On Data Compaction of Scanning Curves

A. QUIN[1] AND Y. YANAGISAWA[2*]
[1] *Department of Computing Science, Paisley College of Technology, High Street, Paisley, Renfrewshire PA1 2BE*
[2] *Department of Mathematics, Statistics and Operational Research, Trent Polytechnic, Burton Street, Nottingham NG1 4BU, England*

Data compaction, or making better use of existing data storage and transmission, is one of the most important matters in computer and TV graphics. It has been suggested that different scanning techniques would improve data compaction for 2-D television images.

We prove that there is no difference in the number of run lengths found for any scanning curves for completely random pictures. Simulation is used to compare the data compaction between a discrete Peano curve and a bidirectional scanning curve for 1000 randomly placed ellipses to determine whether spatial coherence favours one scanning curve. The results are analysed statistically. Both the theory and the results of the simulation show that there is no difference between the scanning curves chosen. It is important to note that the bidirectional scanning curve is a special case of a mixed radix method.

Received July 1987, revised October 1987

1. INTRODUCTION

Data compaction is one of the most important factors in the transmission and storage of data. This is especially true when considering the amount of data required to reconstruct two-dimensional TV and computer graphic images. If a picture can be transmitted using the minimum amount of data and subsequently reproduced without any loss of quality, transmission costs can be minimised, either by using a lower bandwidth link or by using the same link for a shorter period of time.

Several studies on data compaction have been made. These may use either hardware or software techniques. Hardware data compaction techniques are covered in Clare.[1] Software techniques may be classified as either raster scan or tree techniques (Samet[9] and Gargantini *et al.*[7]). Raster scanning techniques may use a variety of scanning curves, which include linear raster scan, discrete Peano curves (Peano, Hilbert and Sierpinski models) and mixed radix. Witten and Neal[12] and Cole[3] suggest using discrete Peano curves (the term will be defined in Section 2) instead of traditional linear raster scan. Cole[4,5,6] suggests that a discrete Peano curve scanning technique might achieve 60–80% reduction in data compaction over the present linear raster scan system. Other data compaction techniques use trees, for example quadtrees. Samet[9] reviewed quadtrees and their data compaction. Samet and Tamminen[10] showed that the minimum required data storage is 2^{m+2} for a pointerless quadtree, where the screen size is $2^m \times 2^m$. Perhaps this may be one of the best quadtree techniques in view of data compaction. It is interesting to note that some quadtrees are a special restricted case for the Hilbert model of a discrete Peano curve.

In this paper we investigate the amount of data compaction that can be achieved by using scanning curves (Hilbert model). Numerical investigation is done by comparing the Hilbert model of the discrete Peano curve suggested by Cole with a bidirectional linear raster scanning system often found in printer-plotters. It is found that there is no difference between scanning curves when discussing data compaction.

* To whom correspondence should be addressed.

2. DATA COMPACTION FOR COMPLETELY RANDOM PICTURES

One property of Peano curves is that they are a continuous mapping from the real line $I = \{x : x \in R, 0 \leqslant x \leqslant 1\}$ to the unit square $E^2 = \{(y, z) : y, z \in R, 0 \leqslant y, z \leqslant 1\}$. This function has multiple points almost everywhere on the unit square, E^2, so the inverse function does not exist, i.e. it is a many–one function. Applying Peano curves to the raster device, we will only use a finite set of discrete points from the line I and a finite set of grid points or pixels from the unit square E^2. These sets are defined by $J = \{x_i : i = 1, \ldots, n\}$ and $K = \{(y_j, z_k) : j = 1, \ldots, n_1, k = 1, \ldots, n_2\}$. In this case a finite procedure of Peano curve is not Peano curve, so this finite procedure of that may be defined by a discrete Peano curve. If $n_1 \times n_2 = n$ and $n_1 = n_2 = 2^m$, we can find a discrete Peano curve that will pass through all the pixels once and once only. The discrete Peano curve gives a one-to-one and onto mapping from J to K, that is, it is an isomorphism. Note that two points on J, x_i and x_j, corresponding to two neighbouring pixels (y_i, z_i) and (y_j, z_j) on K, may be widely spaced on the line J, hence x_i and x_j may have many points lying between them. It follows that a coherent area will require several runs to encode it. This characteristic is not only a peculiarity of discrete Peano curves, but also of any scanning system.

