

Texture Structure Analysis

by

Srenivas Varadarajan

A Dissertation Presented in Partial Fulfillment  
of the Requirements for the Degree  
Doctor of Philosophy

Approved April 2014 by the  
Graduate Supervisory Committee:

Lina J. Karam, Chair

Chaitali Chakrabarti

Baoxin Li

Cihan Tepedelenlioglu

ARIZONA STATE UNIVERSITY

May 2014

## ABSTRACT

Texture analysis plays an important role in applications like automated pattern inspection, image and video compression, content-based image retrieval, remote-sensing, medical imaging and document processing, to name a few. Texture Structure Analysis is the process of studying the structure present in the textures. This structure can be expressed in terms of perceived regularity. Our human visual system (HVS) uses the perceived regularity as one of the important pre-attentive cues in low-level image understanding. Similar to the HVS, image processing and computer vision systems can make fast and efficient decisions if they can quantify this regularity automatically. In this work, the problem of quantifying the degree of perceived regularity when looking at an arbitrary texture is introduced and addressed. One key contribution of this work is in proposing an objective no-reference perceptual texture regularity metric based on visual saliency. Other key contributions include an adaptive texture synthesis method based on texture regularity, and a low-complexity reduced-reference visual quality metric for assessing the quality of synthesized textures.

In order to use the best performing visual attention model on textures, the performance of the most popular visual attention models to predict the visual saliency on textures is evaluated. Since there is no publically available database with ground-truth saliency maps on images with exclusive texture content, a new eye-tracking database is systematically built. Using the Visual Saliency Map (VSM) generated by the best visual attention model, the proposed texture regularity metric is computed. The proposed metric is based on the observation that VSM characteristics differ between textures of differing regularity. The proposed texture regularity metric is based on two texture regularity scores, namely a textural similarity score and a spatial distribution score. In order to evaluate the performance of the proposed

regularity metric, a texture regularity database called *RegTEX*, is built as a part of this work. It is shown through subjective testing that the proposed metric has a strong correlation with the Mean Opinion Score (MOS) for the perceived regularity of textures. The proposed method is also shown to be robust to geometric and photometric transformations and outperforms some of the popular texture regularity metrics in predicting the perceived regularity.

The impact of the proposed metric to improve the performance of many image-processing applications is also presented. The influence of the perceived texture regularity on the perceptual quality of synthesized textures is demonstrated through building a synthesized textures database named *SynTEX*. It is shown through subjective testing that textures with different degrees of perceived regularities exhibit different degrees of vulnerability to artifacts resulting from different texture synthesis approaches. This work also proposes an algorithm for adaptively selecting the appropriate texture synthesis method based on the perceived regularity of the original texture. A reduced-reference texture quality metric for texture synthesis is also proposed as part of this work. The metric is based on the change in perceived regularity and the change in perceived granularity between the original and the synthesized textures. The perceived granularity is quantified through a new granularity metric that is proposed in this work. It is shown through subjective testing that the proposed quality metric, using just 2 parameters, has a strong correlation with the MOS for the fidelity of synthesized textures and outperforms the state-of-the-art full-reference quality metrics on 3 different texture databases. Finally, the ability of the proposed regularity metric in predicting the perceived degradation of textures due to compression and blur artifacts is also established.

*To My Parents*

## ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Dr.Lina Karam, whose invaluable guidance and support have enabled me to complete this research work. I am very much grateful to Dr.Karam for teaching me the various aspects of research and hand-holding me through the most effective one-on-one research meetings. My sincere thanks to her for the extra-ordinary kindness, encouragement and support she has shown me, especially during the most stressful times along my Ph.D. journey.

I would like to extend my thanks to all the members of my committee, Dr.Chaitali Chakrabarti, Dr.Cihan Tepedelenlioğlu and Dr.Baoxin Li for their kind guidance and constructive feedback on my research.

I also thank the ECEE Graduate Program Chair, Dr.Joseph Palais, for supporting my educational funding with a Teaching Assistantship during the initial phase of my Ph.D.. My special thanks to the EECE graduate advisor, Ms.Esther Korner for clearly, patiently and promptly explaining me all the guidelines of the ECEE department and the graduate college.

I truly acknowledge and appreciate the kind help of Dr.Dmitry Chetverikov and Dr.Damon Chandler in sharing with us the authentic implementation of their research work.

I also express my utmost thanks and appreciation to Dr.Judit Martinez of Qualcomm, for her excellent contributions towards the development my research skills and research ideas in several ways.

I am grateful to my fellow Ph.D. students, Milind Gide and Sam Dodge for all the interesting and informative research discussions on visual attention and eye-

tracking. I would like to thank Qian Xu and Jinjin Li for collaborating with me on some of the research projects.

I also express my sincere thanks to my amazing friends Sivaraman Dasarathan, Divya Venkatraman, Emmanuel Jillela and Suresh Muthumani who have given me extra-ordinary help and support during the crests and troughs of my Ph.D. wave. I also thank all the IVU lab students, fellow researchers and my colleagues at Intel who participated in the subjective testing experiments that helped in gathering the vital ground-truth data for my research.

I would like to thank my managers Mr.Krishanan Rajamani, Mr.Kumar Narasimha, Mr.Sivaji Gottumukkala and Mr.Craig Spalding for extending their kind support in my relocation back to Arizona and continuing my Ph.D. while working full-time at Intel. I would like to express my sincere thanks to my talented colleagues, Mr.Thad Letnes, Mr.Keith Hazelet and Mr.Jesus Menchaca, whose guidance and help at office played a vital role for me to balance job and Ph.D.

Finally, I dedicate this work to my parents for their innumerable sacrifices towards my education in general and my Ph.D. in particular. I am also thankful to my wife for her enormous support in the pursuit of my Ph.D. dream.

# TABLE OF CONTENTS

	Page
TABLE OF CONTENTS . . . . .	vi
LIST OF FIGURES . . . . .	viii
CHAPTER	
1 INTRODUCTION . . . . .	1
1.1 Applications of Texture Structure Analysis . . . . .	2
1.2 Visual Attention Models . . . . .	5
1.3 Summary of Contributions . . . . .	6
1.4 Outline of the report . . . . .	8
2 BACKGROUND AND EXISTING APPROACHES . . . . .	9
2.1 Background . . . . .	9
2.1.1 Statistical Methods . . . . .	9
2.1.2 Model-based methods . . . . .	12
2.1.3 Signal Processing Methods . . . . .	17
2.1.4 Geometric Methods . . . . .	22
2.2 Existing Approaches for Texture Regularity Measurement . . . . .	24
2.3 Limitations of Existing Approaches . . . . .	25
3 VISUAL ATTENTION MODELS ON TEXTURES . . . . .	28
3.1 Eye Tracking Database . . . . .	29
3.2 Performance Metrics . . . . .	31
3.3 Performance Evaluation Results . . . . .	34
4 PROPOSED PERCEPTUAL TEXTURE REGULARITY METRIC . . . . .	39
4.1 Proposed Texture Regularity Metric . . . . .	39
4.1.1 Textural Similarity Score . . . . .	40
4.1.2 Spatial Distribution Score . . . . .	43
4.2 Simualtion Results . . . . .	49
4.3 Robustness to Geometric and Photometric Transformations . . . . .	55

CHAPTER	Page
5 ADAPTIVE TEXTURE SYNTHESIS BASED ON TEXTURE REGULARITY . . . . .	59
5.1 Existing Approaches for Texture Synthesis . . . . .	60
5.2 Proposed Adaptive Texture Synthesis based on Texture Regularity . .	65
5.3 Simulation Results . . . . .	69
6 A REDUCED REFERENCE PERCEPTUAL QUALITY METRIC FOR TEXTURE SYNTHESIS . . . . .	73
6.1 Existing Reduced Reference and Texture Quality Metrics . . . . .	73
6.2 Proposed Reduced Reference Metric based on Texture Regularity . .	76
6.2.1 Effect of Texture Synthesis Artifacts on Regularity . . . . .	76
6.2.2 Effect of Texture Synthesis Artifacts on Granularity . . . . .	79
6.2.3 Proposed Texture Granularity Metric . . . . .	79
6.3 SynTEX Database . . . . .	84
6.4 Simulation Results . . . . .	85
7 EFFECT OF TEXTURE REGULARITY ON PERCEIVED TEXTURE DISTORTIONS . . . . .	91
7.1 Effect of Texture Regularity on Perceived Compression Artifacts . . .	91
7.2 Simulation Results . . . . .	95
7.3 Effect of Texture Regularity on perceived Blur . . . . .	98
8 Conclusion . . . . .	101
8.1 Summary of Contributions . . . . .	101
8.2 Future Research Directions . . . . .	103
REFERENCES . . . . .	104



## LIST OF FIGURES

Figure	Page
1.1 Examples of irregularity in textures due to (a) placement, (b) size, shape or color, (c) directionality, and (d) fine-granularity of primitives. . . . .	2
2.1 Clique-types for a second-order neighborhood system. . . . .	12
2.2 A three-level analysis filter bank for implementing the DWT. $L(z)$ and $H(z)$ denote the Lowpass and the Highpass analysis filters, respectively. . . . .	21
2.3 Limitations of pixel domain methods to predict perceived texture regularity: (a) An irregularly perceived texture - greynoise; (b) Binary Gray Level Co-occurrence matrix (BCM) elements as a function of the horizontal displacement for the texture in (a); (c) Expected value of the Extended Gray Level Difference Histogram (EGLDH) along $\alpha = 0^\circ$ as a function of displacement for the texture in (a). . . . .	25
3.1 Validating the eye-tracker system and the software for computing fixations: (a) Test Pattern (red-dot); (b) Fixation Overlaid (blue-cross); (c) Ground-truth Saliency Map. . . . .	30
3.2 Performance evaluation of VA models: (a) AUC with randomly chosen locations as the negative set (AUC I); (b) AUC with non-fixated locations as the negative set (AUC II); (c) Shuffled AUC with fixated locations on other images as the negative set; (d) Normalized Scan path Saliency (NSS); (e) 2D Correlation between predicted and ground-truth saliency maps. . . . .	35
3.3 ROC curves for different VA models. . . . .	36
4.1 Block diagram of the proposed texture regularity metric. . . . .	41
4.2 An Illustration of relationship between the saliency map and the histogram of the saliency map for a regular texture. . . . .	42

Figure	Page
4.3 Examples of affine transformations: (a) Original texture; (b) Affine transform 1; (c) Affine transform 2. . . . .	56
4.4 Examples of color transformations: (a) Original texture; (b) Hue transform; (c) Saturation transform; (d) Contrast transform. . . . .	56
5.1 The general framework for video codecs based on texture synthesis. . . . .	61
5.2 Block diagram of the proposed adaptive texture synthesis algorithm. The regularity thresholds $Th_1$ and $Th_2$ are used for classifying the textures as irregular or hybrid. . . . .	70
6.1 Block diagram of the proposed texture synthesis quality metric. . . . .	82
6.2 Performance of quality metrics on the QualTEX database for different texture synthesis algorithms. . . . .	87
6.3 Performance of the quality metrics on the SynTEX database for different texture synthesis algorithms. . . . .	88
6.4 Performance of quality metrics on the SynthPQA database for different texture synthesis algorithms. . . . .	90

## Chapter 1

### INTRODUCTION

Textures are present in almost everything we see around us, both in natural and man-made objects. Barks of trees, leaves, grass, flowers and ripples of water are all examples of natural textures. Tiles on the floor, carpets, and all types of printed fabrics can be cited as examples for man-made textures that we see every day. Each of these objects has a spatially repetitive pattern of visual properties that characterize the specific object and help us in their recognition, classification and segmentation. This repeated pattern of pixel intensities or color constitute a visual texture. The patterns can be the result of physical surface properties such as roughness or oriented strands which often have a tactile quality. The patterns can also be the result of reflectance differences such as the color on a surface.

Textures are represented by describing this pattern of pixel intensities. The popular approaches for texture representation and analysis are reviewed in Chapter 2. These include the statistical approaches, model-based approaches, geometry-based approaches and structural approaches. In this work, a structural approach is taken for the representation and analysis of textures.

The pattern of visual intensities that is spatially repeated throughout the texture in some regular or irregular manner is called a primitive. The primitives of a texture exhibit varying degrees of similarity in the visual properties like size, shape, color and orientation. These properties of the primitives along with the degree of periodicity in their placements, determine the overall perceived regularity of the primitives. As mentioned in [1], the texture structure can also be quantified through the local properties of the primitives as well as through the organization amongst



Figure 1.1: Examples of irregularity in textures due to (a) placement, (b) size, shape or color, (c) directionality, and (d) fine-granularity of primitives.

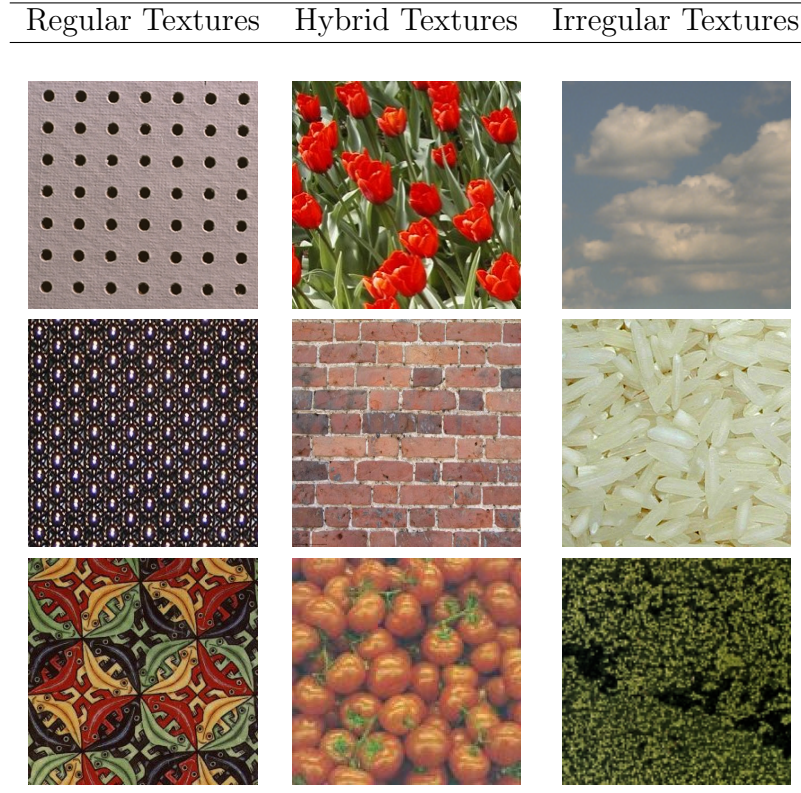
the primitives. Texture structure manifests itself as the perceived regularity of the primitives. Examples of irregularity in textures due to each of these properties are illustrated in Fig. 1.1. Sometimes the irregularity in a texture is so high that, it becomes very difficult to locate and define a primitive. Texture Structure Analysis essentially involves understanding the regularity present in a texture.

Based on the combined regularities in each of the visual properties, textures can be broadly classified as regular, hybrid and irregular textures. Examples of regular, hybrid and irregular textures are shown in columns 1, 2 and 3 of Table 1.1, respectively. It is proven that the regularity of textural properties is one of the important visual cues in early vision [2]. This cue helps the Human Visual System (HVS) in the acceleration of image processing tasks like image segmentation. If the regularity of textures can also be inferred quickly by computer vision systems, then there can also be a significant speed-up in automated decision making. Hence one of the key motivations of this work is to quantify the perceived regularity of textures through the proposed texture regularity metric, which would help in many image processing and computer vision problems as described below.

### 1.1 Applications of Texture Structure Analysis

Texture analysis in general, plays an important role in applications like automated pattern inspection, image and video compression, content based image retrieval, remote-sensing, medical imaging and document processing. Texture Structure Analysis is the process of studying the structure present in the textures and is the main

Table 1.1: Examples of regular, hybrid and irregular textures.



focus of this work. This structure is quantified in terms of a Texture Regularity Metric that represents the amount of perceived regularity when looking at an arbitrary texture. This regularity metric can be of significant use in many compelling applications as explained below.

### *Image and Video Compression*

In natural images and video sequences, a significant amount of the background consists of textures. Since most of the viewers attention is on the foreground, it is unnecessary to compress the background regions with high fidelity. Also, since textures have significant high-frequency components, a lot of bits would be consumed to compress them using conventional DCT-based transform domain methods. In texture-based coding frameworks like [3], [4] and [5], a small patch or a set of patches called exemplars, representing the background textures are compressed and sent as

a part of the bit-stream. The background texture is synthesized from these exemplars at the decoder using a texture synthesis algorithm like [6] and [7]. There are a number of approaches for texture synthesis that offer a trade-off between compression efficiency, visual quality and speed of processing. Some of the texture synthesis algorithms are suitable for stochastic or random textures while others are more appropriate for synthesizing highly deterministic textures. Hence, a Texture Regularity Metric would help in applying the appropriate method for texture synthesis.

#### *Content Based Image Retrieval*

The process of using a small image patch to retrieve similar images from a database is called Content Based Image Retrieval (CBIR). This involves extracting the features of the image-patch and matching with the features of each image in the database. When the size of the database becomes large, feature matching can become an extremely slow operation. Since textures constitute a major proportion of natural images, textural features can be very handy in searching these images. Regularity has been used as a key feature in characterizing images in a database as shown in approaches [8] and [9]. A broad classification of textures based on regularity can play a vital role in reducing the search space drastically and hence accelerate the retrieval process.

#### *Automated Pattern Inspection and Defect Detection*

Patterned textures are very common in many industries like the fabric industry. Due to the high volume and speed of production, human evaluation of every sample is impossible. The automated computer vision systems for monitoring quality need to be extremely quick and robust in defect detection for sample isolation. Based on the defect-free samples, an automated inspection system can be first trained for an expected value of regularity. Whenever the regularity of the observed sample deviates

a lot from this expected value, a defect is detected. The efficacy of regularity metrics in automated pattern inspection is described in [10] and [11].

### *Texture-based Region Classification and Object Recognition*

When a natural image containing a number of objects is viewed, early processing of visual information helps us to classify the different regions and each region may correspond to an object. Textures constitute an important visual cue that helps us in identifying these homogenous regions and hence are used for object recognition through approaches like [12]. The goal of automated texture classification is to produce a texture map having the same size of the original image but whose values are the texture indices. Each index corresponds to a texture class. This early classification can be used for object segmentation or computing the object shape from textures.

In order to quantify the perceived regularity of textures, the proposed texture regularity metric uses a visual attention model. The following section introduces the concept of visual attention and the visual saliency map (VSM).

## 1.2 Visual Attention Models

Visual attention is a low-cost pre-processing step by which artificial and biological visual systems select the most relevant information from a scene, and relay it to higher level cognitive areas that perform complex processes such as scene understanding, action selection, and decision making. Visual attention (VA) can be modelled as a bottom-up or a top-down process. Visual saliency refers to the bottom-up processes that render certain image regions more conspicuous compared to their surroundings. For instance, a single red dot among several blue dots is visually salient. Bottom-up saliency has been studied in search tasks such as finding an odd item among distractors in pop-out and conjunction search arrays, as well as in eye movement prediction on free-viewing of images or videos. In contrast to bottom-up, top-down

attention deals with high-level cognitive factors that make image regions relevant, such as task demands, emotions, and expectations. The VA models can also be divided into spatial [13] or spatial-temporal models [14] depending on whether the stimulus is a still image or a video sequence. The still-image based models classify the pixels into saccades and fixations.

The VA models can also be categorized as being space-based or object-based. Object-based models [15] try to segment or detect objects to predict salient regions. This is supported by the finding that objects predict fixations better than early saliency. In contrast, in space-based models, all operations happen at the image level (pixels or image patches), or in the image spectral phase domain. For the space-based models [16], the goal is to create saliency maps that predict which locations have higher probability of attracting human attention. As reviewed in later chapters of this thesis, the performance of various visual attention models can be evaluated by observing the correlation of the predicted saliency maps produced by these models with ground-truth saliency. Although such comparative performance evaluation had been performed earlier on databases with natural images, no such work exists for images containing exclusive textures. A performance benchmarking of visual attention models on textural images is performed as a part of this work.

### 1.3 Summary of Contributions

The most significant contribution of the work is in proposing a no-reference perceptual texture regularity metric based on visual saliency. This is the first ever work that introduces the concept of perceived regularity and systematically captures it through subjective testing. A texture regularity database, named *RegTEX*, is built as part of this work. The *RegTEX* database consists of images of regular, hybrid and irregular textures and their corresponding subjective regularity mean opinion scores. Using visual attention to quantify texture regularity is a novelty in itself.



Simulation results show that the proposed no-reference regularity metric is robust to geometric and photometric transformations and has a high correlation with the scores obtained from subjective testing.

In order to use the most effective visual attention model for computing the proposed texture regularity metric, the performance of the visual attention models to predict fixations on exclusive texture images have been evaluated for the first time. For this purpose, a texture eye-tracking database, consisting of texture images and ground-truth eye-tracking data for these textures, is constructed and used for this evaluation.

The work also contributes an adaptive texture synthesis algorithm based on the regularity of the original texture. The efficacy of the proposed algorithm is measured through the development of a synthesized textures database, named *SynTEX*, containing synthesized textures along with their corresponding subjective visual quality mean opinion scores.

A reduced reference (RR) texture quality metric (TQM) for assessing the perceived quality of synthesized textures, is proposed as part of this work. The proposed RR TQM uses the change in regularity and the change in granularity between the original and synthesized textures to estimate the amount of loss in fidelity. A novel granularity score is also proposed in this work. The RR TQM uses just 2 parameters but outperforms many full-reference and RR metrics over three different databases.

This work also establishes the influence of texture regularity on other texture distortions like compression artifacts and blur.

## 1.4 Outline of the report

This thesis is organized as follows. Background concepts on the representation and analysis of visual textures are presented in Chapter 2. Some of the prior work on texture regularity analysis is also reviewed in Chapter 2. The performance evaluation of the state-of-the-art visual attention models on textures is elucidated in Chapter 3. The proposed no-reference perceptual texture regularity metric is presented in Chapter 4. Simulation results are also presented in Chapter 4. The application of the proposed texture regularity metric for adaptive texture synthesis is described in Chapter 5. A reduced reference quality metric for measuring texture synthesis quality is proposed in Chapter 6. The influence of texture regularity on the perceived texture distortions resulting from compression or blur is discussed in Chapter 7. Finally, a conclusion summarizing the contributions of this thesis and suggesting future research directions is presented in Chapter 8.

## Chapter 2

### BACKGROUND AND EXISTING APPROACHES

This chapter presents the background concepts related to the representation and analysis of textures. Some of the popular existing approaches for quantifying the regularity of textures are also described in this chapter.

#### 2.1 Background

Natural textures like rocky surfaces, sand, barks of trees, grass, leaves etc., originate from various physical phenomena like the withering or erosion of rocks, aging of trees, to name a few. The physical processes that give rise to the observed textures are very difficult to model. Naturally, the most widely used representation of textures is through the pixel intensities. Textures are basically patterns of pixel intensities. The pattern that is spatially repeated throughout the texture in some regular or irregular manner is called a primitive. Texture representation is achieved by describing the primitives and their placement in a condensed form. Some of the most popular approaches for texture representation and analysis are described below.

##### *2.1.1 Statistical Methods*

#### **Gray Level Co-occurrence matrices**

A spatial Gray Level Co-occurrence Matrix (GLCM) estimates image properties related to second-order statistics. The use of GLCM for texture analysis was suggested by Haralick [1]. For an image of size  $N \times M$  with  $G$  gray levels, the GLCM  $P_d$ , for a displacement vector  $d = (d_x, d_y)$  is defined as follows:

$$P_d(i, j) = |\{(s, t), (u, v) : I(s, t) = i; I(u, v) = j\}| \quad (2.1)$$

where

$$(s, t), (u, v) \in N \times M,$$

$$(u, v) = (s + dx, t + dy),$$

$$(i, j) \in G \times G$$

and  $|\cdot|$  is the cardinality of the set.

For an image with  $G$  gray levels, the GLCM is a matrix  $P_d$  of size  $G \times G$ . The  $(i, j)^{th}$  entry of the matrix  $P_d$  is the number of co-occurrences of the pair of gray levels  $i$  and  $j$  which are a distance  $d$  apart. The GLCM is not a symmetric matrix in general. A number of texture features like energy, entropy, contrast and homogeneity can be computed from the GLCM as shown below:

$$\begin{aligned} Energy &= \sum_i \sum_j P_d^2(i, j) \\ Entropy &= - \sum_i \sum_j P_d(i, j) \log(P_d(i, j)) \\ Contrast &= \sum_i \sum_j (i - j)^2 P_d(i, j) \\ Homogeneity &= \sum_i \sum_j \frac{P_d(i, j)}{1 + |i - j|} \end{aligned} \tag{2.2}$$

The advantage of GLCM based approaches is that they are invariant to monotonic gray tone variations [1]. GLCM based approaches for texture analyses assume that the texture is inherently periodic with an unknown periodicity. This requires computing a GLCM matrix for each possible displacement vector  $d$  and hence requires a large memory and a large number of computations. Since the GLCM is completely based on statistics of the pixel intensities without a perceptual factor, sometimes wrong conclusions may be derived about the regularity of the textures. An example of such anomaly is presented in Section 2.3.

