TWO DIMENSIONAL RANDOM PATTERNS

By

CHAKRADHARA REDDY CHINTHAPANTI

Bachelor of Science in Computer Science and Engineering

Vellore Institute of Technology

Vellore, Tamil Nadu

2008

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
July, 2011

TWO DIMENSIONAL RANDOM PATTERNS

Thesis Approved:

Dr. Subhash Kak

Thesis Adviser

Dr. Johnson P. Thomas

Dr. David Cline

Dr. Mark E. Payton

Dean of the Graduate College

ACKNOWLEDGMENTS

It gives me immense pleasure and joy in acknowledging all those people, who played an important role in the completion of my thesis work as well as my graduation.

I owe my deepest gratitude and respect to my thesis advisor, Dr. Subhash Kak, without whom this thesis is impossible. He guided me throughout the period and inspired me a lot. It was his patience and coolness which made me to work efficiently and complete my thesis effectively. New ideas were always discussed and his guidance is the key in exploring this approach.

I would also like to thank Dr. Rathin Sarathy, Ardmore Professor in MIS, under whom I worked for almost one year. His guidance was very helpful in so many ways. I am involved in an interesting project under him.

I would also like to thank Dr. Johnson P. Thomas and Dr. David Cline for their valuable feedback which helped me to lead my thesis work in the right way.

I would like to thank all of my friends for their constant support and help which made the OSU and Stillwater, a memorable place to live and study.

Finally, I would like to dedicate my thesis to my parents, Chinthapanti Somasekhara Reddy and Bachu Ramasubbamma, without whom, nothing could have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

PN sequences and Decimal sequences

A pseudo-random binary sequence (PN sequence), a sequence of binary digits of length 2^k-1, is also called a maximal-length shift-register sequence [18]. PN sequences are widely used in cryptography, signal design, scrambling, fault detections and simulations (e.g., Monte Carlo methods) [9]. A decimal sequence (d-sequence) is nothing but a decimal expansion of a rational number which is represented in a sequence of digits. The d-sequence generated from a particular number can be periodic, irregular or it may terminate. The papers by Kak [1] - [6] give the theory of d-sequences and their properties.

A pseudo random sequence can be represented as a d-sequence of a rational number. Hence, d-sequences are also used in the place of pseudo random sequences. A binary d-sequence is generated by using the formula, $a(i) = 2^i \bmod p \bmod 2$, where p is a prime number and a maximum-length sequence is generated with period p-1, when 2 is a primitive root of p. These sequences are periodic and therefore examining the sequence in one period is enough to get to know the random properties. D-sequences cannot be used directly in computationally secure random number generator (RNG) applications. The reason being it is easy to find i given $\log_2 p$ bits of a(i) [2]. Hence by adding two or more different d-sequences (obtained by using primes p1,

p2…) mod 2, non-linearity is introduced in the generation of random sequence [6]. The resulting sequence is a better option to be used as random sequence than the previous one.

Coming back to the problem of randomness, pseudo random sequences and decimal sequences appear to be random but they are completely deterministic. The physical nature of randomness is discussed in [19]-[22]. The question of randomness defined in terms of the complexity of the algorithm needed to generate it is given in [23]. Random sequences for scrambling and for two-dimensional patterns are given in [7], [8].

The pseudo-random sequence appears to be random in the sense that the binary values and groups or runs of the same binary value occur in the sequence in the same proportion they would if the sequence were being generated based on a fair "coin tossing" experiment. In the experiment, each head could result in one binary value and a tail the other value. The PN sequence appears to have been generated from such an experiment. The sequence is not truly random in that it is completely determined by a relatively small set of initial values. It was also shown in [2] that it is easy to find i given $\log_2 p$ bits of a(i), therefore, d-sequences cannot be directly used in RNG applications. Further studies tried to increase the period but still the decimal sequence would be generated from the prime reciprocals. This thesis presents a new approach to generate random sequence from Delaunay triangulation or Voronoi diagram drawn from random points which would be more random in nature. The randomness is shown by performing autocorrelation function and diehard tests.

Two dimensional random patterns

Some applications call for two dimensional (2D) random sequences. Two dimensional random arrays and patterns have applications in a variety of areas including scrambling and fault detection. MacWilliams and Sloane [9] considered a n1 × n2 array, where n1 an n2 are relatively prime numbers and they used pseudo-random sequences to produce the two dimensional array.

Two dimensional patterns are basic to visual perception and it is not known how exactly such patterns are coded and recalled [15], although there is evidence that coding is unary in certain situations for one-dimensional patterns [16],[17]. We can also replace pseudo-random sequences by prime reciprocal sequences which gives the same results. By using the idea of prime reciprocals, generalization to two dimensional patterns by using two dimensional polynomials rather than primes is also studied [8].

In our approach, random sequences generated from Delaunay triangulation or Voronoi diagram [10] - [13] drawn from random points are used to generate the 2D random patterns. It is proposed that 2D patterns would be more randomized when generated using this approach. The 2D patterns may be generated using any number of base random sequences. The autocorrelation tests and diehard tests [14] validate the randomness of the pattern.

Delaunay triangulation and Voronoi diagrams

In brief, a Voronoi diagram is a special kind of decomposition of a metric space determined by distances to a specified discrete set of objects in the space, e.g., by a discrete set of points. It is obtained by drawing the bisectors for the two nearest points and the vertex of the generated polygon would be the center of the circle on which the corresponding closest points would be lying. In this thesis, the edges which have their length to infinity and which have their intersection outside the working plane are restricted to the working plane by clipping the polygons accordingly as shown in figure 1.

A Delaunay triangulation for a set of points in the plane is a triangulation such that no point in the set of points considered would lie inside the circumcircle of any triangle in the triangulation. Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation and they tend to avoid skinny triangles. The Voronoi diagram and

Delaunay triangulation are called duals as one can be generated from the other. A Delaunay triangulation generated for 20 random points is shown in figure 2.

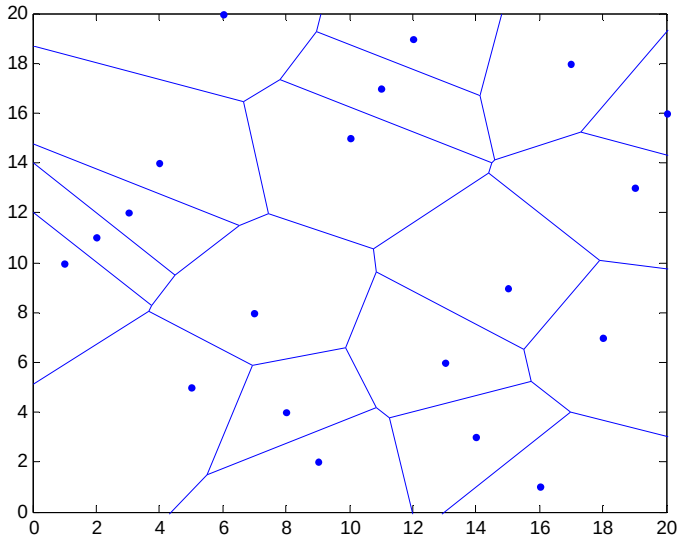

Figure 1. Clipped voronoi diagram for 20 random points (working plane is (0, 0, 20, 20)).

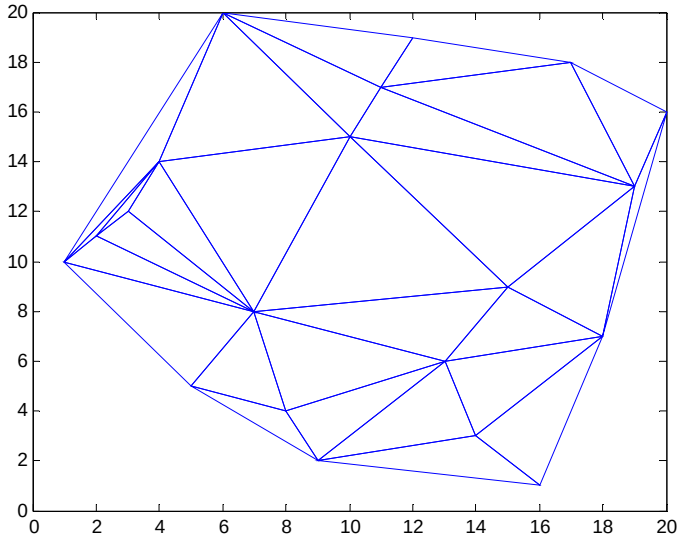

Figure 2. Delaunay triangulation for 20 random points.

CHAPTER II

REVIEW OF LITERATURE

A detailed study on the decimal sequences from various papers [1] - [6] was very helpful in generating random sequences. A study on Voronoi diagrams and Delaunay triangulation [10]-[13] laid the foundation in coming up with a new idea of generating random sequences. Other studies on two dimensional random arrays and patterns [8], [9] show connection between the theory of 2D random patterns and that of one dimensional random sequences.

