A fast algorithm for texture feature extraction from gray level aura matrices

Zohra Haliche, Kamal Hammouche and Jack-Gérard Postaire

***Abstract*—**The Gray Level Aura Matrix (GLAM) is an efficient tool for texture analysis. Considering all the possible couples of gray levels in an image, it indicates, for each couple, how many pixels with one of the two gray levels have, in their neighborhood, pixels with the other gray level. The neighborhood is defined by a structuring element as in mathematical morphology. In the image segmentation process, a GLAM is determined for each pixel and is scanned once or twice in order to extract texture features. To avoid a high computation time, we introduce a strategy, based on a so-called Gray Level Aura Hybrid Structure (GLAHS), which combines a linked list and an integrated hash table. The linked list is used to store the non zero elements of the GLAM and the hash table allows to access quickly to the nodes of that list. The length of the linked list is generally short, allowing a fast extraction of texture features.

***Keywords*—** Fast texture feature extraction, Gray level aura matrices, Linked list, Hash table, Segmentation.

I. Introduction

The segmentation of textured images is a fundamental problem in computer vision [1]. It consists in partitioning an image into homogeneous regions with respect to texture properties. The success of the segmentation depends mainly on the texture features selected to characterize the pixels of the image.

A great number of texture features have been proposed in the literature [2]. Among them, those derived from Gray Level Co-occurrence Matrices (GLCM) are very popular [3]. Generally, these features yield good results [4-5], although the GLCM reflects only the relationships between neighboring pixels taken two by two.

A generalization of the GLCM, called Gray Level Aura Matrix (GLAM), has been proposed in [6]. The GLAM is defined in a framework based on the set-theoretic concept of “aura set”. This matrix allows to quantify the importance of a set of pixels with a specified gray level standing in the neighborhood of another set of pixels having another gray level. The amount of neighboring pixels with the specified gray level is quantified by means of a measure defined on this set: the "aura measure”. The GLAM counts, by means of the aura measures, the occurrences of pairs of gray levels associated with a pixel and its neighbors belonging to a neighborhood defined by a structuring element. Although the computational schemes for determining the GLCM and the GLAM are almost similar, the GLAM should not be confused with the GLCM [6-18]. Indeed, the GLAM differs significantly from the GLCM which simply counts the number of pairs of different gray levels associated with two pixels separated by a distance *d* in a given direction θ .

The GLAMs are used for texture representation [7], image retrieval [8-9], texture synthesis [10],[11], classification [12]-[17] and segmentation [18]. However, despite the evident interest of this texture characterization procedure, the implementation of this new tool becomes time consuming when the number of gray levels in the images is important. In this paper, an algorithm is developed for extracting the texture characteristics from the GLAMs at a reduced cost.

In the next section, following a scheme that has been recently introduced by the authors [18], we recall how a GLAM is generated for each pixel of an image and is used to extract texture features for image segmentation. This feature extraction procedure requires a high running time if it is not considered attentively. Section 3 is devoted to the proposed method for fast extraction of the texture features from a GLAM. Although the proposed fast algorithm yields exactly the same features, it is fundamentally different from the conventional computational technique. Experimental results are reported in section 4. They demonstrate the computational savings that are achieved through the use of the proposed fast algorithm. Concluding remarks are given in the last section.

II. Texture Image Segmentation Based on Gray Level Aura Matrices

The segmentation of a textured image using the GLAMs can be considered as a pixel classification procedure. In this section, we recall how a local GLAM is determined in order to compute a set of characteristic features for each pixel of the image. We focus our attention on the computational efforts required by this feature extraction process. The clustering of the pixels having similar texture features is beyond the scope of this paper. This last step of the segmentation procedure is implemented with standard and well-established classification algorithms.

Z. Haliche is with the Electrical and Computer Engineering Department at Mouloud Mammeri University of Tizi-Ouzou, Algeria. (e-mail: zohrahaliche@yahoo.fr).

K. Hammouche is with Electrical and Computer Engineering Department at Mouloud Mammeri University of Tizi-Ouzou, Algeria. (Phone:+21371832430; e-mail: kamal_hammouche@yahoo.fr).

J.-G. Postaire is with the Laboratoire LAGIS, Université de Lille 1 – Sciences et Technologies, Cité Scientifique – Bâtiment P2, 59655 Villeneuve d’Ascq Cedex, France (e-mail: jack-gerard.postaire@univ-lille1.fr).

A. Aura set

In the feature extraction step, a local GLAM is determined for each pixel of the image. A window S of size $(2w+1)^2$ is centered on each pixel of the image, so that the features attached to the considered pixel reflect the local properties of the texture. Let N be a structuring element. When centered on any pixel $P_s \in S$, this structuring element defines a set N_s of neighbors. The gray level of a pixel P_s is denoted $I(P_s)$. Let S_g and $S_{g'}$ be two subsets of the window S $(S_g, S_{g'} \subseteq \text{S})$, where $S_g = \{P_s \in S / I(P_s) = g\}$ and $S_{g'} = \{P_s \in S / I(P_s) = g'\}$ are the sets of pixels with gray levels g and g', respectively, such as $\bigcup_{g=0}^{G-1} S_g = S$ and $S_g \cap S_{g'} = \emptyset$ for $g \neq g'$, where G is the total number of gray levels in the window S.

