Representing Texture Images Using Asymmetric Gray Level Aura Matrices

Xuejie Qin Yee-Hong Yang
{xuq, yang}@cs.ualberta.ca
Department of Computing Science
University of Alberta

Abstract

In this paper, we present a new mathematical framework for modeling texture images. Under this new framework, we prove that the Asymmetric Gray Level Aura Matrices (AGLAMs) of a given image have the necessary and sufficient information to represent the image. Using AGLAMs, a new similarity measure is defined, which is a one-to-one metric in the sense that zero distance between two images will guarantee that the two images are the same. To the best of our knowledge, none of the existing measures has this property. Applications such as learning for image retrieval and texture synthesis can be applied using AGLAMs. The experimental results show that the new AGLAM-based distance measure outperforms existing distance measures in the above mentioned applications.

1 Introduction

In statistical image modeling, one of the most popular and powerful tools is the gray level cooccurrence matrix (GLCM) approach. Indeed, image analysis, synthesis, segmentation, and classification have been studied extensively using GLCMs [3, 5, 11, 13, 14, 19, 38]. However, one limitation of GLCMs is that they only capture the relationship between two pixels, and thus do not work well for structural textures [15]. The problem can be resolved by using gray level aura matrices (GLAMs) [9, 25, 24], which generalize GLCMs by using neighborhood systems to model the relationship between a set of pixels. However, in previous studies on GLAMs [9, 25, 24, 28], for no obvious reasons other than simplicity, the neighborhood systems are always assumed to be symmetric. In the paper, we demonstrate that this assumption or restriction must be removed to allow neighborhood systems of arbitrary shapes for modeling general textures. Our work is the first attempt to give a systematic study on GLAMs that are defined on neighborhood systems with arbitrary shapes, the approach of which is called Asymmetric Gray Level Aura Matrices (AGLAMs). In fact, our work can be considered as a generalization of GLAMs for general texture modeling.

With the above motivation, this paper addresses the following problem: given a texture image X, can we represent it using a set of gray level aura matrices $A(X)$ so that texture analysis (e.g. texture similarity measure and texture classification) can be performed on $A(X)$ only and that the original image X can be faithfully synthesized from $A(X)$?

To further this goal, we have developed a mathematical framework for modeling texture images using AGLAMs. Under this framework, we demonstrate that the AGLAMs of a given image have the necessary and sufficient information to represent the given image. The fundamental theorem we prove in this paper is that: two images are the same if and only if their corresponding AGLAMs on all possible single site neighborhood systems are the same.

Based on the above theorem, the total number of AGLAMs required to accurately represent a given image is the same as the total number of pixels in the image. In practice, this is computationally expensive. However, for texture modeling, we have shown that a small number of AGLAMs calculated from a neighborhood system, which has a much smaller size than that of the image, can be used to effectively represent a texture image.

Texture image retrieval and learning as well as texture synthesis are described using AGLAMs. Other applications like texture image classification, compression, and restoration can be performed under the same AGLAM-based framework. In texture image retrieval, the similarity between two texture images is measured by the sum of the distances between their corresponding basic AGLAMs, where the distance of two matrices is the Manhattan distance of the two matrix vectors. It is proved that the proposed AGLAM-based similarity measure is one-to-one in the sense that zero distance will guarantee the two images are the same. None of the existing proposed measures will guarantee this one-to-one condition. The experimental results have shown that better performances have been obtained with the new AGLAM-based distance measure in similarity learning for texture image retrieval than existing distance measures. Additionally, the results also show that it is possible to perform texture image retrieval without learning using the new measure.

In texture synthesis, it is shown that for a given input texture image, a similar texture can be synthesized by sampling only from the normalized AGLAMs of the input. Experimental results have shown that a broad range of textures can be synthesized using the proposed AGLAMs.

In summary, the main contributions of the paper include: (1) a theoretical framework based on AGLAMs to demonstrate that the gray level aura matrices of an image have the necessary and sufficient information to represent the image for the purpose of general texture modeling if the asymmetry is incorporated into the neighborhood systems, (2) applications of the approach in texture similarity measure and learning, and texture synthesis. In both applications, very promising experimental results are presented to demonstrate the potential of the AGLAM-based framework.

The paper is organized as follows. The related work is described in the next section. In Section 3, we present the AGLAM-based framework for modeling images. In Section 4, two applications are described: texture similarity measure and learning, and texture synthesis using AGLAMs. The experimental results are also presented in this section. Finally, conclusions are given in Section 5.

2 Related Work

Our work is closely related to statistical texture modeling. One of the most influential statistical approaches is the Markov random field (MRF) models [1, 2, 4, 10]. Only a limited range of textures can be modeled with earlier MRF techniques (e.g. the Ising models) because of the small-size cliques and low-order statistics used in modeling. To address the above problems, Zhu et al. propose the FRAME (Filters, Random Fields and Maximum Entropy) model, which incorporates filtering theory into the MRF models for general texture modeling [36]. The conventional MRF texture models are also generalized by Popat and Picard to the cluster-based probability model [26] and by Paget to the strong MRF model [23] for modeling textures with high order statistics. Different from Zhu et al's FRAME model, both approaches are nonparametric. This nonparametric scheme has motivated recent successful texture synthesis techniques through pixel-based sampling (e.g. [7, 16, 35]) or patch-based sampling (e.g. [8, 17, 18]).

