Literature review

The practical applications of the Hilbert Curve in different computer science areas

Queensland University of Technology

Student Name and ID

Viet Hoang Do – n10329935

Introduction

A Space-filling curve is a method to map between a multi-dimensional space and one-dimensional space (Mokbel et al., 2003). It is a continuous path that passes through each point in a 2D space exactly once and never crosses itself. The first description of the Hilbert curve was written by Hilbert in his paper of 1981. He introduced the curve as a variant of the space-filling Peano curves and used a binary radix to represent the coordinates of points (Hilbert, 1981). The Hilbert Curve is one of the most well-known space-filling curves that is used to map high dimensional features while preserving the locality - the points which are close to each other in 1D space would remain to be close to each other in 2D space. This locality property enables the wide use of the Hilbert Curve in computer science applications. This literature review will discuss the practical applications of the Hilbert curve in multiple computational-related areas.

Previous Studies

The applications of the Hilbert curve in computer vision and image processing:

- A method for pixel-based motion estimation using Hilbert Curve was described in the paper of Trimeche et al. (2006). The paper indicated that more robust performance was generated by using the Hilbert Curve rather than the conventional scan method. The method was simplicity itself and required relatively low computational power.
- Velho and Gomez (1991) describe a digital halftoning technique with Space-Filling Curves (Peano, Hilbert, and Sierpinski curves) to generate periodic patterns of clustered dots. The technique could produce impressive results with low-resolution devices such as a 300dpi laser printer. The gray level of images is well rendered as well as the fine details are captured neatly. However, the algorithm of the technique might require high memory usage because of the non-standard access pattern of images' buffers.
- Valantinas (2005) proposes the idea of using space-filling curves to change image dimensionality. These curves are used as image scan trajectories, produce results that are useful to other image processing techniques (encoding, filtering, etc.).
- Songa and Roussopoulosb (2002) utilize the Hilbert Curve to accelerate the speed of subset queries on uncompressed images. They store the image pixels in the Hilbert order and follow some steps such as analyze the clustering property of the Hilbert Curve, increase query range to avoid bad columns, and retrieve useful pixels.
- Yasser Ebrahim et al., (2009) present a technique using a space-filling curve to solve shape representation and description problems with linear-time complexity. The recognition rate provided by the proposed method is significantly high, up to 95.6% on certain datasets

The applications of the Hilbert curve in genomics fields:

- Wong et al., (2003) propose the idea using the Hilbert Curve visualization technique to demonstrate the structural features of very large whole bacterial genomes. The method does not require high-performance computation and produces the result fast and effectively.

- Deng et al., (2008) also use the DNA Hilbert-Peano curve as a tool to map DNA sequences into 2D space and visualize large-scale genome sequences. The final output is successful to highlight particular features of genome sequences. However, it cannot be used to identify the characteristics of a genome.
- Simon Anders, (2009) claims that the Hilbert curve can be used in conjunction with genome browsers to gain further insights into the data structure. The author demonstrates this by outlining examples from various use cases. HilbertVis, an open-source tool, is introduced, which lets users create and interact with such graphs.

Literature Review

1. Background

The Hilbert curve is built in a recursive manner: the curve is divided into four sections in the first iteration, which are mapped to the four quadrants of the square. The following iteration divides each quadrant into four sub-quadrants, each of which holds 1/16 of the curve, and the quadrants of these sub-quadrants each retain 1/256, and so on.

2. Shape representation and description using the Hilbert curve (Yasser Ebrahim et al., 2009)

Many tasks, such as object detection, matching, registration, and picture retrieval, need shape matching. Users are more interested in retrieval by form than colour and texture, according to a user survey on the cognition elements of picture retrieval (Lambert et al., 1999). For decades, researchers in this field have been working on this subject and have reported on a variety of matching algorithms. Yasser Ebrahim et al., (2009) offers a new shape representation and description methodology that implemented a region-based technique. In contrast to contour-based approaches, region-based techniques take into consideration all of the pixels in the picture to generate the shape representation, rather than just the boundary information. Depending on shapes are broke into subparts or not, region-based shape representation methods may be separated into global and structural techniques. The aim is to scan the object picture with the Hilbert curve (HC), resulting in a 1D version of the image that is smoothed and sampled to obtain the picture's shape feature vector (SFV). The suggested method is linear time and it does not depend on the translated, scaled, or stretched images.