In order to introduce the theorem, we define the word 'scanning curve'. The scanning curve through all the points in K (all the grid points or pixels) must pass through any pixel once and once only. This is a one-to-one and onto mapping from set J to set K. These scanning curves include all existing scanning systems in use such as linear raster scan and bidirectional scanning systems as well as discrete Peano curves (Peano model, Hilbert model and Sierpinski model) and mixed radix systems.

Consider two different scanning curves whose corresponding functions are f and g.

$$f : J \to K$$
$$g : J \to K$$

These are one-to-one and onto mappings, i.e. iso-

21-2

morphisms. So the inverse function f^{-1} and g^{-1} exists and the composite function

$$f^{-1}g : J \to J$$

also exists. Hence $f^{-1}g$ is a one-to-one and onto mapping from J to J.

We now restrict our discussion to monochrome raster displays. Each grid point or pixel on the display can have the value 0 or 1. Any picture is composed of 0's and 1's at the grid points corresponding to each pixel.

Because f, g and $f^{-1}g$ are isomorphisms, each possible pattern of 0's and 1's on K can be mapped to a unique pattern on J, and all possible patterns on K are mapped to all possible patterns on J. It follows that the expected number of runs is the same for a completely random picture whatever the scanning curve chosen. In other words, there is no difference between scanning methods used in terms of data compaction. Therefore we have the following theorem.

Theorem

The probability distribution of run length in a completely random picture is independent of the scanning method chosen. Also the probability distribution of the number of runs in a completely random picture is independent of the scanning method used.

This theorem is also true for colour raster displays, if the number of colours involved is finite. In the above, all the possible pictures or patterns of 0's and 1's on K are considered, but some of them have no information content for TV scan and computer graphics. Confining our discussion to meaningful aggregated pictures, we can consider data compaction under different scanning curves. The theory of the number of runs corresponding to data compaction is discussed by Wilks.[11] However, this theory is not relevant when applied to random aggregated pictures using any scanning curves, because these pictures form only a subset of all possible pictures. In order to compare the number of runs for each scanning curve, we generated random aggregated pictures on the screen, then applied selected scanning curves.

3. DATA COMPACTION FOR AGGREGATED PICTURES

It is impractical to check data compaction for all possible aggregated pictures on the set of points K, because of the limitations imposed by practical computer systems. We restricted ourselves to pictures constructed from a single random ellipse. The size of each ellipse and its position on the display surface were determined using uniform pseudo-random numbers. Each pixel or grid point in K lying inside or on the boundary of an ellipse is assigned the value 1, otherwise it is given the value 0. The number of runs is counted for each scanning curve used for each simulation. In order to avoid generating ellipses which are either too small or too large for the display surface,