The neighborhood V_{S_g} of S_g is defined by the union of all the pixels that belong to all the sets N_s of neighbors $P_s \in S_g$:

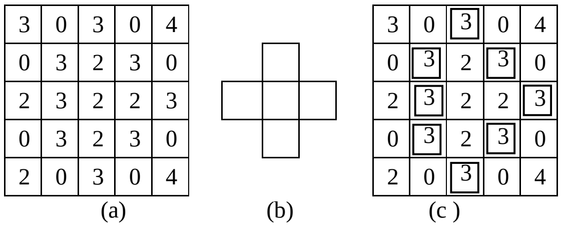
$$V_{S_g} = \bigcup_{P_s \in S_g} N_s \qquad (1)$$

The aura of the set S_g with respect to the set $S_{g'}$ is the set constituted of the elements of the set $S_{g'}$ present in the neighborhood of the set S_g . The aura indicates how a subset is more or less present in the neighborhood of another subset. This set is denoted $\vartheta_{S_{g'}}(S_g)$ such as [6]:

$$\vartheta_{S_{g'}}(S_g) = \bigcup_{P_s \epsilon S_g} (N_s \cap S_{g'}) \qquad (2)$$

Fig. 1 illustrates the construction of an aura set. The window of size (5x5) contains 25 pixels with 5 different gray levels ranging from 0 to 4. The sets S_2 and S_3 of pixels with gray levels 2 and 3, respectively, are easily identified in this figure. The structuring element N chosen for this example is the cross defined in Fig. 1 (b).

The aura of the set S_g with respect to the set $S_{g'}$ is constituted of the elements marked as double boxes in Fig. 1(c).



3	0	3	0	4
0	3	2	3	0
2	3	2	2	3
0	3	2	3	0
2	0	3	0	4

(a)

(b)

3	0	$\boxed{3}$	0	4
0	$\boxed{3}$	2	$\boxed{3}$	0
2	$\boxed{3}$	2	2	$\boxed{3}$
0	$\boxed{3}$	2	$\boxed{3}$	0
2	0	$\boxed{3}$	0	4

(c)

Fig 1 Example of an aura set. a) Window S of size (5X5) , b) Structuring element N, c) The set of pixels marked as double-boxes is the aura of the set of pixels with gray level 2 with respect to the set of pixels with grey level 3.

B. Aura Measure

The aura measure $m(S_g, S_{g'})$ of a set S_g with respect to a set $S_{g'}$ is defined as [6]:

$$m(S_g, S_{g'}) = \sum_{P_s \in S_g} |N_s \cap S_{g'}| \qquad (3)$$

where $|X|$ indicates the number of elements in the set X.

Note that $m(S_g, S_{g'})$ does not indicate exactly the number of elements in the aura of S_g with respect to $S_{g'}$ because, in general, $m(S_g, S_{g'}) \geq \left| \vartheta_{S_{g'}}(S_g) \right|$. For the example of figure 1, $m(S_2, S_3) = 11$ while $\left| \vartheta_{S_3}(S_2) \right| = 8$.

The aura measure $m(S_g, S_{g'})$ evaluates roughly the amount of elements of the set $S_{g'}$ present in the neighborhood of the set S_g. A large value of $m(S_g, S_{g'})$ indicates that the elements of the two sets S_g and $S_{g'}$ are closely interlaced, while a small value indicates that the sets S_g and $S_{g'}$ are more or less separated one from each other.

C. Gray Level Aura Matrix (GLAM)

The local GLAM, denoted M, is a matrix of dimension (GxG) composed of the aura measures between all the couples of subsets of pixels associated with all the available couples of gray levels present in a window S centered on the processed pixel such as [18]:

$$\text{M} = [m(S_g, S_{g'})] \qquad 0 \leq g, g' \leq G-1 \qquad (4)$$

For the sake of simplicity, the elements of this matrix are noted $m_{g,g'}$ in what follows, so that the matrix M can be written as:

$$\text{M} = [m_{g,g'}] \qquad (5)$$

A straightforward way to compute the local GLAM is to first determine all the sets of pixels with different gray levels S_g , g=0, 1, ..., G-1 , within the window S. Then, the aura measure $m_{g,g'}$ is computed as in (3) for each pair of subsets S_g and $S_{g'}$, where g , g' = 0, 1, ... , G-1. However, in order to reduce the computational effort, a GLAM is generally determined in the same way as a GLCM. For each pixel $P_s \in S$ with gray level g, the gray level of each pixel $P_r \in N_s$ is checked. If $I(P_r) = g'$ the value of $m_{g,g'}$ is incremented by 1. This procedure takes advantage of the fact that the measure $m_{g,g'}$ is computed on the basis of all the pixels of gray level g having pixels of gray level g' as neighbors within the structuring element [6].