Decimal Sequences

A decimal sequence is obtained when a number is represented in a decimal form in base 'r' and it may terminate, repeat or be aperiodic. For a certain class of decimal sequences of 1/q , q prime, the digits spaced half a period apart add up to r −1, where r is the base in which the sequence is expressed.

The following section describes the properties of decimal sequences [1, 2]:

Theorem 1: Any positive number x may be expressed as a decimal in the base r

$A_1A_2\ldots.A_{s+1}.a_1a_2\ldots$ where $0 \leq A_i < r$, $0 \leq a_i < r$, not all A and a are zero, and an infinity of the a_i are less then (r-1). There exists a one-to-one correspondence between the numbers and the decimals and

$$x = A_1 r^{s} + A_2 r^{s-1} + ... + A_{s+1} + \frac{a_1}{r} + \frac{a_2}{r^2} + ...$$

That the decimal sequences of rational and irrational numbers may possibly be used to generate pseudo-random sequences is suggested by the following theorems of decimals of real numbers.

Theorem 2: Almost all decimals, in any base, contain all possible digits. The expression almost all implies that the property applies everywhere except to a set of measure zero.

Theorem 3: Almost all decimals, in any base, contain all possible sequences of any number of digits.

Theorems 2 and 3 guarantee that decimal sequence missing any digit is exceptional. The binary d-sequence is generated by means of the algorithm: $a(i) = 2^i \bmod p \bmod 2$, where p is a prime number and a maximum-length sequence is generated with period p-1 when 2 is a primitive root of p. When the binary d-sequence is of maximum length, then bits in the second half of the period are the complements of those in the first half. It is easy to generate d-sequences, which makes them attractive for many engineering applications. It was shown in [2] that it is easy to find i given $\log_2 p$ bits of a(i), therefore, d-sequences cannot be directly used in random number generator applications.

By adding together two or more different binary d-sequences (obtained using primes), non-linearity in the generation process can be introduced and the resulting sequence becomes a good candidate for use as random sequence. This was considered in [6] to generate a sequence with long period.

$a(i) = 2^i \bmod p_1 \bmod 2 + 2^i \bmod p_2 \bmod 2 + 2^i \bmod p_3 \bmod 2…..$

Note: here + means modular 2 addition and p denotes a (ideally very large) large prime number.

If the individual sequence is of maximum length, then the period of the sum will be lcm{(p_1-1), (p_2-1),….}, where lcm (a, b) means the Least Common Multiple of a and b. For randomly chosen primes we do not know if the starting number is a primitive root, therefore, the actual period would be a divisor of be lcm{(p_1-1), (p_2-1),….}. If we choose a seed S, which is relatively prime to each pi, and the order of S does not divide (pi-1) for all i, then the power-exponent random number generate bits according to the following algorithm:

$a(0) = S \bmod p_1 \bmod 2 + S \bmod p_2 \bmod 2$

$a(1) = S^2 \bmod p_1 \bmod 2 + S^2 \bmod p_2 \bmod 2$

$a(2) = S^4 \bmod p_1 \bmod 2 + S^4 \bmod p_2 \bmod 2$ and so on.

One may replace p_1 and p_2 by n_1 and n_2 that are products of primes. For better security, the primes should each be congruent to 3 (mod 4) as in the BBS generator [24].

In [25], a recursive random number generator is proposed based on [6]. The main idea in [25] is to increase the period of sum of d-sequence not only by a factor of lcm{(p_1-1), (p_2-1),….} but by a multiple of it and also to smoothen the auto-correlation function. The recursive formula proposed is as follows:

$(S^i \bmod p_{11} + S^i \bmod p_{12} + \ldots + S^i \bmod p_{1n})^k \bmod p_{21} \bmod 2 + (S^i \bmod p_{11} + S^i \bmod p_{12} + \ldots + S^i \bmod p_{1n})^k \bmod p_{22} \bmod 2 + \ldots + (S^i \bmod p_{11} + S^i \bmod p_{12} + \ldots + S^i \bmod p_{1n})^k \bmod p_{2m} \bmod 2,$

where S is the seed and p_{fg} is a prime number and S and p_{fg} are relatively prime to each other. The first subscript distinguishes the loops i and k and second subscript is number of that element in its respective loop.

Note: + symbol used outside the bracket is meant for modular 2 additon.

Delaunay Triangulation and Voronoi diagrams

Let P be a set of n distinct points (sites) in the plane. The Voronoi diagram of P is the subdivision of the plane into n cells, one for each site. A point q lies in the cell corresponding to a site pi Î P iff ||q-pi || < ||q-pj ||, for each pi € P, j ≠i. Voronoi Diagram is a line that extends infinitely in both

directions, and the two half planes on either side (shown in figure 3). A Voronoi vertex is the center of an empty circle touching 3 or more sites. A point q lies on a Voronoi edge between sites piand pj iff the largest empty circle centered at q touches only pi and pj.

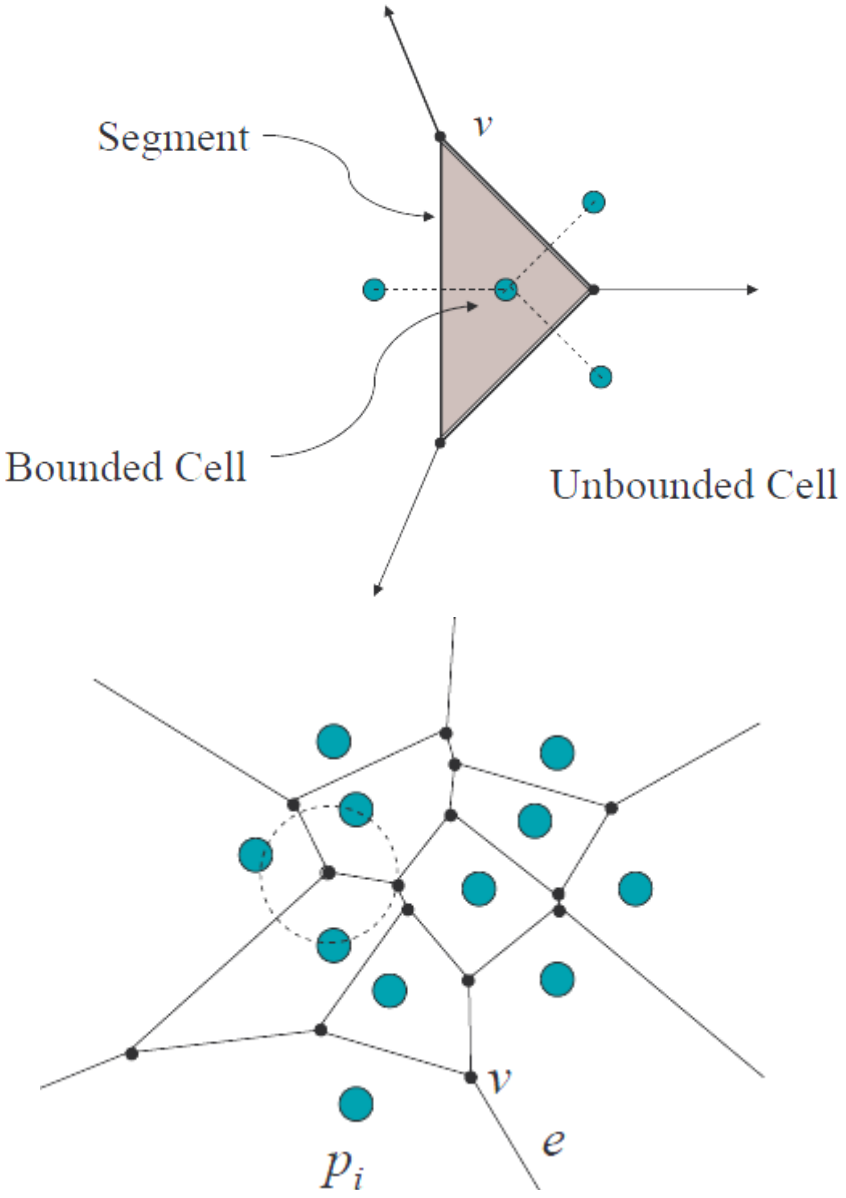

Figure 3. Voronoi diagram and its terminology

Voronoi Diagrams and Delaunay triangulation are duals of each other as said earlier. So Delaunay triangulation can be constructed using the Voronoi diagram. If two sites p_i and p_j share

an edge (p_i and p_j are adjacent), create an arc between v_i and v_j, the vertices located in sites p_i and p_j. Finally, straighten the arcs into line segments. The resultant graph is Delaunay triangulation. The transformation is shown in figure 4.