Textures can also be characterized by features in multiresolutions. In the work by Simoncelli and Portilla [27, 31], it is shown that texture images can be modeled by a set of joint statistics of complex wavelet coefficients, and that new textures can be synthesized by matching the corresponding joint statistics of the input and output image pyramids. Rather than using global joint statistics, De Bonet and Viola use joint occurrence of local features in multiresolutions to model texture images [6].

Besides MRF models, another influential class of statistical approaches is texture modeling using gray level cooccurrence matrices (GLCMs). Research studies (e.g. [3, 5, 11, 14, 13, 19, 38]) have shown that GLCMs can be used as a powerful tool for texture analysis, synthesis, segmentation and classification. The disadvantage of a GLCM is that it only contains cooccurrence information between two pixels, and thus cannot capture the spatial relationship between three or more pixels in the image. This problem can be addressed by using GLAMs [9, 25, 24], which incorporate neighborhood systems to model the relationship between the target pixel and its neighboring pixels, and thus are capable of capturing the relationship between any number of pixels. In the recent work of Qin and Yang [28], GLAMs have been successfully used in the application of texture image retrieval using learning , and a significantly better performance over existing approaches has been reported.

In all the previous studies on GLAMs, the neighborhood systems are assumed to be symmetric, and hence cannot model anisotropic textures. Our work demonstrates that AGALMs, which generalize GLAMs, can be used for general texture modeling.

Other successful statistical image modeling techniques include the unified statistical framework to visual patterns [37], the multiscale structural similarity method [34], and the Bayesian framework for image parsing [32].

3 AGLAM Framework

In this paper, an image X is modeled as a finite rectangular lattice S of $m \times n$ grids with a neighborhood system $\boldsymbol{N} = \{N_s, s \in S\}$, where N_s is the neighborhood at site s. The neighborhood N_s at site s can be viewed as a translation of a basic neighborhood [9, 25], denoted by E, which is called the *structuring element* for the neighborhood system $\boldsymbol{N}$ after the terminology in mathematical morphology. A single site neighborhood system $\boldsymbol{N} = \{N_s, s \in S\}$ is a system with a structuring element that contains a single neighboring site.

3.1 Background Knowledge

Aura: [9] Given two subsets $A, B \subseteq S$, the *aura* of A with respect to B for neighborhood system $\boldsymbol{N} = \{N_s, s \in S\}$, denoted by $\vartheta_B(A, \boldsymbol{N})$ (or simply $\vartheta_B(A)$), is given by:

$$\vartheta_B(A) = \vartheta_B(A, \boldsymbol{N}) = \bigcup_{s \in A}(N_s \cap B). \quad (1)$$

Aura Measure: [9] With the same notations as in Eq. 1, the *aura measure* of A with respect to B, denoted by $m(A, B, \boldsymbol{N})$ (or simply $m(A, B)$), is given by:

$$m(A, B) = m(A, B, \boldsymbol{N}) = \sum_{s \in A} | N_s \cap B |, \qquad (2)$$

where for a given subset $A \subseteq S$, $| A |$ is the total number of elements in A.

Gray Level Aura Matrix (GLAM): [9] Let $\boldsymbol{N}$ be the neighborhood system over *S*, and $\{S_i, 0 \le i \le G-1\}$ be the gray level sets of an image over *S*, then the gray level aura matrix of the image over $\boldsymbol{N}$, denoted by $\boldsymbol{A}(\boldsymbol{N})$ (or simply $\boldsymbol{A}$), is given by:

$$\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{N}) = [a(i,j)] = [m(S_i, S_j)], \qquad (3)$$

where *G* is the total number of gray levels in the image, $S_i = \{s \in S \mid x_s = i\}$ is the gray level set corresponding to the i^{th} level, $m(S_i, S_j)$ is the aura measure between S_i and S_j given by Eq. 2, and $0 \le i, j \le G-1$.

Intuitively, the aura $\vartheta_B(A)$ gives an interpretation of how a subset *B* is present in the neighborhood of subset *A*. The aura measure $m(A,B)$ evaluates the amount of mixing between subsets *A* and *B*. A large value of $m(A,B)$ implies that the subsets *A* and *B* are mixed together. A small value implies that *A* and *B* are separate from each other.

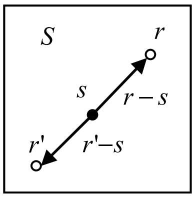
3.2 AGLAM Concepts

In previous studies on GLAMs [9, 25, 24], the neighborhood system is assumed *symmetric* (i.e. for any $s, t \in S$, $s \in N_t$ if and only if $t \in N_s$), by which anisotropic textures cannot be well captured. To address this problem, the concepts of asymmetric neighborhood systems and AGLAMs are introduced below.

Definition 1 The neighborhood system $\boldsymbol{N}$ is *asymmetric* if its structuring element *E* is not symmetric.

Definition 2 Let $r, r', s \in S$, r and r' are symmetric to each other with respect to *s* if $r - s = -(r' - s)$ (i.e. $r' = 2 * s - r$).

In the above definition, sites r, r', $s \in S$ are viewed as points with two coordinates in 2D space. The figure to the right gives an explanation of the relationship between *r* and its corresponding symmetric site r'. From the figure, one can see that the condition of *r* and r' being symmetric to each other with respect to *s* is equivalent to $r - s = -(r' - s)$, i.e. $r' = 2 * s - r$.



