EUROGRAPHICS '2000 / M. Gross and F.R.A. Hopgood
(Guest Editors)

Volume 19, (*2000*), *Number 3*

Context-based Space Filling Curves

Revital Dafner, Daniel Cohen-Or and Yossi Matias

Department of Computer Science,
Tel-Aviv University, Israel

Abstract

A context-based scanning technique for images is presented. An image is scanned along a context-based space filling curve that is computed so as to exploit inherent coherence in the image. The resulting one-dimensional representation of the image has improved autocorrelation compared with universal scans such as the Peano-Hilbert space filling curve. An efficient algorithm for computing context-based space filling curves is presented. We also discuss the potential of improved autocorrelation of context-based space filling curves for image and video lossless compression.

1. Introduction

A space-filling curve (SFC) is a continuous scan that traverses every pixel of an image exactly once (see Figure 1). SFC's are attractive to many image-space algorithms which are based on the spatial coherence of nearby pixels. As depicted in Figure 2, an image is scanned using a SFC. The resulting sequence of pixels is processed as required by the particular application. For instance, the sequence may be compressed using lossless or lossy compression, it may be processed for halftoning, analysis, pattern recognition or texture analysis, and it may be converted into an analog form and be transmitted through channels with limited bandwidth. To obtain the image after processing, the (possibly modified) pixel-sequence is placed back in a frame along the same SFC. In the above applications and others, it is important that the intraframe correlation in the image translates to a favorable autocorrelation within the pixel-sequence [4, 9, 2, 10, 1, 13, 6, 5].

The scan-line is a standard scanning method, which traverses a frame line by line. It is well known, however, that SFC's, which are defined recursively, end up with more favorable properties than the scan-line. Intuitively, the recursive nature of the SFC requires it to traverse neighboring pixels before moving to more distant ones, resulting in better exploitation of the two-dimensional locality. The most popular recursive SFC is the Peano-Hilbert curve, which has been considered for numerous applications (e.g. [9, 2, 10, 1, 13]). The Peano-Hilbert curve is particularly appealing as it has an inherently strong locality property: it never leaves its current quadrant, at any level of refinement, before traversing all the pixels of the quadrant (see [12]).

Lempel and Ziv showed [16] that, for images generated by suitably random sources, the entropy of the pixel-sequence obtained using the Peano-Hilbert curves converges asymptotically to the two-dimensional entropy of the image. Hence, compressing the sequence using the Lempel-Ziv encoder [15] results in an image compression scheme that is optimal in the information theoretic sense. Matias and Shamir considered [11] the relationship between the two-dimensional autocorrelation of an image and the one-dimensional autocorrelation of the pixel-sequence. They showed that, for first-order Markov isotropic images, the autocorrelation of the pixel-sequence is a function of the fractal-dimension ν of the SFC. Hence, the Peano-Hilbert curve for which $\nu = 2$ (highest possible), gives the best autocorrelation, compared with a random SFC (as in Figure 1) for which $\nu \approx 4/3$, and with the scan-line, for which $\nu = 1$ (lowest possible). These studies support the approach that recursive SFCs, such as the Peano-Hilbert curve would be a good choice as a universal SFC that would work well (statistically) for large families of images.

In this paper we propose the use of *context-based space filling curves*, that are to be computed so as to exploit inherent coherence in the image. That is, rather than relying on a universal SFC that works statistically well, our approach is to select a SFC that would work well for the particular im-

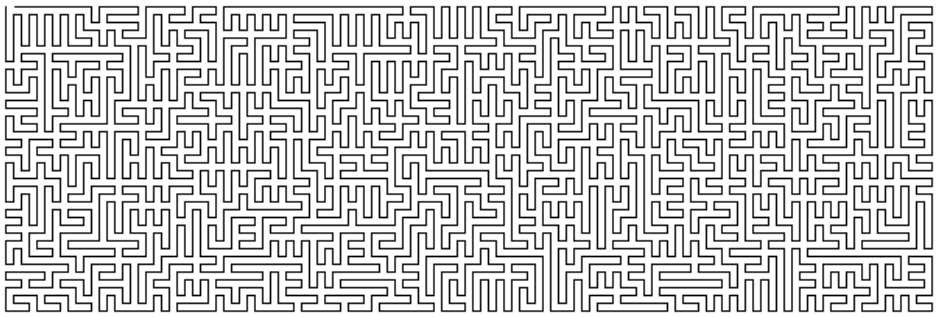

Figure 1: *A pseudo-random space filling curve (this image and others must be printed on a high resolution printer).*