For example, the local GLAM for the window shown in figure 1 is:

$$\text{M} = \begin{bmatrix} 0 & 0 & 5 & 16 & 4 \\ 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 4 & 11 & 0 \\ 16 & 0 & 11 & 4 & 0 \\ 4 & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (6)$$

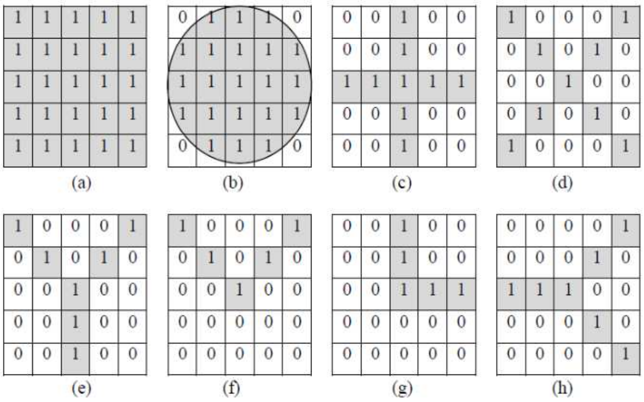
The GLAMs are rich in information. Many authors have considered the elements of the GLAM themselves as features for each pixel [12]-[17]. However, as the dimension (GxG) of a GLAM can be very high in real gray-level images, it is interesting to extract texture features as those proposed by Haralick for the GLCM to summarize this information [2] (see Table I).

Table I. Texture features extracted from a GLAM

Contrast	$\frac{1}{N^2}\sum_{g=0}^{G-1}\sum_{g'=0}^{G-1}(g-g')^2 m_{g,g'}^2 \quad N=\sum_{g=0}^{G-1}\sum_{g'=0}^{G-1} m_{g,g'}$
Dissimilarity	$\frac{1}{N}\sum_{g=0}^{G-1}\sum_{g'=0}^{G-1}(g-g')^2 m_{g,g'}$
Uniformity	$\frac{1}{N^2}\sum_{g=0}^{G-1}\sum_{g'=0}^{G-1} m_{g,g'}^2$
Entropy	$-\frac{1}{N}\sum_{g=0}^{G-1}\sum_{g'=0}^{G-1} m_{g,g'}\log\left(\frac{m_{g,g'}}{N}\right)$
Correlation	$\frac{1}{N}\sum_{g=0}^{G-1}\sum_{g'=0}^{G-1}\frac{\left(g-\mu_g\right)\left(g'-\mu_{g'}\right)m_{g,g'}}{\sigma_g\sigma_{g'}}$ $\mu_g=\sum_{g=0}^{G-1} g\sum_{g'=0}^{G-1}\frac{m_{g,g'}}{N}$ $\mu_{g'}=\sum_{g'=0}^{G-1} g'\sum_{g=0}^{G-1}\frac{m_{g,g'}}{N}$ $\sigma_g^2=\sum_{g=0}^{G-1}\left(g-\mu_g\right)^2\sum_{g'=0}^{G-1}\frac{m_{g,g'}}{N}$ $\sigma_{g'}^2=\sum_{g'=0}^{G-1}\left(g'-\mu_{g'}\right)^2\sum_{g=0}^{G-1}\frac{m_{g,g'}}{N}$
Inverse difference moment	$\frac{1}{N}\sum_{g=0}^{G-1}\sum_{g'=0}^{G-1}\frac{m_{g,g'}}{1+(g-g')^2}$

D. Structuring Elements

A GLAM can be defined with structuring elements of various geometrical shapes and sizes [7], [10], [18]. Some symmetric and asymmetric structuring elements of size (5x5) are represented as binary masks in Fig. 2.


(a)
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1

(b)
0 1 1 1 0
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
0 1 1 1 0

(c)
0 0 1 0 0
0 0 1 0 0
1 1 1 1 1
0 0 1 0 0
0 0 1 0 0

(d)
1 0 0 0 1
0 1 0 1 0
0 0 1 0 0
0 1 0 1 0
1 0 0 0 1

(e)
1 0 0 0 1
0 1 0 1 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0

(f)
1 0 0 0 1
0 1 0 1 0
0 0 1 0 0
0 0 0 0 0
0 0 0 0 0

(g)
0 0 1 0 0
0 0 1 0 0
0 0 1 1 1
0 0 0 0 0
0 0 0 0 0

(h)
0 0 0 0 1
0 0 0 1 0
1 1 1 0 0
0 0 0 1 0
0 0 0 0 1

Fig 2 Examples of structuring elements of sizes (5x5):
(a) to (d) Symmetric structuring elements,
(e) to (h) Asymmetric structuring elements.

III. Fast texture feature extraction from gray level aura matrices

In order to characterize the texture around a pixel, the associated GLAM must be scanned once or twice to evaluate the double sums that are used to compute the texture features listed in Table 1. Hence, the processing time becomes prohibitive when the number *G* of gray levels is large.

To cope with this problem, we have examined two solutions proposed in the texture analysis literature. The first one consists in reducing the number *G* of quantization levels [19]. It can easily be adapted to decrease the size of the GLAM. However, this solution may alter the discriminating power of the texture features extracted from the smaller GLAM. We have adapted the second solution that consists in storing only the non-zero elements of the matrix in a list whose size is shorter than that of the matrix [20]-[22]. A GLAM often contains several null elements that are not taken into account when computing the texture features. Hence, instead of storing the aura measures $m_{g,g'}$ between two subsets S_g and $S_{g'}$ in a matrix of size (*G*x*G*), a linked list can be used to store only the non-zero aura measures. For example, the GLAM of figure 1, given in (6), contains 10 non-null elements among a total of 25, but only 6 non-null different elements have to be stored if we consider the symmetric form of the structuring element.

