Texture Filters and Fractal Dimension on Image Segmentation

Beatriz Marrón

Universidad Nacional del Sur, Bahía Blanca, Argentina

Email: beatriz.marron@uns.edu.ar

Abstract

Texture analysis is important in several image segmentation and classification problems. Different image textures manifest themselves by dissimilarity in both the property values and the spatial interrelationships of their component texture primitives. We use this fact in a texture discrimination system. This paper focuses on how to apply texture operators based on co-occurrence matrix, texture filters and fractal dimension to the problem of object recognition and image segmentation.

Keywords

Textural Features, Co-Occurrence Matrix, Texture Spectrum, Texture Classification, Fractal Dimension

1. Introduction

Unsupervised image segmentation is a fundamental issue in image analysis and computer vision. The purpose of segmentation is to partition the image into regions of similar attribute like luminance, color or texture. Texture plays an important role in numerous computer vision applications, particularity in segmentation of images. Many useful properties for image description and interpretation are gained through texture observation and analysis. Texture classification categories are based sometimes on distinguishing feature as shown in [1] [2]. Markov Random Fields (MRF) defines a class of statistical models which enable to describe both the local and global properties of texture. Early attempts to non-supervised texture segmentation have made use of texture measures, features derived from co-occurrence matrices for instance. The purpose of statistical methods is to characterize the stochastic properties in the spatial distribution of the gray level in the image.

On the other hand, there are some structural methods for texture analysis that can be thought as the regular repetition of a micro texture and the fractal dimension may be appropriate for their characterization.

The aim of this report is to compare the effectiveness of some segmentation methods based on co-occurrence matrices, moreover the texture filters introduced by [3], with a method based on fractal dimension.

The algorithms described in this paper are unsupervised in the following sense:

- They do not require a prior knowledge on the textures in the image, in particular no learning step is necessary;
- The number or regions or texture classes needs not to be known either.

The techniques discussed are neighborhood operators that access pixels in a small area around each central pixel and derive a new value for it, performing some calculation with those values. The operator is repeated throughout the image and the values obtained produce a new image.

The paper is organized as follows, the co-occurrence matrix is introduced in Section 2, the Texture filters in Section 3 and the Method of Range to calculate fractal dimension in Section 4. The performance of each method is illustrated in each section. Conclusions are presented in Section 5.
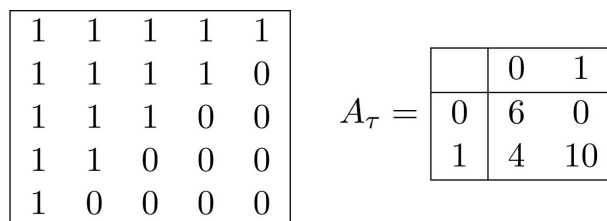
2. Co-Occurrence Matrix

The gray level co-occurrence matrix is defined as a joint distribution of the gray levels of two pixels separated by a given displacement τ. In Cartesian coordinates the displacement can be chosen as a vector $\tau = (\Delta x, \Delta y)$. Each element in the matrix, $A_\tau(g, g')$, is defined as

$$A_\tau(g, g') = \{(g, g') \mid image(x, y) = g \wedge image(x + \Delta x, y + \Delta y) = g'\}$$

where g and g' are gray levels in the image. The size of the co-occurrence matrix A_τ is $G \times G$, denoting G the number of gray levels in the image. This concept is illustrated with a binary model for $\tau = (0,1)$ in **Figure 1** and a model with three levels for gray $\tau = (1,1)$ in **Figure 2**.

Haralick *et al.* [4], described fourteen not independent features extracted from the grey level co-occurrence matrix, and their properties have been showed by several authors in many papers. The texture features that are applied to this research and their physical meaning are presented below.

1	1	1	1	1
1	1	1	1	0
1	1	1	0	0
1	1	0	0	0
1	0	0	0	0

$$A_\tau = \begin{array}{c|c|c} & 0 & 1 \\ \hline 0 & 6 & 0 \\ \hline 1 & 4 & 10 \end{array}$$

Figure 1. Co-occurrence matrix for 5×5 binary image and $\tau = (0,1)$.

$$
\begin{array}{|ccccc|}
\hline
2 & 1 & 2 & 0 & 1 \\
0 & 2 & 1 & 1 & 2 \\
0 & 1 & 2 & 2 & 0 \\
1 & 2 & 2 & 0 & 1 \\
2 & 0 & 1 & 0 & 2 \\
\hline
\end{array}
\qquad
A_\tau =
\begin{array}{c|ccc}
 & 0 & 1 & 2 \\
\hline
0 & 0 & 2 & 2 \\
1 & 2 & 1 & 2 \\
2 & 2 & 3 & 2 \\
\end{array}
$$

Figure 2. Co-occurrence matrix for 5×5 three level of gray image and $\tau = (1,1)$.