### **Autocorrelation and Power Spectral Density**

Autocorrelation measures the self-similarity of a signal. For an image  $I$ , the 2D autocorrelation function for a 2D displacement of  $(x, y)$  is given by:

$$\rho(x, y) = \frac{\sum_u \sum_v I(u, v) I(u + x, v + y)}{\sum_u \sum_v I^2(u, v)} \quad (2.3)$$

In case of textures, the peaks in the 2D autocorrelation function reveal the periodicity of the textural primitives and also the granularity of the texture. If the texture is coarse, then its primitives are large in size. Also in coarse natural textures, there is gradual change in pixel intensities from the primitive to non-primitive regions. As a result, the autocorrelation function will drop off slowly with variations in  $x$  and  $y$  in case of a coarse texture. If the texture is fine, autocorrelation drops off rapidly as a function of displacement. The autocorrelation function of a regular texture is periodic and this period equals the placement between primitives. Kaizer [17] found that there is a high correlation between the displacement  $d = \rho^{-1}(1/e)$  and the subjective coarseness of textures, where  $\rho$  is the autocorrelation function and  $d$  is the distance at which the value of autocorrelation reaches  $(1/e)$ . Further he also observed that a relatively flat background can be interpreted as a fine or a coarse texture. For any smooth gray-tone surface, there exists a scale such that when the surface is examined, it has no texture. It simply appears as a flat region with a constant color and intensity. Then as resolution increases, it takes on a fine texture and then a coarse texture. Also, fine textures are perceived to be less regular (more random) compared to coarse textures [1].

The Fourier Transform of the autocorrelation function gives the Power Spectral Density (PSD). Some approaches like [18] compute the regularity of a texture by computing the Fourier Transform and analyzing the PSD. When the power spectrum is expressed in polar co-ordinates, directional textures have peaks in the phase spectrum, while blob like textures have a peak in the radial spectrum.

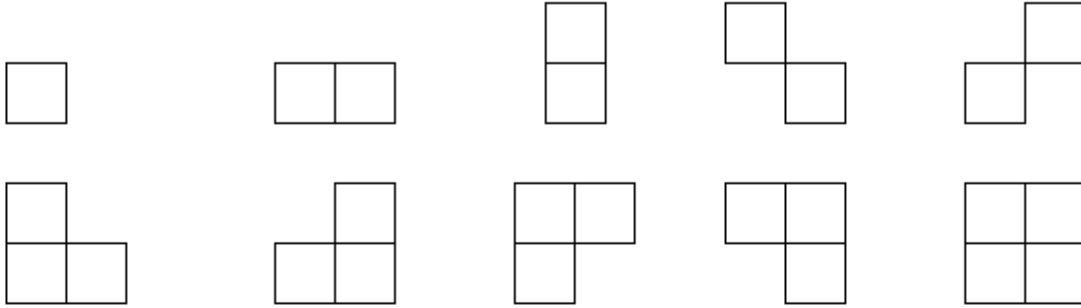


Figure 2.1: Clique-types for a second-order neighborhood system.

### 2.1.2 Model-based methods

#### Markov Random Fields

A discrete Markov random field is a random field whose probability mass function has the properties of positivity, Markovianity, and homogeneity [19]. Textures can be modeled as a MRF as they exhibit the following properties:

1. Locality: The probability of a pixel taking a particular value is dependent only on the pixel values of its immediate neighbors and is independent of other pixel values.
2. Stationarity: The dependency of a pixel value on its neighbors expressed through the conditional probability density, is independent of the location of the pixel.

A texture can be considered to be a lattice  $S$  and each pixel location in the texture corresponds to a site in the lattice. The pixel intensity of each site in the lattice can be denoted by the random variable  $X$ . The lattice can also be viewed as an undirected graph in which every pixel corresponds to a node and an edge is present between a pair of nodes in the graph if the corresponding pixels are adjacent to each other in the texture. A clique is a subset of vertices of this graph in which every pair of vertices are connected by an edge. A neighborhood system  $N_{sys}$  determines the set of all possible clique types. For example the clique types for a second order neighborhood system is shown in Fig. 2.1.

Every clique  $C$  is associated with an energy, that indicates the joint probability of the pixels in the clique possessing a particular set of pixel intensities. This energy is called the clique potential and denoted by  $V_C$ . The higher the joint probability of pixels in the clique, the lower is the clique potential.

As seen earlier, the locality of textures ensures that the probability of a pixel intensity  $X$  assuming a particular value is completely determined by the conditional probability density of the pixel, given its neighboring pixels. The conditional probability density of pixel intensities can also be specified by the clique potentials. The Hammersley-Clifford theorem [20] states that for every MRF, there is a unique Gibbs Random Field. The set of random variables  $X$  is said to be a Gibbs Random Field (GRF) on the lattice  $S$  with respect to a system of neighborhoods  $N_{sys}$  if and only if its configurations obey a Gibbs distribution, whose probability mass function is given as follows:

$$P(X = x) = \frac{1}{Z} e^{-U(x)} \quad (2.4)$$

where  $Z$  is the normalization constant called the partition function and  $U(x)$  is the energy function, expressed through the clique potentials as follows:

$$U(x) = \sum_{c \in Q} V_c(X) \quad (2.5)$$

where  $Q$  is a set of all possible cliques for the considered neighborhood system  $N_{sys}$ . By parameterizing the clique potentials, representation of the entire texture is achieved. Some of the possible parameterizations are proposed in [20] and [21]. The model parameters can be used in texture synthesis [22]. As mentioned in [22], MRF models have been found effective for synthesizing micro-textures but do not result in a good visual quality when synthesizing coarse and inhomogeneous textures.

## Simultaneous Auto-Regressive Models

The non-deterministic nature of textures is best analyzed and characterized by Simultaneous Auto-Regressive (SAR) models as in [23] and [24]. Let  $g(s)$  be the gray level value of a pixel at site  $s = (s_1, s_2)$  in an  $M \times M$  textured image. The SAR model can be expressed as follows:

$$g(s) = \mu + \sum_{i \in D} \theta_i g(s + i) + \epsilon(s) \quad (2.6)$$

where  $D$  is the set of neighbors of the pixel at site  $s$ . A second-order neighborhood is chosen. In (2.6),  $\epsilon(s)$  is an independent Gaussian random variable with zero mean and variance  $\sigma^2$ ;  $\theta(i), i \in (1, \dots, D)$  are the model parameters characterizing the dependence of a pixel to its neighbors, and  $\mu$  is the bias which is dependent on the mean gray value of the image. The standard deviation,  $\sigma$ , is directly proportional to the visually perceived granularity of the texture. These model parameters, estimated using least squares error (LSE) technique or the maximum likelihood estimation (MLE) method, are often used as features for texture classification and segmentation.

The work by Mao et al. [25] improves the SAR model by first suggesting a rotation invariant SAR (RISAR) model. A pixel located at position  $s$  with an original gray level of  $g(s)$  is mapped to a new rotation invariant space  $X$ . Some of the sample points in the new circular grid fall between the pixel locations of the original rectangular grid. As a result, they are estimated by interpolating the available neighboring pixels as follows:

$$x_i(s) = \frac{1}{8i} \sum_{r \in N_i} w_i(r) g(s + r) \quad (2.7)$$

where  $N_i$  is the neighborhood pixel locations of the  $i^{th}$  circle around  $s$  and  $i \in (1, \dots, p)$



$W_i(r)$  indicates the weight or the contribution of  $r^{th}$  neighboring pixel to the  $i^{th}$  parameter. The weights are picked up from a constant table based on the radius of the  $i^{th}$  circle. The weight of a neighborhood pixel is directly proportional to the nearness of the pixel to the interpolated location.

The RISAR model can be obtained by re-arranging the terms in (2.7) as follows:

$$g(s) = \mu + \sum_{i=1}^p \theta_i x_i(s) + \epsilon(s) \quad (2.8)$$

The parameters  $\theta_1 \dots \theta_p$  along with  $\mu$  need to be estimated at each scale.

There are 2 problems with the RISAR model:

1. Since the textures can be at many different scales, the extent of spatial dependence of textures also differs. As a result, a single neighborhood size cannot give an effective representation for all textures.
2. The window over which SAR parameters are estimated must contain homogeneous texture regions. This window size cannot be the same for all textures.

The Multi-Resolution SAR (MRSAR) model described below avoids both the problems of the RISAR model. A Gaussian pyramid of  $L$  levels is constructed by recursively low-pass filtering and subsampling the original image to generate the image sequence,  $G_l, l = 0, 1, \dots, L - 1$ .  $G_0$  represents the original image. The neighborhood size at each level is maintained constant. However, a neighborhood of a considered size in a higher scale covers a larger region compared to the neighborhood of the same size at a lower scale. Hence the MRSAR parameters effectively model the textures belonging to multiple scales.

At each level of the pyramid, the RISAR model parameters are computed. At the lowest scale (highest resolution), the RISAR model parameters capture the properties of textural primitives while at the higher scales, the model parameters

capture the spatial dependency of the primitives. This representation can be used for texture segmentation into a set of homogenous regions as described in [25]. For every pixel, using a neighborhood size of  $25 \times 25$ , at any specific level  $l$  of the Gaussian pyramid there are 5 RISAR model parameters excluding  $\mu$ . Considering all  $L$  levels, each pixel has a total of  $5L$  MRSAR model parameters. Using these  $5L$  elements in the feature vector of each pixel, texture segmentation can be achieved by K-means clustering in the feature space.

### Fractals

A fractal can be defined as a fragmented geometric shape that can be subdivided into parts, each of which is approximately a reduced-size copy of the whole [26]. Fractals are generally self-similar and independent of scale. A fractal dimension is a ratio providing a statistical index of complexity that signifies how detail in a pattern changes with scale. The Euclidean dimension of an object is the dimension of the object in a Euclidean space. The Euclidean space is the real line in one dimension, the Cartesian plane in two dimensions and a coordinate space with three or more real number coordinates in higher dimensions.

The most common fractal dimension is the Hausdorffs dimension [26]. Consider an object that possesses a Euclidean dimension of  $E$ . The Hausdorffs fractal dimension  $D$  can be computed by the following expression:

$$D = \lim_{\epsilon \rightarrow 0} \frac{N(\epsilon)}{\log(\epsilon^{-1})} \quad (2.9)$$

where  $N(\epsilon)$  is the number of hyper-cubes of dimension  $E$  and length  $\epsilon$  that cover the object.

Mandelbrot [27] has pointed out that natural patterns are characterized by similar structures at different scales and textures also possess this property. Textures can be interpreted as fractals because they also exhibit a lot of self-similarity due

to the repetitive primitive patterns. The fractal dimension gives a measure of the roughness of a surface. Intuitively, the larger the fractal dimension, the rougher the texture is. Pentland [28] introduced a fractal-based image description and modeled an image by a fractal set. Since then, a variety of fractal-dimension counting methods have been proposed, such as box dimension [29] and blanket dimension [30], fractal Brownian motion (FBM)-related dimension [31]. Because visually different textures may have indistinguishable fractal dimensions, Peleg et al. [30] improved fractal dimension to fractal signature. Fractal signature is a set of fractal dimensions at different scales. Compared with fractal dimension, the fractal signature provides a more discriminative texture characterization.

### *2.1.3 Signal Processing Methods*

The Human Visual System (HVS) perceives the characteristics of texture images through a set of frequency and orientation selective filters. A similar approach is taken for texture analysis in spatial or frequency domain-based signal processing methods. The filter responses act as feature descriptors and the feature descriptors are further used in tasks like texture classification and segmentation.

#### **Spatial domain methods**

Textures are considered high frequency 2D signals with a lot of edges. The edginess or the number of edges per unit area can readily characterize textures. For example, fine textures have more edginess than coarse textures and this can aid in texture classification and segmentation. The edges in an image can be found by convolving the image with a set of orthogonal edge masks like the Sobel operator as shown below.

$$H_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad H_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2.10)$$

The orthogonal masks  $H_x$  and  $H_y$  extract the gradients along the horizontal and vertical directions, respectively. These masks are mutually orthogonal because the dot product of any column vector from  $H_x$  and any column vector from  $H_y$  is zero. The magnitude and direction of the gradient are computed from the gradient components at each pixel location. The edge pixels are located by thresholding the gradient magnitude. The number of edges in a neighborhood region around a pixel acts as a texture descriptor for the considered pixel.

Pre-attentive visual processing is the process through which the HVS inspects a large portion of a visual field in a very small time and achieves an early understanding of the scene. In all the image data collected by the HVS during this phase, only the information having a significant saliency is selected for further analysis by conscious (attentive) processing. As mentioned in [2], the pre-attentive features of a texture are those the HVS extracts during the first 50ms of looking at the texture. During this phase, the texture is segmented into "Textons" or elongated blobs. The properties of the blobs like color, orientation, size etc., are considered to be the pre-attentive features. Both pre-attentive and attentive features are used by the HVS for texture discrimination.

A model for pre-attentive texture perception in the HVS based on spatial filtering was proposed by Malik and Perona [32] for the segmentation of an image into homogeneous textural regions. Their proposed model consists of three stages: (i) convolution of the image with a bank of even-symmetric filters followed by half-wave rectification, (ii) inhibition of spurious responses in a localized area, and (iii)

detection of the boundaries between the different textures. The even-symmetric filters they used consist of differences of offset Gaussian (DOOG) functions. The half-wave rectification and inhibition (implemented as leaders-take-all strategy) are methods of introducing a non-linearity into the computation of texture features. A non-linearity is needed in order to discriminate texture pairs with identical mean brightness and identical second-order statistics. The texture boundary detection is done by a straightforward edge detection method applied to the feature images obtained from the inhibition stage (ii). This method works on a variety of texture examples and is able to discriminate natural as well as synthetic textures. Unser and Eden [33] have also looked at texture features that are obtained from spatial filters and a nonlinear operator. Reed and Wechsler [34] review a number of spatial/frequency domain filter techniques for segmenting textured images.

### Frequency domain methods

The frequency content of textures can be analyzed by computing the 2D Fourier Transform. The Fourier Transform only gives the global frequency content of the textures without specifying the spatial locations at which those global frequencies occur. Many applications require frequency localization in the spatial domain. A windowed Fourier Transform, centered on a particular pixel  $(\tau_x, \tau_y)$ , can get the spatial frequencies in the local neighborhood of the considered pixel as follows:

$$F_{\tau}(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) w(x - \tau_x, y - \tau_y) e^{-j2\pi ux} e^{-j2\pi vy} dx dy \quad (2.11)$$

where  $w$  is the window function.

When the windowed function is a Gaussian, the above transform becomes the Gabor transform. The proposal to use the Gabor filters in texture analysis was made by Turner [35] and Clark et al. [36]. A two-dimensional Gabor function consists of a sinusoidal plane wave of a certain frequency and orientation modulated by a Gaussian

envelope. It is given by the following:

$$g(x, y) = e^{-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right)} \cos(2\pi u_0 x + \theta) \quad (2.12)$$

where  $u_0$  and  $\theta$  are the frequency and phase of the sinusoidal wave. The values  $\sigma_x$  and  $\sigma_y$  are the sizes of the Gaussian envelope in the  $x$  and  $y$  directions, respectively.

### Wavelet Transform

The Short-Time (or space) Fourier Transform (STFT) despite giving good frequency localization, suffers from the problem of constant window size. Consider the STFT,  $F(u)$  of a temporal signal  $f(t)$ . Let  $\Delta t$  and  $\Delta u$  be the window size in the temporal and frequency domains. The limits on the resolution in the time and frequency domain of the window Fourier transform are determined by the time-bandwidth product or the Heisenberg uncertainty inequality given by:

$$\Delta t \Delta u \geq \frac{1}{4\pi} \quad (2.13)$$

Once a window is chosen for the window Fourier transform the time-frequency resolution is fixed over the entire time-frequency plane. To overcome the resolution limitation of the window Fourier transform, one lets the  $\Delta t$  and  $\Delta u$  vary in the time-frequency domain. Intuitively, the time resolution must increase as the central frequency,  $f_c$ , of the analyzing filter is increased. That is,  $\Delta t$  must decrease and as per the time-bandwidth constraint  $\Delta u$  must increase with the central frequency. As a result, the relative bandwidth, defined as  $\frac{\Delta u}{f_c}$ , must be kept constant in a logarithmic scale. This is accomplished by using the Wavelet Transform. It intrinsically introduces a scale factor  $s$ , which controls both the scale and frequency selectivity simultaneously as shown below:

$$F(\tau, s) = \frac{1}{\sqrt{(s)}} \int_{-\infty}^{\infty} x(t) \psi^* \left( \frac{t - \tau}{s} \right) dt \quad (2.14)$$

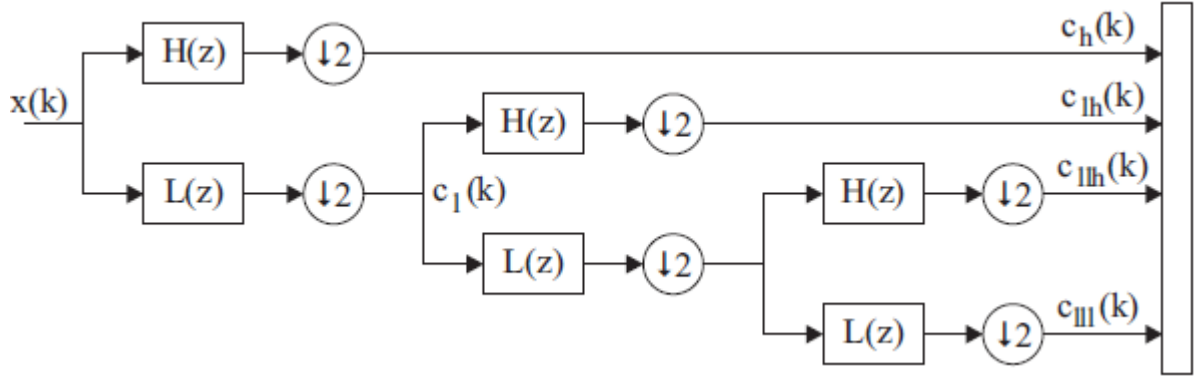


Figure 2.2: A three-level analysis filter bank for implementing the DWT.  $L(z)$  and  $H(z)$  denote the Lowpass and the Highpass analysis filters, respectively.

where  $\psi(t)$  is the mother wavelet function. As the scale factor increases the temporal window is dilated and this also selects low frequency components. A larger temporal window helps in the analysis of low frequencies. On the contrary, as the scale factor decreases the temporal window is small and localized. A small temporal window is suitable for analyzing high frequencies. Thus the wavelet transform achieves high frequency resolution for low frequencies and high temporal resolution for high frequencies in the signal.

The Discrete Wavelet Transform is implemented through a set of multi-resolution filter banks. A three-level analysis filter bank is shown in Fig. 2.2. At each level there is a highpass ( $H(z)$  in Fig. 2.2) and a lowpass filter ( $L(z)$  in Fig. 2.2) that extract the detail and approximation components of the signal input at that level. The high frequency component is down-sampled by a factor of 2 and appended to the output. The low frequency component is also down-sampled by a factor of 2 and becomes the input for the next level. The spatial resolution decreases by a factor of 2 with each successive level as is the case with a Dyadic Wavelet with doubling scale factors. The highpass outputs at each level (except the first level) are actually the band-pass outputs of the original signal.

#### 2.1.4 Geometric Methods

The geometric methods visualize texture as a collection of smaller regions called texture elements. The texture elements are also referred to as primitives, as defined above. Some of the popular geometry based methods are described below.

##### **Voronoi Diagram based methods**

These methods achieve the segmentation of a textural image into smaller homogenous regions through a process called Voronoi Tessellation [37]. They are composed of the following steps:

1. An image filter like the Laplacian of the Gaussian (LoG) is applied to transform the texture into feature space.
2. The local maxima in the feature space are located. This is represented by a binary mask in which a 1 indicates a location where the magnitude of the feature vector is a local maximum in its 6-point or 8-point local neighborhood.
3. Connected components are formed by doing an 8-point connected component analysis on the binary mask. Each connected component represents a texture token.
4. A Voronoi Tessellation is performed using the locations of these texture tokens, to generate a set of open or closed Voronoi polygons or cells.
5. Features of each Voronoi cell are extracted and tokens with similar features are grouped to construct uniform texture regions. Moments of area of the Voronoi polygons serve as a useful set of features that reflect both the spatial distribution and shapes of the tokens in the textured image.

The texture features based on Voronoi polygons have been used for segmentation of textured images. The segmentation algorithm [37] merges the adjacent Voronoi polygons into connected regions based on the similarity of their texture tokens.



## Structural Methods

Structural methods consider a textural image to be a collage of textural elements (Primitives) which are arranged according to a definitive or a random placement rule. The primitives can be arbitrary shaped regions and are locally stationary. Only when there are a significant number of primitives in a textural image, the image is perceived as a texture. Otherwise, the primitives are perceived as individual objects.

As mentioned in [1], a texture can be characterized through the statistical properties of pixels of a primitive and the spatial dependence of primitives with respect to each other. The granularity or the size of the textural primitives plays a role in determining the regularity of the texture. Harlick [1] defines fine and coarse textures and differentiates their properties as follows:

- Fine Texture: When the spatial pattern of tonal primitives is random and the gray tone variation between primitives is large
- Coarse Texture: When the spatial pattern of the primitives is regular and the tonal regions involve more pixels.

As mentioned in Chapter 1, the regularity of textures can be quantified through properties like size, shape, color and orientation. Based on these visual properties of the primitives along with their placement regularity, textures can be classified into 3 broad classes as follows:

1. Regular Textures: The class of textures for which the primitives are well-defined, significantly coarse in size and exhibit a good degree of regularity with other primitives in properties like size, shape, color and orientation, constitute the class of regular textures.

2. Irregular Textures: The textures in which the primitives are either hard to define or extremely small in size constitutes the class of irregular textures. The

textures in which the primitives exhibit a lot of variations in visual properties like size, shape, color and orientation, can also be classified as irregular textures. In this class of textures, the irregularity is either very high in a single property or a moderate degree of irregularity is observed in several properties.

3. Hybrid Textures: Hybrid textures have an intermediate degree of regularity between Regular and Irregular texture classes. The primitives resemble each other in general but exhibit some degree of variation in one of the visual properties like placement, size, shape, color or orientation.

This work takes a structural approach to the analysis of texture regularity.

## 2.2 Existing Approaches for Texture Regularity Measurement

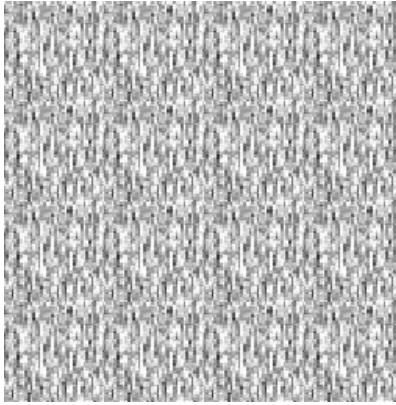
Many approaches have been proposed in the past to quantify the regularity or randomness of textures.

A measure of spatial periodicity for regular textures, derived from the Gray Level Co-occurrence Matrix (GLCM) is proposed in [1]. A faster version of this approach that acts on a Binary Co-occurrence Matrix (BCM) is proposed in [38].

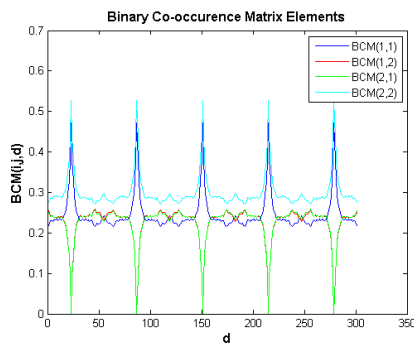
A 2D Wold decomposition of homogeneous random fields is employed in [39] to extract the periodic, random and directional components of textures.

In [40], the spatial regularities in the pixel intensities and the placement of the primitives along a direction, together quantify the directional regularity. The overall regularity metric is taken as the maximum of the directional regularities.

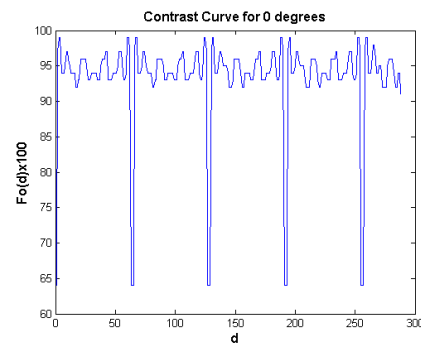
In [41], the randomness of the observed texture is represented by a Kolmogorov Stochasticity parameter measured between the empirical and a modeled distribution of wavelet packet coefficients.



(a)



(b)



(c)

Figure 2.3: Limitations of pixel domain methods to predict perceived texture regularity: (a) An irregularly perceived texture - greynoise; (b) Binary Gray Level Co-occurrence matrix (BCM) elements as a function of the horizontal displacement for the texture in (a); (c) Expected value of the Extended Gray Level Difference Histogram (EGLDH) along  $\alpha = 0^\circ$  as a function of displacement for the texture in (a).

### 2.3 Limitations of Existing Approaches

None of the above mentioned approaches take human perception into account but directly operate on the pixel domain [38] or on the spectral domains [42]. Some of the approaches like [11], assume that they act on a patterned regular texture and are thus not suitable for stochastic textures.

The inability of sheer pixel domain approaches, like those based on GLCM to predict the perceived regularity of textures is exemplified through a *greynoise* texture shown in Fig. 2.3(a). The texture is deliberately constructed by taking a noisy patch of size  $64 \times 64$  and repeating it 6 times along X and Y directions. Due to the local randomness in the primitive placements coupled with the high granularity of primitives, the image is expected to be perceived as an irregular or a hybrid texture. The values of the 4 elements of the BCM [38] exhibit significant periodicity for displacements along both the X and Y directions. The variation of the elements along the X-direction is shown in Fig. 2.3(b) and indicates that the texture has a periodicity of 64 pixels along the horizontal direction. It can be shown through a similar plot along the Y axis that the texture also exhibits a spatial periodicity of 64 along the vertical direction. These results suggest that the *greynoise* texture is perceived as a periodic regular texture, which is not true. The approach in [40] computes a directional regularity from the periodicity in the minima of the expectation of the EGLDH. Even though the *greynoise* texture has a very low perceived regularity, the EGLDH based contrast curve [40] exhibits a significant periodicity as shown in Fig. 2.3(c). The Pattern Regularity metric based on EGLDH has a very high score of 0.984 for the *greynoise* texture, suggesting a high degree of regularity. An improved Pattern Regularity metric based on autocorrelation suggested in [43], gives a score of 0.867 for this texture, which also indicates a high degree of regularity. But as seen in Fig. 2.3(a), the *greynoise* texture does not have a significant degree of perceived regularity. The Mean Opinion Score for this texture is 0.175 in a normalized range of (0,1) where, 1 signifies maximal regularity. The proposed regularity metric turns out to be 0.365 for the *greynoise* texture and this correlates better with the perceptual score.