The linked list is constituted of nodes that contain the gray level g , the gray level g' and the corresponding aura measure $m_{g,g'}$. Let L_k be the k^th^ node of the list in which the values of g , g', and $m_{g,g'}$ are denoted $L_k(g)$, $L_k(g')$, and $L_k(m_{g,g'})$, respectively. Two pointers, "Next" and "Prev" , are also included in each node. The first one points to the next node and the other to the previous node (cf. Fig. 3).

In order to build a GLAM for a P_r pixel of an image, we begin by examining its neighbors located in a window S of size $(2w+1)^2$ centered on that pixel. Let $P_{r'}$ be a neighbor of P_r that belongs to the structuring element N_r attached to pixel P_r. Let the gray levels of P_r and $P_{r'}$ be $I(P_r)=g$ and $I(P_{r'})=g'$, respectively. We search for the node containing the pair (g,g') in the linked list. If this node is found, then the corresponding aura measure $m_{g,g'}$ is incremented by one. Otherwise, a new node is added at the end of the list by storing the values of g and g' and by setting $m_{g,g'}$ to 1. Note that the linked list is initially empty. When all the pixels of the window have been examined, the resulting list contains only the non-zero aura measures. Let Λ denotes the length of that list. The linked list corresponding to the example of Fig. 1 is shown in Fig. 3.

When searching for a particular node, the linked list must be scanned from its beginning to its end. A hash table is used in order to access directly to each node of the linked list and, subsequently, to reduce the processing time necessary to explore the list. This hybrid data structure combining a linked list with a hash table is called hereafter the Gray Level Aura Hybrid Structure (GLAHS). The hash table is a (*G*x*G*) bi-dimensional structure, containing the memory addresses of the linked list nodes. The accesses to the hash table are provided by the pairs of gray levels (g, g'). A null pointer indicates that the particular pair of gray levels (g, g') does not have a representative node in the linked list. In this case, a new node is created and inserted at the end of the linked list. In order to find this node later, its address memory is introduced at the position (g, g') of the hash table. If the pointer is not null, then it points to the corresponding node in the linked list.

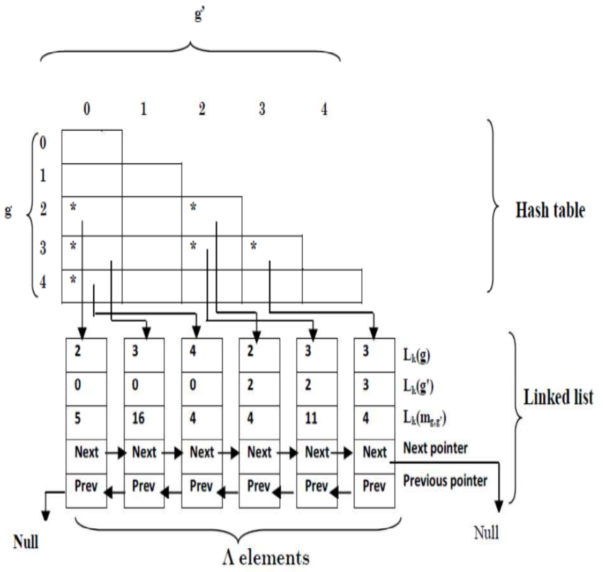


Fig. 3 The GLAHS structure for determining the texture features for the image of Fig. 1(a).

The texture features listed in Table II are extracted from a GLAHS by scanning the linked list from the beginning to the end. These features are identical to those of Table I which are directly extracted from the GLAM, although the scanning of the list is significantly shorter than the scanning of the whole GLAM.

IV. Analysis of the Algorithm

Although the algorithm based on the GLAHS does not introduce new genuine texture features, it is fundamentally different from the conventional computational technique based directly on the GLAM. The characterization of a pixel in terms of texture is a two steps process. A GLAM or a GLAHS is generated in the first step, while the texture features are extracted in a second step.

The computational efforts required for the generation of a GLAM or a GLAHS associated with each pixel of an image are similar and depend on the number of the couples of pixels encountered within a window S of side length $(2w + 1)$. Let n be the number of pixels constituting the structuring element N. As each of the $(2w + 1)^2$ pixels constituting the window has (n-1) neighbors in the associated structuring element N, $(n - 1)(2w + 1)^2$ couples of pixels are considered to compute the aura measure at each pixel of the image. Hence, the determination of all the aura measures, either with the conventional or the hybrid structures, requires $(n - 1)(2w + 1)^2|S|$ operations.