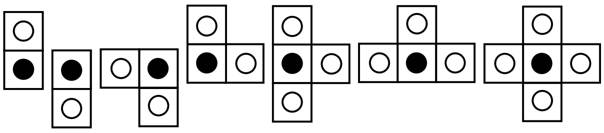


Figure 1: Examples of asymmetric and symmetric neighborhood structures. From left to right, the first six structures are asymmetric, and the last is symmetric, where $s =$'•' is the target pixel and $r =$'∘' is a neighboring pixel of *s*.

Figure 1 gives some examples of asymmetric and symmetric neighborhood structures. The definitions of AGLAM and basic AGLAM are given below.

Definition 3 Given a lattice system *S*, an *AGLAM* on *S* is a *GLAM* computed from an asymmetric neighborhood system $\boldsymbol{N}$.

Definition 4 A *basic AGLAM* is an AGLAM computed from a single site neighborhood system.

Definition 5 Given an arbitrary neighborhood system $\boldsymbol{N}$ over *S* with a structuring element *E*, its *single site neighborhood system decomposition* is a set of single site neighborhood systems defined over *E* as:

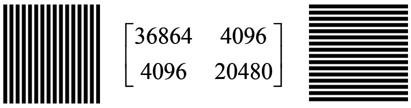
$$\{\boldsymbol{N}_r, r \in E\},$$

where $\boldsymbol{N}_r = \{N_s^r, s \in S\}$ and $N_s^r = \{s + r\}$.

In the rest of the paper, we use SGALM, AGLAM, and GLAM to represent a gray level aura matrix computed using symmetric, asymmetric, and arbitrary (i.e. either symmetric or asymmetric) neighborhood systems, respectively. We assume that all texture images have the same size, i.e. they are defined on the same lattice *S*. This assumption will not cause any loss of generality because for texture images of different sizes, normalized aura matrices can be used.

3.3 AGLAM Theory

Textures cannot be effectively differentiated using SGLAMs. Two images with different textures may have the same SGLAMs (see Figure 2). In this subsection, we prove that a set of basic AGLAMs can give the necessary and sufficient information to differentiate between images.



$$\begin{bmatrix} 36864 & 4096 \\ 4096 & 20480 \end{bmatrix}$$

Figure 2: An example of the inefficiency of SGLAMs for differentiating textures. The right stripe-texture image is a rotation of the left image by 90 degrees, and both images have the same SGLAM that is shown in the middle. Both images are binary, and the nearest four-neighbor neighborhood system is used to compute the SGLAM.

Theorem 1 Any GLAM can be represented as a sum of basic AGLAMs.

Proof: Let *S* be the lattice, $\boldsymbol{A}$ be a GLAM over an arbitrary neighborhood system $\boldsymbol{N}$ with a structuring element *E*, $\{\boldsymbol{N}_r, r \in E\}$ be the single site neighborhood system decomposition of $\boldsymbol{N}$ (see Definition 5, which implies that $N_s = \bigcup_{r \in E} N_s^r$ for any $s \in S$), and A_r be the gray level aura matrix computed from the neighborhood system $\boldsymbol{N}_r$, $\forall r \in E$. By the definitions of aura matrix and aura measure, we have $\boldsymbol{A} = [m(S_i, S_j, \boldsymbol{N})]_{0 \le i, j \le G-1}$, where

$$\begin{aligned} m(S_i, S_j, \boldsymbol{N}) &= \sum_{s \in S_i} | N_s \cap S_j | = \sum_{s \in S_i} | (\bigcup_{r \in E} N_s^r) \cap S_j | \\ &= \sum_{s \in S_i} | \bigcup_{r \in E} (N_s^r \cap S_j) | = \sum_{s \in S_i} \sum_{r \in E} |N_s^r \cap S_j| . \quad (4) \\ &= \sum_{r \in E} \sum_{s \in S_i} |N_s^r \cap S_j| \end{aligned}$$

For any $r \in E$, we have $\boldsymbol{A}_r = [m(S_i, S_j, \boldsymbol{N}_r)]_{0 \le i, j \le G-1}$, where

$$m(S_i, S_j, \boldsymbol{N}_r) = \sum_{s \in S_i} |N_s^r \cap S_j| . \qquad (5)$$

By Eq. 4 and 5, we have $\boldsymbol{A} = \sum_{r \in E} \boldsymbol{A}_r$. Since each $\boldsymbol{N}_r$ is a single site neighborhood system, each $\boldsymbol{A}_r$ is a basic AGLAM. ■

Lemma 1 Let $\boldsymbol{X}_1$ be an image defined on *S*, *s* be a given site in *S*, and ***N*** be any single site neighborhood system. Suppose image $\boldsymbol{X}_2$ is obtained from $\boldsymbol{X}_1$ by changing the intensity value of *s* from g_1 to $g_2 \neq g_1$ (all other pixels' intensity values remain unchanged). Let $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ be the AGLAM of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ over ***N***, respectively, then $\boldsymbol{A}_1 \neq \boldsymbol{A}_2$.