Perceptual quality metrics for textures proposed in the past like STSIM [44], are full reference metrics used to assess the similarity between a texture and a reference, and cannot quantify the structure in a newly observed texture without the presence of a reference. Furthermore, they cannot assess the degree of regularity in a texture image. A rarity-based VA model for texture description is suggested in [45]. The method classifies the regions of an image into regular and irregular textures by considering irregular texture regions to be highly salient relative to regular texture regions. However, this assumption does not generally hold since regular textures can also be highly salient (Table 4.1). Further, it is also shown through performance evaluation that the VA model suggested in [45] has very low correlation with the ground-truth saliency maps obtained from eye-tracking.

## VISUAL ATTENTION MODELS ON TEXTURES

The Human Visual System (HVS) has variant spatial resolution, with high resolution sampling only in the center. This requires that different regions of a scene are inspected sequentially. Our gaze locations move from region to region via a series of fast eye movements (saccades), interspersed with periods of relatively static eye positions (fixations) during which visual information is acquired. When viewing a visual scene, the HVS fixates on salient points in that scene. This visual saliency also called Visual Attention (VA), can be captured through a Visual Saliency Map (VSM) whose values quantify the extent to which each region grabs the human attention. The saliency map is normalized to 1 and shown as an image in which the brightest pixels (close to 1) correspond to highest attention and the darkest pixels (close to 0) correspond to lowest attention. There are a number of earlier works like [46], that evaluate the performance of various VA Models in predicting the true visual saliency on natural images. These images are generally a mixture of natural scenes, man-made objects and sometimes human faces. But images with exclusive textural patterns have image characteristics that differ a lot from the natural images. For example, in a highly regular texture, primitives with identical image characteristics are uniformly distributed throughout the image. This kind of a pattern does not typically occur solely by itself in natural images. Hence, there is an inherent need to re-evaluate the performance of the popular VA models on textures and employ the best VA model for computing the proposed texture regularity metric. In the following sections, the process of constructing an eye tracking database for textures and the performance evaluation of the VA models for textures are described in detail.

Table 3.1: Eye-tracker set up details.

Eye Tracker Sample Rate	40 Hz
Tracking Method	Binocular tracking
Angular Resolution	0.5 degrees
Display Size	15 × 11.25 inches
Display Resolution	1280 × 960
Position of the chin-rest from screen	33 inches
Angle subtended by screen	32 degrees
Pixels per degree	50

### 3.1 Eye Tracking Database

A set of 21 textures representing a wide range of regularity were chosen as the data set. The original size of the textures varied from  $256 \times 256$  to  $800 \times 600$ . These textures had primitives that differed in the degrees of placement, color, orientation and shape regularities. Also, they represented a wide range in the granularity of the primitives. The textures were displayed for 8 seconds one after the other on a 60 Hertz LCD display. A chin rest was placed at a distance of 33 inches in front of the display. An EyeTech TM3 eye tracker was used to track the gaze positions of the subject looking at a displayed texture.

The center of the display, chin rest and the camera of the eye-tracker were precisely aligned in a straight line, along the normal to the plane of the display. The details of the eye tracker set up are enclosed in Table 3.1. A test pattern as shown in Fig. 3.1(a) was used to validate the accuracy of the system. A subject was first asked to focus on the small red-circle for 8 seconds and from the eye-tracking data collected during this period, fixation points were computed and overlaid on the image as shown in Fig. 3.1(b). This result establishes the validity of both the eye-tracking set up as well as the software for fixation computation (explained below). In addition, as a part of separate eye tracking experiments, the eye tracker accuracy was validated

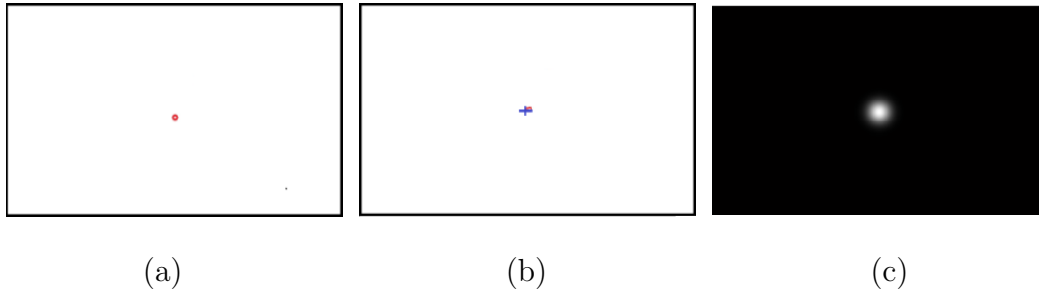


Figure 3.1: Validating the eye-tracker system and the software for computing fixations: (a) Test Pattern (red-dot); (b) Fixation Overlaid (blue-cross); (c) Ground-truth Saliency Map.

by its ability to reproduce the ground-truth eye tracking data of publicly published databases for natural images.

During the calibration phase, the eye-tracker was set in the binocular mode and a 16 point routine was displayed. The subject was asked to focus on each point one after another. The eye tracker tracks the size and location of the subject's pupil and cornea by emanating IR light into the subject's eyes and capturing the relative positions of the four captured IR reflections (so called Purkinje images). During calibration, the eye-tracker learns how to estimate the gaze position from the reflected IR images. After calibration is performed, the computer then validates the calibration information by determining whether the estimation of eye positions is actually close to the known positions of the targets in the 16-point routine. If errors were found to be less than a threshold, calibration is indicated as successful and the subject is allowed to continue with the free-viewing testing on textures. The selective qualification of subjects was done to ensure the alignment and the biological correctness of both eyes before collecting the eye-tracking data. The calibration error threshold was fixed at 2% of the screen resolution in our testing.

For each subject, the data collection lasted for about 20 mins for the 21 textures including breaks and recalibration times. The subjects were asked to take



a break for 2 minutes after viewing every 7 textures. Each time they resumed, they had to do a recalibration which lasted for about 2 minutes. The recorded eye-tracking data of each subject, while free viewing each texture, was analyzed for fixations and saccades using the Dispersion-Duration-Threshold (DDT) algorithm [47] implemented in [48]. The dispersion threshold was fixed at 1.2 degrees and the duration threshold fixed at 200 milliseconds as in [48]. The DDT algorithm computes a set of fixation points for each texture and each subject, based on the eye-tracker’s gaze points and the dispersion and duration thresholds. The fixation points of all the subjects, when free-viewing each of the texture images, were stored for further analysis. This constitutes the eye-tracking database.

The fixation points were stored in a 2D binary VA mask which has the same size as that of the texture image. A 1 in the mask indicates a fixation while a 0 indicates a non-fixation point. For each considered texture  $i$ , the union of the VA masks from various subjects gives the texture’s final ground truth VA mask,  $GTMask_i$ .

### 3.2 Performance Metrics

The ability of the most popular VA models to predict the VA on textures is evaluated in terms of the performance metrics described below. These metrics were also used for evaluation in [46] for natural images.

Receiver Operating Characteristics (ROC) and AUC:

The ROC is used for evaluating the performance of any binary classifier. A predicted saliency map for each texture is obtained from each VA model by applying the corresponding algorithm on the textural image. The predicted saliency map that is generated by a model can be viewed as a binary classifier at any given threshold. In other words, all the pixels having a saliency value higher than the threshold are classified as fixated while those having a saliency below the threshold are classified

as not fixated. A binarized predicted saliency map in which a 1 indicates a fixation and 0 indicates non-fixation, is obtained at a given threshold. Using these predicted fixations and ground-truth fixations, the true and false positive rates (TPR and FPR) are obtained for a particular threshold level. By varying the threshold, a series of TPR and FPR combinations can be obtained for a considered texture. Averaging the TPR and FPR values over all textures at each VA threshold signifies the performance of the VA model. Plotting average TPR versus average FPR generates the ROC and the area under the ROC is the AUC metric.

The 1s in the  $GTMask_i$ , correspond to the salient pixels and constitute the positive set. The  $GTMask_i$  is denoted by  $G \in \{0, 1\}$  for simplicity. The predicted saliency map is binarized at a sweeping threshold (between 0 and 1) to generate a predicted mask,  $P \in \{0, 1\}$ . TPR is computed from the positive sample set of true fixations, as shown below:

$$\text{TPR} = \frac{N(A)}{N(G)} \quad (3.1)$$

where  $A = P \cap G$ , represents the common set of saliency points in the ground truth and predicted saliency masks and  $N(X)$  is the number of 1s in the binary mask  $X$ . The FPR is obtained from the negative set. The negative set can be formed in 3 ways as described below.

(i) Randomized set of locations within the same image:

The negative set can be obtained by considering the visual saliency at a random set of locations within the same image. The number of randomly selected points equals the number of fixations. The values of the binarized predicted saliency map at these random locations form the randomized binary saliency set,  $R \in \{0, 1\}$ . Then the FPR is given by:

$$\text{FPR} = \frac{N(R)}{T(G) - N(G)} \quad (3.2)$$

where  $T(X)$  is the total number of elements in  $X$ .

(ii) Non-fixated locations on the same image:

If we consider the negative set to consist of all locations in the current image at which there are no fixations, then the FPR is computed as follows:

$$\text{FPR} = \frac{N(P) - N(A)}{T(G) - N(G)} \quad (3.3)$$

where  $A$ ,  $P$  and  $G$  are as defined above.

(iii) Fixated locations on the other images in the database (Shuffled AUC):

Some of the models compared here have an inherent center-bias and these models may score better when predicting fixations on a database in which most images have visually attentive regions near the center. To eliminate center-bias, a shuffled AUC metric was suggested by Zhang [16]. For a particular texture and a human subject, the positive set consists of the true fixations of the subject on the texture while the negative set consists of fixations by all subjects on all other images. A random subset, whose cardinality equals the number of elements in the positive set, is chosen from the negative set and the FPR is computed. Plotting TPR versus FPR generates the ROC, the area under which is the shuffled AUC metric for the considered VA model when acting on the considered texture and with respect to the considered subject. Averaging the Shuffled AUC score over all subjects and all textures, gives the overall Shuffled AUC score for the considered VA model.

Normalized Scan-path Saliency (NSS):

A normalized predicted saliency map can be obtained from a model's predicted saliency map by making the map zero mean and unit standard deviation. The true fixations are obtained from the ground-truth data collected while building the eye-tracking database. Each fixation point is assigned an NSS score, which equals the saliency value of the point in the normalized predicted saliency map. The average of

the NSS scores for all the fixations gives the NSS score for the considered texture. For a single texture, NSS=1 indicates that the scan path of the subjects for that texture fell on a region whose predicted saliency values were at least one standard deviation above the average saliency. Averaging the NSS score over all textures gives the NSS score for the considered VA model.

Pearson’s Linear Correlation Coefficient (PLCC):

A 2D correlation coefficient measured between a Predicted Saliency Map ( $PSM$ ) produced by a model and the Ground-truth Saliency Map ( $GSM$ ) quantifies the efficacy of a model’s accuracy in predicting human eye gaze. For a considered texture  $i$ , the  $GSM$  was obtained from the corresponding  $GTMask_i$  by convolving a 2D Gaussian with 2D Dirac-delta functions located at each of the fixations. The width of the 2D Gaussian was equal to the number of pixels in 1 degree of visual angle. The 2D correlation is expressed as follows:

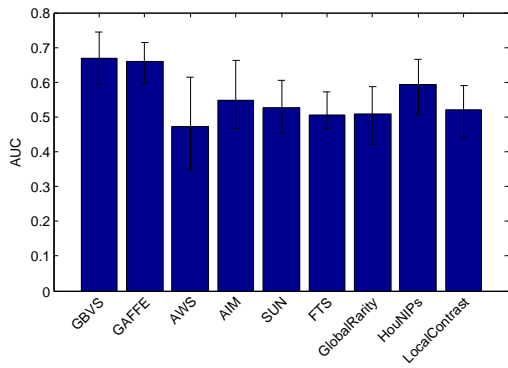
$$\rho_{GSM,PSM} = \frac{cov(GSM, PSM)}{\sigma_{GSM} \cdot \sigma_{PSM}} \quad (3.4)$$

where  $cov(GSM, PSM)$  is the covariance of  $GSM$  and  $PSM$ , and  $\sigma_X$  denotes the standard deviation of the map  $X$ . The value of  $\rho_{GSM,PSM}$  varies from -1 to +1. The higher the value of  $\rho_{GSM,PSM}$ , the better is the VA model.

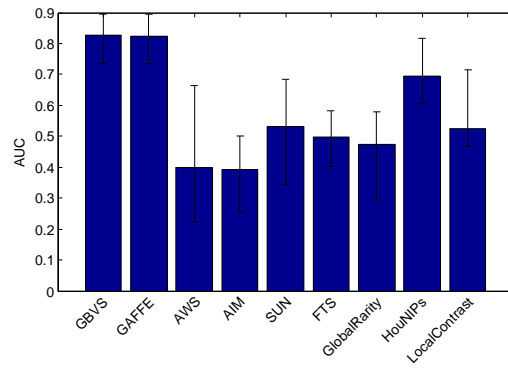
### 3.3 Performance Evaluation Results

In this work, the performance of 9 VA models is compared. These models include GBVS [13], GAFFE [49], AWS [50], AIM [51], SUN [16], FTS [52], GlobalRarity [45] and HouNIPS [53]. Also, the ability to predict visual saliency when using simple local contrast [54] is evaluated for reference.

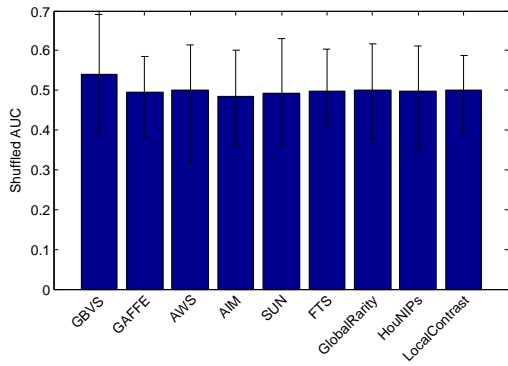
Since GBVS is inherently center-biased, a 2D inverted Gaussian function is multiplied with the activation map to remove the central bias in GBVS while



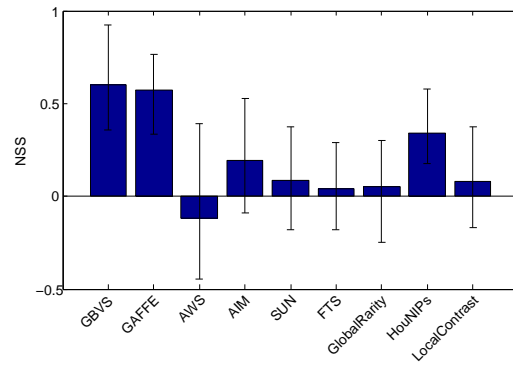
(a)



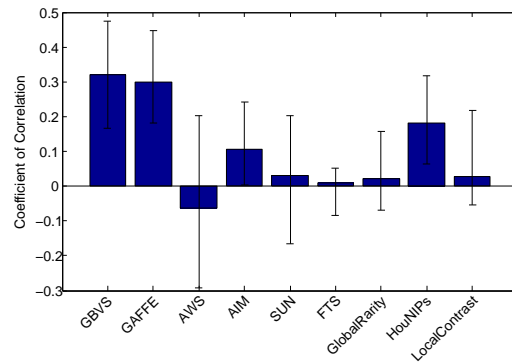
(b)



(c)



(d)



(e)

Figure 3.2: Performance evaluation of VA models: (a) AUC with randomly chosen locations as the negative set (AUC I); (b) AUC with non-fixated locations as the negative set (AUC II); (c) Shuffled AUC with fixated locations on other images as the negative set; (d) Normalized Scan path Saliency (NSS); (e) 2D Correlation between predicted and ground-truth saliency maps.

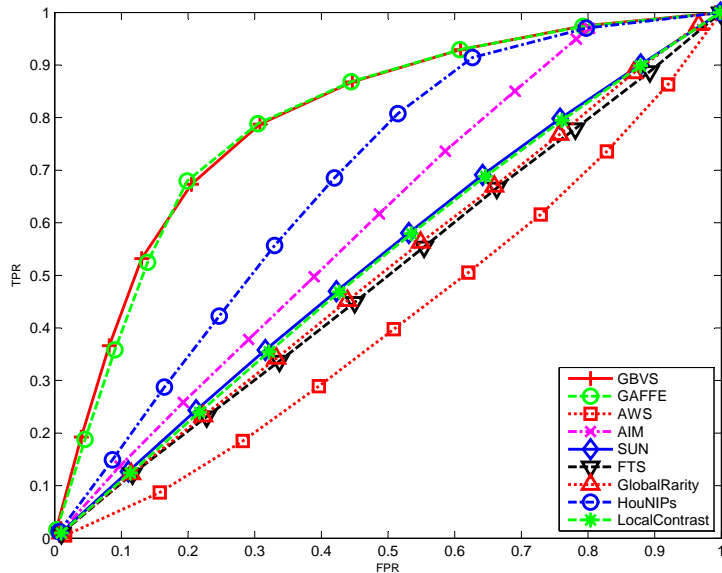


Figure 3.3: ROC curves for different VA models.

generating the predicted saliency map. Figs. 3.2 and 3.3 present the obtained performance evaluation results.

The following can be concluded from our performance analysis:

1. Amongst all the VA models compared in this work, GBVS [13] has the highest overall performance in terms of AUC I, AUC II, shuffled AUC, NSS and Correlation Coefficient, as seen in Fig. 3.2 (a), (b), (c), (d) and (e), respectively. GAFFE [49] is the second best followed by HouNIPS [53].
2. The negative NSS values seen for AWS (Fig. 3.2(c)) are due to border effects. Borders of images have high contrast due to sharp transitions in pixel values. As mentioned in [46], when an image filter lies partially off the edge of an image, the filter response is not well defined. This leads to an increase in saliency at the boundaries and corners of the image, compared to the central regions. So, most of the central pixels have a saliency lower than the mean saliency of the PSM. Hence, the normalized saliency of the central pixels become negative. On removing the peripheral regions (15% from all sides) from both the GSM and PSM before evaluation, both NSS and PLCC become close to zero for AWS.

Table 3.2: P-values for the correlation between the performance metrics of GBVS and other saliency models.

<b>Metric</b>	GAFFE	AWS	AIM	SUN	FTS	Global Rarity	Hou- NIPs	Contrast
AUC I	0.001	0.834	0.056	0.069	0.482	0.891	0.123	0.744
AUC II	0.000	0.492	0.509	0.430	0.807	0.892	0.004	0.593
Shuffled AUC	0.000	0.000	0.004	0.001	0.043	0.046	0.003	0.137
NSS	0.000	0.604	0.103	0.193	0.610	0.957	0.062	0.919
PLCC	0.000	0.895	0.014	0.375	0.268	0.663	0.029	0.479

3. Our oculomotor system biases the fixations towards the center. So there are many VA models that give a higher saliency near the center. But if our database has images with more salient points near the image center, then the models with center-bias would score better than the models without center-bias, irrespective of their true capacity to detect saliency. So in order to compare VA models on fair grounds, it is recommended to remove the central bias as stated in [46]. The shuffled AUC score accounts for central bias and border effects. As shown in Fig. 3.2(a) and Fig. 3.2(b), the shuffled AUC score of AWS (Fig. 3.2(b)) is higher than its corresponding AUC score (Fig. 3.2(a)). However, GBVS, with center inhibition exhibits the highest performance even when using the shuffled AUC metric.
4. The ROC curves shown in Fig. 3.3, indicates that GBVS has the highest AUC. This further supports that GBVS is the best performing VA model on textures.

From our evaluation results, it can be concluded that GBVS predicts the fixations on textures more accurately compared to other models. In order to conclude the statistical significance of these results, a P-value analysis is performed. In statistical significance testing, the null hypothesis is that the two measured quantities under consideration are uncorrelated. The P-value is a number between 0 and 1 representing the probability that this data would have arisen if the null hypothesis were true. The P-values for the correlation between the performance metrics of GBVS

and other saliency models is shown in Table 3.2. If the P-value is greater than 0.05 between a pair of models for a considered metric, then there is statistical difference in the performance of the considered models in predicting the true saliency, with respect to that metric. As seen in Table 3.2, GBVS and GAFFE do not perform statistically different based on the 5 metrics. Also, except for local contrast, all models appear statistically equivalent when compared in terms of Shuffled AUC score. But based on the AUC I & II, NSS and PLCC metrics, GBVS is statistically different (except for GAFFE) and better performing than all the other models. Hence, the visual saliency map (VSM) produced by GBVS is used for computing the proposed texture regularity metric.



## PROPOSED PERCEPTUAL TEXTURE REGULARITY METRIC

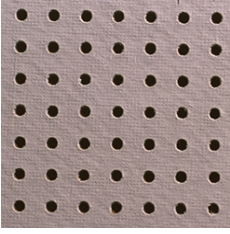

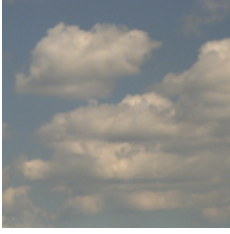
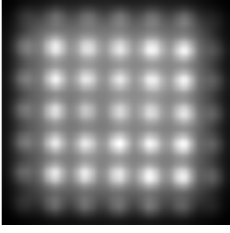
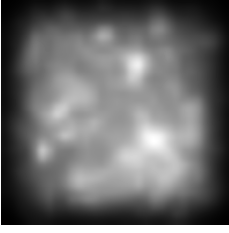
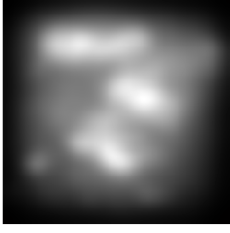
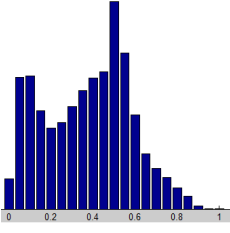
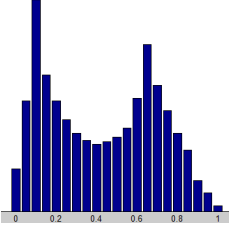
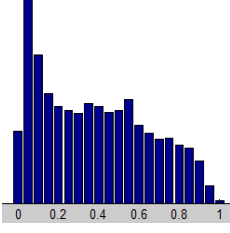
The performance evaluation of the popular visual attention models on textures was described in the previous chapter. A texture regularity metric based on the visual saliency map generated by the most efficient visual attention model, is proposed in this work. This chapter describes the details of the proposed texture regularity metric. Simulation results evaluating the performance of the proposed metric to predict the subjective regularity of textures are also enclosed in this chapter.

## 4.1 Proposed Texture Regularity Metric

The proposed metric makes use of the characteristics and the distribution of the Visual Saliency Map (VSM) in order to assess the degree of perceived regularity in texture images. The motivation for the proposed approach comes from the fact that the VSM characteristics differ between regular, irregular and hybrid textures as illustrated in Table 4.1. Table 4.1 shows sample regular, hybrid and irregular textures (row 1) and their corresponding predicted VSMS (row 2). Table 4.1 also shows the normalized histograms of the VSMS,  $P_{VA}$ , associated with each texture image (row 3). The VSM has the same size as the input image and its value  $VA$  at each pixel corresponds to the probability of attention at that pixel location. For example, a VSM value  $VA = 0.5$  indicates that 50% of the users looked at this location. Thus in a VSM, an attention level  $VA \leq 0.5$ , corresponds to a low attentive area. As shown in Table 4.1 (row 3), the normalized histograms of the VSMS of the three classes of textures differ in their shape, peakedness and the location of the peaks. This fact can be exploited in quantifying the degree of regularity of a texture image.

A flowchart of the proposed texture regularity metric is shown in Fig. 4.1. A VSM is generated by applying the GBVS [13] model on the given texture. A histogram of the visual saliency values is formed from which a Textural Similarity

Table 4.1: Visual Saliency Analysis for regular, hybrid and irregular textures.

	Regular Texture (Tile)	Hybrid Texture (Tulips)	Irregular Texture(Clouds)
Texture			
Saliency Map obtained from GBVS [13]			
Histogram of the Saliency Map			

Score is computed. The fixation points are also determined from the VSM from which a Spatial Distribution Score of the primitives is computed. The proposed texture regularity metric is a combination of these two scores which together influence the perceived regularity. The following sub-sections elucidate their computation.