Table II. Texture features extracted from a linked list

Contrast	$\sum_{k=1}^{\Lambda}[L_k(g) - L_k(g')]^2\left(\frac{L_k(m_{g,g'})}{N_L}\right)^2$ $N_L = \sum_{k=1}^{\Lambda} L_k(m_{g,g'})$
Dissimilarity	$\sum_{\substack{k=1 \\ g=g'}}^{\Lambda}[L_k(g) - L_k(g')]^2\left(\frac{L_k(m_{g,g'})}{N_L}\right)$
Uniformity	$\sum_{k=1}^{\Lambda}\left(\frac{L_k(m_{g,g'})}{N_L}\right)^2$
Entropy	$-\sum_{k=1}^{\Lambda}\left(\frac{L_k(m_{g,g'})}{N_L}\right) log\left(\frac{L_k(m_{g,g'})}{N_L}\right)$
Correlation	$\sum_{k=1}^{\Lambda}\left[L_k(g) - \mu_g\right]\left[L_k(g') - \mu_{g'}\right]\left(\frac{L_k(m_{g,g'})}{N_L}\right)$ $\mu_g = \sum_{k=1}^{\Lambda} L_k(g)\left(\frac{L_k(m_{g,g'})}{N_L}\right)$, $\mu_{g'} = \sum_{k=1}^{\Lambda} L_k(g')\left(\frac{L_k(m_{g,g'})}{N_L}\right)$
Inverse difference moment	$\sum_{k=1}^{\Lambda}\frac{1}{1 + [L_k(g) - L_k(g')]^2}\left(\frac{L_k(m_{g,g'})}{N_L}\right)$

The second step is significantly different for the GLAM and GLAHS based algorithms. A GLAM must be scanned at least once to extract some of the features of Table I, and sometimes twice for the other features. $(G \times G)$ operations are necessary to explore the whole aura matrix.

On the other hand, the linked list associated with a GLAHS is scanned with Λ operations. The length Λ of the linked list depends upon the window size $(2w + 1)^2$ and the size of the structuring element N since the numbers of distinct gray levels of subsets S_g increase with these two parameters. However, the number Λ of operations necessary to explore the linked lists remains always significantly smaller than the number $(G \times G)$ of operations necessary to explore the aura matrices, so that the computational load for computing the features can be significantly reduced by means of the GLAHS.

To be more specific, the total complexity of the texture characterization using the aura for the whole image is equal to $O(|S|((n - 1)(2w + 1)^2 + (\alpha + \beta)G^2))$ with the conventional GLAM based algorithm, where α and β are the numbers of features using 1 and 2 loops, respectively. It is equal to $O(|S|((n - 1)(2w + 1)^2 + (\alpha + \beta)\Lambda))$ with the fast GLAHS based algorithm.

V. Experimental results

In order to help quantify the above discussion, the results of some experiments are now presented. The running times of the proposed algorithm and the conventional one are compared for various synthetic and real images.

A. Running Times

For the first tests, we use the (256x256) synthetic image displayed in Fig. 4, which is composed of two different textures (D_{24}, D_{92}) taken from the Brodatz album [23]. All the procedures are implemented with the C++ Builder 6 language on a 1.66 GHZ computer with 1 GB memory. Although the running times are computer dependent, they allow to compare the speed of the two algorithms.

The running times are evaluated for different sizes $(2w+1)(2w+1)$ of the window S, and for different numbers G of gray levels. As the complexity of the texture characterization is locally dependant on the image, we have determined the mean running times per pixel for each phase of the procedures. The mean values $\bar{\Lambda}$ of the lengths Λ of the linked lists produced by the GLAHS based procedures are also indicated for each experiment.

Table III indicates these mean running times for different values of w, and for G varying from 4 to 256. The structuring element is the (5x5) symmetrical cross defined in Fig. 2 (c).

Table IV shows the influence of the size of the structuring element and the window size on the running times. These results are obtained by using the symmetrical cross like structuring element and a number of gray levels $G = 256$.

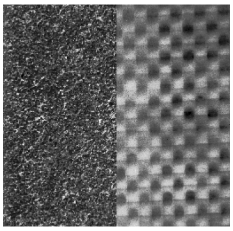


Fig. 4 Test image

As expected, the running times for the generation of the aura matrices with the GLAM based method and for the generation of the linked lists with the GLAHS based method are almost similar. They mainly depend on the size $(2w+1)(2w+1)$ of the window S and on the size of the structuring element N. As the number of considered couples of pixels increases with these two parameters, the running times vary in the same way. Conversely, these running times are not affected by the number G of gray levels.

The main improvement appears in the feature extraction process, especially, when the number G of gray levels is high. For example, with $G = 256$ and with a (17x17) window size, the extraction of the features with the GLAHS is more than 7 times faster than the extraction of the same features directly from the GLAM (see. Table III). This improvement is due to the mean length $\bar{\Lambda}$ of the linked lists which is equal to 629 while the whole aura matrix contains 65536 elements.

As the boundary between the two regions constituting this synthetic image is perfectly known, we have evaluated the quality of the segmentation on the test image as a function of the window size, the size of the structuring element N and the number G of gray levels. This evaluation is carried out on the basis of the classification rate, estimated as the ratio of the number of correctly classified pixels to the total number of pixels. In this simple example, the classification rate is not very sensitive to the adjustment of these parameters. When the size of the window increases, the classification rate begins to increase to reach a maximum value. Then, due to a lack of precision in the localization of the border between the two different textures, it tends to slightly decrease. The study of the influence of the window size, the shape and the size of the structuring element N has been thoroughly analyzed in [18]. Table III shows also that the classification rate decreases with the number G of gray levels.