2.1. Energy

$$\sum_{g,g'} A_\tau (g,g')^2 \tag{1}$$

This feature is also called angular second moment and measures textures uniformity, that is, pixel pairs repetition. When the image is homogenous, this is only similar gray level pixels are present, energy reaches its maxi-mum. Thus, high values of energy occur when the gray level distribution over the window has either a constant or a periodic form.

2.2. Entropy

$$-\sum_{gg'} A_\tau (g,g') \log \left(A_\tau (g,g') \right) \tag{2}$$

Entropy measures the disorder of an image. It assumes larger values when the image is not texturally uniform. It is important to remark that conceptually energy and entropy are inversely correlated.

2.3. Local Homogeneity

$$\sum_{g,g'} \frac{A_\tau (g,g')}{1 + (g-g')^2} \tag{3}$$

Local homogeneity, also called inverse difference moment, measures images homogeneity. It assumes larger values for smaller gray tone difference in pair elements. The correlation between the local homogeneity and the energy is very strong but in an inverse way.

2.4. Correlation

$$\sum_{g,g'} \frac{(g-\mu_1)(g'-\mu_2) A_\tau (g,g')}{\sigma_1 \sigma_2} \tag{4}$$

Correlation is a measure of gray level linear dependencies in the image and high correlation values imply a linear relationship between the gray levels of pixel pair. Thus the correlation is uncorrelated to energy and entropy, that is, to pixel pair repetitions.

Several co-occurrence matrices in different directions, for example, 0°, 45°, 90°, and 135°, this is for different displacement τ, can be calculated. This could, however, present some drawbacks in practice. Firstly, the choice of this displacement vector depends on experimental cases. The suggested displacement is the one who maximizes the statistic

$$\chi^2(\tau) = \sum_{g,g'} \frac{A_\tau^2(g,g')}{A_\tau(g,\cdot)A_\tau(\cdot,g')}, \tag{5}$$

that can be interpreted as a normalization of the scene energy to the gray level linear dependence in the image. Some of these operations are obviously easier than others to calculate for all the pixels in an image. The resulting values are scaled to create a new image that can be segmented by brightness threshold. Sometimes experimentation with several texture operators is required to find the one that gives the best separation be-tween the object and its surrounding and in consequence the best segmentation. The results of applying some of these operators are illustrated in Figure 3. Each operator was calculated over windows of size 3×3 and $\tau = (1,1)$.

The resulting texture characteristics are often redundant and too numerous, particularly when using more displacement vectors. To overcome these drawbacks, in the 90's approach texture spectrum, based on texture units characterizing local texture information in the eight directions proposed. This method has been applied to the extraction of texture characteristics, classification of textures, detection of edges and filtering of textures [5] [6].

(a)

(b)   (c)

(d)   (e)
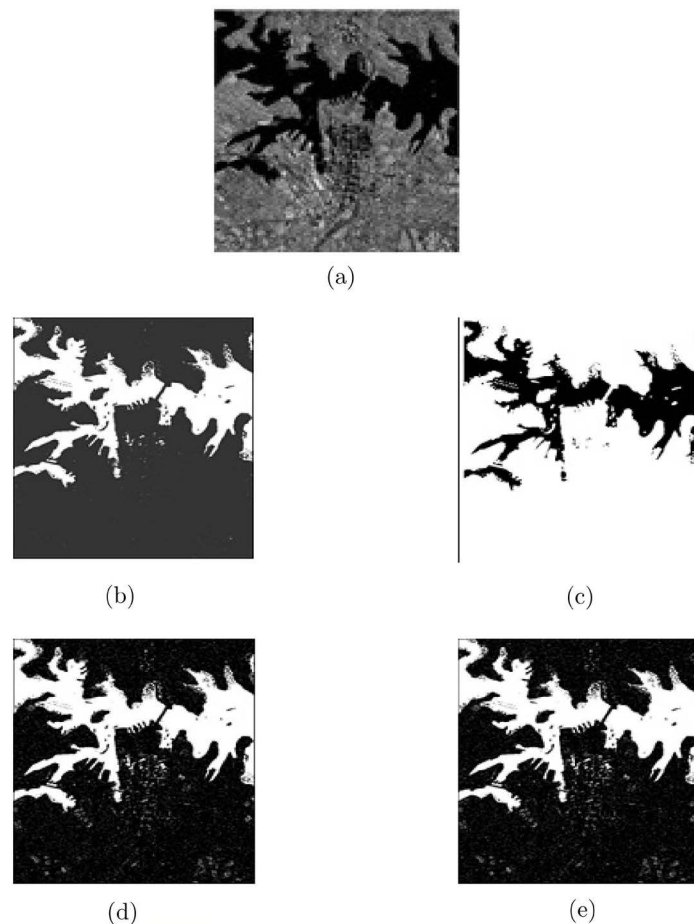
Figure 3. (a) Original image; (b) Energy operator; (c) Entropy operator; (d) Local homogeneity operator; (e) Correlation operator.