Proof: Only the outline is given below, and the detailed proof can be found in the supplemental material that accompanies the paper. Let $\boldsymbol{A}_1 = [a_1(i,j)]$ and $\boldsymbol{A}_2 = [a_2(i,j)]$ as given in Eq. 3, *r* be the only neighboring pixel of *s* (note that ***N*** is a single site neighborhood system), and *r*' be its symmetric site of *r* with respect to *s* (see Definition 2). Let $g = \boldsymbol{X}_1(r)$ and $g' = \boldsymbol{X}_1(r')$ be the gray levels of *r* and r', respectively, in $\boldsymbol{X}_1$. It can be shown that $\boldsymbol{A}_2$ can be obtained from $\boldsymbol{A}_1$ in the following two steps:

1) Initialize $\boldsymbol{A}_2$ as $\boldsymbol{A}_1$, i.e. $\boldsymbol{A}_2 \leftarrow \boldsymbol{A}_1$.
2) Update $\boldsymbol{A}_2$:

$$\begin{aligned} a_2(g_1, g) &\leftarrow a_2(g_1, g) - 1 \\ a_2(g_2, g) &\leftarrow a_2(g_2, g) + 1 \\ a_2(g', g_1) &\leftarrow a_2(g', g_1) - 1 \\ a_2(g', g_2) &\leftarrow a_2(g', g_2) + 1 \end{aligned} \qquad (6)$$

For all possible values of g and g', after checking each step in Eq. 6 and by noting $g_1 \neq g_2$, one concludes that $\boldsymbol{A}_1 \neq \boldsymbol{A}_2$. ■

Theorem 2 Two images are the same if and only if their corresponding GLAMs on all possible neighborhood systems are the same.

Proof: Suppose that two images $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are defined on *S*, and that they are the same, i.e. $\boldsymbol{X}_1(s) = \boldsymbol{X}_2(s)$ for any $s \in S$. It is obvious that their corresponding GLAMs on all possible neighborhood systems are the same.

Suppose that the corresponding GLAMs of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ on all possible neighborhood systems are the same, we have to show $\boldsymbol{X}_1 = \boldsymbol{X}_2$. This is equivalent to prove that if $\boldsymbol{X}_1 \neq \boldsymbol{X}_2$ then there exists a neighborhood system ***N*** such that the corresponding GLAMs $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ over ***N*** are not equal (i.e. $\boldsymbol{A}_1 \neq \boldsymbol{A}_2$).

Assume that $\boldsymbol{X}_1 \neq \boldsymbol{X}_2$. Let $S = C \cup D$ be a partition on *S*, where $C = \{s \in S \mid \boldsymbol{X}_1(s) = \boldsymbol{X}_2(s)\}$ is the region in which each site has the same gray level in $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, and $D = \{s \in S \mid \boldsymbol{X}_1(s) \neq \boldsymbol{X}_2(s)\}$ is the region in which each site has a different gray level in $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$. Since $\boldsymbol{X}_1 \neq \boldsymbol{X}_2$, we have $D \neq \phi$. Let $|D| = n$ and $D = \{r_i \in S \mid 1 \le i \le n\}$. Choose a neighborhood system ***N*** such that its structuring element *E* is large enough to contain *D*, i.e. $E \supseteq D$. Let $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ be the GLAMs of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ over ***N***, respectively. Using Lemma 1 and mathematical induction, one can prove that $\boldsymbol{A}_1 \neq \boldsymbol{A}_2$ based on the fact that $\boldsymbol{A}_2$ can be obtained from $\boldsymbol{A}_1$ by iteratively applying the update process (see Eq. 6) described in the proof of Lemma 1 on each site in *D*. ■

The above theorem indicates that two images can be differentiated by their corresponding GLAMs over a specific neighborhood system (either symmetric or asymmetric). From the proof, in the worst case, the structuring element of the neighborhood system could be as large as *S*. It is impractical to test all possible neighborhood systems on *S* since the number of possible neighborhood systems is exponential with respect to the size of *S*. However, by Theorem 1, any GLAM can be represented as a sum of basic AGLAMs that are AGLAMs defined on single site neighborhood systems. The following corollary indicates that two images can be differentiated by their corresponding basic AGLAMs over a set of single site neighborhood systems, by which the computation cost can be significantly reduced.

Corollary 1 Two images are the same if and only if their corresponding basic AGLAMs on all possible single site neighborhood systems are the same.

Proof: Suppose $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ are two images over *S*, we only have to prove that if $\boldsymbol{X}_1 \neq \boldsymbol{X}_2$ then there exists a single site neighborhood system $\boldsymbol{N}_r$ such that $\boldsymbol{A}_1 \neq \boldsymbol{A}_2$, where $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are the AGLAMs of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ over $\boldsymbol{N}_r$, respectively.

Suppose $\boldsymbol{X}_1 \neq \boldsymbol{X}_2$, then by Theorem 2, there exists a neighborhood system $\boldsymbol{N}$ such that their corresponding GLAMs $\boldsymbol{A}_1(\boldsymbol{N})$ and $\boldsymbol{A}_2(\boldsymbol{N})$ are not equal. Let $\{\boldsymbol{N}_r, r \in E\}$ (where E is the structuring element of $\boldsymbol{N}$) be the single site neighborhood system decomposition of $\boldsymbol{N}$ (see Definition 5), then by Theorem 1, we have:

$$\boldsymbol{A}_1(\boldsymbol{N}) = \sum_{r \in E} \boldsymbol{A}_1(\boldsymbol{N}_r), \ \boldsymbol{A}_2(\boldsymbol{N}) = \sum_{r \in N} \boldsymbol{A}_2(\boldsymbol{N}_r) \quad (7)$$

Since $\boldsymbol{A}_1(\boldsymbol{N}) \neq \boldsymbol{A}_2(\boldsymbol{N})$, there must exist $r \in E$ such that $\boldsymbol{A}_1(\boldsymbol{N}_r) \neq \boldsymbol{A}_2(\boldsymbol{N}_r)$ because otherwise we have $\boldsymbol{A}_1(\boldsymbol{N}) = \boldsymbol{A}_2(\boldsymbol{N})$ by Eq. 7. ■

In the worst case, the total number of basic AGLAMs required to accurately represent a given image is the same as the lattice size. In practice, this is still computationally expensive. However, we found that only a small number (much smaller than the image size) of basic AGLAMs are required to effectively represent a texture image for satisfactory outputs. As an example, for the texture similarity measure and learning described in the next section, we use 80 AGLAMs to represent a 64x64 image, which are calculated from a square window of size 9x9 around a target pixel.