Method [Figure 1]: The intensity variation of pixels distinguishes an object from its surroundings. The segmented object picture is scanned with the Hilbert Curve to capture this variance, which contains shape information. A vector V is used to keep the intensity value of each visited pixel. The Hilbert Curve's locality-preserving property produces a vector that represents the image's pixel clustering. The wavelet transform is performed to V to smooth out noise while keeping the major shape features intact, resulting in the vector WV, which is then sampled to create the vector SWV. Then the method normalizes the vector SWV to generate the object's shape features vector (SFV).

To allow the Hilbert Curve to visit each pixel in a picture, the picture must be 2^n x 2^n pixels in size, where n is the curve iteration. In some cases, the image is rectangular, the Hilbert Curve is stretched to cover. Therefore, the unvisited pixels are spread uniformly on both dimensions, as illustrated in the Figure 2, to reduce the effect of missing the unvisited pixels. In the Figure 2, the curve is stretched to match the large photo, despite the fact that both curves are of the same level. However, the two SFVs are almost similar, having a 0.94 cross-correlation.

Choosing the Hilbert Curve level: The implementation of a Hilbert Curve with the highest level ensures maximum sensitivity to details. In other cases, though, a more generic depiction may be preferable. No wavelet smoothing is applied [Figure 3], however, the HC level is changed between 6 (highest) and 2 (minimum).

Choosing the wavelet approximation level: The wavelet transform is used as a method of smoothing the SFV in order to keep just the most important elements of the shape. It is clear that lowering the HC causes a rapid and abrupt loss of details, including some key characteristics, but raising the wavelet approximation level keeps the primary characteristics while smoothing out just the minor details. Experiments on approximation levels 1–7 reveal that approximation levels 1–4 produce the greatest outcomes [Figure 4].

Choosing the SFV size: The greater the SFV size, the more information is maintained, but the cost of storing and comparing SFVs increases. A larger SFV provides multi-scale support in addition to detail preservation. An SFV with a size of 1024, for example, can be sampled to generate representations having sizes of 512, 256, and 128.

Using key feature points (KFPs) to optimizing the representation: Key Feature Points are SFV points that distinguish the object class. The KFPs for each class is calculated by computing the standard deviation of each SFV across the class and get the lowest ones. Their positions are stored in a KFPs vector. By detecting the similarity of items in the same class, the usage of KFPs enhances the retrieval accuracy significantly. The KFPs vector size is also considered carefully. If the KFPs vector's size is near to that of the SFV, the KFPs have a minor effect on extracting the essential characteristics of a class's shapes. In contrast, having too few KFPs is unfavorable since it captures only a subset of these characteristics, raising the risk of misclassification. There is no efficient method to figure out what size of KFPs vector is optimal. In this study, when the KFPs vector size is 0.1–0.3 the size of the SFV size, the best results are achieved.

The suggested methodology outperforms others in the literature, according to empirical observations. It also demonstrates how the Hilbert curve scan outperforms the traditional raster scan in terms of shape representation and description. The suggested approach is robust to rotations of up to 10 degrees and occlusions of up to 15%. The results of experiments also reveal that the

appropriate wavelet approximation level varies between different datasets and SFV sizes. More study is needed to identify the dataset properties that can determine these variables automatically.

3. **Hilbert Curve method for mapping high level Internet scale traffic** (B. Irwin and N. Pilkington, 2007)

The authors outline a tool to provide high-level Internet-scale network traffic visualization. The demand for a tool that could show massive amounts of IP data gathered at network telescopes while also conveying the semantic and sequential connection between the nodes representing networks motivated the creation of the tool. This study was built on the implementation of Hilbert curve mapping IP address methodology, which was then improved to generate higher-order and hence finer-grained curves. The levels 4, 8, 12, and 16 of the Hilbert curves are particularly intriguing since they have 256, 65536, 16,777,216, and 4,294,967,296 points, respectively. Within the range of Ipv4 addresses space, these values are the number of addresses in the network blocks grouping by Class A (/8), Class B (/16), and Class C (/24). The design for visualizing class A network blocks using a fourth-order curve is shown in the figure below. The top left and top right corners of the curve are mapped to 0.0.0.0/8 and 255.0.0.0/8, respectively [Figure 8].