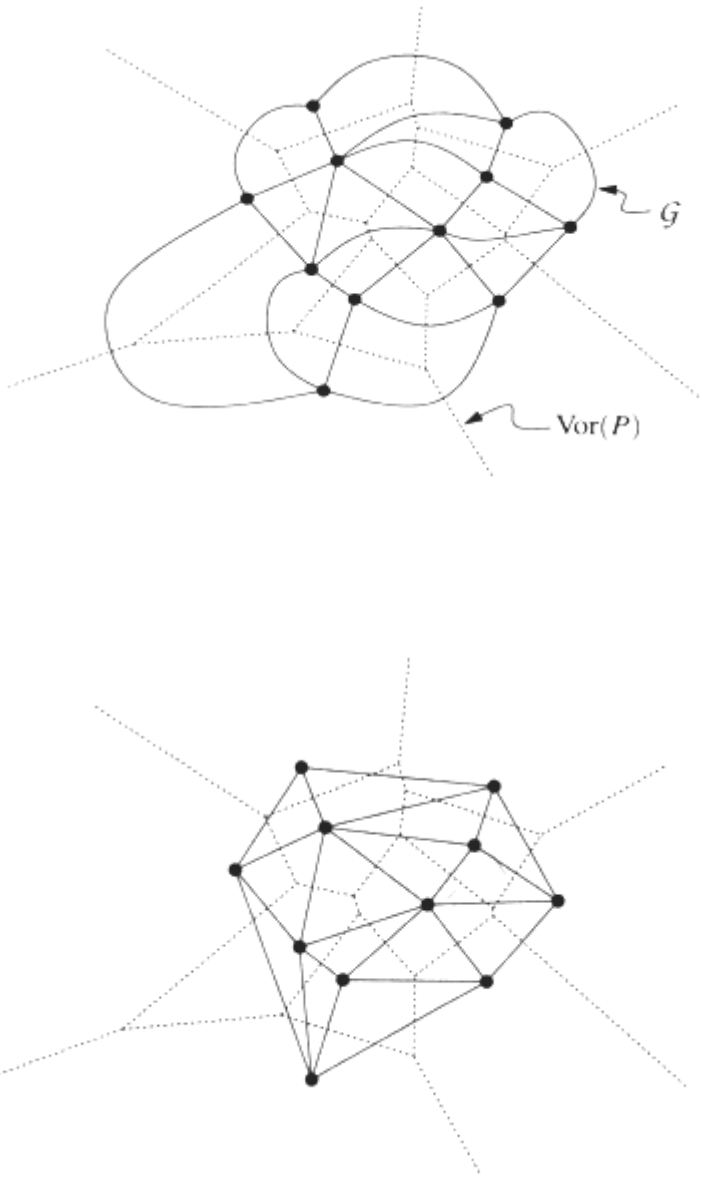

Figure 4. Delaunay triangulation from the voronoi diagram.

Auto-correlation function and Die hard tests

The Autocorrelation function and die hard tests are the two important tests which test the quality of a random sequence. The Autocorrelation (one dimensional) of a binary maximum length decimal sequence is

$$C(j) = (1/k)\sum_{i=1}^{k} a_i a_{i+j} \ ,$$

where k is the maximum length, j= 0, 1, 2, 3… (only positive values are considered). In the binary decimal sequence of 1's and 0's, 0 is replaced by -1 and the test is carried out. The graph is a symmetric one with the maximum value of 1 for j= 0 and maximum length and all the remaining values being 0. This autocorrelation function is used to test the randomness of the generated random sequences.

The two dimensional autocorrelation function of a binary maximum length decimal sequence is given as,

$$C(i, j) = (1/k1)(1/k2)\sum_{m=1}^{k1}\sum_{n=1}^{k2} a(m,n).a(m+i, n+j) \quad ,$$

where k1 is the row size and k2 is the column size, i and j take values 0, 1, 2, 3…. In the binary sequence of 1's and 0's, 0 is replaced by -1 and the test is carried out. The graph should be a symmetric one with a maximum value of 1 for j=0, k (here k=k1*k2) and all the remaining values being 0. The two dimensional autocorrelation function is used to test the randomness of the generated 2D random patterns.

Kak's randomness measure for a maximum length sequence, X, is given as,

$$R(X) = 1 - \frac{\sum_{k=1}^{n-1} |C(k)|}{n-1} \ ,$$

where C (k) is the autocorrelation function of X and n is the maximum length of X. If R(X) =1 (or closer to one), then X is a maximal random sequence and if R(X) =0, then it is completely deterministic.

The diehard tests proposed by George Marsaglia are a series of statistical tests for measuring the quality of a random sequence [14]. The tests need to be performed on the generated random sequences. The die-hard tests are discussed below:

- Birthday spacings: Choose random points on a large interval. The spacing between the points should be asymptotically exponentially distributed. The name is based on the birthday paradox.
- Overlapping permutations: Analyze sequences of five consecutive random numbers. The 120 possible orderings should occur with statistically equal probability.
- Ranks of matrices: Select some number of bits from some number of random numbers to form a matrix over {0, 1}, then determine the rank of the matrix. Count the ranks.
- Monkey tests: Treat sequences of some number of bits as "words". Count the overlapping words in a stream. The number of "words" that don't appear should follow a known distribution. The name is based on the infinite monkey theorem.
- Count the 1s: Count the 1 bits in each of either successive or chosen bytes. Convert the counts to "letters", and count the occurrences of five-letter "words".
- Parking lot test: Randomly place unit circles in a 100 x 100 square. If the circle overlaps an existing one, try again. After 12,000 tries, the number of successfully "parked" circles should follow a certain normal distribution.
- Minimum distance test: Randomly place 8,000 points in a 10,000 x 10,000 square, and then find the minimum distance between the pairs. The square of this distance should be exponentially distributed with a certain mean.

- Random spheres test: Randomly choose 4,000 points in a cube of edge 1,000. Center a sphere on each point, whose radius is the minimum distance to another point. The smallest sphere's volume should be exponentially distributed with a certain mean.
- The squeeze test: Multiply 2^{31} by random floats on [0, 1) until you reach 1. Repeat this 100,000 times. The number of floats needed to reach 1 should follow a certain distribution.
- Overlapping sums test: Generate a long sequence of random floats on [0, 1). Add sequences of 100 consecutive floats. The sums should be normally distributed with characteristic mean and sigma.
- Runs test: Generate a long sequence of random floats on [0, 1). Count ascending and descending runs. The counts should follow a certain distribution.
- The craps test: Play 200,000 games of craps, counting the wins and the number of throws per game. Each count should follow a certain distribution.

These tests would prove the randomness of the generated random sequences.

2D Random Patterns

Two dimensional patterns are basic to visual perception and it is not known how exactly such patterns are coded and recalled [15], although there is evidence that the coding is unary in certain situations for one-dimensional patterns [16], [17].

One way to create a random array is to map a random sequence into a two-dimensional pattern and an obvious choice is the use of shift-register sequences [9] or to use prime reciprocal sequences. One dimensional binary random sequences are obtained as expansions of the prime reciprocals, 1/p. Shift register (maximum length) sequences are obtained using the expansion of 1 divided by an irreducible polynomial [18]. Thus $1/1+x+x^3$ generates the periodic random sequence 0100111.

MacWilliams and Sloane [9] used the following procedure to map a sequence into a n1×n2 array: Start down from the main diagonal and continue from the opposite side whenever an edge is reached. Thus the shift –register random sequence 000100110101111 produces the

	0	1	1	1	1		1	7	13	4	10
array	0	0	1	1	0	using the term numbers	11	2	8	14	5
	0	1	0	0	1		6	12	3	9	15

A sequence may be mapped into an array in many other different ways. In theory, any arbitrary mapping scheme is as good as any other. Other straightforward mapping include mapping by rows or columns.

My thesis is to generate a 2D random pattern using a random sequence or number of base random sequences instead of a polynomial as shown above. The random sequence generated from either `Delaunay triangulation or Voronoi diagram would have more random nature, as we see in the next section, hence, generation of 2D random patterns using the decimal sequence works better than these approaches.

CHAPTER III

METHODOLOGY

As already discussed briefly in the introduction, pseudo random sequences and decimal sequences appear to be random but they are completely deterministic. The PN sequence appears to be random in the sense that the binary values and groups or runs of the same binary value occur in the sequence in the same proportion they would if the sequence were being generated based on a fair "coin tossing" experiment. In the experiment, each head could result in one binary value and a tail the other value. The PN sequence appears to have been generated from such an experiment. The sequence is not truly random in that it is completely determined by a relatively small set of initial values. It was shown in [2] that it is easy to find i given $\log_2 p$ bits of a(i), therefore, d-sequences cannot be directly used in random number generator applications. Further studies tried to increase the period but still the decimal sequence would be generated from the prime reciprocals. My approach is to generate random sequence from Delaunay triangulation or Voronoi diagram which would be more random in nature. Later, using these random sequences, 2D random patterns would be generated.

GENERATION OF RANDOM SEQUENCES

- Generate both the X and Y co-ordinates of the points using a RNG.
- For the generated random points, Delaunay triangulation or Voronoi diagram is generated.

- In Delaunay triangulation, the mean area of all the triangles is calculated and for each individual triangle, if its area is more than the mean area, 1 is considered and if it is less than the mean area, 0 is considered.
- If a Voronoi diagram is chosen, the edges which have their length to infinity are restricted to the working plane and clipped polygons are considered. Now, the mean area of all the polygons is calculated and if the individual polygon area is more than the mean area, 1 is considered and if it is less than the mean area, 0 is considered.
- The sequence is generated with these 1's and 0's in the same order as the individual triangles or polygons (their areas), as they are considered for the comparison with the mean area.