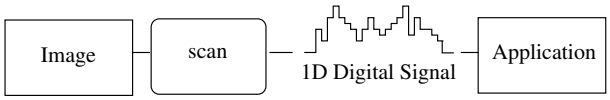

Figure 2: *The framework for image scan.*

age, or group of images, it needs to serve. To illustrate the motivation for a context-based SFC, consider the simple example of the black&white deer image of Figure 3. A favorable SFC would traverse the pixels within the deer image for as long as possible before moving across the border edges to traverse the pixels outside the deer. That is, the number of crossings should ideally be minimized. As can be seen, the Peano scan maintains locality but gives a substantial number of edge crossings. In contrast, a context-based SFC is tailored to avoid edge crossings to a considerable extent, resulting in a smoother pixel-sequence.

The advantage of context-based SFC is illustrated by the three scans of the car image in Figure 4. The pixel-sequences are displayed as a raster scan into the displayed frame, and consist of the following SFCs: a scan line (a) (hence showing the original image), Peano scan (b), and the context-based scan (c). As can be observed, the pixel-sequence resulting from the context-based SFC in (c) is smoother than the one based on the Peano curve (b). The first coefficients of the autocorrelation function of the three sequences are given in (d), demonstrating a clear advantage to the context-based SFC.

Getting back to Figure3, the context-based SFC may be quite intriguing: how can it be tailored to avoid crossing edges as it does, and why does it not do a better job in that? Ideally, we would have liked the context-based SFC to fill the entire deer before starting to traverse the background. Unfortunately, given an arbitrary shape, the problem of finding such a SFC is NP-hard, as implied by a hardness result on computing Hamiltonian paths on general grid graphs [8]. For practical purposes, it would suffice to find a SFC that tends to minimize the number of crossings, and it is essential that the computation of such SFC be efficient. Another consideration is that, unlike with universal SFCs, the selected context-based SFC needs to be encoded along with the pixel sequence, to enable retrieval at a later stage. In fact, it can be useful for applications where the image is stored scrambled in public, and the key is given in private. In other applications, such as compression, it would be desirable to minimize the size of the encoded SFC, implying an interesting tradeoff between its size and the advantage obtained in autocorrelation.

Finding a good context-based SFC can be related to the following approximation problem: The input picture is represented by a rectangular weighted grid graph, where vertices represent pixels and weights reflect differences between the corresponding pixel values. The objective then is to find a Hamiltonian path over the grid graph for which the total weight is small. Finding a path with the minimum total weight is the travelling salesperson problem, which is NP-hard, and for which known approximation are too slow to be considered in most applications of interest.

We present a simple, efficient algorithm for computing context-based SFCs. Given an image, the algorithm first defines a weighted grid graph over the image, so that the weight of an edge in the graph represents the resemblance between neighboring pixels near the edge. Then, a minimum-weight spanning tree is computed in near-linear time. Edges with weight that is relatively high for their neighborhood (corresponding to edges in the original image) would typically not be selected for the minimum spanning tree. Finally, the tree is replaced with a SFC which inherit the locality properties from the minimum spanning tree.

In Section 2 we introduce the algorithm for computing context-based SFC. In Section 3 we discuss the encoding of the curve. In Section 4 we discuss the autocorrelation and the redundancy of the context-based scans. We conclude with

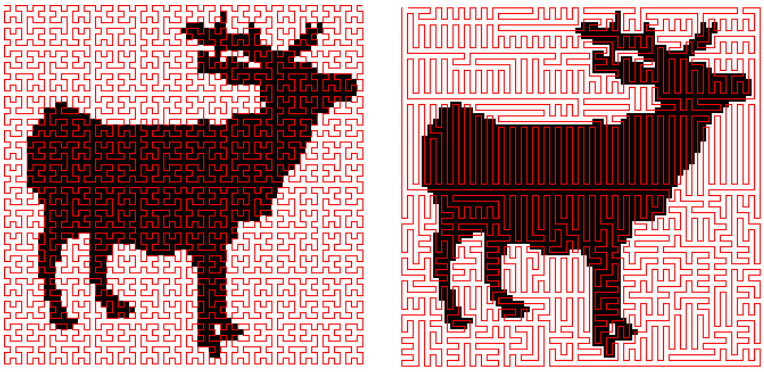

Figure 3: *Scanning a deer image using a Peano scan (left) and a context-based scan (right).*