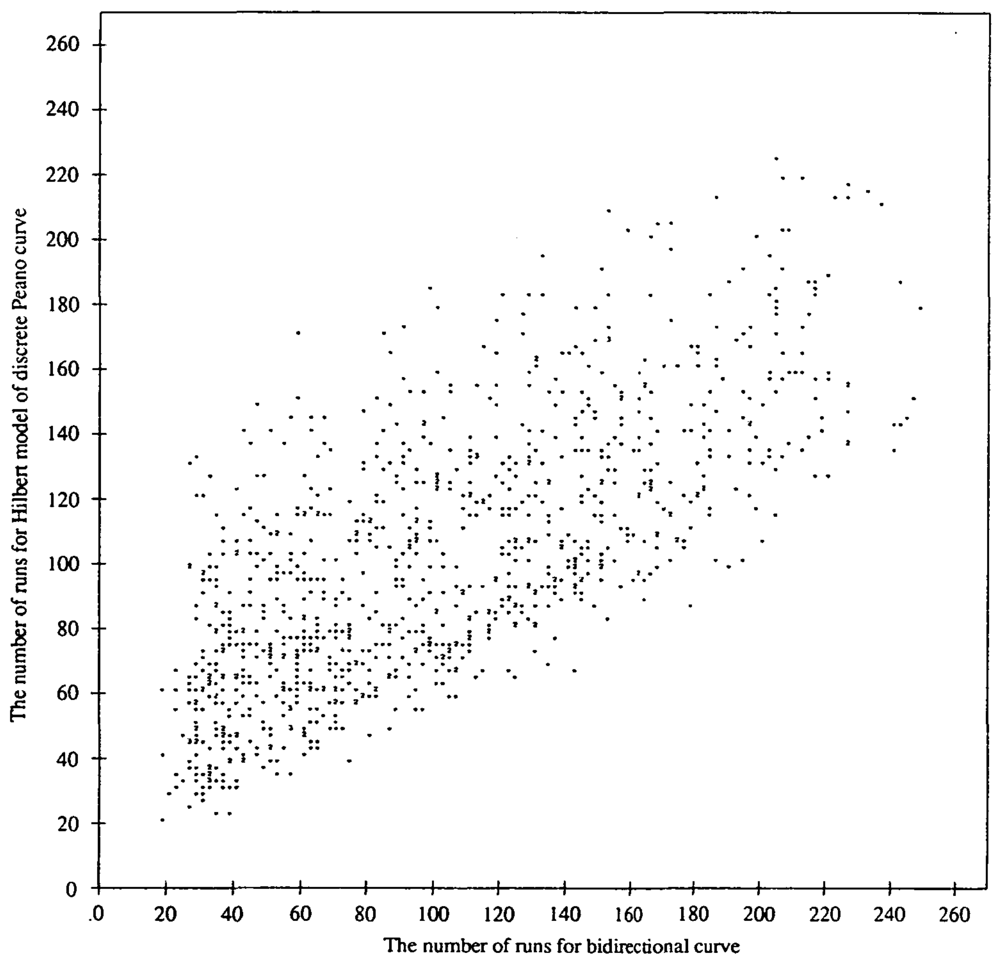

Figure 1. The dot plot for the number of runs for the same 1000 simulations.

Interval			Mid-point	Freq.	Rer. freq.
-110.0000	~	-100.0000	-105.0000	1	0.001
-100.0000	~	-90.0000	-95.0000	9	0.009
-90.0000	~	-80.0000	-85.0000	6	0.006
-80.0000	~	-70.0000	-75.0000	15	0.015
-70.0000	~	-60.0000	-65.0000	27	0.027
-60.0000	~	-50.0000	-55.0000	37	0.037
-50.0000	~	-40.0000	-45.0000	62	0.062
-40.0000	~	-30.0000	-35.0000	82	0.082
-30.0000	~	-20.0000	-25.0000	72	0.072
-20.0000	~	-10.0000	-15.0000	84	0.084
-10.0000	~	0.0000	-5.0000	99	0.099
0.0000	~	10.0000	5.0000	109	0.109
10.0000	~	20.0000	15.0000	90	0.090
20.0000	~	30.0000	25.0000	83	0.083
30.0000	~	40.0000	35.0000	77	0.077
40.0000	~	50.0000	45.0000	47	0.047
50.0000	~	60.0000	55.0000	40	0.040
60.0000	~	70.0000	65.0000	27	0.027
70.0000	~	80.0000	75.0000	13	0.013
80.0000	~	90.0000	85.0000	9	0.009
90.0000	~	100.0000	95.0000	7	0.007
100.0000	~	110.0000	105.0000	3	0.003
110.0000	~	120.0000	115.0000	1	0.001