#### 4.1.1 Textural Similarity Score

In the cases of regular and hybrid textures, where the primitives are easily discernible, the VA predicted by the VA model is largest at the center and decreases from the center to the periphery of each texture primitive (Fig. 4.2). The high VA at the central pixels of the primitives correspond to the tail of the histogram. Furthermore, the neighborhood pixels of the central pixels of the primitives are relatively larger in number and the VA at these neighborhood pixels correspond to the last peak, *lpeak*,

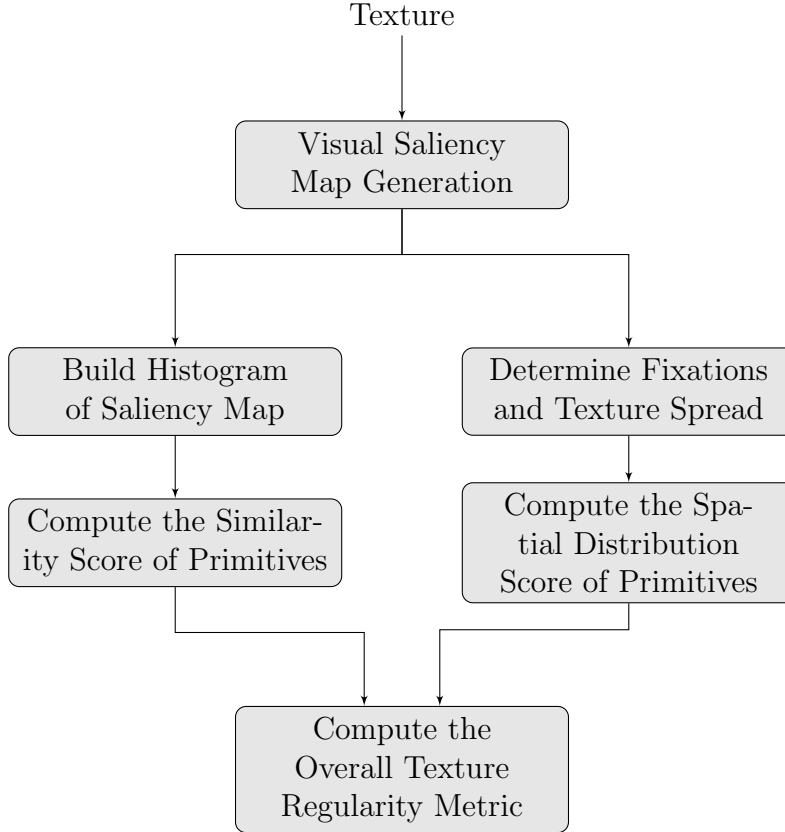


Figure 4.1: Block diagram of the proposed texture regularity metric.

in the histogram. The histograms of the VSMs of regular textures are more peaked near the  $VA_{l_{peak}}$  value of the VSM histogram, compared to hybrid and irregular textures. The histograms of regular textures also exhibit a faster decay beyond the  $VA_{l_{peak}}$  compared to irregular and hybrid textures (Table 4.1, third row). This is illustrated through a regular texture (*tile*) in Fig. 4.2. The saliency map of the regular texture (Fig. 4.2 (a)) is shown in Fig. 4.2 (b) while the correspondence between various texture regions and their VA values in the histogram is shown in Fig. 4.2 (c). When the texture primitives are identical in size, shape and color, pixels of one primitive would have similar VA levels as the corresponding pixels of another primitive. This gives a high value of the histogram at  $VA_{l_{peak}}$ .

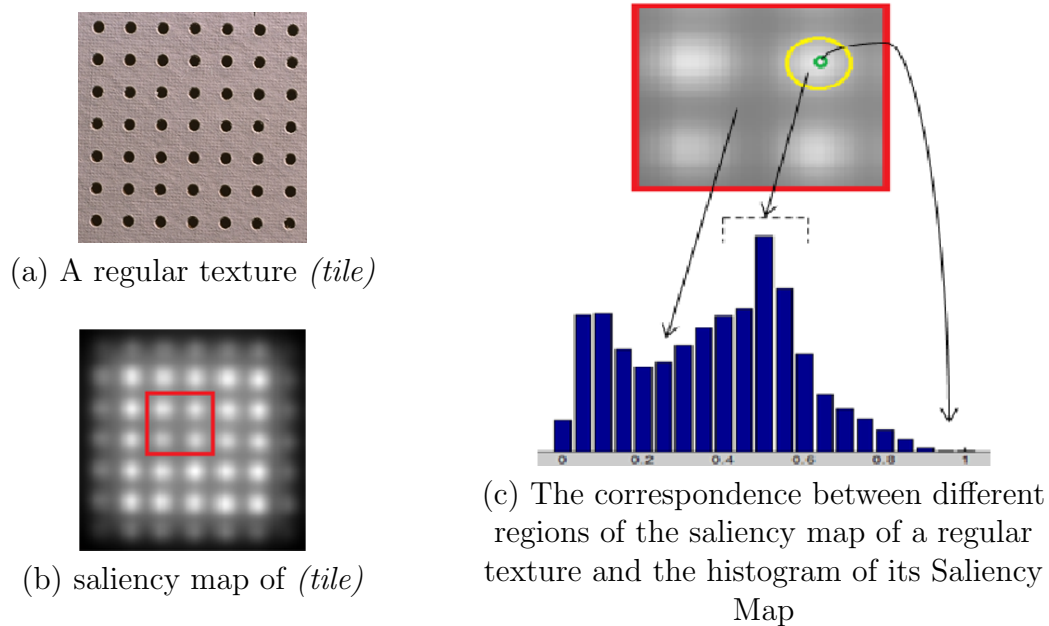


Figure 4.2: An Illustration of relationship between the saliency map and the histogram of the saliency map for a regular texture.

The histogram of a regular texture also shows a faster decay beyond the  $VA_{lpeak}$  because the central primitive pixels corresponding to very high VA values are fewer in number relative to the pixels in their neighborhood. In the case of textures with irregularly shaped, sized or colored primitives, the VA values at the peripheral regions of the primitives are not identical to each other. Hence, the last peak of the histogram is relatively smaller and there is a spread of higher VA values. This leads to a gradual decay of the histogram.

The decay rate  $d$  of the histogram contributes to the quantification of the texture regularity and is found by fitting an exponential function from  $VA = VA_{Thresh}$  to the tail of the histogram as follows:

$$(a, d) = \underset{a, d}{\operatorname{argmin}} \left( \sum_{VA \geq VA_{Thresh}}^1 (P(VA) - a \cdot e^{-d \cdot VA})^2 \right) \quad (4.1)$$

where

$$VA_{Thresh} = \begin{cases} VA_{lpeak}, & \text{if } 0.5 \leq VA_{lpeak} \leq 0.65 \\ 0.6, & \text{otherwise} \end{cases} \quad (4.2)$$

In (4.1), for computing the decay rate, we constrain the  $VA_{lpeak}$  to be greater than 0.5 for the region to be salient. Also in order to account for the decay in the whole salient area in the texture, the  $VA_{lpeak}$  is constrained to be less than 0.65. This higher threshold is fixed at 0.65 because, as mentioned in [55], the threshold of visual detection is often taken as the intensity of the stimulus at which 63% of subjects noticed its presence. In case the  $VA_{lpeak}$  falls outside the  $[0.5,0.65]$  saliency range, the value of the VSM histogram starting at  $VA = 0.6$  is used for computing the decay rate.

A peakedness parameter,  $p$ , that is also used to quantify the textural similarity is computed as follows:

$$p = P(VA_{Thresh}) \quad (4.3)$$

where  $VA_{Thresh}$  is defined in (4.2).

Finally the texture similarity score is computed as the average of the peakedness parameter and decay parameter as follows:

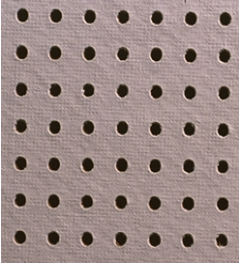

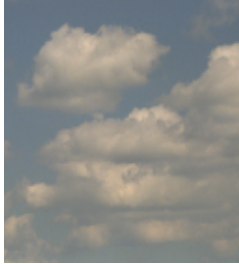
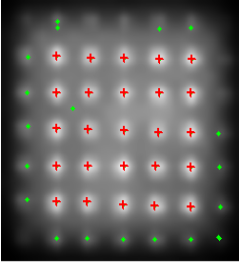
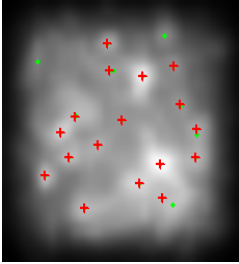
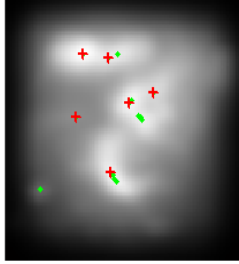
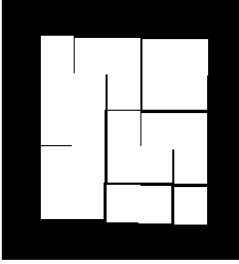
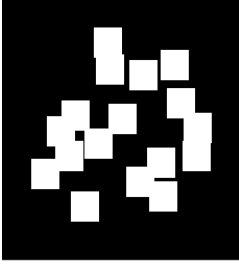
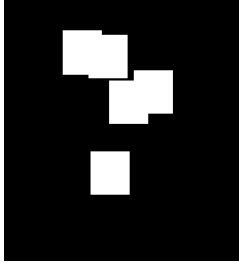
$$S_{Similarity} = \left(\frac{1}{2}\right) \cdot \left(\frac{\min(p, max_p)}{max_p} + \frac{\min(d, max_d)}{max_d}\right) \quad (4.4)$$

where  $max_p$  and  $max_d$  can be calibrated to saturate the peakedness and decay rate, respectively, to their maximum values and to normalize the similarity score to the range  $[0,1]$ . In our implementation,  $max_p = 0.16$  and  $max_d = 12.5$  were used.

#### 4.1.2 Spatial Distribution Score

The above mentioned texture similarity score quantifies the proportion of visually attentive pixels and the similarity between the primitives as measures of regularity.

Table 4.2: Fixation Points and Spread Masks for regular, hybrid and irregular textures.

	Regular Texture (Tile)	Hybrid Texture (Tulips)	Irregular Texture(Clouds)
Texture			
Saliency Map with Predicted Fixation Points (crosses – strongpeaks) (dots – weakpeaks)			
Region of Support for the Texture Spread Mask			

But even when these two measures are high, regularity falls when the primitives do not span the entire image and are not spatially distributed in a periodic or quasi-periodic manner. A spatial distribution score is attributed to the texture for this purpose. To compute the spatial distribution score, the local peaks in the VSM, which act as local maxima simultaneously along both the X and Y directions, are first located. However, some of the local peaks are insignificant in the sense that their VA value is relatively small. We want to retain only the 2D local peaks that

correspond to the center of the primitives and hence local peaks with very low VA values should be discarded.

As mentioned in Section 4.1.1, for center-biased VA models such as GBVS [13], the VA at the 2D local peaks occurring in the peripheral portions of the texture is relatively smaller than the VA at the 2D local peaks that occur in the central parts of the texture. In order to account for attenuated but still significant peaks in the periphery, we qualify the 2D local peaks at which the VA is greater than a significance threshold,  $VA_{sig}$ , as significant peaks and these correspond to the primitive centers. To accommodate for the sensitivity of the VA model to photometric and geometric transformations, a higher and a lower threshold are incorporated in determining  $VA_{sig}$  as follows:

$$VA_{sig} = \begin{cases} VA_{low}, & \text{if } |Peaks_{VA_{low}}| \geq \beta * |Peaks_{VA_{high}}| \\ VA_{high}, & \text{otherwise} \end{cases} \quad (4.5)$$

where  $|Peaks_{VA_{low}}|$  and  $|Peaks_{VA_{high}}|$  denote, respectively, the number of 2D local peaks at which the VSM value  $VA$  exceeds  $VA_{low}$  and  $VA_{high}$ .  $VA_{high} = 0.55$ ,  $VA_{low} = 0.45$  and  $\beta = 2$  were used in our implementation for selecting the visually significant peaks. In row 2 of Table 4.2, the red pluses indicate visually significant large peaks (fixation points) while the green dots indicate insignificant 2D local peaks. In addition to the insignificant peaks, 2D local peaks that fall in a  $8 \times 8$  neighborhood of a significant peak are also eliminated.

A score characterizing the spread of the primitives is computed as follows. The located visually significant VSM peaks,  $p_i$ , act as high saliency fixation points. For each such fixation point  $p_i$ , the distance of its closest neighboring fixation point,

$cdist_i$ , is determined as follows:

$$cdist_i = \min(p_i - p_j), \text{ where } 1 \leq i, j \leq N \text{ and } i \neq j \quad (4.6)$$

where  $N$  is the total number of fixation points.

Let the closest neighbor distances of all the fixation points together constitute the closest neighbor distance set,  $D$ , as follows:

$$D = \{cdist_i, 1 \leq i \leq N\} \quad (4.7)$$

The region of support for the texture spread mask, shown in row 3 of Table 4.2, is obtained by convolving 2D Dirac-delta functions at the fixation points with a rectangular filter. The 2D impulse response of the rectangular filter is approximated by a square block of size  $L \times L$ , where  $L$  is computed as follows:

$$L = mean(D) \quad (4.8)$$

where  $D$  is the closest neighbor distance set.

The region of support of the texture spread mask gives a measure of the spread of the texture primitives. The higher the spread, the higher is the perceived regularity. A texture spread score is thus computed as follows:

$$S_{TextureSpread} = \frac{\text{Spread Mask Area}}{\kappa \cdot \text{Total Image Area}} \quad (4.9)$$

where

$$\text{Spread Mask Area} = \bigcup_{1 \leq i \leq N} r(m - x_i, n - y_i) \quad (4.10)$$

In (4.10),  $(x_i, y_i)$  is the location of the peak  $p_i$  and  $r(m, n)$  is given by:



$$r(m,n) = \begin{cases} 1, & \text{if } -(L-1)/2 \leq m, n \leq (L-1)/2 \\ 0, & \text{otherwise} \end{cases} \quad (4.11)$$

where  $L$  is the width of the convolved rectangular filter.

In (4.9), the term  $\kappa$  is introduced due to the inherent center-bias of the GBVS model [13] which leads to a reduced region of support for the computed spread mask. As it can be noticed in row 2 of Table 4.2, the predicted saliency has low values in the periphery of the textural image. Assuming a peripheral region of 15% of the image dimension on all 4 sides where the visual saliency of GBVS becomes relatively low, the central region is a rectangle whose dimensions are 70% of the height and width of the original texture. In other words the VA model can only account for 49% of the original texture's area and hence the value  $\kappa = 0.5$  was used in our implementation.

In addition to the textural spread, a score characterizing the placement regularity of the primitives is computed as follows. The standard deviation of the closest neighbor distance set,  $std(D)$ , is higher for textures with irregularly placed primitives and lower for regularly placed primitives. Therefore,  $std(D)$  gives a measure of placement irregularity. However, among textures exhibiting the same  $std(D)$ , the perceived irregularity tends to be smaller for textures with large primitives compared to textures with small primitives. In order to better account for the perceived irregularity,  $std(D)$  is normalized by the primitive size. The primitive size is approximated as the mean of the closest neighbor distance set,  $mean(D)$ . The placement regularity score is computed as follows:

$$S_{PlacementRegularity} = 1 - \frac{PlacementIrreg}{MaxPlacementIrreg}, \quad (4.12)$$

where

$$PlacementIrreg = \frac{std(D)}{mean(D)}, \quad (4.13)$$

and  $MaxPlacementIrreg$  is a constant that is used to saturate  $PlacementRegularity$  to a maximum value and to normalize the placement regularity score to the range  $[0,1]$ . In our implementation,  $MaxPlacementIrreg$  is set to 100.

When the number of fixations ( $N$ ) is very small ( $N < N_{Th}$ ),  $std(D)$  is not a dependable measure of placement irregularity. This typically happens in two scenarios, namely, (i) when there are a few large sized primitives in a regular texture in which case the textural spread score is high and (ii) when there are a large number of small random primitives in an irregular texture. In the latter case, the fixations are mostly in the center, closer to each other and smaller in number, resulting in a very small texture spread score. Consequently, in these cases, the  $PlacementRegularity$  score can be estimated using solely the  $TextureSpread$  score. The  $TextureSpread$  is highest for regular textures having primitives distributed throughout the texture and lowest for irregular textures exhibiting very small number of fixations. Therefore, when the number of fixations is very small, the placement regularity score can be computed by quantizing the  $S_{TextureSpread}$  score into 3 levels (low, medium, high) as follows:

$$S_{PlacementRegularity} = \Delta \cdot \left\lfloor \frac{|S_{TextureSpread}|}{\Delta} + \frac{1}{2} \right\rfloor \quad (4.14)$$

where  $\Delta = 1/3$  is the quantization step size.

When the  $S_{TextureSpread}$  is quantized as shown in equation (4.14) with  $\Delta = 1/3$ , the  $[0,1]$  range of the texture spread score is divided into three equal non-overlapping bins and a score corresponding to the bin center is approximated as the placement regularity score.

The spatial distribution score is then computed as follows:

$$S_{SpatialDistribution} = S_{TextureSpread} \cdot S_{PlacementRegularity} \quad (4.15)$$

The overall texture regularity measure is obtained as follows:

$$S_{Regularity} = S_{Similarity}^{\alpha_1} \cdot S_{SpatialDistribution}^{\alpha_2} \quad (4.16)$$

In (4.16), the values  $\alpha_1 = 1$  and  $\alpha_2 = 0.50$ , were empirically found to be the best set of parameters for obtaining a high degree of correlation with the subjective regularity scores.

## 4.2 Simualtion Results

Subjective testing was conducted on 21 textures shown in Table. 4.3. These textures were obtained from two databases, namely, the MIT Vistex database [56] and the Graph-Cut texture synthesis database [57]. The textures were chosen such that they distinctly represented one or more of the primitive irregularities as mentioned in Fig. 1.1. Also, to understand the difference in the perceived regularity between fine granular and large sized primitives, textures from both these classes were chosen in the test set. Ten subjects with normal to corrected-normal vision participated in the subjective tests. The textures, equally distributed amongst the broad classes of regular, irregular and hybrid textures, were randomly displayed one after another to each subject. The subjects were asked to score the overall regularity for each observed texture using a three-scale score with 1 corresponding to lowest and 3 to highest. In addition to observing the overall texture regularity, subjects were asked to observe and score using a three-scale score (1 being the lowest and 3 the highest) five visual properties of the texture primitives, namely (i) ease in locating the primitive, (ii) regularity in the placement of the primitives, (iii) regularity in size, shape and color of the primitives, (iv) regularity in the direction of the primitives, and (v) average size of the primitives. The subjects also gave a final overall regularity score for each displayed texture image.

Table 4.3: Textures used for evaluating the performance of the proposed metric.

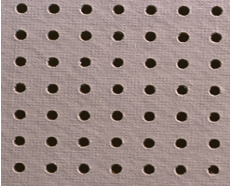


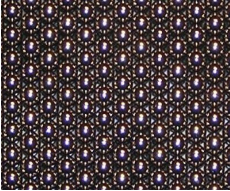
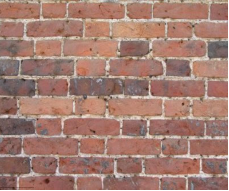




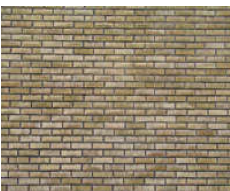






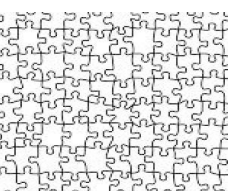
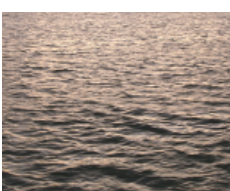
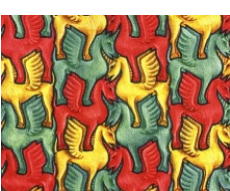

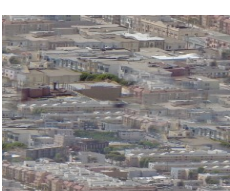
Regular Textures	Hybrid Textures	Irregular Textures
		
		
		
		
		
		
		

Table 4.4: Mean Opinion Score (MOS) and the proposed metric for textures. The textures are listed in the order of decreasing perceived regularity according to the MOS Overall Regularity.

<b>Input</b>	<b>MOS Overall Regularity</b>	<b>MOS Avg Property</b>	<b>Pattern Regularity Metric [43]</b>	<b>Proposed Texture Regularity Metric</b>
marbles	3.00	2.81	0.80	0.74
tile	3.00	2.83	0.76	0.67
gecko	2.88	2.68	0.56	0.42
blocks	2.88	2.43	0.68	0.22
keyboard	2.81	2.79	0.23	0.44
fabric	2.81	2.71	0.12	0.44
horses	2.69	2.64	0.30	0.31
puzzlepieces	2.69	2.53	0.10	0.30
bricks	2.63	2.43	0.25	0.26
red-peppers	2.31	2.18	0.22	0.37
tomatoes	2.13	1.85	0.20	0.30
tulips	2.00	2.01	0.08	0.29
fresh-blueberries	2.00	1.93	0.0	0.23
lobelia	2.00	1.78	0.07	0.21
flowers	1.94	1.75	0.12	0.18
rice	1.88	1.71	0.08	0.19
northbeach	1.31	1.28	0.10	0.07
misc	1.31	1.18	0.04	0.13
water	1.25	1.16	0.02	0.05
long_island	1.19	1.33	0.0	0.03
clouds	1.13	1.38	0.04	0.06

The primitives in a regular texture are very easy to find compared to those in an irregular texture. Also, small sized primitives are perceived as less regular than larger ones [1]. An average of the five property scores gives the Average Property score for each subject. The Overall Regularity and the Average Property scores were separately averaged over all 10 subjects to generate the respective Mean Opinion Scores (MOS Property Regularity and MOS Overall Regularity) as shown in Table 4.4. The textures in Table 4.4 are listed in the order of decreasing perceived regularity according to the MOS Overall Regularity. The proposed texture regular-

ity metric is shown in the last column of Table 4.4. For comparison, Table 4.4 also shows the Pattern Regularity Metric of [43]. To account for extreme values at the ends of the testing range (very high and very low regularity), each metric value  $M_i$  is transformed into a predicted MOS ( $MOS_{p_i}$ ) value using a four-parameter logistic function [58]:

$$MOS_{p_i} = \frac{\beta_1 - \beta_2}{1 + e^{\left(M_i - \frac{\beta_3}{|\beta_4|}\right)}} + \beta_2 \quad (4.17)$$

For calibrating the texture regularity metric, the maximal values  $max_p$ ,  $max_d$  and  $MaxPlacementIrreg$  were computed from the maximal peakedness, decay and placement irregularities, respectively, over all the 21 textures. This results in the texture regularity metric given in Table 4.4. However, the other way to calibrate the proposed metric is based on the peakedness, decay and placement irregularities of the perceptually most regular texture in the database.

The performance of the proposed Texture Regularity Metric to quantify the perceived regularity is shown through the Pearson Linear Correlation Coefficient (PLCC) and the Spearman Rank Order Correlation Coefficient (SROCC) between  $MOS_p$  and MOS. As shown in Table 4.5, for the original set of 21 textures, the proposed Texture Regularity Metric results in a PLCC of 91.4% and SROCC of 90.1% for the Overall Regularity MOS, while the Pattern Regularity metric [43] gives a PLCC and SROCC of 83.5% and 86.6%, respectively. When correlating with the Average Property MOS, the proposed metric has PLCC and SROCC values of 91.7% and 93.3%, respectively, as shown in Table 4.6. These values were again significantly higher than the corresponding values of the Pattern Regularity metric [43]. Tables 4.5 and 4.6 also show the Root Mean Square Error (RMSE), the Mean Absolute Error (MAE) and the P-values for PLCC and SROCC while correlating the proposed and the compared Pattern Regularity [43] metrics with the MOS. The P-value is the probability of getting a correlation as large as the observed value by random chance,

Table 4.5: Correlation with Overall Regularity MOS.

<b>Texture Regularity</b>	<b>PLCC</b>	<b>SROCC</b>	<b>RMSE</b>	<b>MAE</b>	$P_{PLCC}$	$P_{SROCC}$
Pattern Regularity Metric [43]	83.46	86.59	0.35	0.26	0.000026	0.000000
Proposed Regularity Metric	91.41	90.10	0.27	0.19	0.000000	0.000000
Proposed Regularity Metric (without outliers)	97.04	97.83	0.16	0.12	0.000000	0.000000

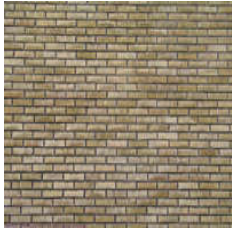
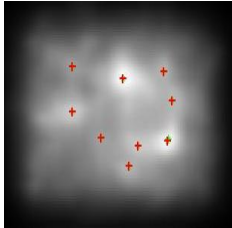
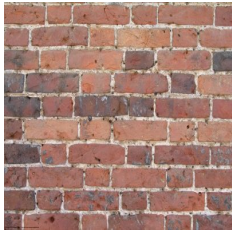
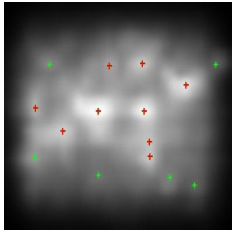

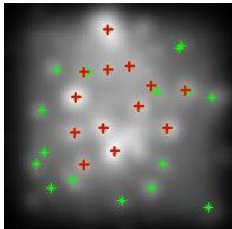
while the variables are unrelated. If the P-value is less than 0.05 then the correlation is significant. The P-values reported in Tables 4.6 and 4.6 indicate that all the correlation scores are statistically significant.

Table 4.6: Correlation with Average Regularity Property MOS.

<b>Texture Regularity</b>	<b>PLCC</b>	<b>SROCC</b>	<b>RMSE</b>	<b>MAE</b>	$P_{PLCC}$	$P_{SROCC}$
Pattern Regularity Metric [43]	79.50	78.93	0.35	0.28	0.000016	0.000021
Proposed Regularity Metric	91.65	93.28	0.23	0.17	0.000000	0.000000
Proposed Regularity Metric (without outliers)	95.99	95.87	0.17	0.11	0.000000	0.000000

As it can be seen from Table 4.4, *bricks*, *blocks* and *red-peppers* are outliers. These textures along with their saliency maps produced by the GBVS model [13] are shown in Table 4.7. The metric failed to accurately predict the perceived regularity on *bricks* and *blocks* due to the presence of artifacts leading to a non-uniform visual saliency map produced the VA model (GBVS [13]) these textures. As shown in Table 4.7, the presence of visual irregularities in the chroma components of these textures results in a high VA at these artifacts. This leads to a reduction of visual attention at the primitive locations and few VA peaks in the normalized VSM. Hence the corresponding spatial distribution score is very low and this results in a very low predicted overall regularity, while the subjective perceived regularity is higher for

Table 4.7: Visual Saliency Maps of outlier textures, produced by GBVS [13]. Fixation points are shown as crosses.