This classification rate has been evaluated similarly for tuning the two algorithms now compared in terms of running times for the segmentation of some synthetic images displayed in Fig. 5. The size of all these images is (256x256).

Table V gives the classification rate as well as the running times obtained by the GLAM and GLAHS based procedures. The shape and the size of the structuring element and the size of the window used for each image are also indicated in the Table V. The GLAHS based algorithm is 4 to 9 times faster than the conventional GLAM based algorithm. For all images, the mean values $\bar{\Lambda}$ of the lengths of the linked lists produced by the GLAHS based procedure are smaller than the size GxG of the GLAM. Fig. 5 shows also the images segmented using the aura matrices.

For the real images of Fig. 6, the parameters of the algorithm, given in Table V, have been adjusted by visual examination of the results. With this strategy, the segmentation with the features extracted from the aura matrices are always better than those obtained from the co-occurrence matrices, as already demonstrated by the authors in [18]. However, in the framework of this paper, the main interest of these results is to quantify the improvement of the running times between the standard GLAM based algorithm and the GLAHS based procedure for the different images. For the same result of the segmentation process, the improvement of the computational times can reach a ratio of 10 between the GLAM and GLAHS based algorithms (see Table V).

VI. Conclusion

In this paper, we have proposed a fast feature extraction procedure for textured image segmentation from gray level aura matrices. This procedure takes advantage of a Gray Level Aura Hybrid Structure (GLAHS) that combines a linked list and hash table. It consists in storing only the non-zero elements of the GLAM in a linked list instead of considering the whole matrix itself. The size of this list is generally shorter than the number of elements of the GLAM, so that it can be rapidly scanned to compute the features. The hash table provides a rapid access to the content of that linked list.

Experimental results show that the running time can be significantly reduced by means of this hybrid structure, especially for large values of the number of gray levels.

References

[1] J. M. Tuceryan and A.K. Jain, “Texture analysis,” Chapter 2.1, The Handbook of Pattern Recognition and Computer Vision (2nd Edition), by C. H. Chen, L. F. Pau, P. S. P. Wang (eds.), Word Scientific Publishing Co., 1998, pp. 207-248.

[2] X. Xie, and M. Mirmehdi, “A Galaxy of Texture Features,” Chapter 13, Handbook of Texture Analysis, by M. Mirmehdi, X. Xie and J. Suri (Ed.). Imperial College Press,2008, pp. 375-406.

[3] R.M. Haralick, K. Shanmugam and I.H. Dinstein, “Textural features for image classification,” *IEEE Transactions on Systems Man and cybernitics*, Vol. SMC-3, n°6, pp. 610-621, 1973.

[4] C.I. Christodoulou, C.S. Pattichis, M. Pantziaris and A. Nicolaides, “Texture-based classification of atherosclerotic Carotid Plaques,” *IEEE Transactions on Medical Imaging*, 22(7), pp. 902-912, 2003.

[5] C.I. Christodoulou, S.C. Michaelides and C.S. Pattichis, “Multifeature texture analysis for the classification of clouds in satellite imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, 41(11), pp. 2662-2668, 2003.

[6] I.-M. Elfadel and R.-W. Picard, “Gibbs random fields, co-occurrences, and texture modeling,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp.24-37, 1994.

[7] X. Qin and Y.-H. Yang, “Representing texture images using asymmetric gray level Aura matrices,” Tech. Rep. TR05-27, University of Alberta, Canada, *2005*.

[8] X. Qin and Y.-H. Yang, “Similarity measure and learning with gray level Aura matrices (GLAM) for texture image retrieval,” *IEEE Computer Vision and Pattern Recognition*, vol. 1, pp. 326-333, 2004.

[9] S.Wiesmüller, and D.A.Chandy, “Content based mammogram retrieval using Gray Level Aura Matrix,” Proceedings of the International Joint Journal Conference on Engineering and Technology (IJJCET 2010), 2010, pp. 217-221.

[10] X. Qin and Y.-H. Yang, “Basic gray level Aura matrices: theory and its application to texture synthesis,” IEEE International Conference on Computer Vision, 2005, pp. 128-135.

[11] X. Qin and Y.-H Yang, “Aura 3D textures,” IEEE Transaction on Visualization and Computer Graphics, vol. 13, no. 2, pp.379-389, 2007.

[12] M.T. Syrjäsuo, E.F. Donova, X. Qin, and Y.-H. Yang, “Automatic classification of auroral images in substorm studies,” Int. Conf. Substorms,Vol.8, 2006, pp.309-313.

[13] X. Qin and Y.-H. Yang, “Texture image classification using basic gray level Aura matrices,” Technical Report TR06-14, University of Alberta, Canada, 2006.

[14] S. Liao and A.C.S.Chung, “Texture classification by using advanced local binary patterns and spatial distribution of dominant patterns,”, ICASSP, Vol.1, 2007, pp.1221-1224.

[15] Y.Wang, X.Gao, R.Fu, and Y.Jian, “Dayside corona aurora classification based on X-gray level aura matrices,” Proceedings of the ACM International Conference on Image and Video Retrieval, 2010, pp.282-287.

[16] M. A. Hannan, M. Arebey, R. A. Begum, and H. Basri, “An automated solid waste bin level detection system using a gray level aura matrix,” Waste Management, vol. 32, pp. 2229-2238, 2012.