The binary sequence generated from the above procedure is a better candidate than the previous random sequences. The unpredictability is more, as one cannot predict the next digit in the generated sequence at any length which is highly unlikely compared to d-sequences or pseudo-random sequences. Before going to results, there is one more point to be noted. The main idea is to present a new approach to generate random sequences and not on security issues. Also, there will be no Delaunay triangulation when the random points are on a straight line and no Voronoi diagram when the random points are either on a straight line or on a circle. But it is almost impossible for such a condition to occur as the X and Y co-ordinates would be generated by a RNG. The number of polygons (along with clipped) generated from the Voronoi diagram would be same as the number of random points taken (shown in figure 5). Whereas, the number of triangles generated from the Delaunay triangulation would be 2n-k-2, where n is the number of random points and k is the number of points on the convex hull of the given set of points (shown in figure 6).

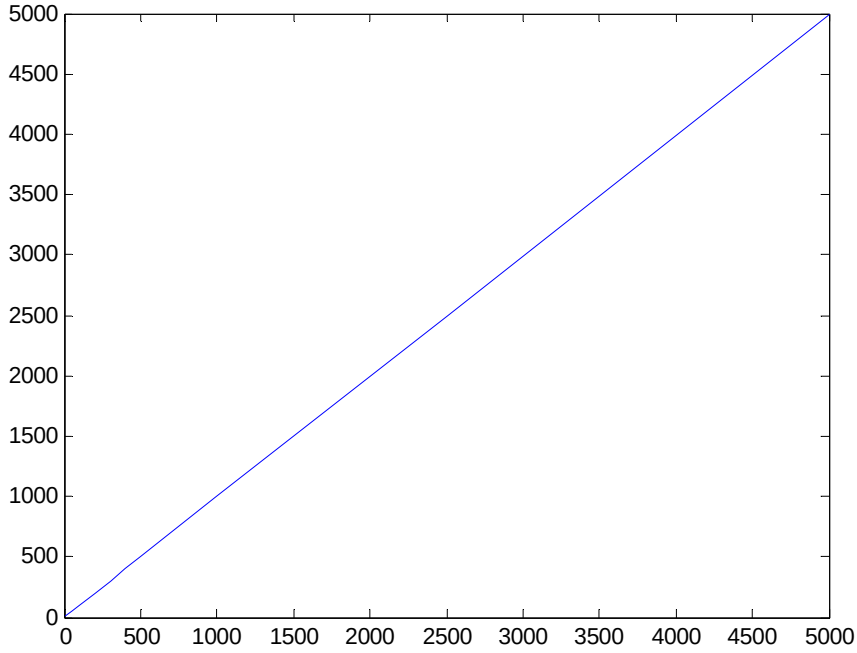

Figure5. Number of random points (X-axis) Vs Number of polygons generated from the Voronoi diagram (Y-axis).

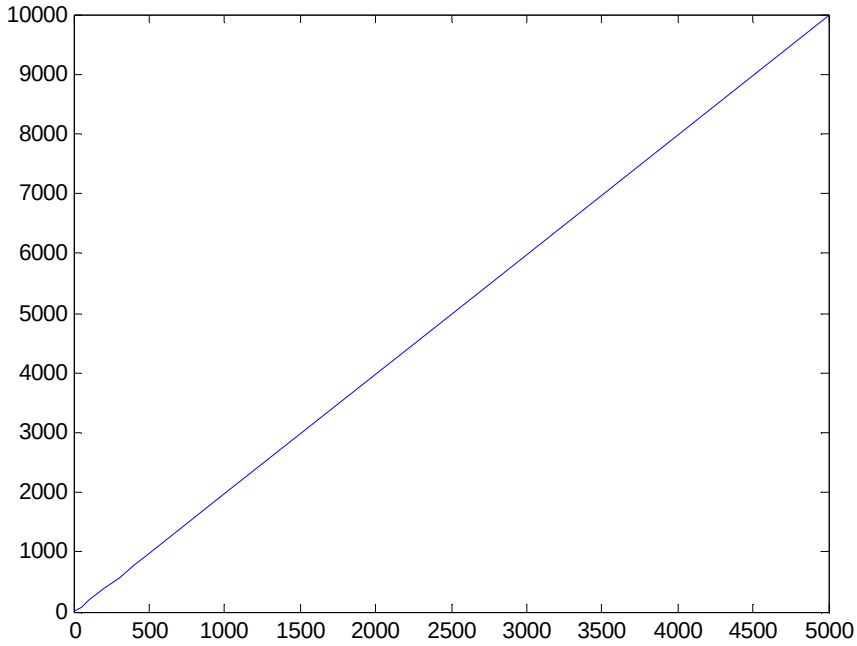

Figure6. Number of random points (X-axis) Vs Number of triangles generated from the Delaunay triangulation (Y-axis).

TWO DIMENSIONAL RANDOM PATTERNS USING RANDOM SEQUENCES

Two dimensional random patterns can be generated from either d-sequences or random sequences generated from the above proposed method. As we know, the decimal sequence of a prime reciprocal, 1/p, is generated using the formula 2^i mod p mod 2. When the sequence is placed into rows and columns (any order can be used), a 2D random pattern of desired size can be generated. By converting the 1's and 0's to black and white pixels, even an image of desired size can be generated.

For example, consider the prime number, 19. The maximal length sequence for its prime reciprocal is 000011010111100101. Figure 7 shows the 6 X 3 image when the decimal sequence is converted into black and white pixels. Consider the prime number 17. The decimal sequence for 17 is 0000111100001111. Figure 8 shows a 4 X 4 image for the sequence.

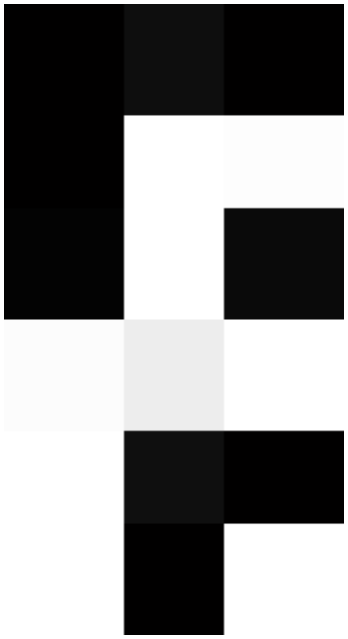

Figure 7. A 6 X 3 image generated from the prime reciprocal of 19.

Figure 8. A 4 X 4 image generated from the prime reciprocal of 17.

Two dimensional random patterns could be generated from random sequences or combination of base random sequences in the same way as mentioned above. The 2D random patterns generated from these sequences can be considered as more randomized than the other patterns as we shall see the results in the next chapter.

CHAPTER IV

FINDINGS

Generation of random sequences

All the simulations are done in MATLAB V 7.10 (R 2010a) and the results are shown only for random sequences generated from Delaunay triangulation. Figure 9, 10, 11 and 12 show the Delaunay triangulations generated for 10, 100, 1000 and 5000 random points and binary random sequences of length 12, 184, 1982 and 9977 are generated respectively. The random sequence generated for 10 random points is 010111110011.

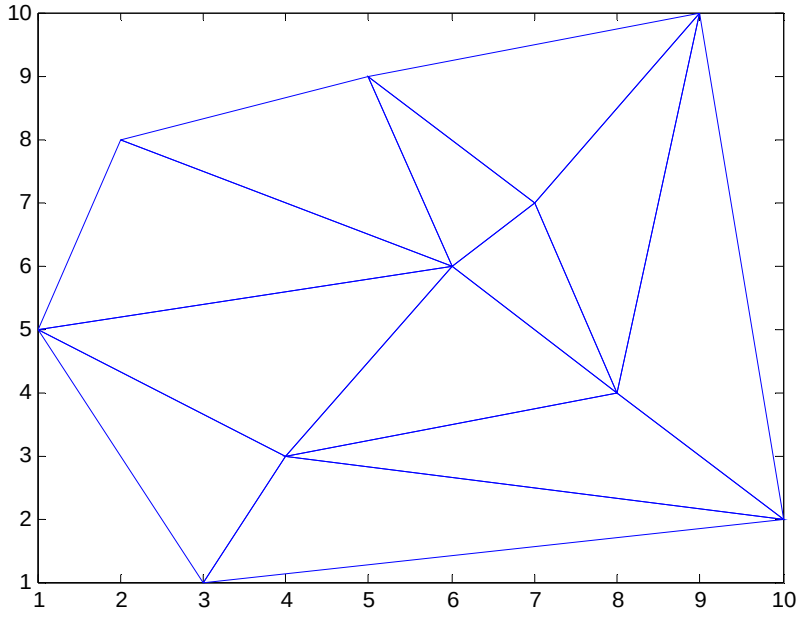

Figure 9. Delaunay triangulation generated for 10 random points

The sequence generated for 100 random points is 0011000100000011101101111100011001
0010001100000000110000000111110010000000110100101110000010101100001010010000
0010010111100000001000100001110110011110011100000111011010000000.