Texture Name	Texture	Predicted Saliency Map with Fixations
<i>blocks</i>		
<i>bricks</i>		
<i>red – peppers</i>		

these textures (Table 4.4). In the case of *red-peppers*, the high similarity in the tonal properties between the primitives results in a VSM in which the VA corresponding to the primitive regions are similar to each other. This results in a high similarity score and hence a high value for the proposed texture regularity metric. But the MOS scores were medium for this case due to irregularities in shape of the primitives. As shown in the last row of Tables 4.5, when eliminating these outliers, the PLCC and SROCC of the proposed metric with the Overall Regularity MOS increases to 97.0% and 98.0%, respectively, for the remaining textures. Similarly, the PLCC and SROCC of the proposed metric with the Average Regularity Property MOS increases to 96.0% and 96.1%, respectively, as shown in the last row of Table 4.6.



The Pattern Regularity Metric for this set of 18 textures without *bricks*, *blocks* and *red – peppers* has a PLCC of 82.1% and a SROCC of 83.5% with the Overall Regularity MOS, while it has a PLCC of 79.9% and a SROCC of 76.9% with the Average Regularity Property MOS.

### 4.3 Robustness to Geometric and Photometric Transformations

The robustness of the proposed texture regularity metric to geometric and photometric transformations was tested by applying a set of affine and color transformations. A 2D affine transformation matrix can be expressed as a decomposition of translation, rotation and scaling matrices as follows:

$$T = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Two sets of affine transformations, as shown in Fig. 4.3(b) and Fig. 4.3(c), were obtained respectively using the following parameter sets 1 and 2:

**Affine Parameter Set 1:**  $s_x = 0.9$ ;  $s_y = 1.1$ ;  $\theta = 5^\circ$ ;  $d_x = 1$ ;  $d_y = -1$

**Affine Parameter Set 2:**  $s_x = 1.05$ ;  $s_y = .95$ ;  $\theta = -5^\circ$ ;  $d_x = 0$ ;  $d_y = 2$

Three sets of color transformations were applied by modifying the hue, saturation or contrast of the 21 textures. The hue transformed textures were obtained by transforming the colors to the CIE *Lab* color space and rotating the color space by  $15^\circ$ . The saturated textures were obtained by increasing the color components in the CIE *Lab* color space by a factor of 2. Finally, contrast transformation was obtained by stretching the V component in the HSV color space. The values within the upper 0.5% and lower 0.5% of the V histogram were saturated to the extremal points of

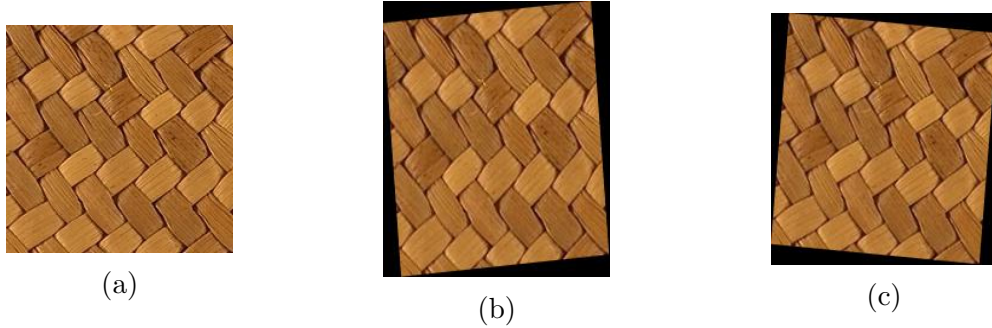


Figure 4.3: Examples of affine transformations: (a) Original texture; (b)Affine transform 1; (c) Affine transform 2.



Figure 4.4: Examples of color transformations: (a) Original texture; (b) Hue transform; (c) Saturation transform; (d) Contrast transform.

the range and the remaining values were stretched between them. Examples of hue, saturation and contrast transformations along with the input texture are shown in Fig. 4.4.

A subjective testing was conducted on the 5 transformed (2 affine and 3 color transformations) images of the same set of 21 textures, described above. Ten subjects with normal to corrected vision participated in the tests and gave an overall regularity score for each of the transformed textures. The proposed texture regularity metric was also computed on these transformed textures. Each metric value was also mapped into a predicted MOS using the 4-parameter logistic function as shown in equation (4.17). The robustness of the proposed metric to geometric and photometric transformations is compared with that of the Pattern Regularity metric [43] as shown in Table 4.8. Results for the original textures are shown in the first row of Table 4.8.

Table 4.8: Comparison of the performance of the proposed Texture Regularity Metric (TRM) with Pattern Regularity Metric (PRM) on textures with geometric and photometric transformations.

<b>Transform</b>	<b>Method</b>	<b>PLCC</b>	<b>SROCC</b>	<b>RMSE</b>	<b>MAE</b>	$P_{PLCC}$	$P_{SROCC}$
Original	TRM	91.41	90.10	0.27	0.19	0.000000	0.000000
	PRM [43]	83.46	86.59	0.35	0.26	0.000026	0.000000
Affine1	TRM	85.11	85.84	0.38	0.29	0.000001	0.000001
	PRM [43]	79.27	75.00	0.44	0.33	0.000018	0.000090
Affine2	TRM	88.19	79.06	0.33	0.27	0.000000	0.000020
	PRM [43]	66.84	59.84	0.51	0.38	0.000926	0.004163
Hue	TRM	78.26	78.83	0.44	0.32	0.000028	0.000022
	PRM [43]	82.90	82.79	0.39	0.28	0.000003	0.000004
Saturation	TRM	80.42	78.54	0.41	0.30	0.000010	0.000025
	PRM [43]	54.98	34.48	0.58	0.44	0.009821	0.125836
Contrast	TRM	86.71	83.99	0.35	0.28	0.000000	0.000002
	PRM [43]	79.60	74.06	0.42	0.33	0.000016	0.000123

Results for the affine transformed textures are shown in rows 2 and 3, while those for the hue, saturation and contrast transformed textures are shown, respectively, in rows 4, 5 and 6 of Table 4.8. In each row, the performance of the proposed regularity metric is presented in the first line while that of the compared pattern regularity metric [43] is given in the second line. As it can be seen in Table 4.8, the proposed metric is more robust to affine and contrast transformations than to hue and saturation transforms. Due to each of these transformations, the VSM produced by GBVS model changes and this leads to a change in the proposed metric. The change in VSM characteristics is higher for hue and saturation transformations, compared to the affine and contrast transformations. However, the proposed regularity metric scores better than the pattern regularity metric [43] under all transformations except the hue transformation.

The execution time of the proposed texture regularity metric will depend on the processor speed, coding language and efficiency of implementation. The algorithm, including the computation of visual saliency, takes about 2.5 seconds per

texture using Matlab R2012b on an Intel Core I5 CPU running at 2.6 GHz with 4 GB of RAM. Since our goal in this work is to achieve a high correlation between the proposed regularity metric and the subjective regularity (MOS), algorithmic optimizations for speed of execution is beyond the scope of this work.

ADAPTIVE TEXTURE SYNTHESIS BASED ON  
TEXTURE REGULARITY

This chapter presents the influence of texture regularity on the perceptual quality of textures synthesized through parametric and non-parametric approaches. It is shown through subjective testing that textures with different degrees of perceived regularity exhibit different degrees of vulnerability to synthesis artifacts. The work also proposes an algorithm for adaptively selecting the appropriate texture synthesis algorithm based on the no-reference texture regularity metric proposed in Chapter 4.

The contrast sensitivity function of the Human Visual System (HVS) indicates that the HVS is less sensitive to visual artifacts in high frequency components. Many perceptually motivated video codecs like [3] and [4], exploit this fact to achieve higher compression ratios. Textures, because of their high spatial frequencies, are very good candidates for lowering the bit-rate at the cost of introducing imperceptible artifacts. Typically, the texture-based video codecs attempt to save on bit-rate by synthesizing the texture regions at the decoder such that the synthesized regions perceptually resemble the original texture. This is achieved by sending a sample texture patch or synthesis parameters which represent the original texture. The general framework for image and video codecs based on texture synthesis is shown in Fig. 5.1. The input image or a frame of video is first segmented into texture and non-texture regions. The texture regions are further analyzed to extract the data needed for texture synthesis. This data may be constraint parameters in parametric texture synthesis approaches like [7] or it could be seed-pixel regions in exemplar-based synthesis approaches like [59]. The quality assessment unit uses the synthesized and the original texture to estimate the fidelity of the synthesized texture. The texture synthesis process is repeated in an iterative manner until a prescribed quality is met.

The final synthesis parameters and/or the seed pixels of the texture regions of the frame are compressed and sent over the network. The standard video codecs like H.264 are used for the compression of non-texture regions.

The various parametric approaches like [7] and [60] and non-parametric approaches like [59] and [61] differ in their speed, perceptual quality and the amount of side information needed for synthesizing textures. In this work, we propose that the perceived loss in fidelity when employing a considered synthesis method is not the same for all types of textures but depends on the regularity of the original texture. In other words, for synthesizing a texture of a given regularity, one method may have a perceptually better performance compared to another. As mentioned in [59] and [62], parametric methods are believed to be more suited for stochastic textures and exemplar-based methods like [59] are appropriate for structured textures. However, it is shown in this paper through examples that even some of the highly structured textures can be synthesized with high fidelity using parametric methods provided that they exhibit a significantly high regularity. This regularity of the original texture is quantified through the no-reference perceptual texture regularity metric proposed in Chapter 4.

### 5.1 Existing Approaches for Texture Synthesis

Textures in video sequences may be classified into static and dynamic textures. Static textures are those textures whose appearance does not change from frame to frame. That is, they show a repetition only in space. Dynamic textures are regions of a video sequence that exhibit a repetition in both space and time. We restrict our analysis to only 2D static textures in this work.

2D Texture synthesis methods can be broadly classified into parametric approaches and non-parametric based approaches. In parametric based approaches, the texture is characterized and described through a set of perceptually motivated

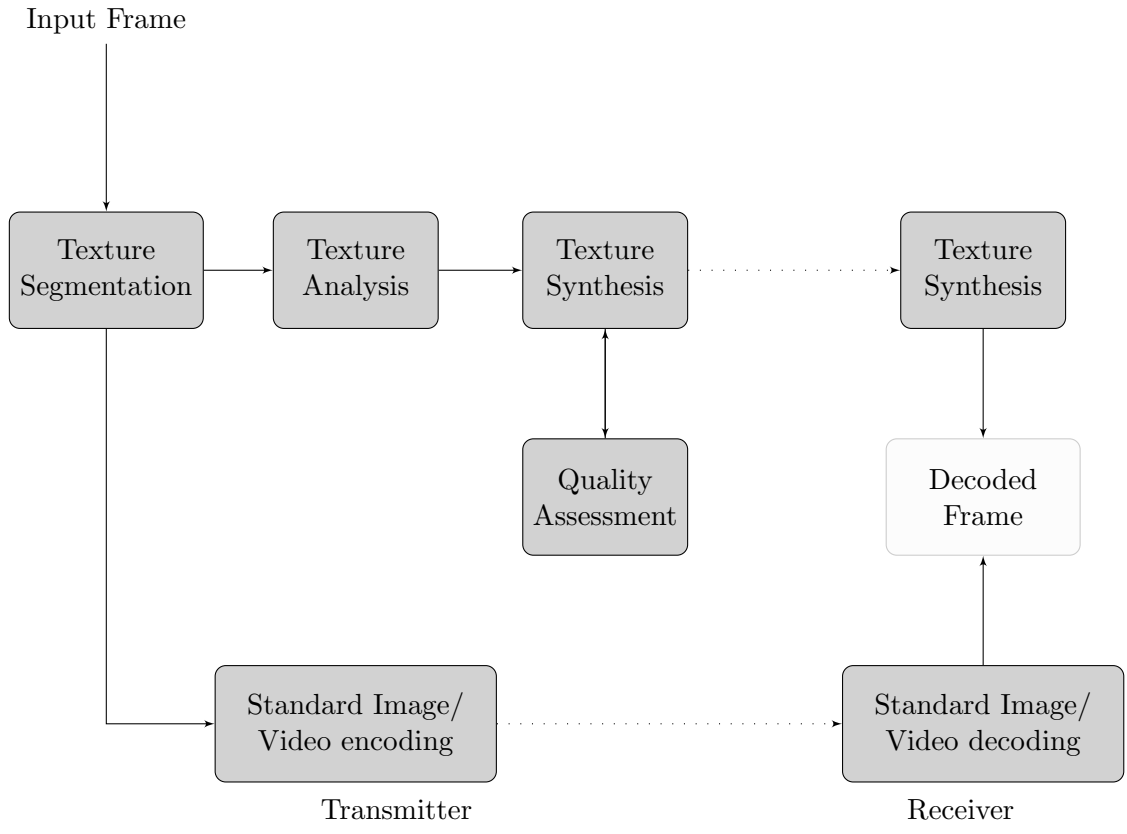


Figure 5.1: The general framework for video codecs based on texture synthesis.

parameters. These methods begin with a uniform random texture and apply the parameters/constraints in an iterative manner to match the appearance of the original texture. In one of the earliest works by Heeger [60], the histograms of the wavelet coefficients of different bands of the steerable pyramid of the original texture are used as parameters. Portilla et al. [7] proposed a more refined model based on matching wavelet coefficients of multi-scale oriented filter responses. The parameters in [7] include the first-order and second-order statistics of the filter coefficients of the neighboring orientation and scale. Being parametric, these methods describe any texture via a few parameters and thus achieve a very compact representation of

textures. This is very useful for attaining high compression efficiency in scenarios like low bit-rate video transmission or compact storage applications.

Non-parametric approaches can be further classified into pixel-based and patch-based approaches. Pixel-based approaches synthesize one pixel at a time while patch-based approaches synthesizes an output patch or a group of neighboring pixels at a time. A region of the original texture called the seed or the exemplar is used for both these approaches. The pixel-based synthesis proposed by Effros et al. [61] works as follows. Given an input exemplar, the output is first initialized by copying a small seed region from the input. The synthesized region is then gradually grown from the initial seed by assigning the output pixels one by one in an inside-out fashion. Each output pixel is determined by a neighborhood search process. To synthesize an output pixel at a particular location, the method considers a neighborhood of user-determined size (3x3 for this toy example) around it and collects the set of already synthesized pixels in that neighborhood. The method then finds the candidate set of good matches from the input with respect to this partial neighborhood composed of these already synthesized pixels, and assigns the output pixel as the center of a randomly selected neighborhood from the candidate set. This process is repeated for every output pixel by growing from the initial region until all the output pixels are assigned. This synthesis method is very slow and also may not give good synthesis quality for arbitrary sized texture primitives using a fixed neighborhood size for finding the candidate regions. In the method proposed by Wei and Levoy [63], both the synthesized texture and the seed texture are represented as image pyramids and reconstruction is done through neighborhood matching at each level. Also, the pixels are synthesized in a raster-scan order instead of inside out. For every output pixel, the causal neighborhood is matched with the input at the corresponding level of the pyramid. The pixel at the center of the matched neighborhood is assigned as the



output pixel at the considered location. The coarser levels are reconstructed first followed by the finer levels. Even with a fixed neighborhood size at each level, a larger range of textural primitives could be accommodated through this multi-resolution approach. The method further accelerates the best neighborhood search through a tree-structured vector quantization and hence increases the speed of synthesis.

Whenever output texture pixels are synthesized based on neighborhood matching as in the above mentioned approaches, it is very likely that adjacent pixels in the input exemplar end up as adjacent pixels in the output. So instead of searching through the entire input exemplar for every output pixel, only a smaller set of candidate locations are considered in the K-coherence based approach [64]. The K-coherence algorithm is divided into two phases: analysis and synthesis. During analysis, the algorithm builds a similarity set for each input texel (texture element), where the similarity set contains a list of other texels with similar neighborhoods to the specific input texel. During synthesis, the algorithm copies pixel from the input to the output, but in addition to colors, the source pixel location is also copied. To synthesize a particular output pixel, the algorithm builds a candidate set by taking the union of all similarity sets of the neighborhood texels for each output texel, and then searches through this candidate set to find out the best match. The size of the similarity set,  $K$ , is a user-controllable parameter (typically in the range 2 to 11) that trades off the overall speed for the quality of synthesis.

The quality and speed of pixel-based approaches can be improved by synthesizing patches rather than pixels. The quality of synthesis is better as the contiguous pixels belonging to a particular patch in the input texture are more likely to be contiguous in the output texture. However, in patch-based algorithms, the issue is more complicated as a patch, being larger than a pixel, usually overlaps with the already synthesized portions, so some decision has to be made about how to handle

the conflicting regions. In [65], new patches simply overwrite over existing regions. By using patches with irregular shapes, this approach took advantage of the texture masking effects of the human visual system and works surprisingly well for stochastic textures. Blending the overlapped regions in patch-based texture synthesis can cause blurry artifacts in some situations. Instead of blending, the work by [59] finds an optimal boundary between adjacent patches in the overlapped regions via graph cut. The graph-cut synthesis first places an exemplar patch in a random location in the synthesized texture grid. It then places successive exemplar patches that overlap with the prior synthesized patches. After each placement, the boundaries of the adjacent patches are then trimmed through a graph-cut algorithm in the overlapping regions. There are two ways in which the exemplar can be placed, namely (i) random placement (P1) and (ii) entire patch matching (P2). The placement algorithm P1, places the exemplar at a random position, while algorithm P2 always places the exemplar at a position such that the sum of the squared difference between the pixels in the overlapping region is minimized. The algorithm P1 is faster while the algorithm P2 promises to have a better synthesis quality as the misalignment artifacts will be minimized.

There is a trade-off between the speed of texture synthesis, the fidelity of texture synthesis and the compression ratio using different texture synthesis methods. As mentioned above, there are two variations of the graph-cut texture synthesis approach [59], one that aims at faster synthesis while the other aims at higher synthesis quality. Also, the parametric approaches often achieve very low bit-rates for the same perceived quality on certain types of textures. Texture regularity can be useful in employing the right method for texture synthesis. In the following sections, a novel method for performing adaptive texture synthesis is proposed.

## 5.2 Proposed Adaptive Texture Synthesis based on Texture Regularity

Textures are basically patterns of pixel intensities. The pattern that is spatially repeated throughout the texture in some regular or irregular manner is called a primitive. The regularity of textures can be quantified through properties like size, shape, color and orientation of the primitives [1]. The overall perceptual regularity of textures is due to the accumulated effect of all these factors. Based on these visual properties of the primitives along with their placement regularity, textures can be classified into regular, irregular or hybrid textures as discussed in Chapter 4. The various texture synthesis methods produce different types of visual artifacts that lead to a loss in fidelity of the synthesized texture compared to the original. It is shown as part of this work that the regularity class of a texture has a direct impact on the perceptual loss in fidelity of the synthesized texture and this loss in fidelity varies from one synthesis method to another.

Effect of regularity on texture synthesis:

We propose that the perceived loss in fidelity using a considered synthesis method is not the same for all types of textures but depends on the regularity of the original texture. This is illustrated through some examples in Table 5.4 which contains the original textures and the synthesized textures. The original textures are shown in column 1 of Table 5.4. The textures synthesized by two variations of graph-cut synthesis (P1 and P2) [59], are shown in columns 2 and 3. The Luma component of the texture synthesized by the parametric approach suggested in [7] is shown in column 4.

As discussed in Chapter 4, a subjective testing was conducted on 18 textures from two databases, namely, the MIT Vistex database [56] and the Graph-Cut tex-

ture synthesis database [57] in order to determine the perceived regularity of these textures. Ten subjects with normal to corrected vision participated in the subjective tests. The textures, equally distributed amongst the broad classes of regular, irregular or hybrid textures, were randomly displayed one after another to each subject. The subjects were asked to score the regularity for each observed texture using a three-scale score with 1 corresponding to lowest and 3 to highest. The regularity scores were averaged for each texture image over all 10 subjects to generate the Regularity Mean Opinion Scores (R-MOS) for the considered texture.

In another experiment, a subjective testing was conducted to analyze the synthesis quality of the same 18 textures from the texture databases mentioned above. Fifteen subjects with normal to corrected vision participated in the subjective tests. For the synthesis, seed texture regions were hand-picked from each of these textures and given as input to Kwatra’s patch-based texture synthesis algorithm [59]. As mentioned before, corresponding to the two versions of the algorithm (P1 and P2) in [59], two sets of textures were synthesized. The tile size was just large enough to capture the periodicity along the X and Y directions. The synthesized texture dimension was set as 6 times the size of the original seed. A third set of textures was synthesized from the original textures using the Portilla’s method [7]. In the implementation of [7] obtained from [66],  $N=4$  multi-resolution levels,  $K=4$  orientations and 75 iterations were used. A  $128 \times 128$  region was used for analysis and a set of 710 parameters as mentioned in [7] was used for synthesis of textures that were of  $192 \times 192$  in dimension. For each considered synthesis method, the synthesized textures along with the corresponding original textures were displayed side-by-side in a subjective testing experiment. To avoid scaling differences, only the top-left portions of the synthesized images having the same dimension as the original textures were displayed. The temporal order of the image pairs were randomized

Table 5.1: Mean Opinion Scores (MOS) for original and synthesized regular textures.

<b>Texture</b>	<b>Regularity MOS Original</b>	<b>Q-MOS Kwarta P1 [59]</b>	<b>Q-MOS Kwarta P2 [59]</b>	<b>Q-MOS Portilla [7]</b>
Marbles	3.00	1.4	1.7	3.2
Tile	3.00	1.5	2.4	4.4
Gecko	2.88	1.8	1.3	1.4
Cream	2.88	3.5	4.2	4.2
Keyboard	2.81	1.9	2.0	1.6
Maille	2.81	2.5	3.2	2.86
Average		2.60	2.80	2.95

and displayed one after another to each subject. The reference texture was always displayed on the left and the synthesized texture on the right. The subjects were asked to score the overall fidelity for each synthesized texture by comparing it to the original, using a five-scale score with 1 corresponding to lowest and 5 to highest. An average of the fidelity scores over all 10 subjects for a considered texture gives the Mean Opinion fidelity Score (Q-MOS) for a considered method. The Regularity MOS (R-MOS) is available from the texture regularity subjective experiments and is shown in the second column of Tables 5.1 to 5.3. The synthesis Quality MOS (Q-MOS) for the regular textures using Kwatra’s P1, P2 and Portilla’s methods are given in columns 3, 4 and 5 of Table 5.1 respectively. Similarly the R-MOS and the synthesis Q-MOS for the hybrid and irregular textures are shown in Tables 5.2 and 5.3, respectively. The Average Q-MOS for the different synthesis methods are shown in the last row of Tables 5.1, 5.2 and 5.3.

As it can be seen from Table 5.1, for highly regular textures, Portilla’s method has the average best performance. The parametric synthesis performs the best when modeling and synthesizing highly regular textures as the autocorrelation constraints in the wavelet coefficients can be well established. The Kwatra’s patch matching methods P1 and P2 exhibit misalignment at the patch boundaries and a partial loss

Table 5.2: Mean Opinion Scores (MOS) for original and synthesized hybrid textures.

Texture	Regularity MOS Original	Q-MOS Kwarta P1 [59]	Q-MOS Kwarta P2 [59]	Q-MOS Portilla [7]
Horses	2.69	3.46	3.57	1.25
Puzzle-pieces	2.69	3.57	3.93	2.00
Red-peppers	2.31	3.75	3.79	2.57
Tomatoes	2.13	3.61	2.93	1.96
Tulips	2.00	3.75	3.18	1.14
Average		3.63	3.48	1.79

Table 5.3: Mean Opinion Scores (MOS) for original and synthesized irregular textures.

Texture	Regularity MOS Original	Q-MOS Kwarta P1 [59]	Q-MOS Kwarta P2 [59]	Q-MOS Portilla [7]
Flowers	1.94	3.32	2.25	2.43
Rice	1.88	2.68	2.82	1.04
Misc	1.31	3.07	3.32	3.80
Water	1.25	2.25	2.82	3.75
LongIsland	1.19	1.00	1.18	1.64
Clouds	1.13	1.50	1.36	1.46
Average		2.30	2.29	2.35

in spatial periodicity for regular textures. In the case of hybrid textures, the performance of Portilla’s method is poor compared to Kwatra’s methods as illustrated in Table 5.2. The parametric synthesis [7] has a poor performance when the primitives have differing orientations or intensities. Also, there is a higher loss in fidelity when the primitives resemble closed blobs or flat polygons with just linear boundaries [7]. These limitations are more pronounced for hybrid textures than for regular or irregular textures. As shown in Table 5.3, all the 3 methods have a similar performance when synthesizing irregular textures. The better local structure produced by the Kwatra’s methods is partially offset by the tiling artifacts caused due to pseudo-periodic repetition of primitives, which was absent in the original textures. Between the two variations of Kwatra’s graph-cut synthesis, the method P1 is based on ran-

dom patch placement and is much faster than P2. The synthesis performance of P1 is however not affected for hybrid or random textures in which there is an inherent randomness already present. However for regular textures, its performance is not as good as P2.