[17] M. Khalid, R. Yusof and A. S. M. Khairuddin, “ Improved tropical wood species recognition system based on multi-feature,” World Academy of Science, Engineering and Technology, vol. 59, 2011.

[18] Z.Haliche, and K.Hammouche, “The gray level aura matrices for textured image segmentation,” *Analog Integrated Circuits and Signal Processing*, Vol.69, pp. 29-38, 2011.

[19] D.A. Clausi, “An analysis of co-occurrence texture statistics as a function of grey level quantization,” *Canadian Journal Remote Sensing*, Vol. 28, no.1, pp. 45-62, 2002.

[20] D.A. Clausi and M.E. Jernigan, “A fast method to determine co-occurrence texture features,” *IEEE International on Geosciences and Remote Sensing*, Vol. 36, no.1, pp. 298-300, 1998.

[21] D.A. Clausi and Y. Zhao, “Rapid co-occurrence texture features extraction using a hybrid data structure,” *Computers and Geosciences*, Vol. 28, no. 6, pp. 763-774, 2002.

[22] K. Hammouche and J.-G. Postaire, “Multidimensional Texture Analysis for Unsupervised Pattern Classification,” Chapter 7, Pattern Recognition Techniques, Technology and Applications, Book edited by: Peng-Yeng Yin, pp. 163-196, , I-Tech, Vienna, Austria.

[23] P. Brodatz, “Texture: A Photograph Album for Artists and Designs,” Dover, New York,1956.

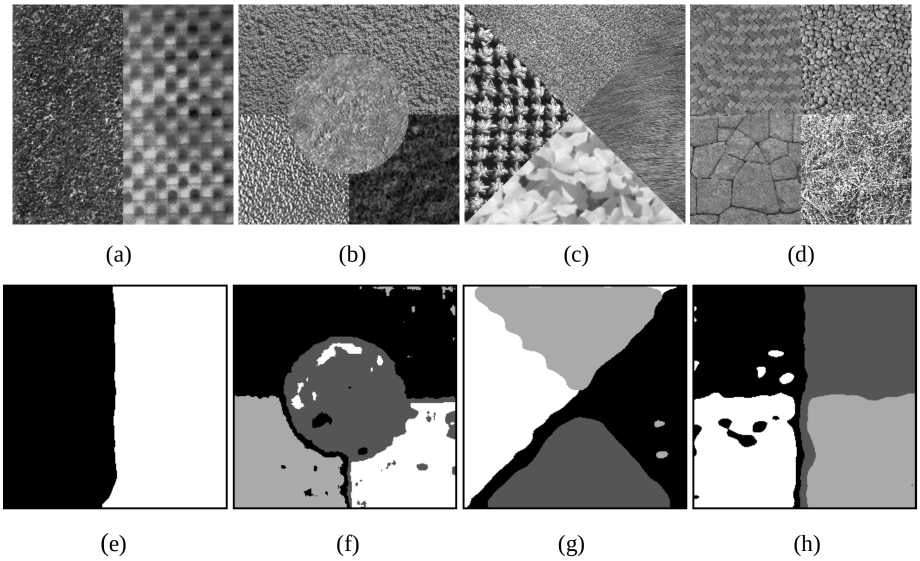


Fig. 5 Segmentation of four synthetic images based on the GLAM
(a)-(d): Original images,
(e)-(h): Segmented images.

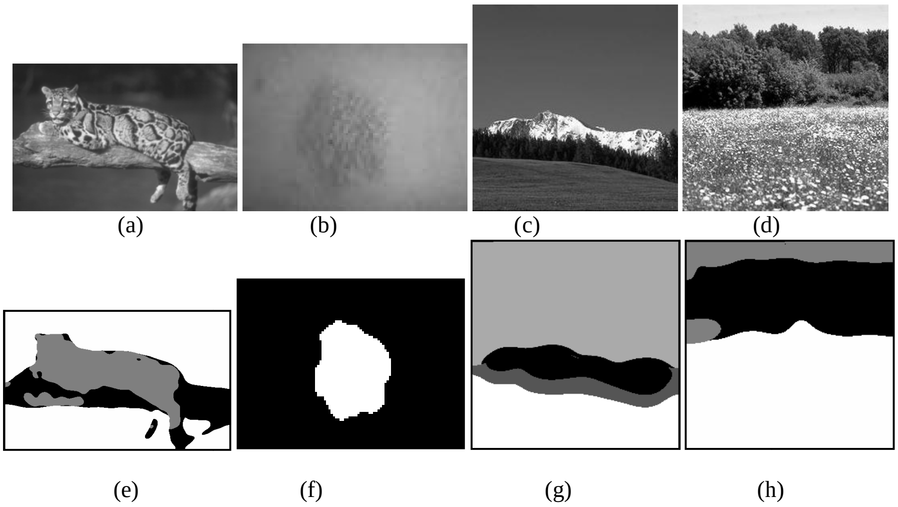


Fig. 6 Segmentation of four real images based on the GLAM
(a)-(d): Original images,
(e)-(h): Segmented images.