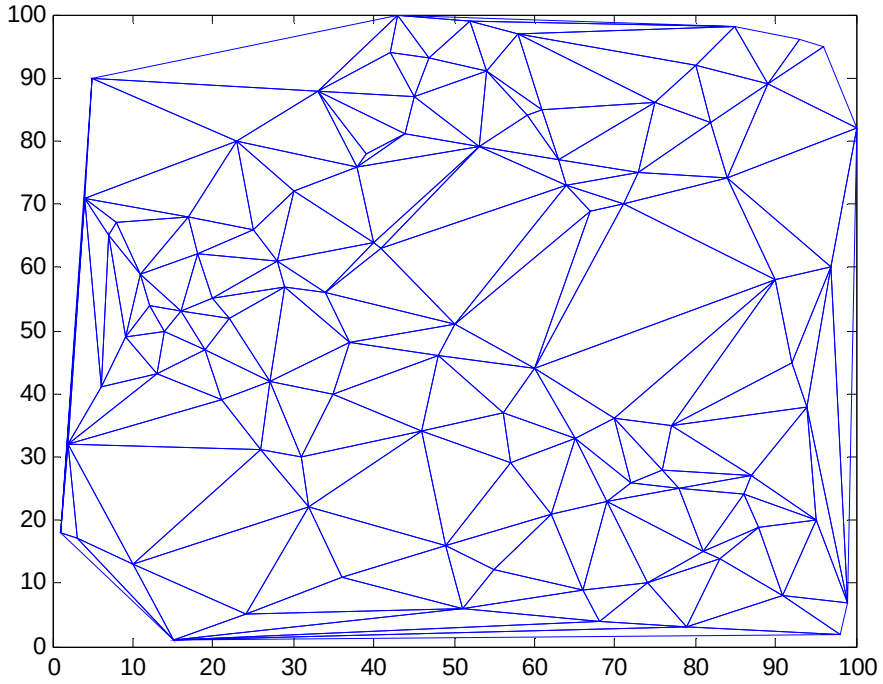

Figure 10. Delaunay triangulation generated for 100 random points

The sequence generated for 1000 random points is 1010110101010010000110100100 10
1001000001010101100011000100000000011110000010000101010101101111000000101111101
0001000100101110000010100111000110100110001011111010010101100110100001110001 1
0001001010101110010010001010000010100000011111010001000100000111011111111010110
0011001110100011100110010000100001100011100100001100000110010000110110010001 10
0001100100110101000101000100000001000011100100111110010111001000010111001100
1010000001000001100100111100101100000110011000000000110011100001000011 00

10011010100110100000010010000010111110000001001000010011110010001001100000000110001100100010011001001011010100010001110011010000010111110110110001110000000000001010001000111100000101111010101111000000110010000000100000000000001111010001010100100110100110000100000000000100001000001100100101011000000000000010110110111011100100001000000001001010000101110100010000001011000010111110010001101010100000111001110101001010010001000010010100000000000000101001111100101011110110110011010001010000100010011111011001000000000100010110001001101000011010101010100101000110001000110100010101000000100111000011001100000000110100000100111000000010001100010000100000111110110000001000011100101101011111110000011000000001001000100001101110100101000111001000011011101111000101000001000010000000100101111000000010111000011101011000000000111001100111001111001000110011010100000000100000100000000110001001110010001010011010010111000000000000100011001000111001100100101001000011011001001110001000011001100110010111011101001011000000100000000000100110110011010010000000000000000101001010111100000100101101110101010011011100111110010110111101000000010111000011100000000000100101010001000100111010110100001101110010000010110001101011101001010000000100001000010000110110000000101010000000011010110101111000000000100100000100100110010100010101000011011010010100000000100111000000100111001000100000001100110101001101011000010010010100000010010010110001000111100110000000001100010. The random sequence generated for 5000 random points is not shown here (as it would consume the whole paper). Unlike d-sequences, one cannot predict the next coming bits or digits when some or all the previous bits of the sequence are known.

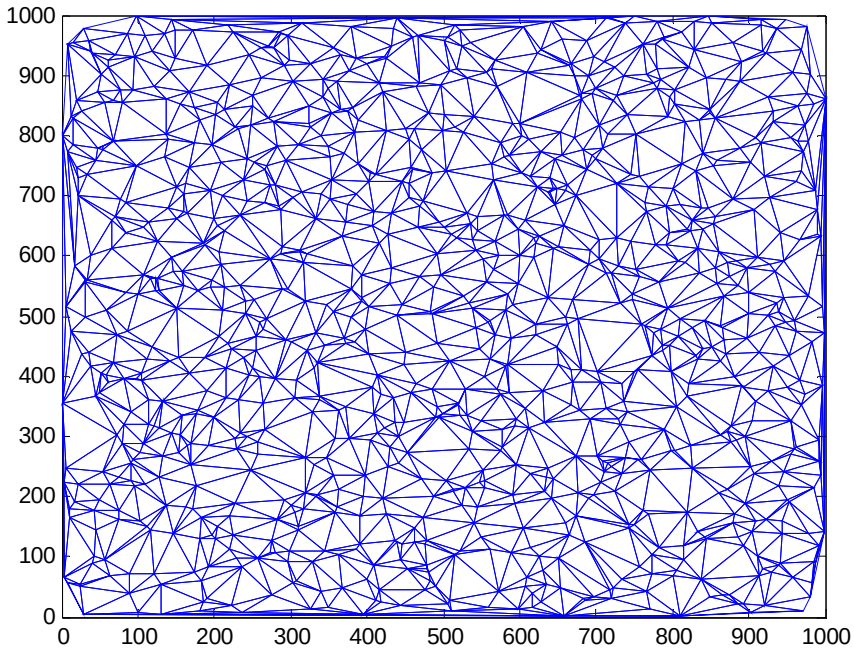

Figure 11. Delaunay triangulation generated for 1000 random points

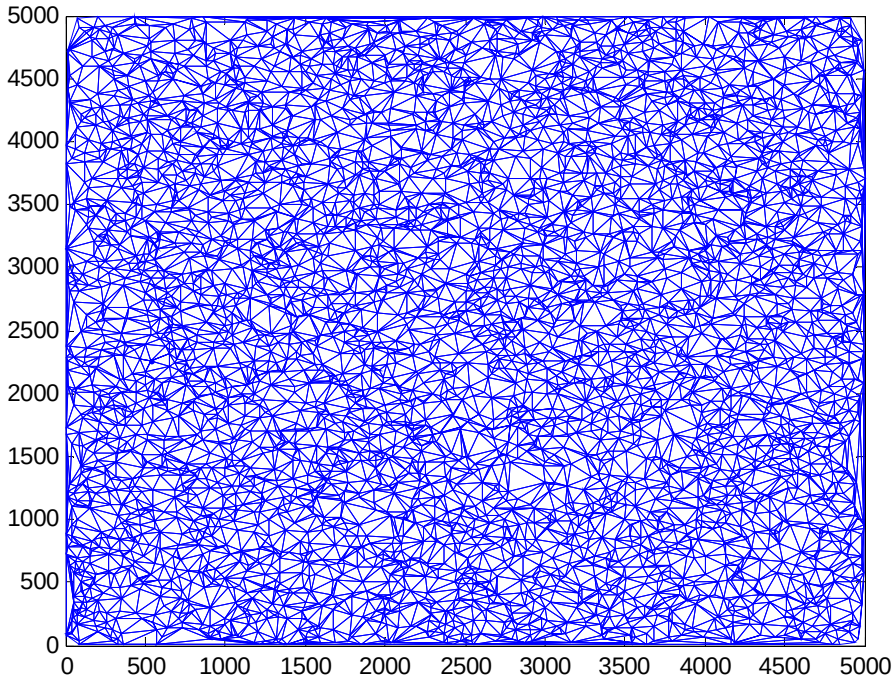

Figure 12. Delaunay triangulation generated for 5000 random points

AUTOCORRELATION FUNCTION

The graph in figure 13 shows the autocorrelation function of the random sequence of length 184 (100 random points). The maximum value of 1 is obtained for 0 and 184, as it should be. The intermediate values are between +0.2 and -0.1. Kak's randomness measure of the sequence is 0.9426 which is a high value.