Adaptive Texture Synthesis Algorithm:

The block diagram of the proposed adaptive texture selection algorithm is given in Fig. 5.2. The input image is first analyzed and segmented into texture and non-texture regions. The non-texture regions are compressed using codecs like JPEG2000 or H.264. The texture regions are further segmented into homogeneous texture sub-regions. The no-reference texture regularity metric,  $R$ , described in Chapter 4, is computed for each texture sub-region. Each texture sub-region is synthesized using one of the three aforementioned algorithms, based on the texture regularity metric for that sub-region. When  $R \leq Th_1$ , the texture is considered an irregular texture and the synthesis quality of all the 3 methods are similar. In this case Portilla’s method [7] is used for synthesis as it involves synthesis parameters that occupy fewer bits compared to the exemplar patches in Kwatra’s methods. When  $Th_1 \leq R \leq Th_2$  then the texture is considered a hybrid texture and is synthesized using Kwatra’s P1 method. Finally, for highly regular textures  $R \geq Th_2$  and in this case, the primitive region is approximated by a square of size  $L \times L$  pixels, as mentioned in Chapter 4. For regular textures with large primitives ( $L \geq Th_3$ ), Kwatra’s P2 algorithm is used. For regular textures with small-sized primitives ( $L \leq Th_3$ ), Portilla’s synthesis [7] is employed. The thresholds  $Th_1 = 0.3$ ,  $Th_2 = 0.5$  and  $Th_3 = 128$  and were used in our implementation.

### 5.3 Simulation Results

The efficacy of the proposed algorithm to select the right synthesis algorithm for a texture under consideration is illustrated through Table 5.4. The original textures

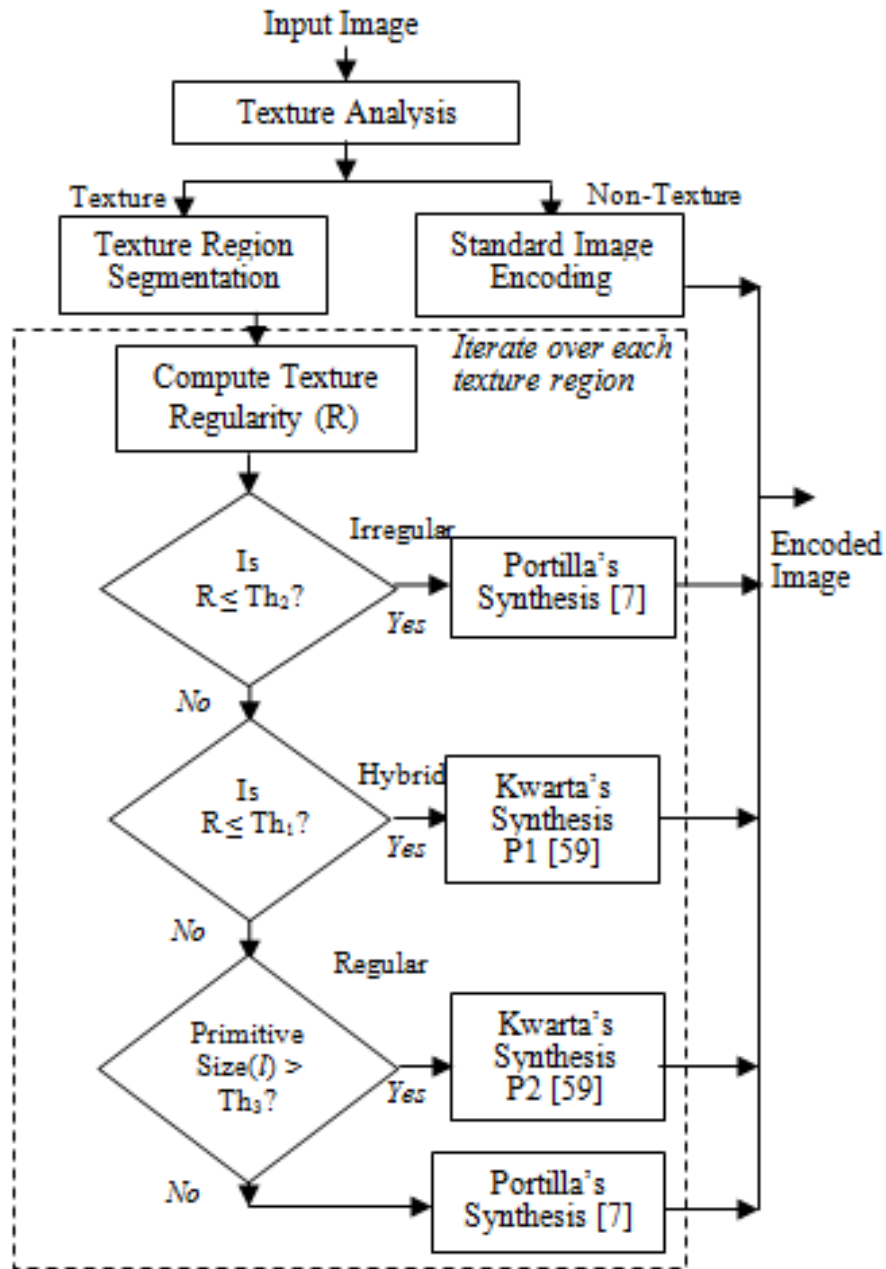
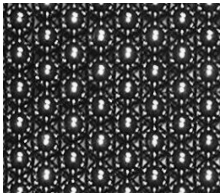
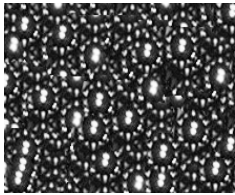
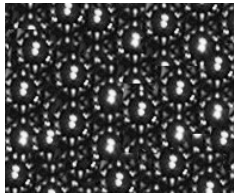
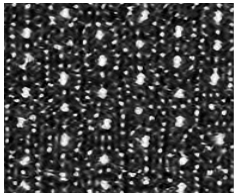
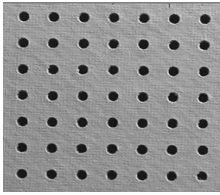
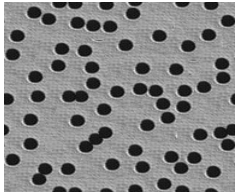
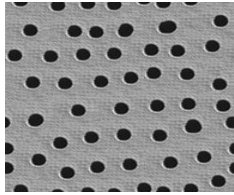
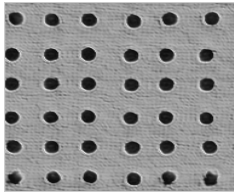

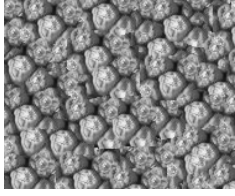
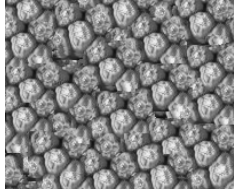
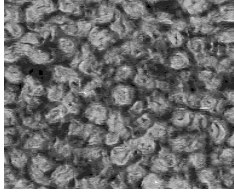




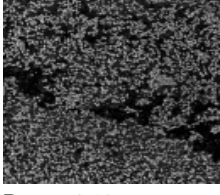
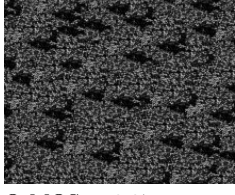
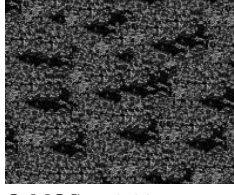
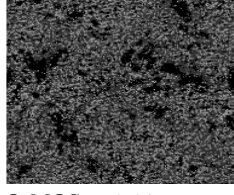






Figure 5.2: Block diagram of the proposed adaptive texture synthesis algorithm. The regularity thresholds  $Th_1$  and  $Th_2$  are used for classifying the textures as irregular or hybrid.



along with the proposed texture regularity metric are shown in column 1 of Table 5.4. The synthesized textures using Kwatra’s P1 and P2 algorithms and those synthesized using Portilla’s method are given in columns 2, 3 and 4. The fidelity MOS (Q-MOS) for each of the synthesized textures is also shown. The results for regular, hybrid and irregular textures are shown in rows 1, 2 and 3, respectively. Portilla’s method outperforms Kwatra’s methods for the regular and the irregular textures. But for hybrid textures, the patch-based method in [59], gives better results. This is in accordance with our proposed algorithm. Also, as observed in Table 5.3, the synthesis quality of irregular textures is almost independent of the synthesis method.

Table 5.4: Original and Synthesized Textures with regularity metric (Reg) and Quality Mean Opinion Scores (Q-MOS).

Original Texture	GraphCut Synthesis P1	GraphCut Synthesis P2	Parametric Synthesis
 Reg = 0.71	 Q-MOS = 2.14	 Q-MOS = 2.27	 Q-MOS = 3.41
 Reg = 0.67	 Q-MOS = 1.77	 Q-MOS = 2.41	 Q-MOS = 4.55
 Reg = 0.33	 Q-MOS = 3.91	 Q-MOS = 3.91	 Q-MOS = 2.64
 Reg = 0.19	 Q-MOS = 4.45	 Q-MOS = 4.41	 Q-MOS = 1.45
 Reg = 0.13	 Q-MOS = 3.27	 Q-MOS = 3.23	 Q-MOS = 3.64
 Reg = 0.05	 Q-MOS = 2.27	 Q-MOS = 2.64	 Q-MOS = 3.86

### A REDUCED REFERENCE PERCEPTUAL QUALITY METRIC FOR TEXTURE SYNTHESIS

In this chapter, a reduced-reference quality metric that quantifies the perceived quality of the synthesized textures is presented. The metric is based on the change in perceived regularity and change in perceived granularity between the original and the synthesized textures. The perceived regularity is quantified through a texture regularity metric based on visual attention. It is shown through subjective testing that the proposed metric has a strong correlation with the Mean Opinion Score for the fidelity of synthesized textures and outperforms the state-of-the-art full-reference quality metrics.

#### 6.1 Existing Reduced Reference and Texture Quality Metrics

Image quality metrics can be broadly divided into 3 classes, namely, full-reference, no-reference and reduced reference metrics. In the case of full-reference metrics, both the distorted and the reference image are completely available. No-reference metrics compute a quality score using only the distorted image assuming a particular type of distortion like blurring, ringing or blocking. Reduced reference (RR) metrics compute the quality of the distorted image from features of the reference image.

We address the problem of measuring texture synthesis quality in this work. Even when the original reference texture is available, the process of evaluating the quality of texture synthesis is ill-posed for two reasons, namely, (i) the sizes of the synthesized and the original texture can be different and (ii) the synthesized texture need not have pixel-wise correspondence with the original texture but can still appear perceptually equivalent. Since the perceptual equivalence of textures stem from their inherent structure, metrics that quantify the structural similarity of the synthesized and the original textures seem to be most relevant.

There are many full-reference quality metrics to measure the structural differences between textures. The SSIM [67] and the multi-scale SSIM (MSSSIM) [68] metrics use the mean, variance, and co-variance of pixels to compute structural similarity. The CWSSIM [69] and WCWSSIM [70] extend SSIM to the complex wavelet domain. A pixel-domain structural texture similarity metric (STSIM) was proposed in [71]. Later, STSIMs involving steerable pyramid decomposition [7] and sub-band autocorrelations and cross-correlations of wavelet coefficients were proposed in [72]. A metric that uses Portilla’s constraints [7] along with the Kullback-Leibler Divergence (KLD) is suggested in [62]. The above mentioned metrics such as [68] and [69], are either devised for near-threshold applications (e.g., image compression) or for supra-threshold applications (e.g., image retrieval) [62]. They may be either highly constrained or excessively accommodative of texture synthesis artifacts. The quality metric in [62] gives 85% of weightage to Portilla’s constraints and hence may not be suitable for non-stochastic textures.

As explained in Chapter 5, there are many texture synthesis based video coding schemes in which texture regions are synthesized at the decoder using parametric or non-parametric synthesis algorithms. Many of these algorithms are iterative approaches in which the synthesized image fidelity needs to be estimated at the end of each iteration. Also, the receiver may have to evaluate the quality degradation of synthesized regions in order to communicate to the sender for alternative synthesis parameters or ‘seed’ textures. Since the original reference is not available at the decoder, RR features representing the original texture can be used to measure image fidelity. RR quality metrics are typically used for quality evaluation at the receiver side of a communication system [73]. They can also be used in other applications like Content Based Image Retrieval (CBIR) where the RR features can act as an efficient representation of the image and help in reducing the computations

for feature-matching. In a communication system, the sender sends the original image data after compression and modulation through a distorted channel. At the sender side the RR features are computed from the original image and sent through a distortion-free auxiliary channel. At the receiver side the data received from the distorted channel is demodulated and decoded. Quality evaluation is performed at the receiver side by extracting the features of the distorted image and comparing the distorted image's features with the RR features from the auxiliary channel. The computed quality degradation can be used for controlling the network streaming sources.

The performance of a RR quality metric to predict the true visual degradation of images depend on the following:

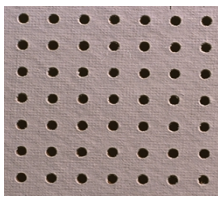
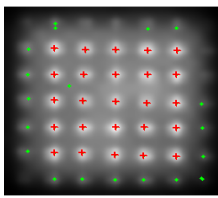
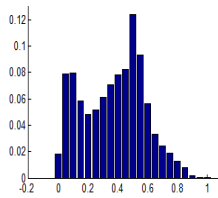
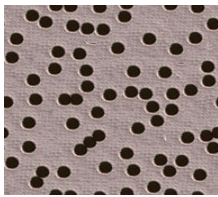
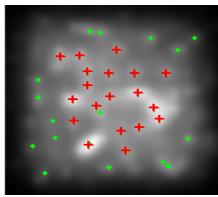
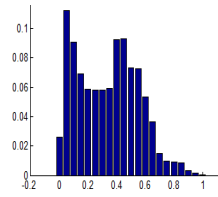

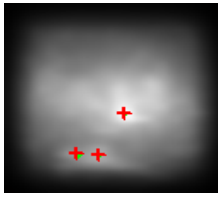
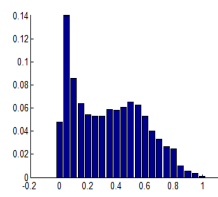

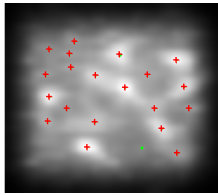
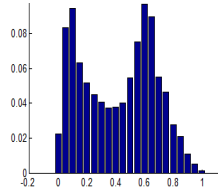
- (i) the ability of the RR metric to efficiently represent the image
- (ii) the sensitivity of the RR metric to various image distortions
- (iii) the visual perceptual relevance of the RR features

The various RR quality metrics can be further classified into 3 classes based on the modelling approach:

- (i) RR metrics based on modelling image distortions
- (ii) RR metrics based on modelling the natural image statistics
- (iii) RR metrics based on modelling the Human Visual System (HVS)

Popular reduced reference metrics include RRIQA [73] and RRSSIM [74]. These metrics require training of parameters that are trained to approximate a target full-reference metric as in the case of RRSSIM, or that are trained to fit subjective quality mean opinion scores (MOS) over a database, as in the case of RRIQA. Both of these metrics also require the extraction of several features as side information (48

Table 6.1: Visual Saliency Map characteristics for original and synthesized textures.

Texture	VSM with VA peaks	Histogram of VSM	Regularity
			0.674
			0.304
			0.054
			0.307

for RRIQA using a 3 level steerable pyramid decomposition and 36 for RRSIM) that are extracted from the original image.

## 6.2 Proposed Reduced Reference Metric based on Texture Regularity

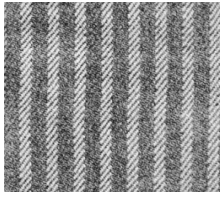
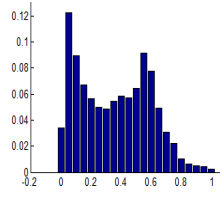
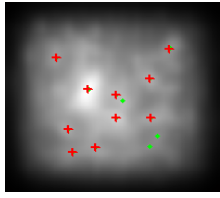
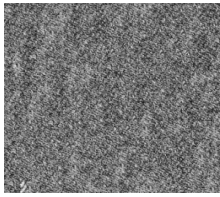
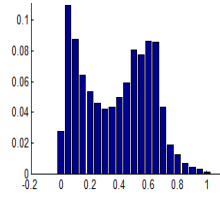
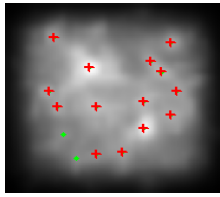
The perceived loss in fidelity due to texture synthesis artifacts can be quantified by the effect of these artifacts on the regularity and granularity of the textures. This is illustrated with some examples in the following subsections.

### 6.2.1 Effect of Texture Synthesis Artifacts on Regularity

The various parametric and non-parametric approaches for texture synthesis lead to different types of artifacts. The parametric approaches like [7] while preserving the

global texture properties lead to a loss of local structure. The patch-based approaches like [59] while preserving the local structure lead to a loss in global structure. The synthesis artifacts include misalignment of texture patterns, blurring artifacts due to overlap of patterns and tiling artifacts resulting from excessive repetition of patterns. These artifacts disrupt the regularity of the original texture. Also, as illustrated in Table 6.1, these artifacts naturally capture our visual attention (VA) in a different way compared to the VA of the original texture. This VA can be captured through a Visual Saliency Map (VSM) whose values directly quantify the extent to which each region grabs the human attention. The saliency map is normalized to 1 and shown as an image in which the brightest pixels (close to 1) correspond to highest attention and the darkest pixels (close to 0) correspond to lowest attention. The effect of texture synthesis artifacts on visual saliency is illustrated through a regular and an irregular texture in Table 6.1. Samples of the original and synthesized textures (column 1) along with the corresponding VSMs (column 2) and the histogram of the VSMs (column 3) are shown in Table 6.1. The textures were synthesized using Graph-Cut synthesis [59]. The VSMs in Table 6.1 were generated using the GBVS model [16]. The impact of misalignment artifacts on visual saliency while synthesizing a regular texture is illustrated through the Tile texture in rows 1 and 2 of Table 6.1. For this regular texture, as shown in Table 6.1, the original texture’s VSM (row 1) exhibits a periodic regular placement of local peaks. The local peaks correspond to the center of the textural primitives where there is maximum VA in the local neighborhood. In the corresponding synthesized texture’s VSM (row 2), there is a higher visual saliency at locations corresponding to the boundary misalignment artifacts. This lowers the relative saliency of other regions without artifacts. This results in a modified VSM in which the location and distribution of the local peaks differ from those of the original VSM. The artifacts also lead to a change in the shape of the VSM histogram characteristics like location of the last peak and the

Table 6.2: Limitation of the texture regularity metric to capture change in granularity.

Texture	Histogram of VSM	VSM with VA peaks	Regularity
			0.332
			0.321

decay rate. A decrease in the perceived texture regularity can also be observed for the synthesized texture (Table 6.1, column 1, row 2) relative to the original one (Table 6.1, column 1, row 1). The objective texture regularity metric (Chapter 4) computed from the VSM characteristics decreases from 0.674 to 0.304 for the Tile texture as a result of texture synthesis artifacts. Similarly, the effect of tiling artifacts on visual saliency and texture regularity is illustrated through the irregular Water texture in rows 3 and 4 of Table 6.1. The irregularity in the original texture arises from the fine-granularity of the primitives. This results in few visually salient regions and less number of local peaks (row 3). The VSM of the synthesized texture (row 4) shows more local peaks due to the tiling artifacts. This leads to a change in the shape of the VSM histogram. It can also be seen from Table 6.1 (row 4) that the artifacts also result in an increase in the perceived texture regularity as also verified by the authors through subjective testing of perceived texture regularity. As expected, the texture regularity metric (Chapter 4) increases from 0.054 to 0.307 for the Water texture due to tiling artifacts.



### 6.2.2 Effect of Texture Synthesis Artifacts on Granularity

Some of the texture synthesis algorithms synthesize one pixel at a time. These approaches sometimes generate highly granular textures in which the shape and contours of the original primitives are not preserved, resulting in a low fidelity score. The synthesis methods, while altering the local characteristics of the original texture, may still generate a texture that preserves the VSM characteristics of the original texture. As a result, the granularity artifacts cannot be captured through the change in texture regularity metric that depends completely on the VSM. This is illustrated in Table 6.2, through the *D11* texture obtained from [75]. The original *D11* texture along with the histogram of its VSM and the VA peaks are shown in row 1 of Table 6.2. The *D11* texture synthesized by the algorithm proposed in [63] along with its VSM characteristics are shown in row 2 of Table 6.2. The texture regularity metric for the two textures, computed using the algorithm proposed in Chapter 4, is given in the last column of Table 6.2. The DMOS score for the two textures is 0.875 in a (0,1) scale which indicates a high loss in fidelity. But since the VSM characteristics of the two textures does not differ much, the change in texture regularity based on VSM characteristics, will not quantify the perceptual loss in texture quality.

### 6.2.3 Proposed Texture Granularity Metric

Granularity is directly related to the size of the textural primitives. A granularity score is computed from the wavelet transform coefficients of the texture image as described below. As mentioned in [76], by applying the Discrete Dyadic Wavelet Transform (DDWT) to the texture image  $g(u, v)$ , two detailed-subband images,  $W_{g(u,v,s)}^1$  and  $W_{g(u,v,s)}^2$ , corresponding, respectively, to the horizontal and the vertical edges and one low-frequency subband image,  $W_{g(u,v,s)}^3$  are obtained at each level  $s$ . Only the low-frequency subband image,  $W_{g(u,v,s)}^3$ , will be used for the decomposition at

the next level. A 5-level non-subsampled DDWT decomposition is performed in the proposed method.

For each of the detailed subbands  $W_{g(u,v,s)}^l$ ,  $l = 1, 2$  and  $s = 1, 2, 3, 4$ , a correlation measure is computed at each location as follows:

$$Corr^l(u, v, s) = W_{g(u,v,s)}^l \cdot W_{g(u,v,s+1)}^l. \quad (6.1)$$

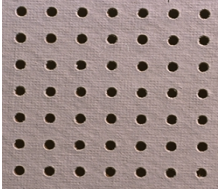
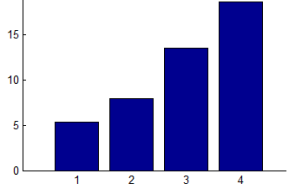
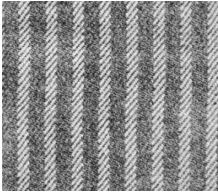
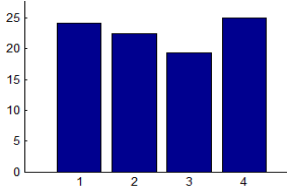
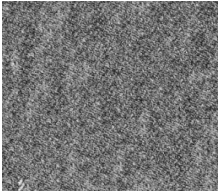
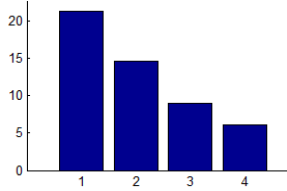
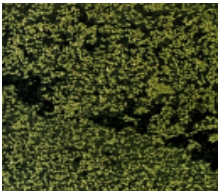
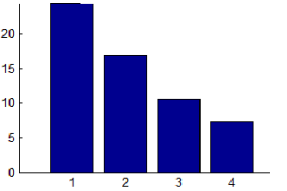
Let  $\mu_{corr}(s)$  denote the average correlation of the detail subband coefficients at a level  $s$  with the corresponding detail subband coefficients at the next level. The granularity score,  $G$ , is measured as follows:

$$G = \frac{\mu_{corr}(1) - \mu_{corr}(4)}{C} \quad (6.2)$$

where  $C$  is a normalization constant so that  $G$  is in the range  $(-0.5, 0.5)$ .  $C = 80$  was used in our implementation.

The variation of the proposed granularity metric with the size of texture primitives is illustrated through some examples in Table 6.3. The textures are shown in column 1 while the average correlation  $\mu_{corr}(s)$  for different scales  $s$  in the 5-level DDWT is shown in column 2 of Table 6.3. The textures are arranged in increasing order of granularity in Table 6.3. The proposed granularity metric is shown in column 3 of Table 6.3. As shown in row 1 of Table 6.3, for a texture with large primitives, the amount of inter-subband correlation increases with scale  $s$ . This results in a negative granularity metric for textures with large primitives. The inter-subband correlation decreases as the scale  $s$  increases for textures with very small sized primitives (micro-textures) as illustrated through the *Misc* texture in row 4 of Table 6.3. This results in a positive granularity metric for textures with very small primitives. The ability of the granularity metric to compensate for the inadequacies of the regularity metric is shown through rows 3 and 4 of Table 6.3. The original *D11* texture

Table 6.3: Examples of textures and their corresponding Granularity Metric  $G$ . The horizontal axes of the plots in the middle column are the DDWT levels ( $s$ ).