Table III. Mean running times of the GLAM and GLAHS based algorithms obtained with a 5x5 cross shaped structuring element as functions of the size of the window S and the number *G* of gray levels.

Window size	Number of Gray Levels (G)	Mean processing times (ms)						Mean length of the linked list (*Λ*)	Classification rate
		Generation procedure (ms)		Feature extraction procedure (ms)		Total time (ms)			
		GLAM	GLAHS	GLAM	GLAHS	GLAM	GLAHS		
9x9	256	0.42	0.55	11.82	1.38	12.24	1.93	242	98.28%
13x13		0.88	1.05	11.96	1.48	12.84	2.53	427	98.65%
17x17		1.50	1.68	11.99	1.63	13.49	3.31	629	98.17%
9x9	128	0.42	0.52	3.94	1.32	4.36	1.84	194	98.42%
13x13		0.88	1.02	3.98	1.40	4.86	2.42	321	98.14%
17x17		1.50	1.62	4.05	1.49	5.55	3.11	446	97.96%
9x9	64	0.42	0.51	1.98	1.28	2.40	1.79	145	97.78%
13x13		0.88	0.98	2.00	1.33	2.88	2.31	222	97.70%
17x17		1.50	1.60	2.02	1.39	3.52	2.99	290	97.69%
9x9	32	0.42	0.48	1.39	1.26	1.81	1.74	86	97.76%
13x13		0.88	0.96	1.40	1.27	2.28	2.23	119	97.61%
17x17		1.50	1.56	1.42	1.28	2.92	2.84	146	97.50%
9x9	16	0.42	0.46	1.27	1.22	1.69	1.68	38	97.43%
13x13		0.88	0.92	1.29	1.23	2.17	2.15	48	97.46%
17x17		1.50	1.50	1.30	1.24	2.80	2.74	56	97.40%
9x9	8	0.42	0.44	1.25	1.20	1.67	1.64	15	97.36%
13x13		0.88	0.90	1.26	1.21	2.14	2.11	18	97.29%
17x17		1.50	1.49	1.27	1.22	2.77	2.71	20	97.12%
9x9	4	0.42	0.43	1.24	1.18	1.66	1.61	6	96.12%
13x13		0.88	0.89	1.25	1.19	2.13	1.08	7	96.11%
17x17		1.50	1.47	1.26	1.20	2.76	2.67	8	96.04%

Table IV. Mean running times of the GLAM and GLAHS based algorithms obtained with G=256 as functions of the size the window S and the size of the cross shaped structuring element.

Window size	Size of the structuring element	Mean running times (ms)						Mean length of the linked list ($\bar{\Lambda}$)	Classification rate
		Generation procedure (ms)		Feature extraction procedure (ms)		Total time (ms)			
		GLAM	GLAHS	GLAM	GLAHS	GLAM	GLAHS		
9x9	3x3	0.20	0.24	11.72	1.28	11.92	1.52	134	97.85%
	5x5	0.42	0.55	11.82	1.38	12.24	1.93	242	98.28%
	7x7	0.63	0.73	11.87	1.42	12.50	2.15	343	98.44%
13x13	3x3	0.42	0.48	11.82	1.34	12.24	1.82	247	97.64%
	5x5	0.88	1.05	11.96	1.48	12.84	2.53	427	98.65%
	7x7	1.33	1.48	11.98	1.60	13.31	3.08	587	98.54 %
	9x9	1.78	2.01	12.05	1.71	13.83	3.72	733	98.46 %
	11x11	2.25	2.46	12.07	1.82	14.32	4.28	871	98.39%
17x17	3x3	0.71	0.82	11.87	1.43	12.58	2.25	376	97.59%
	5x5	1.50	1.68	11.99	1.63	13.49	3.31	629	98.17%
	7x7	2.22	2.49	12.07	1.82	14.29	4.31	844	98.21%
	9x9	2.97	3.28	12.18	1.91	15.15	5.19	1034	97.78%
	11x11	3.78	4.20	12.23	1.96	16.01	6.16	1208	97.76%
	13x13	5.08	5.80	16.16	2.53	21.24	8.33	1368	97.70%
	15x15	5.44	6.68	16.18	2.66	21.62	9.34	1516	97.63%

Table V. Total running times of the GLAM and GLAHS based algorithms for the synthetic and the real images of figures 5 and 6.

Images		Image size	Window size	Structuring element	Size of the structuring element	Mean length of the linked list ($\bar{\Lambda}$)	Total running times (s)		Classification rate
							GLAM	GLAHS	
Synthetic images	**Image 1**	256x256	13x13	Cross	5x5	427	1099s	207s	98.65%
	Image 2	256x256	13x13	Circle	5x5	1727	1537s	507s	92.36%
	Image 3	256x256	13x13	Parallelogram	5x5	823	1175s	328s	92.34%
	Image 4	256x256	9x9	Cross	5x5	325	1344s	193s	93.80%
Real images	**Image 1**	481x321	19x19	Y Form	9x9	1195	3760s	1323s	-
	Image 2	100x75	11x11	Parallelogram	7x7	207	79s	33s	-
	Image 3	256x256	5x5	Y Form	5x5	57	1010s	102s	-
	Image 4	256x256	5x5	Parallelogram	5x5	156	1089s	150s	-