4 Applications

Two applications of using AGLAMs: texture similarity measure and learning as well as texture synthesis are described in this section.

4.1 Texture Similarity Measure and Learning

Given two texture images $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ defined on S, let $\boldsymbol{N}$ be a neighborhood system, E be its structuring element, $\{\boldsymbol{N}_r, r \in E\}$ be the single site neighborhood system decomposition of $\boldsymbol{N}$ (see Definition 5), and $\boldsymbol{A}_{ir}$ be the basic AGLAM of $\boldsymbol{X}_i$ over $\boldsymbol{N}_r$ for $i = 1, 2$. The similarity measure between $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ is defined by the following distance function:

$$d(\boldsymbol{X}_1, \boldsymbol{X}_2) = \sum_{r \in E} \| \boldsymbol{A}_{1r} - \boldsymbol{A}_{2r} \|, \quad (8)$$

where for a given matrix $\boldsymbol{A} = [a(i,j)]_{0 \leq i,j \leq G-1}$, its norm is computed by $\| \boldsymbol{A} \| = \sum_{i,j=0}^{G-1} | a(i,j) |$.

It is a standard requirement that a distance measure is metric by satisfying the properties of non-negativity, symmetry, and triangle inequality though the necessity of the triangle inequality is argued (doubted) by some researchers (see [29]). It is easy to check that the distance measure defined in Eq. 8 is metric.

Most importantly, one unique property of the new distance measure is that it is one-to-one. Namely, if the neighborhood system $\boldsymbol{N}$ used in Eq. 8 is large enough, a distance measure of zero will guarantee that the two images are the same. This can be easily proved using Corollary 1. As far as we know, none of the existing similarity measures is one-to-one.

Similarity learning using AGLAMs can be performed using a SVM (Support Vector Machine)-based approach similar to the one used in Qin and Yang's work [28]. However, the learning in our approach is performed on AGLAMs calculated directly from images without using filters, while in Qin and Yang's work, the learning is performed on SGLAMs calculated in image pyramids (i.e. multiresolutions), which require filters.

Using SGLAMs in multiresolutions, a significantly better performance over existing filter-based approaches (e.g. [12, 20, 21]) has been reported in Qin and Yang's work. In this paper, we compare our approach with theirs although comparisons with other approaches are interesting and will be carried out in the future.

The Brodatz texture database, which is used for our experimentation, contains 112 texture classes, each of which is a 512x512 image. For the ease of comparison, we use the same experimental set-up as in Qin and Yang's work [28]. More precisely, each texture image, which is called a *class*, is divided into 49 subimages of 128x128 pixels each overlapping with other subimages, whose central pixels are on a 7x7 grid over the original image. The first 33 subimages are used as the training set, and the rest of them (16 images) are used for retrieval, in which one of them will be used as a query image. Therefore, we have a database of 3696 images for learning, and a database of 1792 images for retrieval. The 112 texture image classes are grouped into 32 *clusters* manually, where each cluster consists of perceptually similar textures (i.e. a set of classes).

Given a query image, to find all images in the retrieval database that look similar to the query image, the first step is to learn the cluster information from the learning database using a SVM algorithm (e.g. [33]). In this step, the texture feature space is represented by the set of all AGLAMs computed from the texture images in the learning database. The second step is to classify the query database and the query image based on the cluster information learned from the first step. In the third step, all images in the query database that are in the same cluster as the query image are retrieved. Finally, all retrieved images are ranked based on their AGLAM-based distances defined in Eq. 8 to the query image.

Figure 3 gives the evaluation results in texture image retrieval between our AGLAM-based approach and the SGLAM-based approach [28]. For a given query image, the average retrieval accuracy is defined as the average percentage of the number of texture images retrieved in the same class as the query image in the total number of images retrieved. For example, in Figure 3, for a given query image $\boldsymbol{X}$, suppose 20 images have been retrieved from the database, among them 12 images are in the same class as ***X***, then the retrieval accuracy is 12/15 = 80% because there are a total of 15 other texture images in the same class as ***X***.

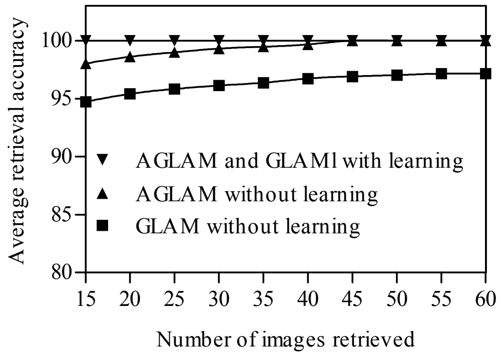


Figure 3: Retrieval performance evaluation.

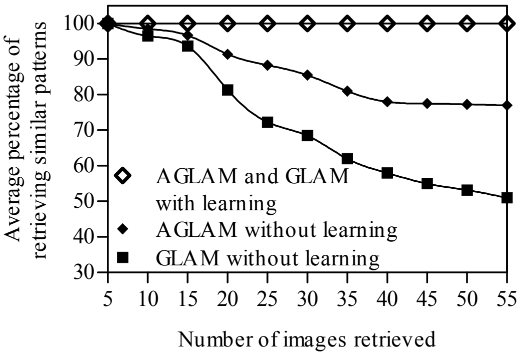


Figure 4: Performance evaluation of retrieving similar textures.