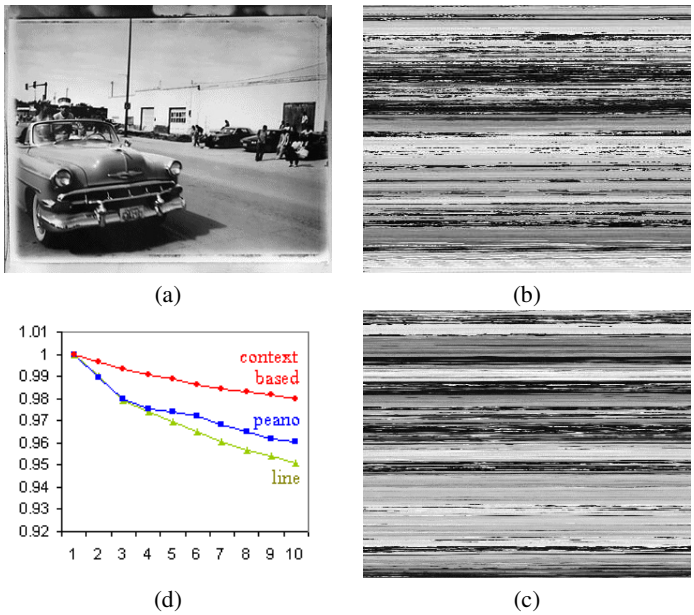

Figure 4: *Scanning an image of a car using a scan line (a), Peano scan (b), and context-based scan (c). The autocorrelation of the three scans (d): The horizontal axis is the distance (in pixels) and the vertical axis is the computed autocorrelation; the best autocorrelation corresponds to the context-based scanned image.*

remarks regarding future directions of research on context-based SFC.

2. Context-based Space Filling Curves

The image is considered as a directed graph whose vertices are the pixel's image, where each two adjacent pixels are connected by an edge. Our goal is to find an *adaptive context-based* curve that traverses every pixel of a given image. This is equivalent to the problem of finding a Hamiltonian path in an undirected graph [8]. Our algorithm utilizes a technique for generating pseudo-random SFC's that was developed to scramble a video signal as an encryption technique for applications requiring protected transmission of video [11]. The method of producing a Hamiltonian circuit in a grid graph is based on the concept of *cover and merge*. The property of a grid graph is required to guarantee the polynomial-time complexity of the Hamiltonian path construction. First, all vertices are covered by small disjoint circuits and then merged into a single Hamiltonian circuit. Let G be a grid graph. Choose disjoint external circuits that cover all the vertices of G . Two circuits A and B are adjacent if there are two edges e_1 in A and e_2 in B and two edges f_1 and f_2 in G that connect e_1 and e_2 to form a square, as shown in Figure 6(a). Merging two such adjacent circuits by replac-

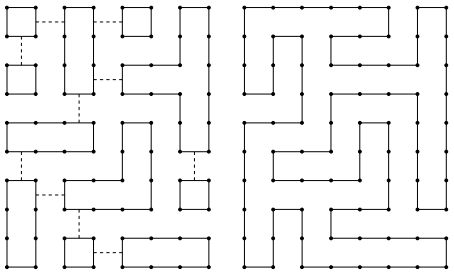

Figure 5: *The Cover and Merge concept: merging the MST (a) into a Hamiltonian cycle(b).*

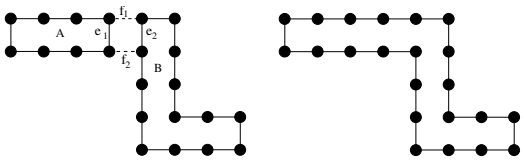

Figure 6: *The merge of two circuits.*

ing e_1 and e_2 with f_1 and f_2 (Figure 6(b)), creates a circuit which covers exactly the same vertices that A and B covered.

Define G' to be the dual graph of G, whose vertices are the initial circuits $C_1,...,C_k$, and its edges are the pairs (C_i,C_j) such that C_i and C_j are adjacent circuits. To merge all these circuits into one connected circuit, a spanning tree on G' is found. For each edge (C_i,C_j) in the spanning tree, merge the circuit C_i with C_j in G to yield a single circuit that spans all the vertices covered by the initial circuits (Figure 5(b)).