Texture	Average Correlation $\mu_{corr}(s)$	Granularity Metric $G$
		-0.166
		-0.012
		0.188
		0.213

along with its granularity score is shown in row 2 of Table 6.3 while the granularity characteristics of the texture synthesized by the algorithm proposed in [63] is shown in row 3 of Table 6.3. As shown in Table 6.3, the synthesized texture appears more grainy compared to the original texture and this results in an increase in the proposed granularity metric from -0.12 to 1.88. Based on these observations, this work proposes an RR texture quality metric that quantifies the loss in texture fidelity due to synthesis through the change in regularity and the granularity of the synthesized texture relative to the original. Fig. 6.1 shows a flowchart of the proposed RR texture synthesis quality metric computation algorithm at the decoder. The proposed

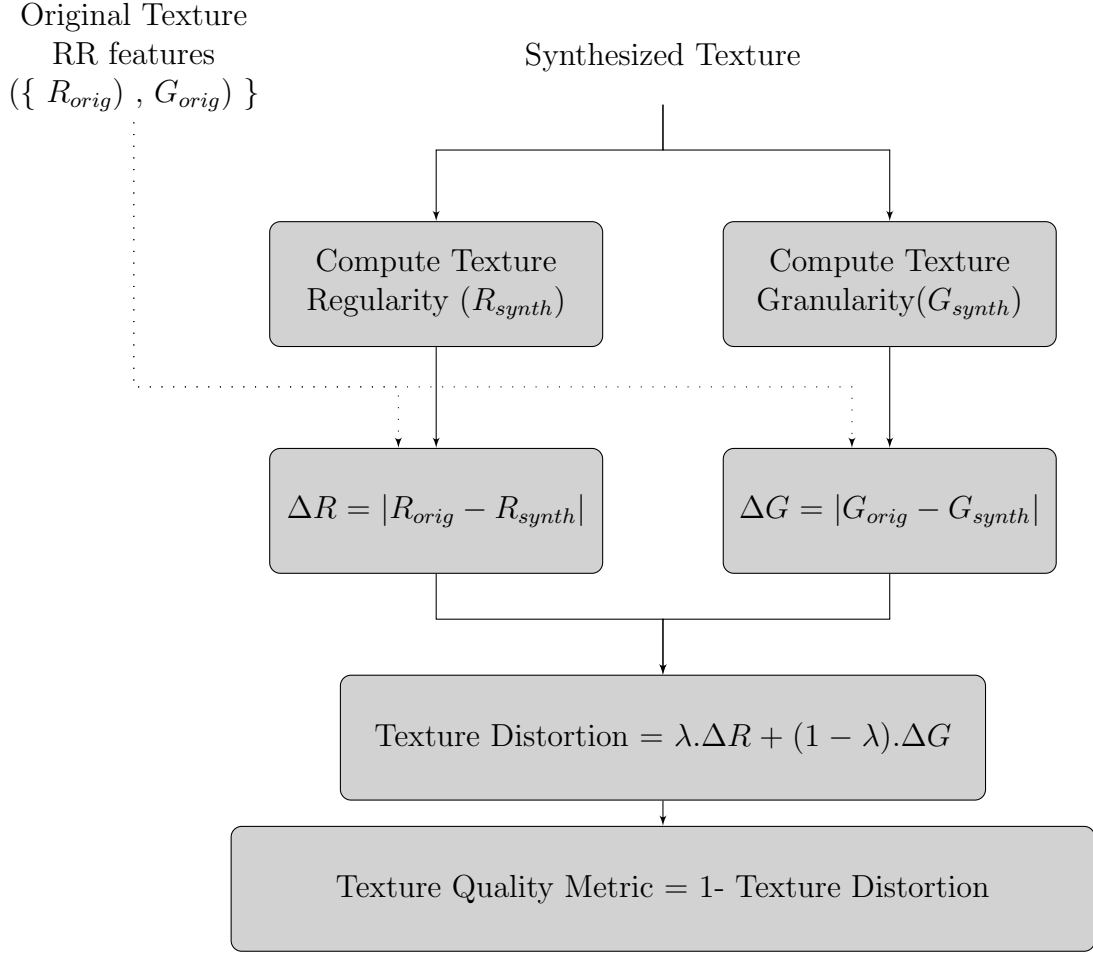


Figure 6.1: Block diagram of the proposed texture synthesis quality metric.

metric makes use of the amount of perceived texture regularity in the original and synthesized textures. The difference in regularities quantifies the loss in fidelity.

Texture Quality Metric:

The proposed reduced-reference texture quality metric makes use of the change in perceived regularity and the change in granularity of the synthesized textures compared to the original textures. The regularity and the granularity of the original texture are available as RR features for quality assessment. The perceived regularities of the original and the synthesized textures are computed using (4.16) as mentioned in the previous chapter. The regularity of the original texture is available

as a RR feature at the receiver. The difference of the perceived regularities between the original and synthesized textures is used as a measure of loss in fidelity. To accommodate both increase and decrease in regularity as a result of synthesis, an absolute value of the difference is computed as the delta-regularity score,  $\Delta R$ , as shown below:

$$\Delta R = |R_{orig} - R_{synth}| \quad (6.3)$$

where  $R_{orig}$  and  $R_{synth}$  are the perceived regularities of the original and synthesized textures, respectively. Since both the regularities are in the normalized (0,1) range,  $\Delta R \in (0, 1)$ . We propose that the higher the value of the delta-regularity score, the higher is the loss in fidelity of the synthesized texture or, equivalently, the lower is the perceived quality of the synthesized texture.

Similarly, the perceived granularities of the original and the synthesized textures are computed using (6.2). The granularity of the original texture is available as a RR feature at the receiver. The difference of the granularities between the original and synthesized textures is also used as a measure of loss in fidelity. An absolute value of the difference is computed as the delta-granularity score,  $\Delta G$ , to accommodate both increase and decrease in granularity as a result of synthesis as shown below:

$$\Delta G = |G_{orig} - G_{synth}| \quad (6.4)$$

where  $G_{orig}$  and  $G_{synth}$  are the granularities of the original and synthesized textures, respectively. Since both the granularities are in the normalized (-0.5,0.5) range,  $\Delta G \in (0, 1)$ . We propose that the higher the value of the delta-granularity score, the higher is the loss in fidelity of the synthesized texture or, equivalently, the lower is the perceived quality of the synthesized texture.

The perceived texture distortion is computed from the change in regularity  $\Delta R$  and the change in granularity  $\Delta G$  scores as follows:

$$TextureDistortion = \lambda.\Delta R + (1 - \lambda).\Delta G \quad (6.5)$$

The value of  $\lambda = 0.25$  was used in our implementation in order to maximize the correlation with subjective loss in fidelity.

The proposed RR Texture Quality Metric (TQM) for synthesized textures is computed from the texture distortion as follows.

$$TQM = 1 - TextureDistortion \quad (6.6)$$

### 6.3 SynTEX Database

A set of 16 textures were obtained from two databases, namely, the MIT Vistex database [56] and the Graph-Cut textures database [57]. The textures were chosen such that they together represented the regular, hybrid and irregular texture classes. The textures exhibit varying degrees of regularities in the visual properties of primitives, like placement, size, shape, color or orientation. Also, to understand the difference in the perceived regularity between fine granular and large sized primitives, textures from both these classes were chosen in the test set. For the patch-based graph-cut texture synthesis algorithm [59], seed texture regions were hand-picked from each of these textures and given as input. There are two versions of the graph-cut synthesis algorithm that differ in the cost function for boundary-matching while placing the successive patches [59]. Corresponding to the two versions, two sets of textures were synthesized. The seed size was just large enough to capture the periodicity along the X and Y directions. The synthesized texture dimension was set as 6 times the size of the original seed. A third set of textures was synthesized from the original textures using the Portilla’s method [7]. In the implementation, N=4 multi-resolution levels, K=4 orientations and 75 iterations were used. So in total,

a set of 48 textures were synthesized from the original 16 textures using the 3 texture synthesis algorithms. A subjective testing was conducted on the 48 synthesized textures. Fifteen subjects with normal to corrected vision participated in the subjective tests. Only the subjects who passed the 20-20 vision test and the color-vision tests were chosen for the subjective testing experiment. For each considered synthesis method, the synthesized textures along with the corresponding original textures were displayed side-by-side. To avoid scaling differences, only the central portions of the synthesized images having the same dimension as the original textures were displayed. The order of the image-pairs were randomized and displayed one after another to each subject. The subjects were asked to score the overall fidelity for each synthesized texture by comparing it to the original, using the ITU recommended five-scale score with 1(very-annoying) corresponding to lowest and 5(imperceptible) corresponding to highest fidelity. An average of the fidelity scores over all subjects for a considered texture gives the Mean Opinion fidelity Score (MOS) for a considered method. The standard deviation of the MOS over all subjects was within 1.5, for each of the 48 textures and within 1.0 for 43 out of 48 textures. The 48 synthesized textures along with their fidelity MOS constitute our Synthesized Textures (*SynTEX*) database.

#### 6.4 Simulation Results

In addition to the SynTEX database, two more texture synthesis databases were used for performance evaluation. The *QualTEX* database [77] contains a set of 20 textures synthesized from an original set of 10 textures, by using quantized and non-quantized Portilla’s parameters. The third database used in our performance analysis was the parametric quality assessment database [78] denoted by *SynthPQA* database in this work. The *SynthPQA* database consists of 567 textures synthesized using 9 algorithms from the original set of 63 textures obtained from the Brodatz

database [75]. The 9 different algorithms are denoted by *Alg1* to *Alg9* and they represent the texture synthesis algorithms in [61], [79], [80], [81], [7], [63], [82], [83] and [59], respectively. While *Alg3* and *Alg5* are parametric algorithms, the rest are all non-parametric approaches. In [62], a set of 42 textures representing 7 different perceptual texture clusters as mentioned in [84], was used for quality evaluation. In this work, 27 of the 42 textures mentioned in [84], were used for quality evaluation by eliminating textures with the following characteristics:

1. Textures without at least 2 primitives along the X and Y directions.
2. Textures whose DMOS scores were abnormally high.
3. Textures with brick-like flat regions for which the VA models fail.
4. Textures with very low contrast for which the VA models fail.

The final set of 27 original textures used from the *SynthPQA* database consists of *D2*, *D4*, *D5*, *D11*, *D12*, *D25*, *D29*, *D30*, *D31*, *D33*, *D35*, *D38*, *D40*, *D41*, *D42*, *D45*, *D46*, *D65*, *D74*, *D84*, *D92*, *D93*, *D97*, *D99*, *D101*, *D104* and *D108*.

The performance of the proposed Texture Quality Metric (TQM) to predict the perceptual quality of synthesized textures was compared against 7 popular full-reference algorithms for measuring image quality. These include MSSIM [68], CWS-SIM [69], WCWSSIM [70], STSIM [71], STSIM1 [72], STSIM2 [72] and Parametric Quality [62]. Each of these algorithms acted on a pair of original and synthesized textures and produced a quality metric. To account for extreme values at the ends of the testing range (very high and very low quality metric), each metric value  $M_i$  is transformed into a predicted MOS ( $MOS_{p_i}$ ) value using a four-parameter logistic function as suggested by VQEG [58]:

$$MOS_{p_i} = \frac{\beta_1 - \beta_2}{1 + e^{\left(M_i - \frac{\beta_3}{|\beta_4|}\right)}} + \beta_2 \quad (6.7)$$



Table 6.4: Performance evaluation on the QualTEX database.

Texture Regularity	PLCC	SROCC	RMSE	MAE	$P_{PLCC}$	$P_{SROCC}$
CWSSIM	0.400	0.423	0.137	0.094	0.081	0.063
WCWSSIM	0.476	0.402	0.137	0.094	0.034	0.079
MSSSIM	0.391	0.349	0.127	0.090	0.088	0.131
STSIM	0.382	0.486	0.137	0.094	0.097	0.030
STSIM1	-0.335	-0.422	0.137	0.094	0.149	0.064
STSIM2	-0.332	-0.464	0.137	0.094	0.152	0.039
Parametric Quality	0.639	0.688	0.105	0.064	0.002	0.001
Proposed TQM	0.911	0.550	0.056	0.031	0.000	0.012

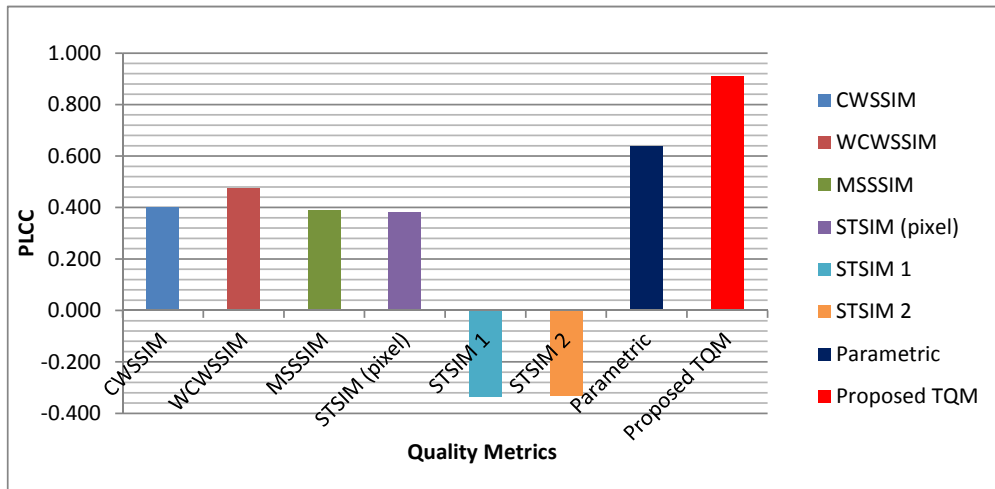


Figure 6.2: Performance of quality metrics on the QualTEX database for different texture synthesis algorithms.

The performance comparison of the proposed TQM with other quality metrics in quantifying the perceptual texture quality is shown through the Pearson Linear Correlation Coefficient (PLCC) and the Spearman Rank Order Correlation Coefficient (SROCC) between  $MOS_p$  and  $MOS$  in addition to the Root Mean Square Error (RMSE) and the Mean Absolute Error (MAE). The quality metrics were evaluated on the QualTEX database consisting of 20 textures synthesized using Portilla's

Table 6.5: Performance evaluation on the SynTEX database.

Texture Regularity	PLCC	SROCC	RMSE	MAE	$P_{PLCC}$	$P_{SROCC}$
CWSSIM	0.405	0.410	0.154	0.123	0.004	0.004
WCWSSIM	0.409	0.397	0.154	0.122	0.004	0.005
MSSSIM	-0.171	-0.171	0.169	0.142	0.245	0.246
STSIM	0.220	0.190	0.169	0.142	0.134	0.196
STSIM1	0.443	0.397	0.151	0.125	0.002	0.005
STSIM2	0.433	0.390	0.169	0.142	0.002	0.006
Parametric Quality	0.196	0.089	0.165	0.138	0.182	0.546
Proposed TQM	0.491	0.592	0.147	0.116	0.000	0.000

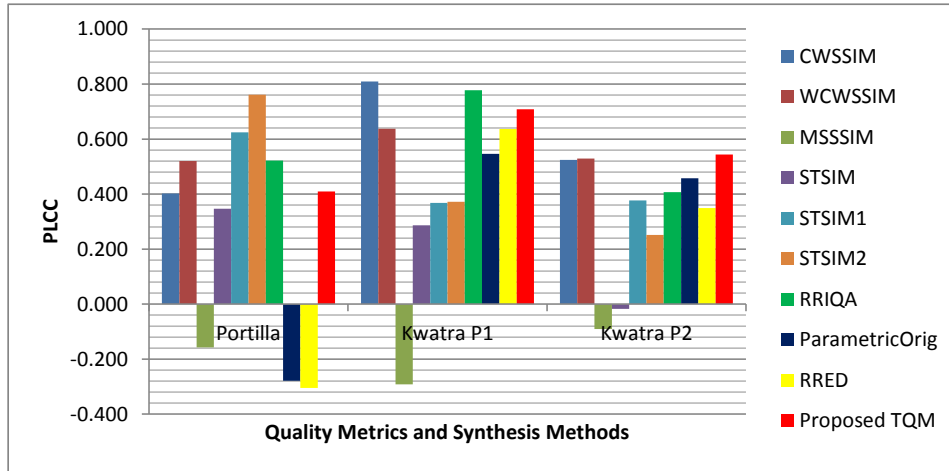


Figure 6.3: Performance of the quality metrics on the SynTEX database for different texture synthesis algorithms.

parametric synthesis as explained above. The proposed TQM outperformed all the state-of-the-art quality metrics in terms of PLCC, as shown in Fig. 6.2 and Table 6.4. The SROCC of the proposed TQM is the second best, lower than the Parametric Quality metric. The proposed RR metric required just 2 parameters, namely Regularity and Granularity, in contrast to hundreds of parameters involved in other image-pyramid based approaches [62].

Table 6.6: Performance evaluation on the Parametric Quality Assessment database.

<b>Texture Regularity</b>	<b>PLCC</b>	<b>SROCC</b>	<b>RMSE</b>	<b>MAE</b>	$P_{PLCC}$	$P_{SROCC}$
CWSSIM	0.146	0.194	0.235	0.190	0.032	0.004
WCWSSIM	0.167	0.190	0.235	0.190	0.014	0.005
MSSSIM	0.089	0.058	0.235	0.190	0.193	0.398
STSIM	0.261	0.263	0.235	0.190	0.000	0.000
STSIM1	0.143	0.221	0.235	0.190	0.036	0.001
STSIM2	0.268	0.278	0.235	0.190	0.000	0.000
Parametric Quality	0.639	0.428	0.181	0.144	0.000	0.000
Proposed TQM	0.546	0.484	0.197	0.157	0.000	0.000

Similarly, the performance comparison of the proposed TQM with other quality metrics was done on the SynTex database consisting of 48 textures synthesized using parametric and patch-based synthesis as explained above. As shown in Table 6.5, the proposed RR TQM has the highest correlation with MOS. The P-value of this correlation is less than 0.05 for both PLCC and SROCC. This proves the statistical significance of the correlation between the proposed TQM and synthesized texture quality. The performance of the quality metrics on the individual algorithms is shown in Fig. 6.3. It can be seen that the proposed quality metric has the highest performance for Kwatra’s Graph-Cut synthesis that employ P2 algorithm for boundary matching. It has a reasonably good performance on Kwatra’s P1 synthesis. However, the proposed approach uses just 2 RR parameters while other metrics are Full-Reference. It should be noted that the Parametric Quality metric [62] can be as well thought of as a RR metric employing hundreds of parameters.

On the *SynthPQA* database, the performance of the proposed RR TQM has a PLCC of 0.546 and a SROCC of 0.484 over all synthesis algorithms, as shown in Table 6.6. The performance of the proposed RR TQM, was the second-best in performance to the parametric quality metric suggested in [62]. However the proposed approach requires just 2 RR parameters per texture and hence is significantly more efficient than the metric in [62]. The performance of the quality metrics on the

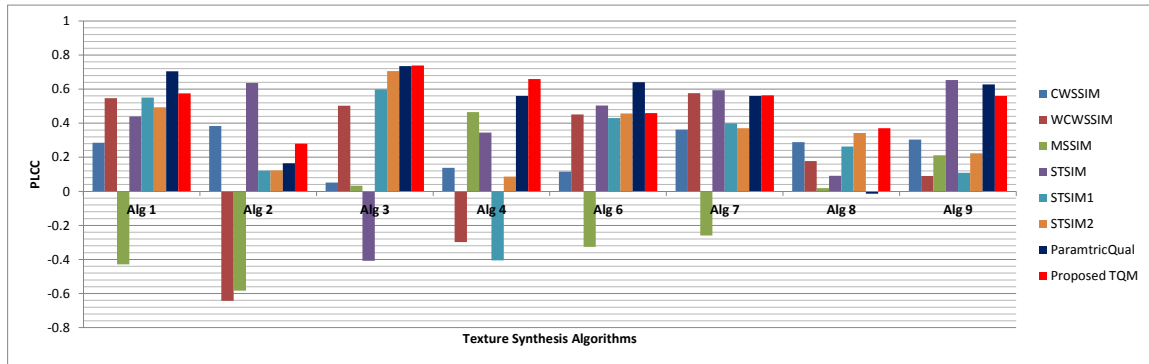


Figure 6.4: Performance of quality metrics on the SynthPQA database for different texture synthesis algorithms.

individual synthesis algorithms is shown in Fig. 6.4. It can be seen from Fig. 6.4, that all metrics perform poorly on textures synthesized using *Alg2* and *Alg8*, which leads to an overall poor score for correlation. Also, the proposed RR TQM performs as good as or even better than the Parametric Quality metric of [62] for *Alg2*, *Alg3*, *Alg4*, *Alg7* and *Alg8*.

EFFECT OF TEXTURE REGULARITY ON PERCEIVED TEXTURE  
DISTORTIONS

This chapter presents some of the effects of texture regularity on perceived distortions in textures. The distortions considered in this chapter are the artifacts due to texture compression and the visual degradation due to Gaussian blur on textures.

7.1 Effect of Texture Regularity on Perceived Compression Artifacts

This section presents the influence of texture regularity on the perceived quality of textures compressed through JPEG2000. It is shown through subjective testing that textures with different degrees of perceived regularity exhibit different degrees of vulnerability to compression artifacts, for the same degree of compression. Further, it is also shown through subjective and objective metrics that the regularity of textures directly affects the perceived loss in fidelity due to compression.

The popular video codecs like MPEG4, H.264 and HEVC use lossy video compression to enable the transmission of high volume video data through low-bandwidth networks. Lossy video compression involves discarding unimportant video data in order to achieve higher compression ratios compared to lossless video coding. The discarding of image data may lead to the appearance of visual artifacts, which can lead to a loss of fidelity. The perceived visual quality depends on the sensitivity of the Human Visual System (HVS) to the various visual artifacts introduced during the compression process. The contrast sensitivity function of the HVS indicates that the HVS is less sensitive to visual artifacts in high frequency components. Many perceptually motivated video codecs like [3], [4] and [5] exploit this fact by encoding regions of high spatial frequencies with a lower number of bits. Textures, because of their high spatial frequencies, are very good candidates for lowering the bit-rate at the cost of introducing imperceptible artifacts. The texture-based video codecs

achieve compression through either of the following approaches:

1. Texture synthesis at the decoder using a sample texture patch and synthesis parameters.
2. Heavily quantizing the texture regions compared to the non-texture regions.

Texture synthesis and texture synthesis based video codecs were reviewed in Chapter 5. Apart from the texture-based video codecs based on texture-synthesis, some schemes like [5] try to quantize the textures heavily compared to the non-textured regions and achieve a better rate-distortion (R-D) performance. The perceptual Macroblock (MB)-level Rate Control (RC) scheme in [85] uses a distortion measure weighted by a perceptual threshold considering luminance adaptation, texture masking, and skin detection in the Rate-Distortion (R-D) model used for obtaining the optimal quantization step size for each MB. The advantage of the rate-control based methods for exploiting texture irrelevancy is that they are backward compatible with existing standard video decoders. On the other hand, implementing texture synthesis based approaches involve modifying both the encoder and decoders to perform texture synthesis.

Most of the popular image and video compression standards like JPEG, MPEG and H.264 achieve data compression by a set of generic steps:

1. Prediction in spatial or temporal domain
2. Transformation of the signal to frequency domain using DCT or DWT
3. Quantization of the frequency domain data.



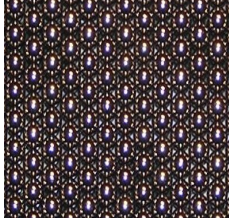


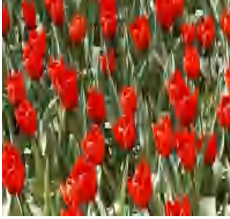



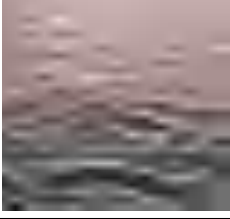


Typically, these operations together produce a set of artifacts on compressed images like color distortions, blockiness, ringing artifacts and blur. The quantization is a lossy operation due to which a large range of values is represented by a single

quantum value. The amount of noise introduced due to quantization is directly proportional to the step size. Since the HVS is less sensitive to visual artifacts in high frequency components compared to those in lower frequencies, a higher step size is used for higher frequencies. Since edges are predominantly high frequency components, the loss of edges due to quantization results in blurred images.

Textures in general contain a lot of edges and consume a lot of bits compared to uniform regions for compression. Often, textures occur in the background and are perceptually unimportant. So many video-codecs like [5] try to isolate texture regions and heavily compress them. But not all types of textures are equally vulnerable to compression artifacts. This is illustrated through the examples in Table 7.1. The original and the JPEG2000-compressed textures are shown in columns 1 and 2 of Table 7.1, respectively. The textures in column 2 were compressed to 1% of the original size with JPEG2000. As it can be observed from Table 7.1, the regular textures are more robust to compression while the irregular or random textures exhibit a much higher loss in structure due to JPEG2000 compression. As shown later in this section, the Mean Opinion Scores (MOS) obtained through subjective testing also indicate this. *Tile* and *Marbles* are regular textures and their corresponding MOS scores are 3.72 and 3.5, respectively. *Water* and *Misc* are irregular textures whose compressed images show a MOS of 1.22 and 1.61, respectively. This is because these random textures with very small-sized textural primitives contain a lot of weak high frequency coefficients which are heavily modified during the compression process. This greatly modifies the regularity and the structure of these textures, leading to a significant loss in fidelity.

The standard image and video codecs compress the data to a specified extent using a compression unit called the Rate-Control (RC) unit. The RC unit takes into account many factors like the target bit-rate, already consumed bits for a frame etc.,

Table 7.1: JPEG2000 Compression Artifacts on Textures.

Original Texture	JPEG2000 Compressed Texture	JPEG2000 Fidelity MOS	Regularity Metric
		3.72	0.69
		3.50	0.81
		2.78	0.39
		2.83	0.33
		1.22	0.20
		1.61	0.19



and computes a QP for a frame and ultimately for a Macro-block or a block. Even though many RC algorithms are content adaptive like [85], none of the algorithms take texture regularity into account. We propose that texture regularity has a strong influence on the perceived compression artifacts and must be taken into account while computing the QP for the texture block. This can potentially lead to a higher perceived quality through a more efficient distribution of bits.

Rate-Control algorithms on video encoders aim to achieve the best perceptual quality by redistributing the bits optimally to achieve the best visual quality at any given bitrate. Since textures contain high frequency data, the textural artifacts are less perceptible compared to non-textured regions. So the RC algorithms tend to heavily quantize the textural regions. The degradation in the perceived image fidelity due to the blur resulting from quantization, is not the same for all class of textures but varies from regular to irregular textures. Therefore, it may be possible to vary the quantization step size based on the regularity of the textures for a higher video fidelity.

This is illustrated through the textures in Table 7.1. The original texture is shown in column 1 while the reconstructed texture after JPEG2000 compression is shown in column 2 of Table 7.1. The fidelity Mean Opinion Score for the reconstructed texture is shown in column 3 and the proposed texture regularity metric is given in column 4 for the original non-compressed texture. As it can be observed, the compression artifacts are much more pronounced on irregular textures as compared to regular textures.