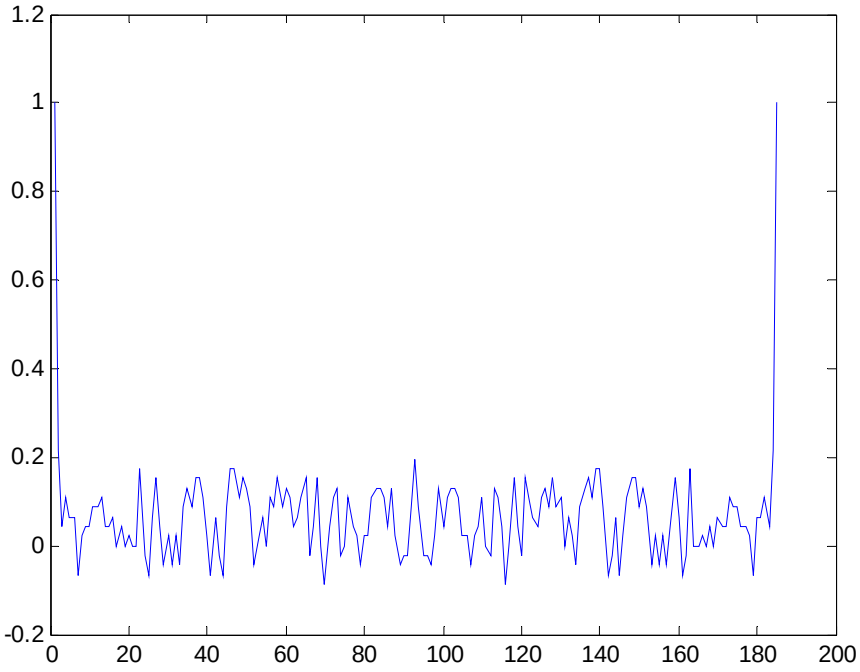

Figure 13. Autocorrelation function of random sequence of length 184.

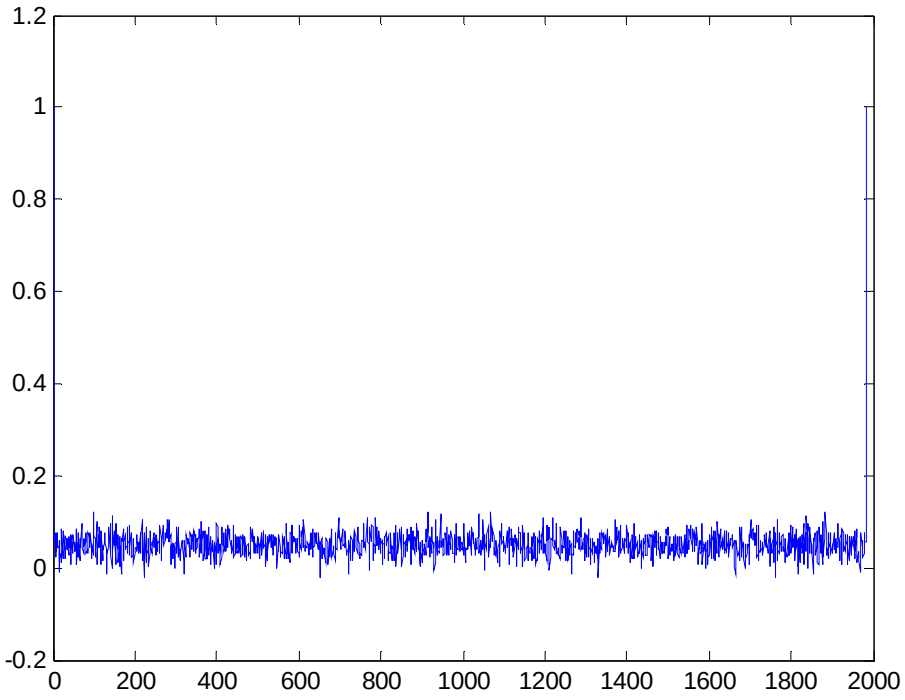

Figure 14. Autocorrelation function of random sequence of length 1982.

As we consider the autocorrelation function of the random sequence of length 1982 (1000 random points), intermediate values lie between +0.1 and 0, as shown in figure 14. Kak's randomness measure of the sequence is 0.9485 and hence it is also very random.

DIEHARD TESTS RESULTS

The diehard tests proposed by George Marsaglia are a series of statistical tests for measuring the quality of a random number generator [14]. These tests can be used to prove the randomness of these random sequences. The diehard tests are simulated in JAVA by Zur Aougav under sourceforge (open source software), and his test suite, jrandtest-0.4, is used to test the random sequence. The test suite also provides diehard tests for so many other RNG algorithms.

Table 1 provides the diehard tests results of random sequences and a comparison is provided with other random number generators namely SHA1 algorithm and JAVA random function. In our simulation, all the tests which follow a distribution are using chi-square (KS) test.

Test	JAVA random function	SHA1 Algorithm	Random sequence
Runs test (Set 1) (sequence of 10000)	RunsUP: KS test for 10 p's: 0.304 Runs DOWN: KS test for 10 p's: 0.460	RunsUP: KS test for 10 p's: 0.455 Runs DOWN: KS test for 10 p's: 0.510	RunsUP: KS test for 10 p's: 0.423 Runs DOWN: KS test for 10 p's: 0.498
Runs test (Set 2) (sequence of 10000)	RunsUP: KS test for 10 p's: 0.127 Runs DOWN: KS test for 10 p's: 0.698	RunsUP: KS test for 10 p's: 0.132 Runs DOWN: KS test for 10 p's: 0.751	RunsUP: KS test for 10 p's: 0.136 Runs DOWN: KS test for 10 p's: 0.798
Squeeze test	Chi-square with 42 degrees of freedom: 52.6387 z-score =1.1608, p-value = 0.1258	Chi-square with 42 degrees of freedom: 42.34 z-score = 0.0371, p-value = 0.4563	Chi-square with 42 degrees of freedom: 38.19 z-score = 0.145, p-value = 0.6
Min distance (done for 100 sets)	KS test on 100 transformed mindist^2's: p-value = 0.2174	KS test on 100 transformed mindist^2's: p-value = 0.2110	KS test on 100 transformed mindist^2's: p-value = 0.3735
Count the 1's	Chisquare =	Chisquare =	Chisquare =

(sample size: 256000)	944,003.9000 z-score = 13,314.8758 p-value = 0.0000	942,790.6199 z-score = 13,297.7175 p-value = 0.0000	963,889.3220 z-score = 13,596.0982 p-value = 0.0471
Birthday spacings (bdays=1024, days/yr=2^24, lambda=16, sample size=500)	degree of freedoms : 17 p-value for KStest on the 9 p-values: 0.0894	degree of freedoms : 17 p-value for KStest on the 9 p-values: 0.3454	degree of freedoms : 17 p-value for KStest on the 9 p-values: 0.2378
Binary rank (32×32 matrices)	chi-square =4.3138 with df = 3; p-value = 0.2077	chi-square =8.0225 with df = 3; p-value = 0.0409	chi-square =7.0140 with df = 3; p-value = 0.0642
Binary rank (31×31 matrices)	chi-square =1.8952 with df = 3; p-value = 0.5460	chi-square =1.9042 with df = 3; p-value = 0.5442	chi-square =1.8344 with df = 3; p-value = 0.5622
Binary rank (6×8 matrices)	KS p-value = 0.8102	KS p-value = 0.6028	KS p-value = 0.5294
Craps Wins	p-value = 0.5088	p-value = 0.4914	p-value = 0.8204
Craps Throws	p-value = 0.1169	p-value = 0.3176	p-value = 0.2965

Table 1. Diehard test results for random sequences, SHA1 algorithm and JAVA random function.

Most of the tests in Diehard return a p-value, which should be uniform on [0, 1) if the input file contains truly independent random bits. Those p-values are obtained by p=F(X), where

F is the assumed distribution of the sample random variable X. But that assumed F is just an asymptotic approximation, for which the fit will be worst in the tails. Thus one should not be surprised with occasional p-values near 0 or 1, such as .0012 or .9983. When a bit stream really fails big, you will get p's of 0 or 1 to six or more places. By all means, a $p < .025$ or $p > .975$ does not mean that the RNG has "failed the test at the .05 level". Such p's happen among the hundreds that Diehard produces, even with good RNG's. The distribution of the p-values from the diehard suite of tests is shown in Table2.

p-value range	Expected Percent	Observed Percent		
		JAVA RNG	SHA1	Random Sequence
0.0 - 0.1	10	13	12	11
0.1 - 0.2	10	12	10	10
0.2 - 0.3	10	12	10	11
0.3 - 0.4	10	10	10	8
0.4 - 0.5	10	10	15	12
0.5 - 0.6	10	9	10	10
0.6 - 0.7	10	10	9	10
0.7 - 0.8	10	7	10	10
0.8 - 0.9	10	10	9	10
0.9 - 1.0	10	7	5	8

Table 2. Distribution of the p-values from the diehard suite of tests.

When we have a look at the above table random sequences are doing a great job compared to JAVA RNG and SHA1 RNG as random sequences are more uniform on [0, 1).

GENERATION OF 2D RANDOM PATTERNS

A 128 × 64 image generated from a single random sequence obtained from the Delaunay triangulation is shown in figure 15. Here the length of the sequence should be 128 × 64 = 8192. Hence, a random sequence with a maximal length of 8192 (generated from 4108 random points) is generated from the above approach. Figure 16 shows the 128 × 64 image generated from four different random sequences of length 1982 (1000 random points), 3971 (2000 random points), 982 (500 random points) and 1257 (637 random points).

