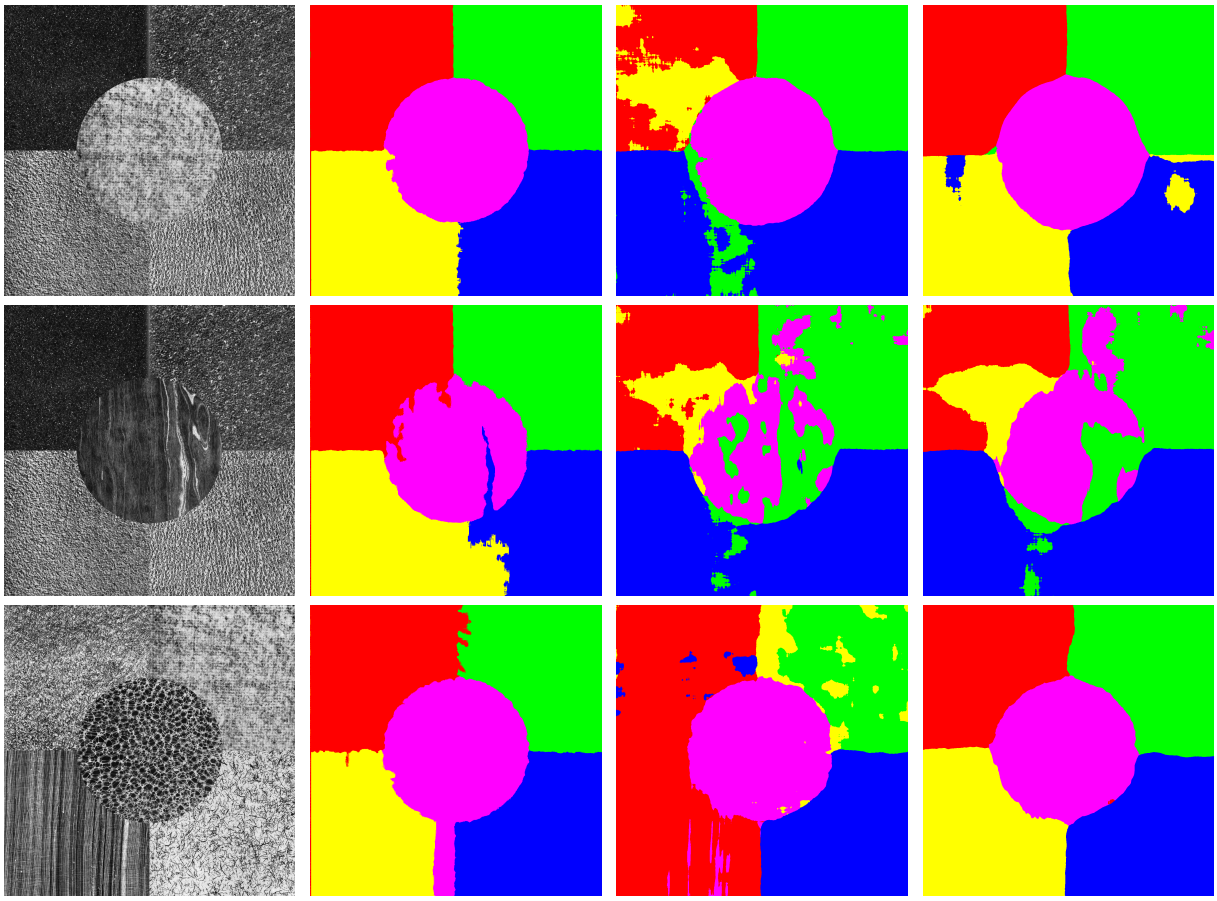

Fig. 3. The first column shows 3 original 1280×1280 mosaics produced by combining five Brodatz Textures, while the second column shows the crisp segmentation maps for our proposed method. The third and fourth columns show the results produced by the method of [18] with integration scales of 60×60 and 120×120, respectively.

[11] G.T Herman. *Geometry of Digital Spaces*. Springer, 1998.

[12] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2Nd Edition)*. Wiley-Interscience, 2000.

[13] B. M. Carvalho, L. M Oliveira, and G. S. Andrade. Fuzzy segmentation of color video shots. In *13th International Conference on Discrete Geometry for Computer Imagery (LNCS)*, volume 4245, pages 494–500, Berlin, 2006. Springer.

[14] Bruno M. Carvalho, Edgar Garduño, Tiago S. Santos, Lucas M. Oliveira, and José F. Silva Neto. Fuzzy segmentation of video shots using hybrid color spaces and motion information. *Pattern Analysis and Applications*, 17(2):249–264, May 2014.

[15] B.M. Carvalho, E. Garduño, and G.T. Herman. Multiseeded fuzzy segmentation on the face centered cubic grid. In S. Singh, N. Murshed, and W. Kropatsch, editors, *Advances in Pattern Recognition: Second International Conference - ICAPR 2001 (LNCS 2013)*, pages 339–348, Rio de Janeiro, Brazil, 2001. Springer-Verlag.

[16] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951.

[17] L. Lee. Measures of distributional similarity. In *Proc. of Ann. Meet. of the ACL on Comput. Linguist.*, ACL '99, pages 25–32, Stroudsburg, PA, USA, 1999.

[20] Greg Hamerly and Charles Elkan. Learning the K in K-means. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 281–288. MIT Press, 2004.

[18] Jiangye Yuan, DeLiang Wang, and Anil M. Cheriyadat. Factorization-based texture segmentation. *IEEE Trans. Im. Proc.*, 24:3488–3497, 2015.

[19] I. Borg and P.J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[21] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover Publications, New York, 1966.

[22] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intel.*, 39(4):640–651, April 2017.

[23] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, and Andrea Vedaldi. Deep filter banks for texture recognition, description, and segmentation. *Int. J. of Comp. Vis.*, 118(1):65–94, May 2016.

[24] L. R. Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297–302, July 1945.

[25] W. R. Crum, O. Camara, and D. L. G. Hill. Generalized Overlap Measures for Evaluation and Validation in Medical Image Analysis. *IEEE Trans. Med. Im.*, 25(11):1451–1461, November 2006.

[26] Rishi Jobanputra and David A. Clausi. Preserving boundaries for image texture segmentation using grey level co-occurring probabilities. *Pat. Rec.*, 39(2):234 – 245, 2006.