Approach: C++ and OpenGL are used as tools to develop the implementation of the Hilbert curve-based layout system. The code is used to map a certain substring of a dotted-quad IP address representation to a certain position on the grid being generated. Only the curves of orders 4, 8, and 12 are implemented due to the limitations on screen size, memory constraints, and complexities of the graph. Simple text files with a single dotted-quad IP address per line are used as the input to the system. A hash table is implemented to contain the IP addressed. It allows the quick retrieval of the required IP addresses from the table to display in the plot. The nodes are created iteratively in the software using the Lindenmayer System representation of the Hilbert - this is the most basic representation that enables the rapid and precise creation of the curve. Additionally, the curve's nodes are generated in numerical order, allowing points to be displayed at the same time as the curve is formed, saving time and effort. Then, graphical output is produced, with nodes colored as containing elements aggregated based on the curve's order.

Result: It takes 60 minutes to render a plot displaying 63 million individual addresses on an order 12 curve (equivalent to /24 network buckets each holding 256 individual IP addresses). The improvement in system performance is required to perform additional zooming and navigating actions. Some overview plots are generated to allow sanity checking or provide a brief summary of input data. The below figures [Figure 5, Figure 6] plot 2.4 million unique data points in the 8th order curve (/16 bins) and 12th order curve (/24 bins) respectively.

After generating the traffic plots, further analysis of the generated pictures is available. Displaying the aggregation of close networks [Figure 7] is one of the advantages of the proposed method. This makes it considerably easier to see the logical and distinctive connection between

these nodes than other classic grid-based and wrapped linear plotting approaches. Further messages about the displayed nodes could be delivered by using color-index-based mappings as a way of colouring graph nodes. The authors also suggest some useful additional features to the toolsets such as IPv6 support, HUD capability, overlay mode, time sequences view and the geographical mapping. The tool presented is beneficial in delivering high-level overviews of massive amounts of traffic while keeping the input data's sequence order. It enables the creation of summative reports from massive amounts of network traffic in terms of pure Information Security perspective.

4. **Visualization of genomic data with the Hilbert curve** (Simon Anders, 2009)

Many genomic studies are interested in genome-position-dependent data. In the case of the ChIP-Seq technique, its scores might be stored in an extremely long vector. Visualizing such a vector to gain a deep understanding of the data is in great demand. A conventional method is implementing a genome browser to process the data, present users with the genes in their interest, and verify if the data meets their expectations. However, this could take a lot of time since users have to look at a wide range of loci to obtain an overview picture from representative samples. Furthermore, if one simply looks at one item at a time, it is difficult to highlight important characteristics of the arrangement and spacing of the features. This study introduces an approach to display the entire chromosome while also allowing access to specific information: using Hilbert Curve to map data from a one-dimensional chromosomal layout to a two-dimensional form.

In the conventional method, a vector of length 1.8 million is visualized as in the Figure 9. It proves challenging to draw any conclusion from this plot since the peaks are all similar and merged. A possible solution is producing many zoomed-in charts, such as the one shown in the Figure 10. An example using the proposed method in this study implements an 8th order Hilbert Curve to represents bins of 1.8 million elements with 256 x 256 pixels [Figure 11]. Each bin contains about 27 consecutive vector elements. The color of a pixel is based on the maximum value within the bin. Due to the symmetric properties of the Hilbert Curve, the appearance of the visualization could help users deduce properties of the spacing in the data.

Due to the locality-preserve properties of the Hilbert Curve, bins on the data vector that are near to each other are represented in the plot by pixels that are near to each other. Adjacent bins, in particular, are mapped to adjacent pixels. Therefore, users can approximate the location of peaks in the vector by examining the dark spots in the figures; they can also study the height of peaks by looking at the level of darkness of the spot. However, some pixels in close proximity would inevitably correspond to distant elements on the vector. In comparison with Peano's original curve or Z curve, the Hilbert Curve is considered as the optimal space-filling curve that minimizes these distortions. In addition, another drawback of this technique is that it is difficult to relate a specific plot point back to a vector position.