Figure 4 gives the results of performance evaluation in retrieving similar textures (i.e. textures in the same cluster as the query image) for the two approaches. Given a query image, let 20 images be retrieved. If 15 of the 20 retrieved images are in the same cluster as the query image, then the percentage of retrieving similar patterns is 15/20 = 75%. Figure 7 gives some examples of retrieving similar textures for a given query texture.

With learning, both approaches have obtained a retrieval accuracy of 100%. Without learning, the AGLAM-based measure outperforms the SGLAM-based measure. In similarity ranking as shown in Figure 7, the new measure has better performance over the SGLAM-based measure. In all the examples shown in this subsection, a SGLAM is computed from the steerable pyramid [30] of an image with 5 levels, while an AGLAM is computed from the image itself with a neighborhood size of 9×9.

Interestingly, as shown in Figure 4, without learning, the AGLAM-based measure has obtained a retrieval accuracy of above 75% no matter what is the number of images retrieved, and the performance becomes stable when the number of retrieved images increases. This suggests that the AGLAM-based measure may be used for image retrieval without learning, more evidence needs to be given by future research.

4.2 Texture Synthesis

In texture synthesis, for a given input texture image, a similar texture can be synthesized by sampling from the normalized AGLAMs of the input only (an aura matrix $A=[a(S_i,S_j)]_{0\le i,j\le n}$ is said to be *normalized* if $\sum_{i,j=0}^{n} a(S_i,S_j)=1$). This is done by iteratively modifying the gray level of each pixel in the output image until the distance (defined in Eq. 8) between the corresponding normalized AGLAMs of the output and those of the input is small enough.

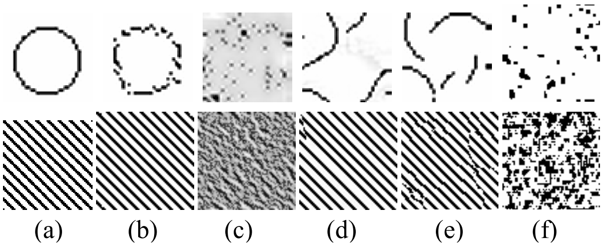


Figure 5: Examples of texture synthesis of the AGLAM-based algorithm compared with existing algorithms, where images in column (a) are the input samples, and images in the next four columns ((b) – (e)) are the synthesized results for the AGLAM-based algorithm, the Heeger and Bergen algorithm [16], the Wei and Levoy algorithm [35], and the Paget and Longstaff algorithm [22]. The images in (f) are synthesized by sampling the SGLAMs of the input textures.

During the synthesis process, the AGLAMs of the output and the distance between the corresponding AGLAMs of the input and output can be iteratively updated in an efficient way without the recalculation using an updating process as described in Eq. 6 in the proof of Lemma 1.

The above is the outline of the AGLAM-based algorithm for texture synthesis. Because of space limitations, we will not describe the details of the algorithm, nor compare it with existing techniques in this paper, as they will be presented in another paper.

Figure 5 gives some comparison results of texture synthesis. More examples of texture synthesis of the AGLAM-based algorithm can be found in Figure 6 and in the supplemental material that accompanies the paper. One can observe that the results are very encouraging.

Our experiments have shown that a wide range of textures can be faithfully synthesized by sampling from AGLAMs of input textures. It is noteworthy that

SGLAMs are inappropriate for texture synthesis as shown in Figure 5 (f).

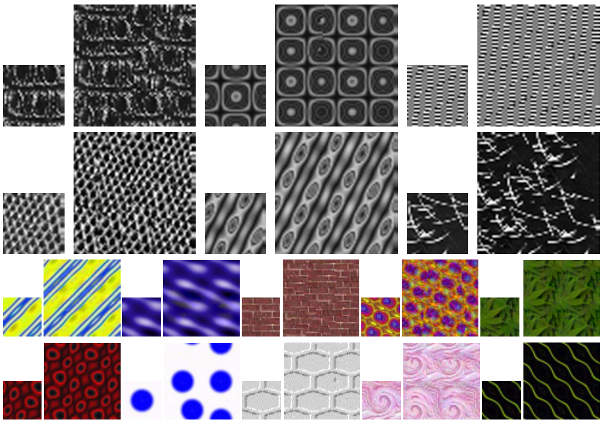


Figure 6: More examples of texture synthesis using the AGLAM-based algorithm, where the smaller images are the input and the larger ones are the output. The sizes of the input and output images are 64x64 and 128x128, respectively, and a neighborhood size of 21x21 is used for calculating AGLAMs for all images in this figure.

5 Conclusions

This paper presents a mathematical framework for modeling texture images based asymmetric gray level aura matrices (AGLAMs). It is shown that the basic AGLAMs of an image have the sufficient and necessary information to represent the image. Based on this fact, a new distance measure is defined using a set of basic AGLAMs, which is one-to-one in the sense that zero distance will guarantee the two images measured are the same. To the best of our knowledge, none of the existing measures guarantees this one-to-one condition.