## 7.2 Simulation Results

A subjective testing was conducted on 9 textures from the texture regularity database of 21 textures constructed from two databases, namely the MIT Vistex database [56] and the Graph-Cut texture synthesis database [57] mentioned in Chapter 4. Ten

Table 7.2: Mean Opinion Scores (MOS) for JPEG2000 compressed images along with MOS and objective scores for regularity of original textures.

<b>Texture</b>	<b>JPEG2000 Fidelity MOS</b>	<b>Average Property Regularity MOS</b>	<b>Average Overall Regularity MOS</b>	<b>Texture Regularity Metric</b>
Tile	3.72	2.83	3.00	0.69
Marbles	3.50	2.81	3.00	0.81
Fabric	2.67	2.71	2.81	0.56
Tulips	2.78	2.01	2.00	0.39
Fresh-blue-berries	2.83	1.93	2.00	0.33
Lobelia	2.39	1.78	2.00	0.32
North-beach	1.56	1.28	1.31	0.22
Misc	1.22	1.18	1.31	0.23
Water	1.61	1.16	1.25	0.20

Table 7.3: Correlation of fidelity MOS of JPEG2000-compressed images with subjective and objective regularity scores.

<b>Texture Regularity</b>	<b>PLCC</b>	<b>SROCC</b>	<b>RMSE</b>	<b>MAE</b>
Average Property Regularity MOS	92.56	88.33	0.246	0.168
Average Overall Regularity MOS	91.95	85.83	0.266	0.195
Proposed Regularity Metric	91.95	85.00	0.082	0.061

subjects with normal to corrected vision participated in the subjective tests. The textures were equally distributed amongst the broad classes of regular, irregular or hybrid textures. The textures were compressed using JPEG2000, implemented in [86], to 1% of their original size and reconstructed back. The reconstructed images exhibit compression artifacts. The decompressed textures along with the corresponding original texture were displayed side-by-side in a subjective testing experiment. The order of the image pairs was randomized and these were displayed one after another to each subject. The subjects were asked to score the overall fidelity for each compressed texture by comparing it to the original, using a five-scale score with

1 corresponding to lowest and 5 to highest. An average of the fidelity scores over all 10 subjects for a considered texture gives its Mean Opinion JPEG2000 fidelity Score. The JPEG2000 MOS for the various textures are given in column 2 of Table 7.2. The Average Property Regularity MOS and Overall Regularity MOS, available from the texture database (as described in Chapter 4), are shown in columns 3 and 4 of Table 7.2, respectively. The no-reference perceptual texture regularity metric proposed in Chapter 4 is given in column 5. To account for extreme values at the ends of the testing range (very high and very low regularity), each metric value  $M_i$  is transformed into a predicted MOS ( $MOS_{p_i}$ ) value using a four-parameter logistic function as mentioned in Chapter 4. The correlation of the fidelity MOS with the subjective and objective regularity metrics is quantified through the Pearson Linear Correlation Coefficient (PLCC) and the Spearman Rank Order Correlation Coefficient (SROCC) as given in Table 7.3. For the considered nine textures, the Average Property Regularity MOS had a PLCC and SROCC of 92.6% and 88.3% respectively with the JPEG2000 image fidelity MOS. The Overall Regularity MOS had a correlation of 92.0% and 85.8% with the JPEG2000 image fidelity MOS. The texture regularity metric proposed in Chapter 4 also had a high correlation with the JPEG2000 image fidelity MOS. The PLCC and SROCC for the texture regularity metric were 92% and 85%, respectively when correlating with the image fidelity of textures compressed by JPEG2000.

The effect of texture regularity on the amount of perceived artifacts in compressed textures is analyzed. It is proposed that irregular textures with small primitives are more susceptible to perceived artifacts than regular textures for the same degree of compression and that texture regularity should be taken into consideration in bit allocation and rate control methods. Subjective testing on JPEG2000-compressed

textures validates this proposition. Including the texture degree of regularity may help in predicting the degradation of texture quality due to compression.

### 7.3 Effect of Texture Regularity on perceived Blur

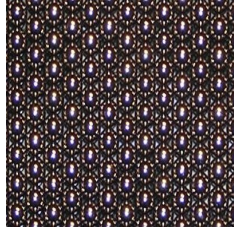
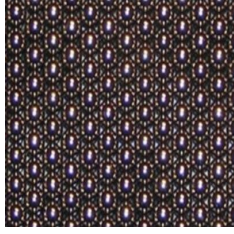
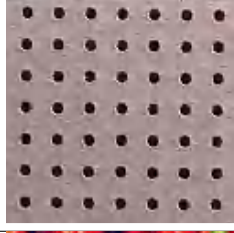
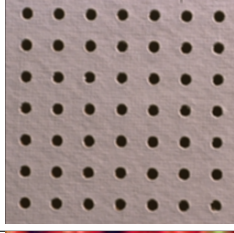

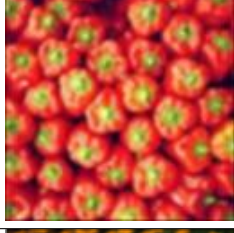
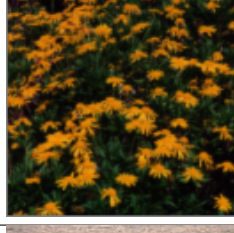
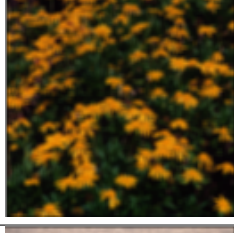
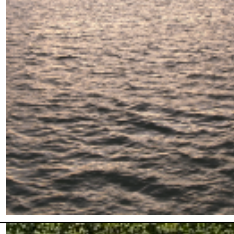
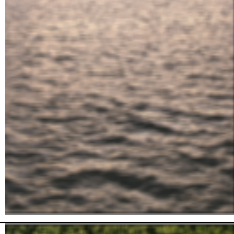

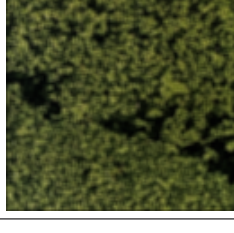
Blur in textures can be due to many reasons like blur due to the camera lens, motion blur due to camera motion during capture, and blurry artifacts as a result of quantizing the high frequency coefficients. The perceived blur is often quantified as the spread of edges in the direction of the gradient.

Regular textures have clearly demarcated primitives and are associated with strong edges. Since the primitive sizes are significant, the spread of the edges does not change the structure or the spatial periodicity of the primitives. Also the regular pattern grabs the human attention and hence reduces the perceptibility of blur. Amongst the regular textures, the perceived blur is higher in textures with a high local contrast. This is because the Just Noticeable Blur (JNB) width tends to decrease with increase in contrast [87].

The perceived blur is relatively high in irregular textures having very small-sized primitives. This is because the small sized primitives (high granularity) in the irregular textures lose their shape and structure completely upon convolving with a lowpass filter such as a 2D Gaussian kernel. This leads to a larger perceived blur.

The correlation of the perceived blur with texture regularity is illustrated through a set of textures in Table 7.4. The original textures are shown in column 1 of Table 7.4. The blurred textures obtained by convolving the corresponding original textures with a Gaussian of standard deviation of  $\sigma = 3.0$  is shown in column 2 of Table 7.4. The proposed Texture Regularity Metric score is given in column 3 of the table. It can be observed that the perceived blur increases with decreasing texture regularity. This is because the very small-sized primitives in these irregular

Table 7.4: Gaussian Blur ( $\sigma = 3.0$ ) on Textures.

Original Texture	Blurred Texture	Regularity Metric
		0.81
		0.69
		0.33
		0.39
		0.20
		0.19

textures lose their form and structure as a result of the blur. This directly affects the placement periodicity of the primitives and result in a high degree of perceived blur. The regular textures on the other hand, are characterized by strong edges at the primitive boundaries and the primitives are medium or large in size. So the shape, structure or the placement periodicity of the primitives in regular textures are not affected significantly due to blur. This fact can be exploited in the design of an improved blur metric for textures based on the texture regularity metric that is proposed as a part of this work.

## Chapter 8

### Conclusion

This work contributes to the areas of visual saliency on textures, texture regularity analysis and applications like texture-based image coding and texture quality assessment. This chapter summarizes the main contributions of this work and some directions for future research.

#### 8.1 Summary of Contributions

The contributions of this thesis are summarized below.

- A no-reference perceptual texture regularity metric is proposed as part of this work. This is the first ever texture regularity metric that takes visual perception into account. The proposed metric is based on the fact that regularity in textural primitives manifests itself as regularity in the visual saliency map. The texture regularity is quantified through a texture similarity score and a spatial distribution score. The proposed regularity metric is compared with one of the state-of-the-art regularity metrics [43] and is shown to have a higher correlation with subjective testing results. The proposed regularity metric is also found to be fairly robust to small geometric and photometric transformations
- The performance of the visual attention models on textures is evaluated. Though there are a number of visual saliency databases for natural images, this is the first work to build the ground-truth eye-tracking data for images with exclusive texture content. Nine popular VA models were evaluated using the acquired ground-truth data.
- An adaptive texture synthesis algorithm is proposed. This algorithm, selects the appropriate method for texture synthesis based on the regularity of the

original texture. The texture regularity metric proposed in this work is used to quantify the texture regularity.

- A reduced reference texture quality metric (RR TQM) is developed and presented in this work for assessing the quality of synthesized textures. The RR TQM uses the change in regularity and change in granularity between the original and synthesized textures to estimate the amount of loss in fidelity of the synthesized textures. The proposed texture regularity metric is employed for computing the regularity score. A novel granularity score that quantifies the amount of perceived fineness or coarseness of textures is also proposed in this work. The proposed RR TQM uses just 2 parameters and is shown to have a high degree of correlation with the Mean Opinion Scores on 3 different texture databases. A new texture synthesis quality database named *SynTEX* is also a contribution of this work. This database was constructed by synthesizing a set of 16 original textures obtained from [56] and [57], through parametric [7] and graph-cut based texture synthesis [59] methods. The database also includes the MOS for the fidelity of the synthesized textures.
- The final contribution of this work is in establishing the influence of texture regularity on distortions like texture compression artifacts and texture blur. It is proposed that irregular textures with small primitives are more susceptible to perceived artifacts than regular textures for the same degree of compression and that texture regularity should be taken into consideration in bit allocation and rate control methods. Subjective testing on JPEG2000-compressed textures validates this proposition. It is also proposed that the perceived blur increases with decreasing texture regularity. The loss in form and structure of small-sized primitives in blurred irregular textures, directly affects the placement periodicity of the primitives and result in a high degree of perceived blurriness.



## 8.2 Future Research Directions

The efforts to apply visual saliency for computing texture regularity are still in their inception. So there are many unsolved problems which can be pursued for future research. Some of the potential directions for future research could be the following:

- Improving the limitations of the proposed regularity metric for detecting shape and direction irregularities.
- Understanding the effect of texture regularity on visual attention models.
- Improving the visual attention models that are able to more closely predict the effect of texture regularity on human visual saliency. For example, the regularities in grid-like patterns cannot be currently detected due to flat regions without high contrast primitive centers.
- Improving the visual quality assessment of textures based on the regularity of textures.
- Improving the texture enhancement and restoration techniques based on the knowledge of the original undegraded texture's regularity.
- Designing an improved blur metric for textures, that takes into account the regularity of the textures.
- Designing visual texture compression systems that adaptively perform texture synthesis or compression based on texture regularity.

## REFERENCES

- [1] R. M. Haralick, “Statistical and structural approaches to texture,” *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.
- [2] B. Julesz and J. R. Bergen, “Textons, the fundamental elements in preattentive vision and perception of textures,” in *Readings in computer vision: issues, problems, principles, and paradigms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1987, pp. 243–256.
- [3] M. Bosch, F. Zhu, and E. J. Delp, “Segmentation-based video compression using texture and motion models,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1366–1377, 2011.
- [4] F. Zhang and D. R. Bull, “A parametric framework for video compression using region-based texture models,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1378–1392, 2011.
- [5] J. Balle, A. Stojanovic, and J. R. Ohm, “Models for static and dynamic texture synthesis in image and video compression,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1353–1365, 2011.
- [6] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, “Texture optimization for example-based synthesis,” *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 795–802, 2005.
- [7] J. Portilla and E. P. Simoncelli, “A parametric texture model based on joint statistics of complex wavelet coefficients,” *International Journal of Computer Vision*, vol. 40, no. 1, pp. 49–70, 2000.
- [8] M. N. Do and M. Vetterli, “Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance,” *IEEE Transactions on Image Processing*, vol. 11, no. 2, pp. 146–158, 2002.
- [9] Y. D. Chun, N. C. Kim, and I. H. Jang, “Content-based image retrieval using multiresolution color and texture features,” *IEEE Transactions on Multimedia*, vol. 10, no. 6, pp. 1073–1084, 2008.
- [10] H. Y. T. Ngan, G. K. H. Pang, and N. H. C. Yung, “Performance evaluation for motif-based patterned texture defect detection,” *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 1, pp. 58–72, 2010.

- [11] H. Y. T. Ngan and G. K. H. Pang, “Regularity analysis for patterned texture inspection,” *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 1, pp. 131–144, 2009.
- [12] L. Liu and P. W. Fieguth, “Texture classification from random features,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 3, pp. 574–586, 2012.
- [13] J. Harel, C. Koch, and P. Perona, “Graph-based visual saliency,” in *Advances in Neural Information Processing Systems 19*. MIT Press, 2007, pp. 545–552.
- [14] S. Marat, T. H. Phuoc, L. Granjon, N. Guyader, D. Pellerin, and A. Gurin-Dugu, “Modelling spatio-temporal saliency to predict gaze direction for short videos,” *International Journal of Computer Vision*, vol. 82, no. 3, pp. 231–243, 2009.
- [15] W. Einhäuser, M. Spain, and P. Perona, “Objects predict fixations better than early saliency,” *Journal of Vision*, vol. 8, no. 14, pp. 1–26, 2008.
- [16] L. Zhang, M. H. Tong, T. K. Marks, H. Shan, and G. W. Cottrell, “Sun: A Bayesian framework for saliency using natural statistics,” *Journal of Vision*, vol. 8, no. 7, pp. 1–20, 2008.
- [17] H. Kaizer, “A quantification of textures on aerial photographs,” Boston University Research Laboratories, Boston University, Boston, MA, Tech. Rep. Tech. Note 121, AD 69484, 1955.
- [18] Y. Huang and K. L. Chan, “Texture decomposition by harmonics extraction from higher order statistics,” *IEEE Transactions on Image Processing*, vol. 13, no. 1, pp. 1–14, 2004.
- [19] R. Kinderman and S. L. Snell, *Markov random fields and their applications*. American mathematical society, 1980.
- [20] J. Besag, “Spatial interaction and the statistical analysis of lattice systems,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 36, no. 2, pp. 192–236, 1974.
- [21] H. Derin and H. Elliott, “Modeling and segmentation of noisy and textured images using Gibbs random fields,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 1, pp. 39–55, 1987.

- [22] G. R. Cross and A. K. Jain, “Markov random field texture models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, no. 1, pp. 25–39, 1983.
- [23] M. Liao, J. Qin, and Y. Tan, “Texture classification and segmentation using simultaneous autoregressive random model,” in *IEEE Symposium on Computer-Based Medical Systems*, 1992, pp. 398–401.
- [24] R. Chellappa, S. Chatterjee, and R. Bagdazian, “Texture synthesis and compression using Gaussian Markov random field models,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-15, no. 2, pp. 298–303, 1985.
- [25] J. Mao and A. Jain, “Texture classification and segmentation using multiresolution simultaneous autoregressive models,” *Pattern Recognition*, vol. 25, no. 2, pp. 173–188, 1992.
- [26] K. Falconer, *Fractal Geometry: Mathematical Foundations and Applications*. Wiley, 2003.
- [27] B. B. Mandelbrot, *The Fractal Geometry of Nature*. New York: W. H. Freeman and Co., 1983.
- [28] A. P. Pentland, “Fractal-based description of natural scenes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 6, pp. 661–674, 1984.
- [29] S. S. Chen, J. M. Keller, and R. M. Crownover, “On the calculation of fractal features from images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1087–1090, 1993.
- [30] S. Peleg, J. Naor, R. Hartley, and D. Avnir, “Multiple resolution texture analysis and classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 4, pp. 518–523, 1984.
- [31] J.-C. Liu, W.-L. Hwang, and M.-S. Chen, “Estimation of 2-D noisy fractional brownian motion and its applications using wavelets,” *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1407–1419, 2000.
- [32] J. Malik and P. Perona, “Preattentive texture discrimination with early vision mechanisms,” *Journal of the Optical Society of America*, vol. 7, pp. 923–932, 1990.

- [33] M. Unser and M. Eden, “Nonlinear operators for improving texture segmentation based on features extracted by spatial filtering,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 20, no. 4, pp. 804–815, 1990.
- [34] T. R. Reed and H. Wechsler, “Segmentation of textured images and Gestalt organization using spatial/spatial-frequency representations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 1–12, 1990.
- [35] M. R. Turner, “Texture discrimination by Gabor functions,” *Biological Cybernetics*, vol. 55, no. 2-3, pp. 71–82, 1986.
- [36] M. Clark, A. C. Bovik, and W. S. Geisler, “Texture segmentation using Gabor modulation/demodulation,” *Pattern Recognition Letters*, vol. 6, pp. 261–267, 1987.
- [37] M. Tuceryan and A. K. Jain, “Texture segmentation using Voronoi polygons,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 2, pp. 211–216, 1989.
- [38] V. Starovoitov, S.-Y. Jeong, and R.-H. Park, “Texture periodicity detection: features, properties, and comparisons,” *IEEE Transactions on Systems, Man and Cybernetics*, vol. 28, no. 6, pp. 839–849, 1998.
- [39] F. Liu and R. Picard, “Periodicity, directionality, and randomness: Wold features for image modeling and retrieval,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 722–733, 1996.
- [40] D. Chetverikov, “Pattern regularity as a visual key,” in *Proc. BMVC*. British Machine Vision Association, 1998, pp. 3.1–3.10.
- [41] A. Atto and Y. Berthoumieu, “How to perform texture recognition from stochastic modeling in the wavelet domain,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 4320–4323.
- [42] X. Liu and D. Wang, “Texture classification using spectral histograms,” *IEEE Transactions on Image Processing*, vol. 12, no. 6, pp. 661–670, 2003.
- [43] D. Chetverikov, “Pattern regularity as a visual key,” *Image and Vision Computing*, vol. 18, no. 12, pp. 975–985, 2000.

- [44] J. Zujovic, T. Pappas, and D. Neuhoff, “Structural similarity metrics for texture analysis and retrieval,” in *IEEE International Conference on Image Processing*, 2009, pp. 2225–2228.
- [45] M. Mancas, C. Mancas-Thillou, B. Gosselin, and B. Macq, “A rarity-based visual attention map - application to texture description,” in *IEEE International Conference on Image Processing*, 2006, pp. 445–448.
- [46] A. Borji, D. N. Sihite, and L. Itti, “Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study,” *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 55–69, 2013.
- [47] D. Salvucci and J. Goldberg, “Identifying fixations and saccades in eye-tracking protocols,” in *Proceedings of the Symposium on Eye Tracking Research & Applications*, 2000, pp. 71–78.
- [48] O. Komogortsev and A. Karpov, “Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades,” *Behavior Research Methods*, vol. 45, no. 1, pp. 203–215, 2013.
- [49] U. Rajashekar, I. van der Linde, A. Bovik, and L. Cormack, “Gaffe: A gaze-attentive fixation finding engine,” *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 564–573, 2008.
- [50] A. Garcia-Diaz, X. R. Fdez-Vidal, X. M. Pardo, and R. Dosil, “Saliency from hierarchical adaptation through decorrelation and variance normalization,” *Image and Vision Computing*, vol. 30, no. 1, pp. 51–64, 2012.
- [51] N. D. B. Bruce and J. K. Tsotsos, “Saliency attention and visual search: An information theoretic approach,” *ARVO Journal of Vision*, vol. 9, no. 3, pp. 1–24, 2009.
- [52] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, “Frequency-tuned salient region detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1597–1604.
- [53] X. Hou and L. Zhang, “Dynamic visual attention: searching for coding length increments,” in *NIPS*. Curran Associates, Inc., 2008, pp. 681–688.

- [54] M. Mancas, “Computational attention: Modelisation and application to audio and image processing,” in *PhD Thesis*. Mons, Belgium: Universite de Mons, 2007.
- [55] A. B. Watson, “Probability summation over time,” *Vision Research*, vol. 19, pp. 515–522, 1979.
- [56] “MIT texture database.” [Online]. Available: <http://vismod.media.mit.edu/vismod/imagery/VisionTexture>
- [57] “Graph-Cut textures database.” [Online]. Available: <http://www.cc.gatech.edu/cpl/projects/graphcuttextures>
- [58] *Final report from the Video Quality Experts Group on the validation of objective models of video quality assessment*, VQEG Std., 2000. [Online]. Available: <http://www.its.bldrdoc.gov/vqeg/projects/frtv-phase-i/frtv-phase-i.aspx>
- [59] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobick, “Graphcut textures: image and video synthesis using graph cuts,” *ACM SIGGRAPH*, vol. 22, no. 3, pp. 277–286, 2003.
- [60] D. J. Heeger and J. R. Bergen, “Pyramid-based texture analysis/synthesis,” in *Proceedings of the 22nd annual conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH, 1995, pp. 229–238.
- [61] A. A. Efros and T. K. Leung, “Texture synthesis by non-parametric sampling,” in *Proceedings of International Conference on Computer Vision- Volume 2*, ser. ICCV ’99, 1999, pp. 1033–1038.
- [62] D. S. Swamy, K. J. Butler, D. M. Chandler, and S. S. Hemami, “Parametric quality assessment of synthesized textures,” *Proceedings of the SPIE*, vol. 7865, 2011.
- [63] L.-Y. Wei and M. Levoy, “Fast texture synthesis using tree-structured vector quantization,” in *Proceedings of the 27th annual conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’00, 2000, pp. 479–488.
- [64] X. Tong, J. Zhang, L. Liu, X. Wang, B. Guo, and H.-Y. Shum, “Synthesis of bidirectional texture functions on arbitrary surfaces,” in *Proceedings of the 29th annual conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’02, 2002, pp. 665–672.

- [65] E. Praun, A. Finkelstein, and H. Hoppe, “Lapped textures,” in *Proceedings of ACM SIGGRAPH*, 2000, pp. 465–470.
- [66] “Implementation of Portilla and Simoncelli’s parametric texture synthesis.” [Online]. Available: <http://www.cns.nyu.edu/eero/texture/>
- [67] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [68] Z. Wang, E. P. Simoncelli, and A. C. Bovik, “Multiscale structural similarity for image quality assessment,” in *IEEE Asilomar Conference on Signals, Systems and Computers*, vol. 2, 2003, pp. 1398–1402.
- [69] Z. Wang and E. P. Simoncelli, “Translation insensitive image similarity in complex wavelet domain,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, 2005, pp. 573–576.
- [70] A. Brooks, X. Zhao, and T. Pappas, “Structural similarity quality metrics in a coding context: Exploring the space of realistic distortions,” *IEEE Transactions on Image Processing*, vol. 17, no. 8, pp. 1261–1273, 2008.
- [71] X. Zhao, M. G. Reyes, T. N. Pappas, and D. L. Neuhoff, “Structural texture similarity metrics for retrieval applications,” in *IEEE International Conference on Image Processing*, 2008, pp. 1196–1199.
- [72] J. Zujovic, T. N. Pappas, and D. L. Neuhoff, “Structural texture similarity metrics for image analysis and retrieval,” *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2545–2558, 2013.
- [73] Q. Li and Z. Wang, “Reduced-reference image quality assessment using divisive normalization-based image representation,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 3, no. 2, pp. 202–211, 2009.
- [74] A. Rehman and Z. Wang, “Reduced-reference SSIM estimation,” in *IEEE International Conference on Image Processing*, 2010, pp. 289–292.
- [75] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*. New York: Dover, 1996.



- [76] C. Q. Zhan and L. J. Karam, "Wavelet-based adaptive image denoising with edge preservation," in *IEEE International Conference on Image Processing*, 2003, pp. 97–100.
- [77] "QualTex: A quality assessment database for textures." [Online]. Available: <https://ivulab.asu.edu/databases/qualtex>
- [78] D. S. Swamy, "Quality assessment of synthesized textures," Master's thesis, Oklahoma State University, Stillwater, OK, USA, 2011.
- [79] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *ACM SIGGRAPH*, 2001, pp. 341–346.
- [80] B. Galerne, Y. Gousseau, and J. M. Morel, "Random phase textures: Theory and synthesis," *IEEE Transactions on Image Processing*, vol. 20, no. 1, pp. 257–267, 2011.
- [81] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin, "Image analogies," in *ACM SIGGRAPH*, 2001, pp. 327–340.
- [82] S. Lefebvre and H. Hoppe, "Parallel controllable texture synthesis," in *ACM SIGGRAPH*, 2005, pp. 777–786.
- [83] P. Harrison, "Image texture tools," in *PhD Thesis*. Victoria, Australia: Monash University, 2005.
- [84] B. Balas, "Attentive texture similarity as a categorization task: Comparing texture synthesis models," *Pattern Recognition*, vol. 41, no. 3, pp. 972–982, 2008.
- [85] X. Yang, W. Lin, Z. Lu, X. Lin, S. Rahardja, E. Ong, and S. Yao, "Rate control for videophone using local perceptual cues," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 4, pp. 496–507, 2005.
- [86] "JasPerlibrary : an open source JPEG2000 codec." [Online]. Available: <http://www.ece.uvic.ca/frodo/jasper/>
- [87] R. Ferzli and L. J. Karam, "A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB)," *IEEE Transactions on Image Processing*, vol. 18, no. 4, pp. 717–728, 2009.