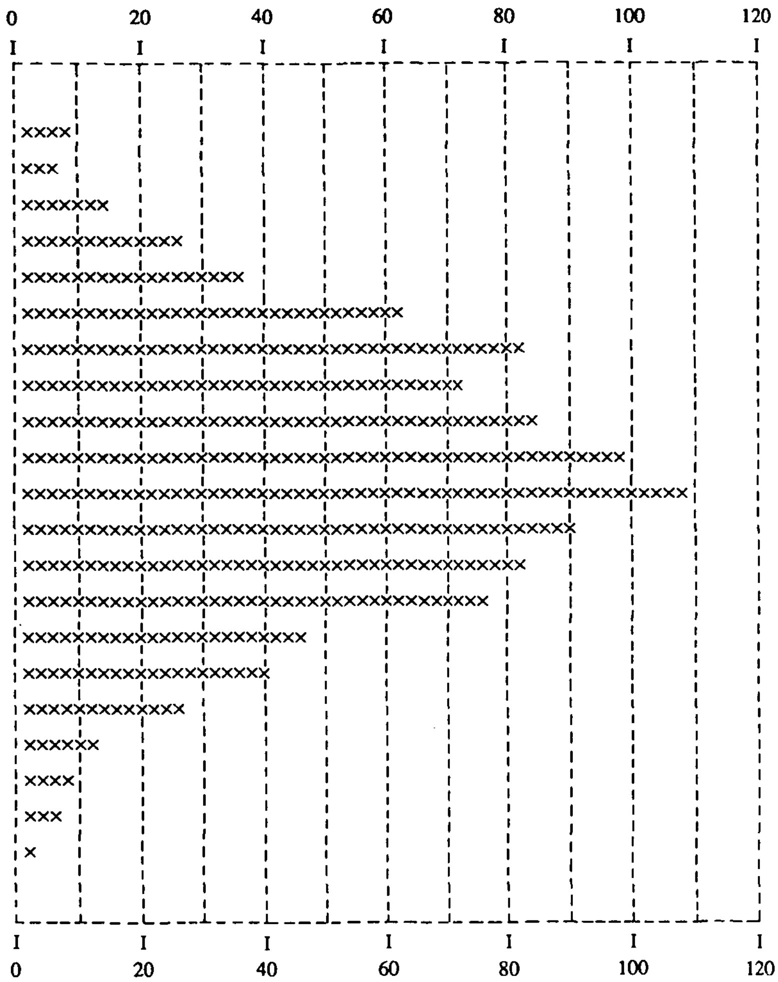

Figure 2. The frequency distribution for the number of runs (difference between the number of runs; Hilbert model of discrete Peano curve – bidirectional curve) for the same 1000 simulations.

the size of the ellipses is restricted to the interval 10% to 90% of the display surface, and the centre of the ellipses is randomly placed within the screen area. Restrictions in CPU time and memory reduced the number of simulations to 1000 ellipses and the screen size to 128 × 128 pixels. The scanning curves selected were the Hilbert model of the discrete Peano curve and the bidirectional linear raster scan. The algorithm used to generate the Hilbert model of the discrete Peano curve is proposed by Cole[3] and is written in S-Algol. The more universal equivalent C algorithm is given below.

```
hilbert (r, u, l, d, i)
char r, u, l, d;
int i;
{
  if (i > 0)
  {
  hilbert (u, r, d, l, i−1);
  move (r);
  hilbert (r, u, l, d, i−1);
  move (u);
  hilbert (r, u, l, d, i−1);
  move (l);
  hilbert (d, l, u, r, i−1);
  }
}
```

The algorithm is invoked by the call

```
hilbert ('e', 'n', 'w', 's', i);
```

For other space-filling curves see Goldschlager.[8] It has been suggested that the Hilbert model gives good data compaction,[4-6] and is better than the linear raster scan. If this is true, the number of runs for the bidirectional linear raster scan, one of the simplest scanning curves, would be expected to be large. It is interesting to note that bidirectional linear raster scan is a special case of mixed radix method. The procedure for simulation is as follows.

Procedure

(1) Four pseudo uniform random numbers lying in the range 0, 1 are generated u_1, u_2, u_3 and u_4.

(2) The centre of the ellipse is calculated from $(64\,(u_1+u_2),\ 64\,(u_3+u_4))$ and the major and minor axes X and Y are $128(0.05+0.4|u_1-u_2|)$ and $128(0.05+0.4|u_3-u_4|)$.

(3) The number of runs for each scanning curve is counted.

(4) Steps 1–3 are repeated 1000 times.