3. Texture Filters

A statistical method for texture analysis focuses on the characterization and discrimination of textures was presented by He and Wang in [3]. First define the concept of "texture unit" (UT) which can explain as the smallest unit that best characterizes the appearance of texture, locally, of a pixel and its nearest neighbors of order two. Then the texture of the image is characterized by the "texture spectrum" which describes the distribution of all the units in the image texture.

3.1. Texture Unit

Each pixel is surrounded by its eight nearest neighbors of order two that represents the smallest complete unit, in the sense that it has the eight directions around the pixel, and where you can extract the information for the local texture. For each pixel s take a 3×3 window centered on the pixel, we note this set $V_S = \{g_0, g_1, \cdots, g_8\}$ where g_0 denotes the intensity or gray level in the central pixel and g_i denotes the intensity or gray level in the remaining pixels in the window for $i = 1, \cdots, 8$. The "texture unit" is defined as the set of eight elements $UT = \{E_1, E_2, \cdots, E_8\}$ where each E_i, for $i = 1, \cdots, 8$, is determined by the formula

$$E_i = \begin{cases} 0 & \text{si } g_i < g_0 \\ 1 & \text{si } g_i = g_0, \\ 2 & \text{si } g_i > g_0 \end{cases} \tag{6}$$

and the element E_i occupies the same relative position as the corresponding g_i.

Since each element of texture unit can take three possible values, the total amount of texture units are $3^8 = 6561$. the The window and the texture unit is represented graphically in **Figure 4**.

There is no a single way to label and sort the 6561 texture units, He and Wang [3] label them using the formula

$$N_{UT} = \sum_{i=1}^{8} E_i 3^{i-1}, \tag{7}$$

that represents the "number of texture unit", with E_i as in (6). In addition, the eight elements of the UT can be ordered in a different way, if we order them in the clockwise direction, they can take eight possible forms depending on where we position the first element E_1.

g_1	g_2	g_3
g_4	g_0	g_5
g_6	g_7	g_8

(a)

E_1	E_2	E_3
E_4		E_5
E_6	E_7	E_8

(b)

Figuer 4. (a) 3× 3 window of an image. (b) Texture unit.

3.2. Texture Spectrum

The 6561 set of texture units describes the local texture appearance of a given pixel, this is the relative relationship between central gray levels and its neighbors and the frequency of occurrence of all the texture units over the image will reveal information on texture. We call "texture spectrum" to the frequency distribution of all texture units. Graphically, the horizontal axis represent the number of texture units, N_{UT}, and the ordinate axis represent frequency of occurrence.

Texture spectrum in the increased percentage of a texture component will result in a tendency to a particular distribution of peaks. And the different textures are composed of texture units in individuals with different distributions textured appearance, thus the texture of an image can be characterized by its texture spectrum. Let us note that although the labeling method chosen can affect the relative position of the texture units in the spectrum of texture, not change the value of its frequency in the spectrum. We also make clear that the local texture of a pixel and its environment is characterized by the corresponding texture unit, while the appearance of the texture reveals its texture spectrum calculated and a suitable window size not depends on window the nature of the image.

Al-Jacobi, in [7] proposes a similar technique in which the eight pixels of the window texture unit are divided into two groups of four elements each. These groups are called "cross texture unit" (*UTC*) and is composed of the nearest neighbors of the first order central pixel "bias texture unit" (*UTD*), composed of the elements that are in the directions of the diagonals, graphically in Figure 5.