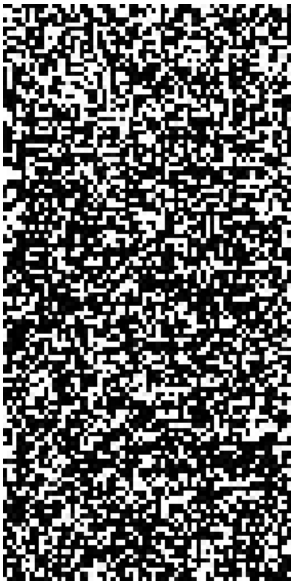

Figure 15. A 128 × 64 image generated from the random sequence obtained from the Delaunay triangulation drawn for 4108 random points.

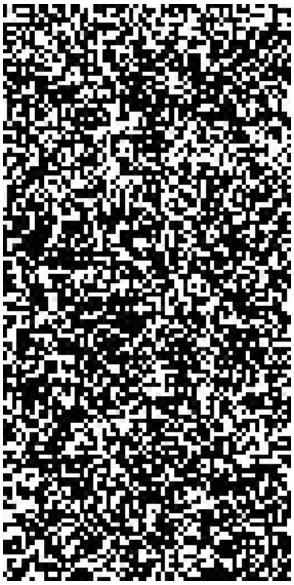

Figure 16. A 128 × 64 image generated from four random sequences.

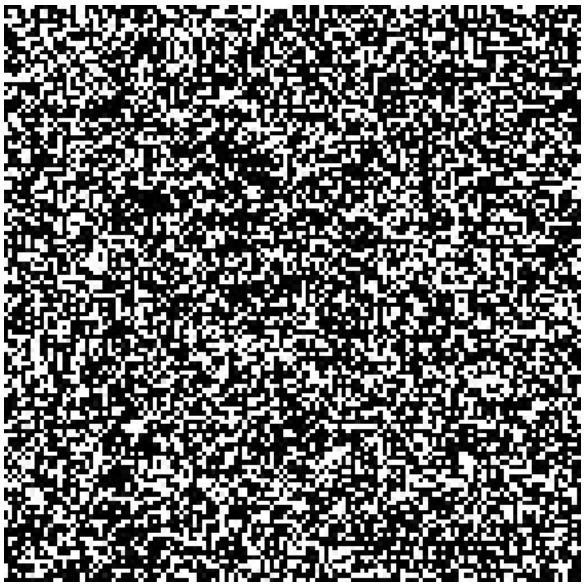

Figure 17. A 128 × 128 image generated from seven random sequences.

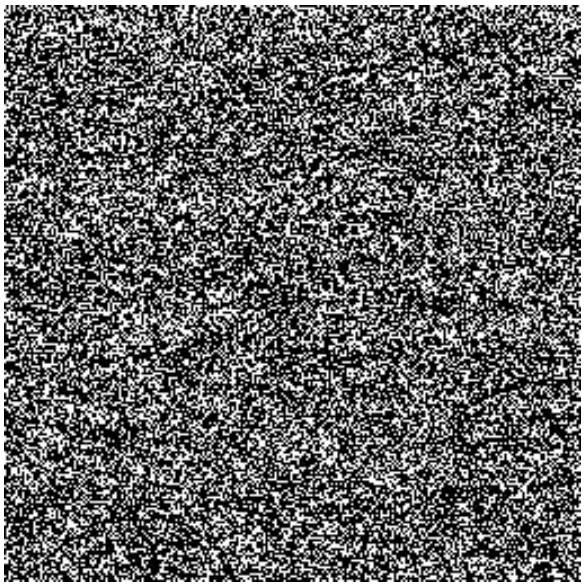

Figure 18. A 256 × 256 image generated from six random sequences.

Figure 17 shows a 128 × 128 image (16284 pixels) generated from seven different random sequences of length 4976 (2500 random points), 583 (300 random points), 1379 (700 random points), 2376 (1200 random points), 984 (500 random points), 2979 (1500 random points) and 3107 (1564 random points). Figure 18 shows a 256 × 256 image (65536 pixels) generated from six different random sequences of length 19973 (10000 random points), 14977 (7500 random points), 9978 (5000 random points), 5975 (3000 random points), 11972 (6000 random points) and 2661 (1343 random points).

TWO DIMENSIONAL AUTOCORRELATION FUNCTION

Figures 19, 20, 21 and 22 show the 2D autocorrelation function results for the images generated in figures 15, 16, 17 and 18 respectively. In all the graphs, the maximum value of 1 is obtained for 0 and maximum length, but the intermediate values lie between +0.1 and -0.1. R(X) for the four patterns is computed as 0.9732, 0.9785, 0.9745 and 0.9815 respectively. Hence all the four patterns can be considered as good random patterns.

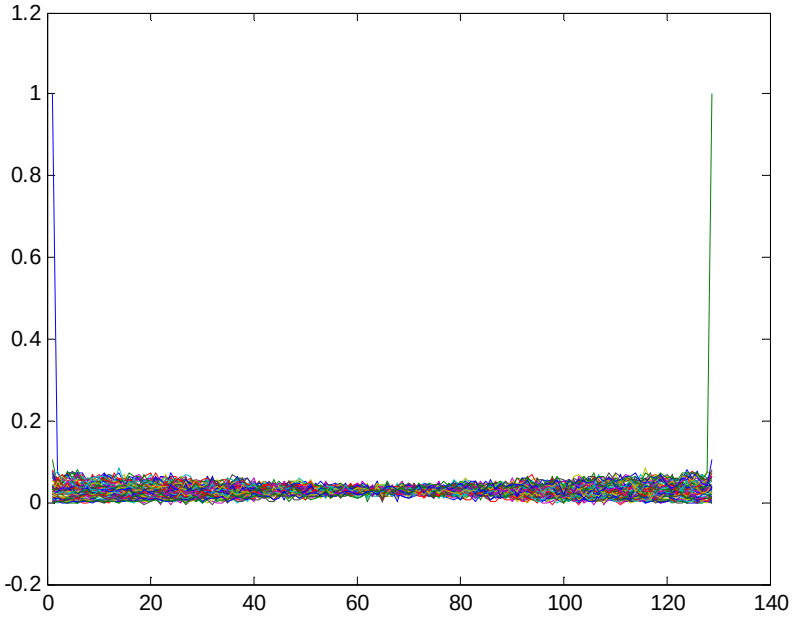

Figure 19. Two-dimensional autocorrelation function of a 128×64 image generated by using only one random sequence.

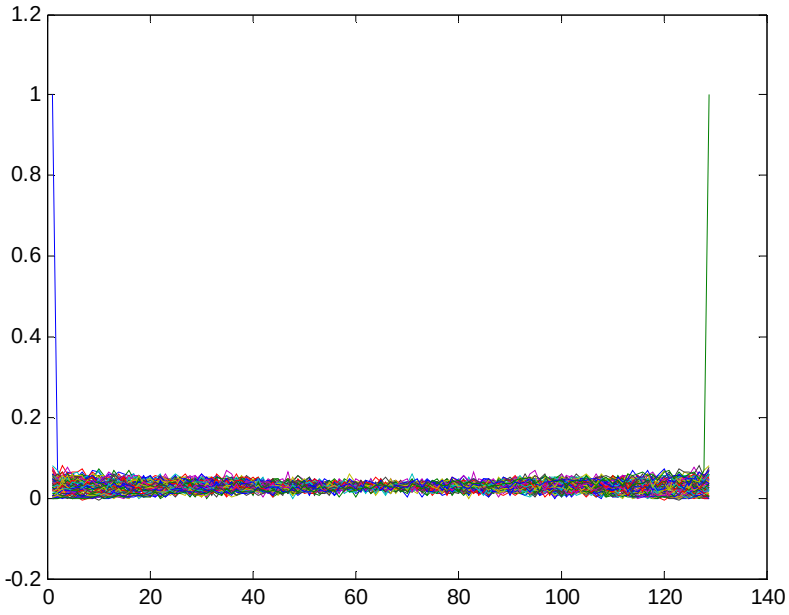

Figure 20. Two-dimensional autocorrelation function of a 128×64 image generated by using four random sequences.

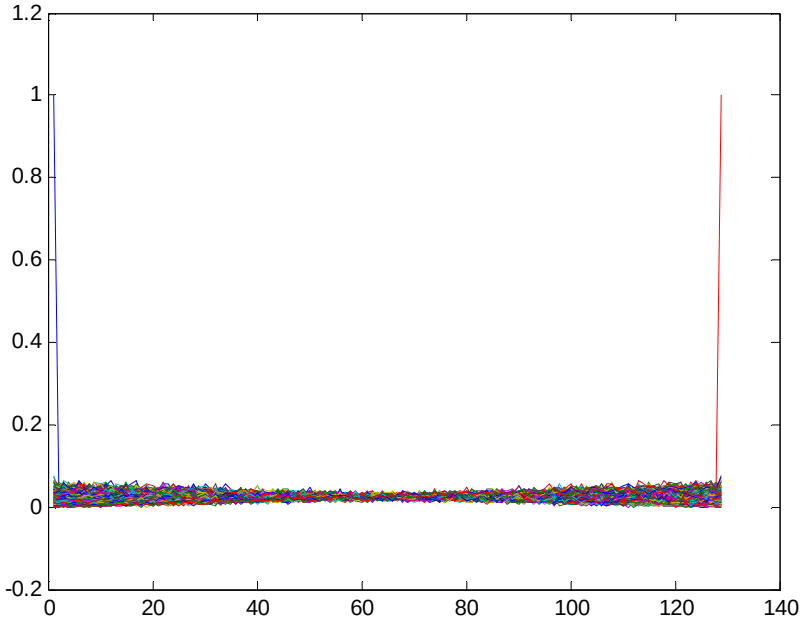

Figure 21. Two-dimensional autocorrelation function of a 128×128 image generated by using seven random sequences.