Each point in Fig. 1 is plotted from the number of runs produced by both scanning curves for the same picture. The number of runs produced by the Hilbert model of the discrete Peano curve is plotted against the number of runs produced by the bidirectional linear raster scan. This scatter diagram is almost symmetrical about the line $y = x$. Data compaction is better for the linear raster scan above the line $y = x$, and the Hilbert model gives better data compaction below. Closer examination of Fig. 1 shows that in the region bounded by $x = 20$–140 and $y = 20$–140, the points lying in the area above the line $y = x$ are scattered over this area, whilst the points below the line $y = x$ lie close to this line. It follows that when the number of runs is relatively small the Hilbert model is more sensitive to slightly changing pictures. The distribution of points about the line $y = x$ projected on the line $y = -x$ is given in Fig. 2. Points that lie below the mean in Fig. 2 give better data compaction for the Hilbert model and vice versa for points that lie above the mean. From Fig. 2 we calculated skewness and kurtosis and tested for normality of the distribution. The following results were obtained. Skewness = 0.026, standard deviation of skewness (sds) = 0.077, kurtosis = −0.245, standard deviation of kurtosis (sdk) = 0.155. Skewness/sds = 0.336 and kurtosis/sdk = −1.588, and they are normally distributed when the observations are taken from a normal distribution. Neither skewness nor kurtosis results enable us to reject normality at the 10% significance level. According to the sample mean (m) = −1.11, sample standard deviation (ssd) = 38.51 we accept normal distribution and test the population mean of Fig. 2 against 0. $(\sqrt{1000})\mathrm{m}/\mathrm{ssd} = 0.91$, so we cannot reject mean = 0 at the 10% significance level. Therefore statistically there is no difference between the number of runs for the Hilbert model and for the bidirectional linear raster scan.

4. CONCLUSION

According to the theory of Section 2 the number of runs corresponding to data compaction is independent of the scanning curve for completely random pictures. The Hilbert model of the discrete Peano curve and the bidirectional raster scan were compared by generating random ellipses. No significant difference could be found between them for data compaction. This comparison is not only of two scanning curves, but of over 30 different scanning curves which were tried. The number of simulations for the other scanning curves is small. However, we found that almost exactly the same conclusion as that in Section 3 was obtained. We also have to point out the following possibility. Our simulations depended on the random ellipses, and the size of the screen was 128 × 128. If the size of the screen was bigger and the pictures were not ellipses, but some other coherence areas, then the numerical results would perhaps have been changed. However, we think this possibility is slight. So both the numerical and the theoretical results show that there is no difference between scanning curves for data compaction. Data compaction has been the subject of many investigations using a range of techniques such as binary trees, quadtrees, octrees and scanning curves. Quantitative techniques for comparing data compaction are necessary, and the above approach is suggested. Simple scanning curves are recommended because there is no theoretical or experimental difference between scanning curves. The bidirectional or linear raster scan is adequate.

Another possibility was suggested by Cole.[5] According to Cole, ignoring short run lengths in any scanning method improves data compaction. Sometimes we require immediate transmission, and high visual quality is not of importance. In this case this method is very important; however, this results in picture degradation.

Acknowledgements

The authors are greatly indebted to Professor A. J. Cole for his valuable comments on a previous draft of the paper. The authors are also grateful to a referee and to Professor M. K. Crowe, Mr C. Irgens and Mr R. A. Shaw for their comments on a previous draft.

REFERENCES

1. C. R. Clare, *Designing Logic Systems using State Machines.* McGraw-Hill, New York (1972).
2. A. J. Cole, A note on space filling curves. *Software – Practice and Experience* **13**, 1181–1189 (1983).
3. A. J. Cole, A note on Peano polygons and Gray codes. *International Journal of Computer Mathematics* **18** (1), 3–13 (1985).
4. A. J. Cole, Multiple radix arithmetic and computer graphics. *Bulletin of Mathematics* **22** (5/6), 71–75 (1986).
5. A. J. Cole, Compaction techniques for raster scan graphics using space-filling curves. *The Computer Journal* **30** (1), 87–92 (1987).
6. A. J. Cole, Data compaction using Murray polygons. *Proceedings of Computer Graphics 87*, 185–193 (1987).
7. I. Gargantini, T. R. Walsh and O. L. Wu, Viewing transformations of voxel-based objects via linear octrees. *IEEE Computer Graphics and Applications* (Oct.), 12–21 (1986).
8. L. M. Goldschlager, Short algorithms for space-filling curves. *Software – Practice and Experience* **11**, 99–100 (1981).
9. H. Samet, The quadtree and related hierarchical data structures. *Computing Surveys* **16** (2), 187–260 (1984).
10. H. Samet and M. Tamminen, Computing geometric-properties of images represented by linear quadtrees. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **7** (2), 229–240 (1985).
11. S. S. Wilks, *Mathematical Statistics.* Wiley, New York, pp. 144–150 (1962).
12. I. H. Witten and R. M. Neal. Using Peano curves for bilevel display of continuous-tone image. *IEEE Computer Graphics and Applications*, 47–52 (1982).