Each element of these new units can take the values 0, 1 or 2 in the same manner as in (6). Now these texture units can take $3^4 = 81$ possible values and numbers of texture units can be defined $N_{UTC} = \sum_{i=1}^{4} E_{C_i} 3^{i-1}$ and $N_{UTD} = \sum_{i=1}^{4} E_{D_i} 3^{i-1}$. Similarly to what happens as we define the N_{UT} can take different values depending on the order that we give the E_{C_i} and E_{D_i}.

Bhattacharya and others proposed in [8] a modification work of Al-Jacobi estimating N_{UT} based on N_{UTD} and N_{UTC}, having a slightly lower computational cost. For this use various functions such as

$$N_{UT} = N_{UTC} * N_{UTD} ; \quad N_{UT} = N_{UTC} + N_{UTD} ;$$

$$N_{UT} = 81 * N_{UTC} + N_{UTD} ; \quad N_{UT} = \frac{1}{4} \sum_{j=1}^{4} N_{UTC} * N_{jUTD} ;$$

$$N_{UT} = \frac{1}{4} \sum_{j=1}^{4} N_{UTC} * N_{jUTD} ; \text{ and } \quad N_{UT} = \frac{1}{4} \sum_{i=1}^{4} \sum_{j=1}^{4} N_{iUTC} * N_{jUTD}$$

	E_{C1}	
E_{C4}		E_{C2}
	E_{C3}	

(a)

E_{D1}		E_{D2}
E_{D4}		E_{D3}

(b)

Figure 5. (a) Cross texture unit, (b) Bias texture unit.

where N_{iUTC} and N_{jUTD} ordinate indicates the ways of calculating N_{UTC} and N_{UTD} respectively, among others.

After estimating the N_{UT}, we have two techniques to replace the value in the pixel and then segment, these are the "mean" and the "median".

Unit Texture Algorithm

- A centered squared 3×3 window is taken for each pixel and the unit of texture is computed using Equation (6) as shown in Figure 4.
- The texture number N_{UT} is calculated using the Equation (7).
- For each value of N_{UT} obtained we can apply one of the following filters.
- Mean filter: every pixel having the same N_{UT} will be replaced by the average value of intensity of these pixel.
- Median filter: every pixel having the same N_{UT} will be replaced by the median intensity value of these pixels.

In Figure 6 the application of the above algorithm is shown. These algorithms provide a profile in the image which emphasizes the separation of regions of different texture. Both algorithms produce very similar images, however the median filter is slightly more effective in reducing the noise.

4. Fractal Dimension

Let $\mathcal{H}(\Re^n)$ be the space of all nonempty compact subsets of the Euclidean space $\Re^n$. A set $A \subset \mathcal{H}(\Re^n)$ is said to be self similar if A is the union of $N(\epsilon)$ distinct and non overlapping copies of itself each of which is similar to A scaled down by a radio ϵ. The fractal dimension D of A can be defined by

$$D = \lim_{\epsilon \to \infty} \frac{\ln N(\epsilon)}{\ln(1/\epsilon)}, \tag{8}$$

in other words $N(\epsilon)$, the number of rules of length ϵ required to cover the set A, is related to the fractal dimension D by the equation $N(\epsilon)\cdot\epsilon^D = 1$.

Several methods are use to calculate the fractal dimension, we propose the Method of Range.

4.1. Method of Range to Estimate the Fractal Dimension

The Method of Range, which has been introduced in [9], is a a variation of the box counting method. The method computes the fractal dimension as a function

(a)   (b)
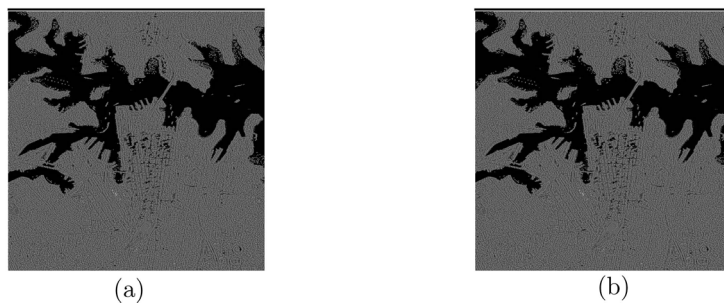
Figure 6. Application of: (a) Mean filter, (b) Median filter.

of the brightness range, keeping the idea of refining the grid over the image, while improving the efficiency of the computation [9] [10].