Two applications of the new framework are discussed: texture similarity measure and learning for image retrieval as well as texture synthesis. Experimental results have shown that better performances have been obtained with the new AGLAM-based distance measure in similarity learning for texture image retrieval than existing distance measures. Experiments also suggest that it is possible to perform texture image retrieval using the new measure without learning, which needs future study to confirm this finding. For texture synthesis, we have shown that a wide range of textures can be faithfully synthesized by sampling only from AGLAMs of input textures without requiring additional information elsewhere.

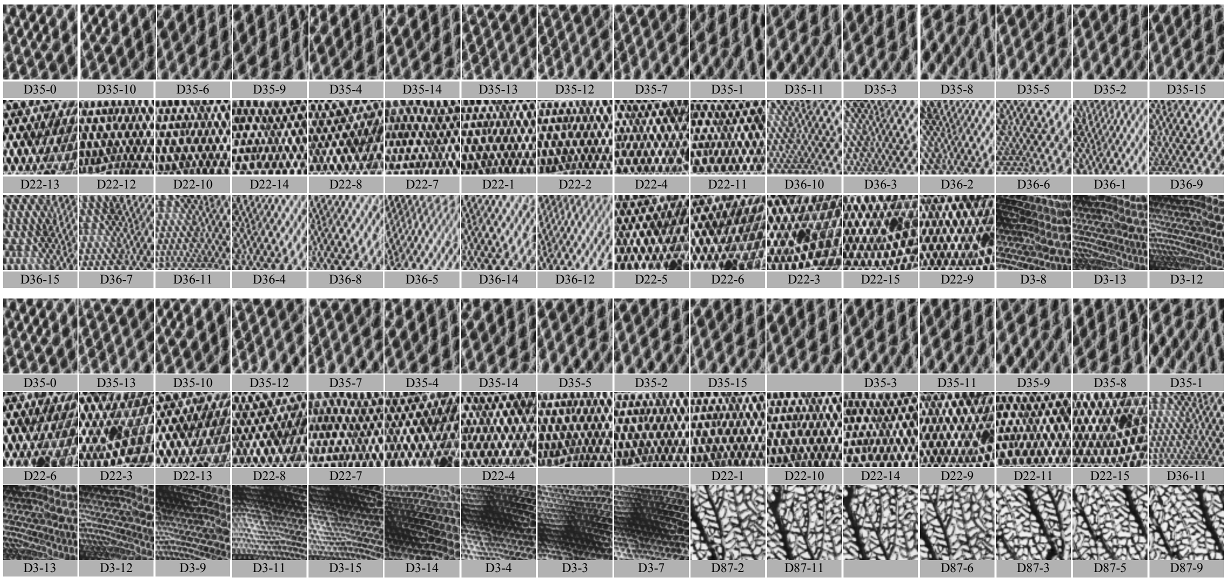
Acknowledgements

The authors would like to thank the anonymous reviewers and appropriate agencies.

References

1. Chellappa, R. and S. Chatterjee, *Classification of Textures Using Gaussian Markov Random Fields.* IEEE ASSP, 1985: p. 959-963.
2. Cohen, F.S. and D.B. Cooper, *Simple Parallel Hierarchical and Relaxation Algorithms for Segmenting Noncausal Markovian Random Fields.* PAMI, 1987: p. 195-219.
3. Copeland, A.C., G. Ravichandran, and M.M. Trivedi, *Texture Synthesis Using Gray-Level Co-occurrence Models: Algorithms, Experimental Analysis, and Psychophysical Support.* Opt. Eng., 2001: p. 2655-2673.
4. Cross, G.C. and A.K. Jain, *Markov Random Field Texture Models.* PAMI, 1983. **5**(2): p. 25-39.
5. Davis, L.S., S.A. Johns, and J.K. Aggarwal, *Texture Analysis Using Generalized Cooccurence Matrices.* PAMI, 1979. **1**(3): p. 251-259.
6. DeBonet, J.S. and P. Viola, *Texture Recognition Using a Non-parametric Multi-Scale Statistical Model.* IEEE CVPR, 1998: p. 641-647.
7. Efros, A. and T. Leung, *Texture Synthesis by Non-Parametric Sampling.* ICCV, 1999: p. 1033-1038.
8. Efros, A.A. and W.T. Freeman, *Image Quilting for Texture Synthesis and Transfer.* Siggraph, 2001: p. 341-346.
9. Elfadel, I.M. and R.W. Picard, *Gibbs Random Fields, Cooccurrences, and Texture Modeling.* PAMI, 1994: p. 24-37.
10. Geman, S. and D. Geman, *Stochastic relaxation, Gibbs distributions, and the Bayesian Restoration of Images.* PAMI, 1984. **6**: p. 721-741.
11. Gotlieb, C.C. and H.E. Kreyszig, *Texture Descriptors Based on Co-occurrence Matrices.* CVGIP, 1990: p. 70-86.
12. Guo, G., S.Z. Li, and K.L. Chan, *Learning Similarity for Texture Image Retrieval.* ECCV, 2000: p. 178-190.
13. Haralick, R.M., K. Shanmugan, and I.H. Dinstein, *Textural Features for Image Classification.* IEEE Trans. Syst. Man Cybern., 1973: p. 610–621.
14. Haralick, R.M., *Statistical and Structural Approaches to Texture.* In Proc. 4th Int. Joint Conf. Pattern Recognition, 1978: p. 45-69.
15. Haralick, R.M. and L.S. Shapiro, *Computer Vision, Vol. 1.* Addison Wesley, 1992.
16. Heeger, A.J. and J.R. Bergen, *Pyramid-Based Texture Analysis/Synthesis.* Siggraph, 1995: p. 229-238.
17. Kwatra, V., et al., *Graphcut Textures: Image and Video Synthesis using Graph Cuts.* Siggraph, 2003. **22**(3): p. 227-286.
18. Liang, L., et al., *Real-Time Texture Synthesis by Patch-Based Sampling.* TOG, 2001: p. 127-150.
19. Lohmann, G., *Co-occurrence-Based Analysis and Synthesis of Textures.* ICPR, 1994. **1**: p. 449-453.