One of the use cases the author mentions is implementing the proposed method with the ChIP-Seq data for histone methylation (H3K4me1 and H3K4me3). It is believed that the peaks of H3K4me1 are considerably more scattered and less clearly localized. While the peaks of H3K4me3 are known to be narrow, sharp, and extremely high. The author states that the Hilbert Curve visualization successfully supports users in the rapid recognition of stated facts and the formation of further tests. Besides, assuming the user has a basic idea of what to expect, the visualization could be utilized to provide an initial quality assessment of the data. The author believes that the proposed method could be implemented in previous studies of the Hilbert Curve in genomics visualizations (Deng et al., (2008); Wong et al., (2003)).

Conclusion

In conclusion, this literature discusses the practical applications of the Hilbert Curve. It is evident that there is a wide range of implementations of the Hilbert Curve in different computer science areas such as computer vision and image processing, networking and cybersecurity, bioinformatics, and so on. Most of the studies exploit the clustering and locality-preserve properties of the Hilbert Curve to generate desired visualizations that meet different requirements of interest. However, more research needs to be conducted to see the overall efficiency of the Hilbert Curve approach in comparison with traditional approaches

References

1. Mokbel, Mohamed & Aref, Walid & Kamel, I. 2003. Analysis of Multi-Dimensional Space-Filling Curves. *GeoInformatica*. 7. 179-209.
2. D. Hilbert. 1891. Über die stetige Abbildung einer Linie auf ein Flächenstück. *Mathematische Annalen*. 38. 459–460.
3. M. Trimeche, M. Tico and M. Gabbouj. 2006. Dense optical flow field estimation using recursive LMS filtering. *14th European Signal Processing Conference*. 1-5.
4. Luiz Velho and Jonas de Miranda Gomes. 1991. Digital halftoning with space filling curves. *SIGGRAPH Comput. Graph*. 25, 4. 81–90.
5. Jonas Valantinas. 2005. The use of space-filling curves in changing image dimensionality Inform. Technol. Control, 34.
6. Zhexuan Songa, Nick Roussopoulosb. 2002. Using Hilbert curve in image storing and retrieving Inform. Systems. 27. 523-536.
7. Yasser Ebrahim, Maher Ahmed, Wegdan Abdelsalam. 2009. Siu-Cheung Chau, Shape representation and description using the Hilbert curve, *Pattern Recognition Letters*. 30, 4. 348-358.
8. Wong PC, et al. 2003. Global visualization and alignments of whole bacterial genomes. *IEEE Trans. Vis. Comput. Graph*. 9. 361-377.
9. Deng X, et al. 2008. DHPC: a new tool to express genome structural features. *Genomics*. 91. 476.

10. Simon Anders. 2009. Visualization of genomic data with the Hilbert curve,. *Bioinformatics*. 25. 10. 1231–1235.
11. Irwin, Barry & Pilkington, Nick. 2007. High Level Internet Scale Traffic Visualization Using Hilbert Curve Mapping. 147-158.

Appendix

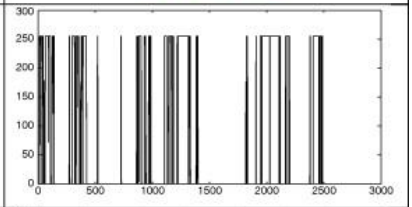
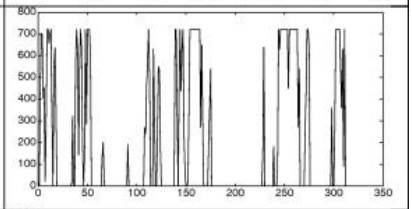
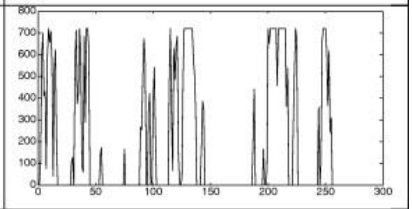
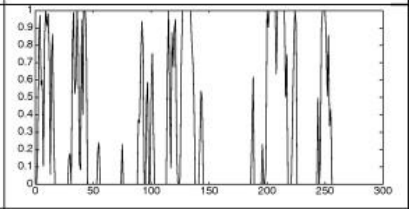
Step	Sample output
Original image	
Crop and HC scan image	
V	
Approximate to produce WV	
Sample WV to obtain SWV	
Normalize SWV to obtain SFV	