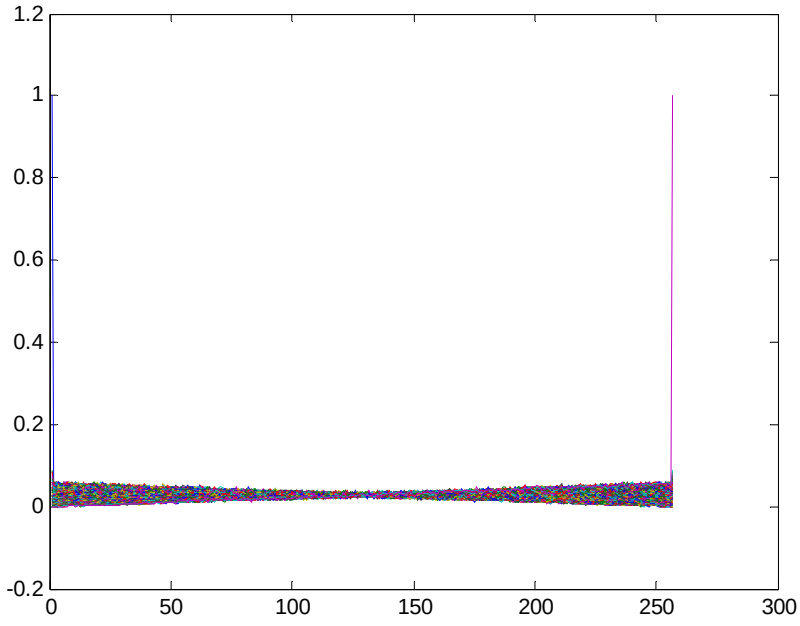

Figure 22. Two-dimensional autocorrelation function of a 256×256 image generated by using six random sequences.

CHAPTER V

CONCLUSION

Pseudo random sequences and decimal sequences appear to be random but they are completely deterministic in nature. The sequences appear to be random in the sense that the binary values and groups or runs of the same binary value occur in the sequence in the same proportion they would if the sequence were being generated based on a fair "coin tossing" experiment. In the experiment, each head could result in one binary value and a tail the other value. The PN sequence appears to have been generated from such an experiment. The sequence is not truly random in that it is completely determined by a relatively small set of initial values. Also, d-sequences cannot be directly used in computationally secure random number generator applications as it is easy to find i given $\log_2 p$ bits of a(i). One can choose to increase the period but still the decimal sequence would be generated from the prime reciprocals.

In the thesis, random sequences are generated by making use of either Delaunay triangulation or Voronoi diagram drawn from random points. These random sequences are more random in nature compared to the d-sequences and PN sequences. The randomness of these sequences is proved by performing the autocorrelation function and diehard tests and the results are satisfactory.

The main focus of the thesis is to present the new idea of generating random sequences and patterns and not on any issues relating to security. The limitations of this new approach are

very minimal. The cases where there is no possibility of a Delaunay triangulation or a Voronoi diagram occur infrequently.

Further work can be done in implementing these sequences and 2D patterns in real world applications. The applications of these sequences and patterns for cryptography and key distribution could also be investigated.

REFERENCES

[1] S. Kak and A. Chatterjee, On decimal sequences. IEEE Transactions on Information Theory IT-27, 647-652, 1981.

[2] S. Kak, Encryption and error-correction coding using D sequences. IEEE Transactions on Computers C-34, 803-809, 1985.

[3] S. Kak, New results on d-sequences. Electronics Letters, vol. 23, p. 617, 1987.

[4] S. Kak, A new method for coin flipping by telephone. Cryptologia, Vol, 13, pp. 73-78, 1989.

[5] N. Mandhani and S. Kak, Watermarking using decimal sequences. Cryptologia 29, 50-58, 2005.

[6] S. Kak, A new random number generator. http://arxiv.org/ftp/arxiv/papers/0907/0907.5226.pdf.

[7] S. Kak and N.S. Jayant, On speech encryption using waveform scrambling. Bell System Technical Journal 56, 781-808, 1977.

[8] S. Kak, The algebra of two dimensional patterns, 2011. http://arxiv.org/pdf/1102.4573.

[9] F. J. MacWilliams and N. J. A. Sloane, Pseudo-random sequences and arrays. Proc. IEEE 64, 1715-1729, 1976

[10] M. de Berg, O. Cheong, M. V. Kreveld and M. Overmars, Computational geometry: algorithms and applications, Springer, 2001.

[11] F. Aurenhammer, Voronoi diagrams - A survey of a fundamental geometric data structure. ACM Comput. Surv 23, 345-405, 1991.

[12] A. Miu, Voronoi diagrams, http://www.diku.dk/students/duff/Fortune.

[13] Delaunay triangulations,

groups.csail.mit.edu/graphics/classes/6.../Delaunay/Delaunay2D.ppt.

[14] G. Marsaglia, Diehard, A battery of tests of randomness, http://stat. fsu. edu/~ geo/diehard.

[15] L. Squire, T. Albright, F. Bloom, F. Gage, and N. Spitzer (eds.), New Encyclopedia of Neuroscience, Elsevier, 2007.

[16] S. Kak, Unary coding for neural network learning, 2010. arXiv:1009.4495.

[17] I.R. Fiete and H.S. Seung, Neural network models of birdsong production, learning, and Coding. Elsevier, 2007.

[18] S.W. Golomb, Shift Register Sequences. Holden-Day, San Francisco, 1967.

[19] S. Kak, On information associated with an object. Proceedings Indian National Science Academy 50, 386-396, 1984.

[20] S. Kak, On quantum numbers and uncertainty. Nuovo Cimento 33B, 530-534, 1976.

[21] S. Kak, Quantum information and entropy. International Journal of Theoretical Physics 46, 860-876, 2007.

[22] S. Kak, The initialization problem in quantum computing. Foundations of Physics 29, 267–279, 1999.

[22] R. Landauer, The physical nature of information. Phys. Lett. A 217, 188- 193, 1996.

[23] M. Li and P. Vitanyi, An Introduction to Kolmogorov Complexity and its Applications. Springer Verlag, 2008.

[24] L. Blum, M. Blum and M. Shub, A simple unpredictable pseudorandom number generator, SIAM J. Computing, 15: 364-383, 1986.

[25] A. Parakh, A d-sequence based recursive random number generator, Proceedings of International Symposium on System and Information Security – Sao Jose dos Campos: CTA/ITA/IEC, 2006; arXiv:cs/0603029v2.

VITA

Chakradhara Reddy Chinthapanti

Candidate for the Degree of

Master of Science

Thesis: TWO DIMENSIONAL RANDOM PATTERNS

Major Field: Computer Science

Biographical:

Education:

Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in July, 2011.

Completed the requirements for the Bachelor of Engineering in Computer Science and Engineering at Vellore Institute of Technology, Vellore, India in 2008.

Experience:

- *Research Assistant* *Oklahoma State University, Stillwater, OK,* August 2010 to present

  Projects:

  Secure Analytics (August 2010 - present): A windows based application which is being implemented simultaneously in JAVA and C#.net.

- Completed training in INFOSYS, a multinational company in India, January 2008 to May 2008.

  Projects:

  Finacle core Estimation And Sizing Tool (FEAST): A web tool which aids in counting/sizing finacle core, a banking software from Infosys, based on function point analysis using JAVA and SQL server 2005.

Name: Chakradhara Reddy Chinthapanti Date of Degree: July, 2011

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: TWO DIMENSIONAL RANDOM PATTERNS

Pages in Study: 36 Candidate for the Degree of Master of Science

Major Field: Computer Science

Scope and Method of Study:

This thesis presents a new approach to the generation of random sequences and two dimensional random patterns in which random sequences are generated by making use of either Delaunay triangulation or Voronoi diagrams drawn from random points taken in a two dimensional plane. Both the random sequences and two dimensional random patterns generated in this manner are shown to be more random when compared to pseudo-random sequences and patterns.

Findings and Conclusions:

Random sequences of different lengths are generated by this approach and also 2D random patterns are generated in various ways by using one or more number of random sequences. The randomness of these sequences and patterns are shown by performing the autocorrelation function and diehard tests. We conclude that the sequences and 2D patterns generated from this approach are effectively random.

ADVISER'S APPROVAL: Dr. Subhash Kak