Method of Range Algorithm

- A centered squared window of size $l_1 = 9$ is taken for each and the brightness range as $r_1 = M_1 - m_1$ is computed, where M_1 is the maximum brightness, and m_1 is the minimum brightness in the window.
- For the same pixel, we take a small squared window of size $l_2 = 5$, centered at that pixel. We compute the brightness range as $r_2 = M_2 - m_2$, where M_2 is the maximum brightness, and m_2 is the minimum brightness in the window.
- We estimate the fractal dimension as $D = \dfrac{r_1 - r_2}{\ln l_1 - \ln l_2}$.
- The histogram with all the values D is constructed and, choosing a threshold h, the image is recolored putting black on each pixel where $D > h$ and white on each pixel where $D \leq h$.

The expression D represents the ratio between the difference of the ranges in each window and the proportion of the length of each window, in log scale. This segmentation technique is illustrated in Figure 7.

5. Conclusions

One of the more important indication of the usefulness of an unsupervised image segmentation algorithm, is its stability. We need segmentation with low bias and variance; this is low variability respect to the parameters.

In this paper each method was applied to the segmentation of the same images to measure the strengths and weaknesses, not in the images where they have their best performance.

Figure 7. Image segmented with the method of range.

About the texture features based on co-occurrence matrices, energy, entropy, local homogeneity and correlation were chosen. All of them were computed for different values of τ and the chosen was $\tau = (1,1)$. The operator energy has been efficient and the operator entropy has been the best to define edges of the coast, but the interior details are lost in both, and their computational coast was too expensive.

In that sense, the algorithms based on Unit Texture provide a profile in the image improving a bit the separation of regions of different texture and with less computational cost. In most of the images in which we have applied them, both algorithms produce very similar results; however, the medium filter is slightly more effective in reducing noise.

The Range Method, based on the fractal dimension, has also been effective in most of the images and is computationally faster. The lower computational weight is not the only advantage of the method, in addition to this method defined the edges very well, and the segmentation was not so homogeneous in the large regions; it not only does profile the coast well but it also detects objects inside.

The Range Method has been shown to be an improvement in statistical methods based on MRF and could be used to build a supervised classifier based on previous segmented images as training blocks with low computational cost.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Suresha, M. and Harisha Naik, T. (2017) A Survey on Image Analysis Based on Texture. *International Journals of Advanced Research in Computer Science and Software Engineering*, **7**, 686-695. https://doi.org/10.23956/ijarcsse/V7I6/0324

[2] Lu, D. and Weng, Q. (2007) A Survey of Image Classification Methods and Techniques for Improving Classification Performance. *International Journal of Remote Sensing*, **28**, 823-870. https://doi.org/10.1080/01431160600746456

[3] He, D.C. and Wang, L. (1990) Texture Unit, Texture Spectrum, and Texture Analysis. *IEEE Transactions on Geoscience and Remote Sensing*, **28**, 509-512. https://doi.org/10.1109/TGRS.1990.572934

[4] Haralick, R.M., Shanmugam, K. and Denstein, I. (1973) Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, **SMC-3**, 610-621. https://doi.org/10.1109/TSMC.1973.4309314

[5] Haralick, R.M. (1979) Statistical and Structural Approaches to Texture. *Proceedings of the IEEE*, **67**, 786-804. https://doi.org/10.1109/PROC.1979.11328

[6] Haralick, R.M. and Shapiro, L.-G. (1992) Computer and Robot Vision. Vol. 1, Addison-Wesley, Reading, MA.

[7] Al-Janobi, A. (2001) Performance Evaluation of Cross-Diagonal Texture Matrix Method of Texture Analysis. *Pattern Recognition*, **34**, 171-180. https://doi.org/10.1016/S0031-3203(99)00206-X

[8] Bhattacharya, A.K., Srivastava, P.K. and Bhagat, A. (2001) A Modified Filtering Technique for Satellite Images. *22nd Asian Conference on Remote Sensing*, 5-9 November 2001, 262-267.

[9] Marrón, B.S. (2012) Modelos Markovianos en Segmentación de Imágenes. Thesis, Departamento de Matemática, Universidad Nacional del Sur, Bahía Blanca.

[10] Marrón, B.S. (2012) Co-Occurrence Matrix and Fractal Dimension for Image Segmentation. *Revista de Matemática: Teora y Aplicaciones*, **19**, 49-63.
https://revistas.ucr.ac.cr/index.php/matematica/article/view/2104/2067