Figure 1

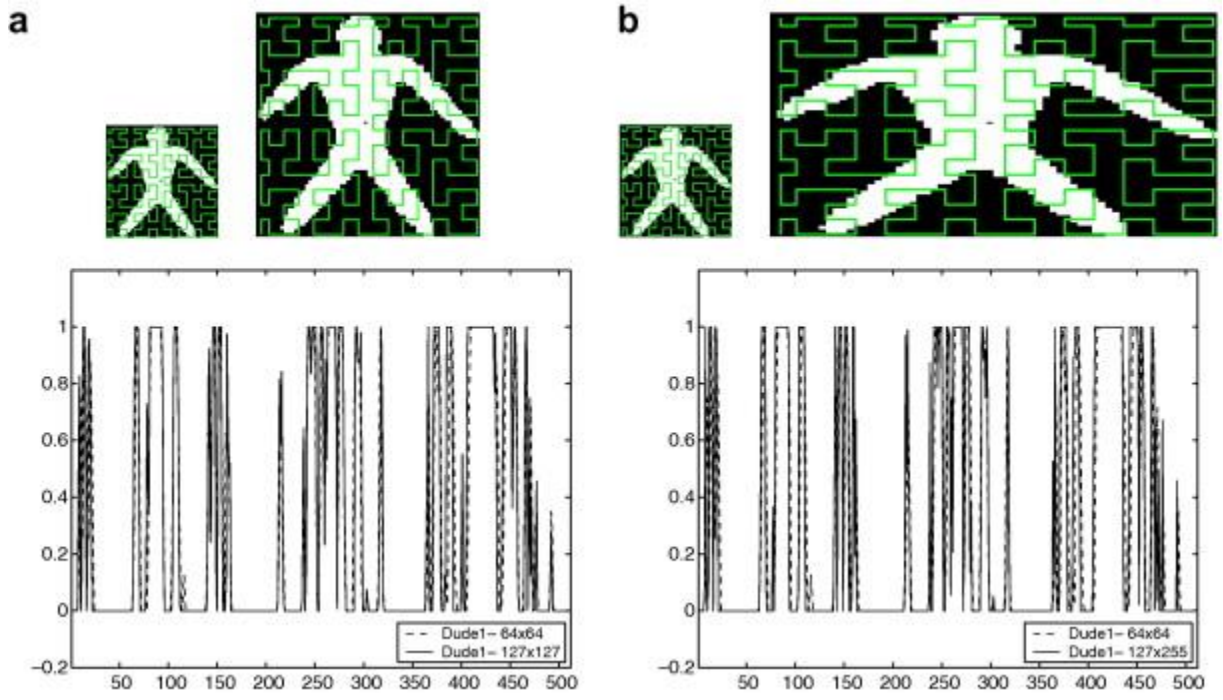

a
b
1
0.8
0.6
0.4
0.2
0
-0.2
50 100 150 200 250 300 350 400 450 500
Dude1- 64x64
Dude1- 127x127
Dude1- 64x64
Dude1- 127x255