20. Ma, W. and B.S. Manjunath, *Texture Features and Learning Similarity.* IEEE CVPR, 1996: p. 425-430.
21. Manjunath, B.S. and W.Y. Ma, *Texture features for browsing and retrieval of image data.* PAMI, 1996. **18**(8): p. 837-842.
22. Paget, R. and I.D. Longstaff, *Texture Synthesis via a Noncausal Nonparametric Multiscale Markov Random Field.* IEEE TIP, 1998. **7**(6): p. 925-931.
23. Paget, R., *Strong Markov Random Field Model.* PAMI, 2004. **26**(3): p. 408-413.
24. Picard, R.W., I.M. Elfadel, and A.P. Pentland, *Markov/Gibbs Texture Modeling: Aura Matrices and Temperature Effects.* IEEE CVPR, 1991: p. 371-377.
25. Picard, R.W. and I.M. Elfadel, *Structure of Aura and Co-occurrence Matrices for the Gibbs Texture Model.* J. of Math. Imaging & Vision, 1992: p. 5-25.
26. Popat, K. and R.W. Picard, *Cluster-Based Probability Model and its Application to Image and Texture Processing.* IEEE TIP, 1997. **6**(2): p. 268-284.
27. Portilla, J. and E.P. Simoncelli, *A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients.* IJCV, 2000. **40**(1): p. 49-71.
28. Qin, X. and Y.H. Yang, *Similarity Measure and Learning with Gray Level Aura Matrices (GLAM) for Texture Image Retrieval.* ICVPR, 2004: p. 326-333.
29. Santini, S. and R. Jain, *Similarity measures.* PAMI, 1999. **21**(9): p. 871-883.
30. Simoncelli, E.P., et al., *Shiftable Multi-Scale Transforms.* IEEE TIT, 1992. **38**: p. 587-607.
31. Simoncelli, E.P. and J. Portilla, *Texture Characterization via Joint Statistics of Wavelet Coefficient Magnitudes.* IEEE ICIP, 1998. **1**: p. 62-66.
32. Tu, Z., et al., *Image Parsing: Unifying Segmentation, Detection, and Recognition.* ICCV, 2003: p. 18-25.
33. Vapnik, V.N., *Statistical Learning Theory*. 1998: John Wiley & Sons, New York, 1998.
34. Wang, Z., et al., *Image Quality Assessment: from Error Measurement to Structural Similarity.* IEEE TIP, 2004. **13**(4): p. 600-612.
35. Wei, L. and M. Levoy, *Fast Texture Synthesis Using Tree-Structured Vector Quantization.* Siggraph, 2000: p. 479-488.
36. Zhu, S.C., Y. Wu, and D. Mumford, *Filters, Random Fields and Maximum Entropy - Towards a Unified Theory for Texture Modeling.* IJCV, 1998: p. 1-20.
37. Zhu, S.C., *Statistical Modeling and Conceptualization of Visual Patterns.* PAMI, 2003: p. 691-712.
38. Zucker, S.W., *Finding Structure in Co-occurrence Matrices for Texture Analysis.* CVGIP, 1980. **12**: p. 286-308.



D35-0 D35-10 D35-6 D35-9 D35-4 D35-14 D35-13 D35-12 D35-7 D35-1 D35-11 D35-3 D35-8 D35-5 D35-2 D35-15
D22-13 D22-12 D22-10 D22-14 D22-8 D22-7 D22-1 D22-2 D22-4 D22-11 D36-10 D36-3 D36-2 D36-6 D36-1 D36-9
D36-15 D36-7 D36-11 D36-4 D36-8 D36-5 D36-14 D36-12 D22-5 D22-6 D22-3 D22-15 D22-9 D3-8 D3-13 D3-12
D35-0 D35-13 D35-10 D35-12 D35-7 D35-4 D35-14 D35-5 D35-2 D35-15 D35-3 D35-11 D35-9 D35-8 D35-1
D22-6 D22-3 D22-13 D22-8 D22-7 D22-4 D22-1 D22-10 D22-14 D22-9 D22-11 D22-15 D36-11
D3-13 D3-12 D3-9 D3-11 D3-15 D3-14 D3-4 D3-3 D3-7 D87-2 D87-11 D87-6 D87-3 D87-5 D87-9

Figure 7: Examples of retrieving similar texture images for a given query image, which is the first image located at the top left corner in each picture with label D35-0. The textures in the top picture, which contains 48 texture images, are obtained by the AGLAM-based measure, and are ordered (ranked) by their AGLAM distances to the query image. The textures in the bottom picture are obtained by the SGLAM-based measure, and are ordered by the SGLAM distances to the query image. One can see that the AGLAM-based similarity ranking method is more accurate than the SGLAM-based ranking method. For example, texture images in the top picture whose labels contain D36 look more similar to the query image than the ones whose labels contain D87 are successfully retrieved by the AGLAM-based measure, but not by the SGLAM-based measure. On the other hand, D87-type texture images, which are less similar to the query image than D36-type texture images, are retrieved and ranked incorrectly with higher similarity degrees by SGLAM-based measure.