Figure 2

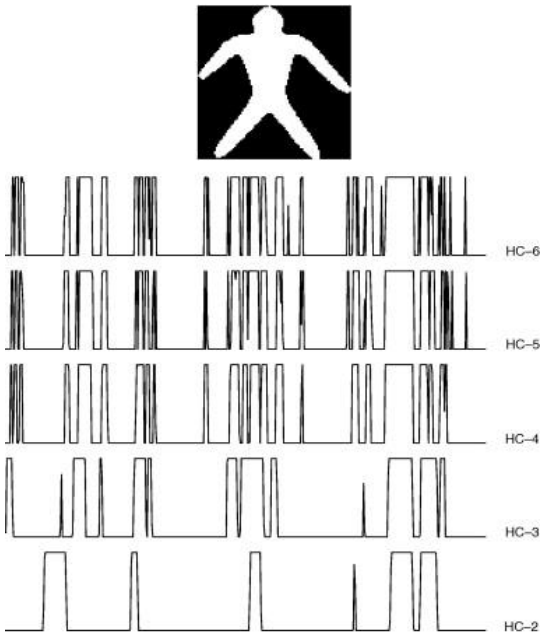


Figure 3

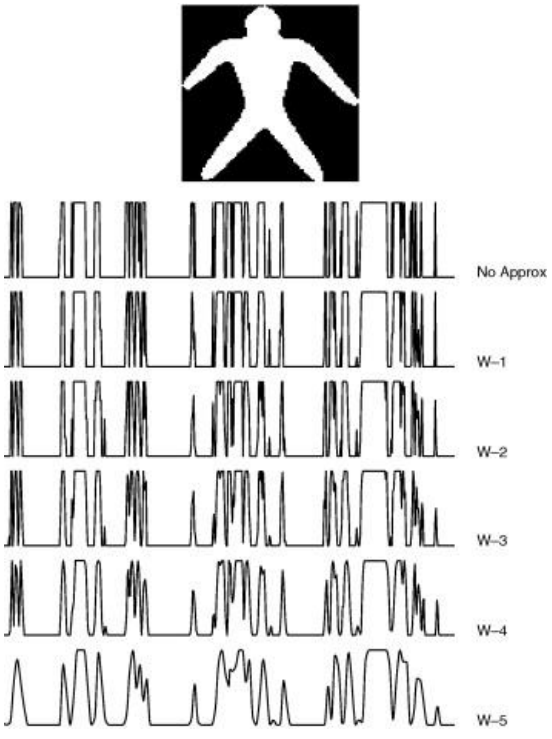


Figure 4

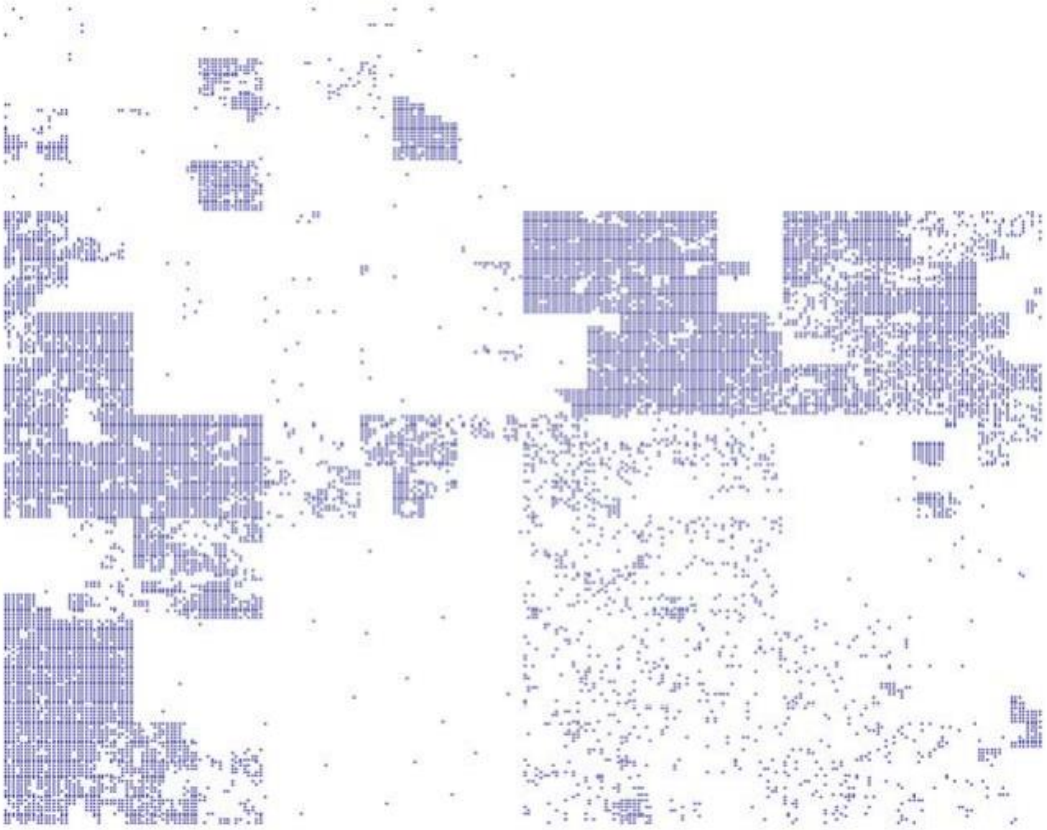


Fig. 5 Plot of eighth order representation of the data in Fig. 4, showing buckets corresponding to /16 networks (Class B)

Figure 5

Fig. 6 Plot of data on a 12th order curve. Points correspond to buckets of size/24 (Class C). The boxed area relates to the zone of discussion in Fig. 7, and represents the network ranges from 23.232.0.0/13 to 24.170.0.0/16

Figure 6

Fig. 7 Close up of the view of the 23.232.0.0/13 and 24.0.0.0/8 networks using /16 buckets (A – left), and then with a finer resolution of /24 buckets in B (right) showing a distinctly different pattern. This area is shaded in Fig. 6

Figure 7

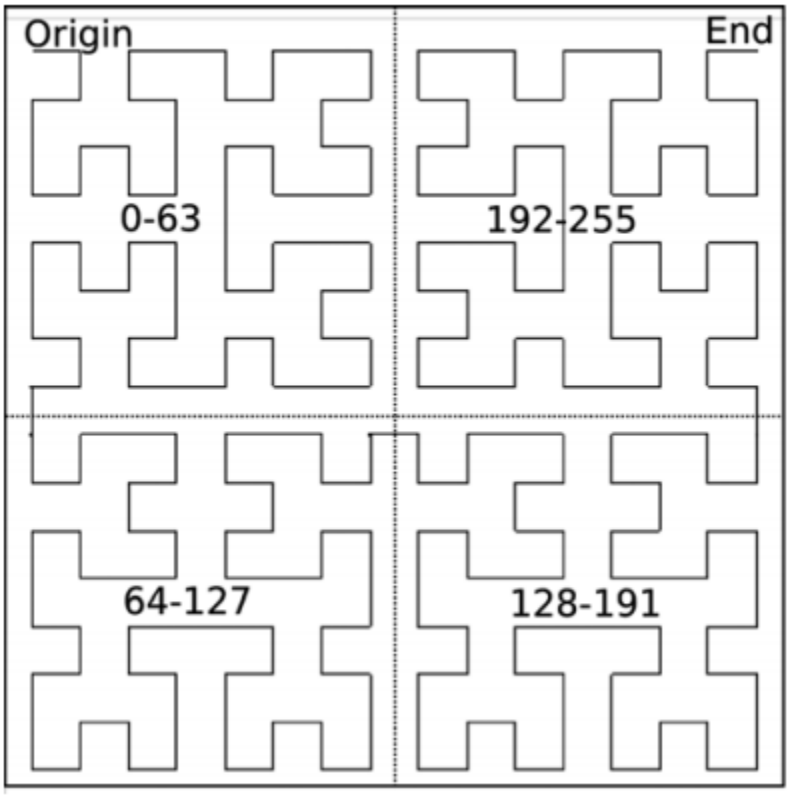


Figure 8

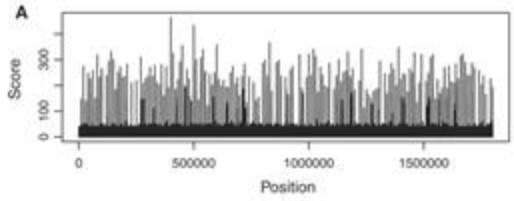


Figure 9

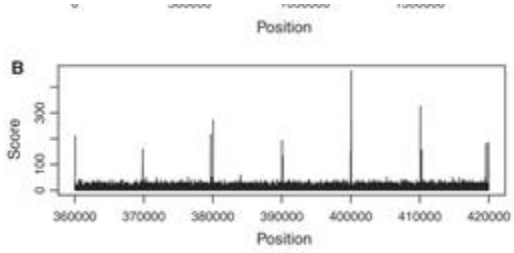


Figure 10

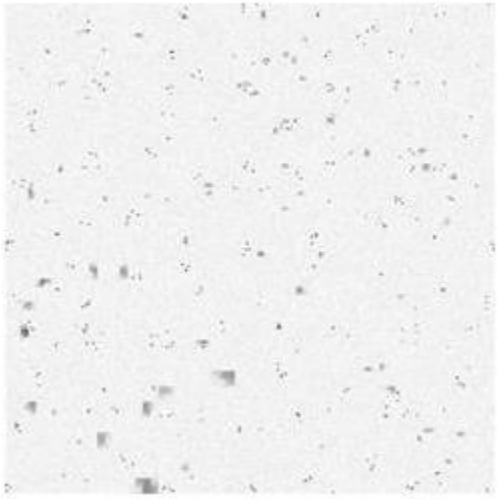


Figure 11