In the following we extend the above basic method to efficiently generate a context-based Hamiltonian path. Our aim is to construct a Manhattan-Hamiltonian path such that rearranging the pixels in the order defined by that path exhibits the inherent 2D coherence that exists in the image. Given an image with n columns and m rows, let G be the Manhattan grid-graph whose set of vertices V consists of all pixels in the image. Each pair of adjacent pixels, in a Manhattan grid graph, is connected by an edge, where even rows and columns are directed in the opposite direction to odd rows and columns, respectively. The reason we use a Manhattan graph is that it has a rather simple encoding, which will be discussed in Section 3.

First, the image is covered with a set of 1×1 circuits as shown in Figure 7(a). These initial circuits are called *small circuits*. Define a dual graph $G' = (V', E')$ such that each small circuit in G defines a vertex in G' and the two edges u, w connecting the two small circuits A and B in G define an edge v' in the dual graph (fig 7(b)). A weight is assigned to each edge v' in G'. The weight is the cost of merging the two associated circuits A and B in G, which correspond to the vertices A' and B' in G'. In other words, the weight between two given circuits A and B is the cost of exchanging the edges e and f with the edges u and w:

$$W(A,B) = |u| + |w| - |e| - |f|$$

where the cost of an edge is the difference between the values of its endpoints.

The algorithm builds a MST by iteratively merging circuits into a growing subtree according to the cost of the merge operation (Figure 8). After all the small circuits are merged into the growing tree, its edges form a Hamiltonian circuit that covers all the vertices in G with Manhattan orientation. The image is scanned along the cycle to form a highly correlated stream of pixels.

Finding the MST can actually be done in time complexity $O(N \log^* N)$ [7], where N is the number of pixels in the image. For practical reasons, in our implementation we use Prim's minimum spanning tree algorithm [3], with a heap for the priority queue which guarantees that the construction time of the MST is $O(N \log N)$.

The choice of weight function ensures that our adaptive curve fills the image area-by-area, where all pixels in a certain area have similar colors, thus preserving important two-dimensional information in a one-dimensional sequence of pixels arranged in the order defined by the curve. Figure 10(a) shows an example were the cost of merging circuit A with cirquit C and the cost of merging circuit B with C are equal. Yet, merging circuit A will result in a smoother curve, while merging circuit B destroys the sequence of black pixels along the Hamiltonian curve. To ensure the addition of merging circuit A we can use the weight function (see Figure 10(b)):

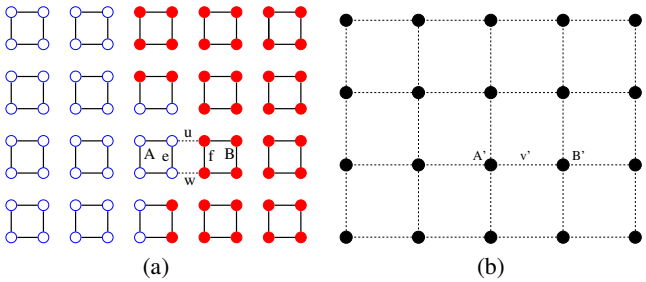

Figure 7: *The graph G of the image (a), and the dual graph G' (b).*

$W(C_i,C_j) = |u| + |w| + |x| + |y| + |z| - |e| - |f|$. In this function, the weight of an edge (C_i,C_j) is the cost of adding the vertex C_j to the spanning tree. Of course, other weight functions can be used to achieve that goal. Moreover, a suitable weight function can be chosen to improve other two dimensional properties. For RGB color images, the weight of each edge $e = (a,b)$ is $w(e) = |a_{red} - b_{red}| + |a_{green} - b_{green}| + |a_{blue} - b_{blue}|$ so that one curve is computed for the Red, Green and Blue components.

3. Encoding the Space Filling Curve

The main advantage of a context-free SFC is that there is no need to explicitly store the order of the scan. Both the encoder and the decoder agree on the next pixel to be processed. In a Peano-Hilbert curve the next pixel to be traversed can be expressed as a function of the coordinates of the current pixel. In a context-based scan the next step is determined by the context of the current pixel and its neighbors. The penalty is that the path along which pixels are encoded must be stored explicitly. In a general grid Hamiltonian path, two bits-per-pixel are necessary to indicate which of the three possible directions is next. Note that there are only three possible directions, as the entry direction cannot be chosen. However, for the restricted, yet large family of Manhattan-Hamiltonian paths, the cost of representing the path is just one bit per pixel. Only one bit is needed to indicate whether the next step is vertical or horizontal. The directions, up, down, left, right are directly implied from the position of the current pixel.

One important enhancement is to encode the MST instead of the path. Instead of the one-bit representation of the Hamiltonian path, it is possible to directly represent the spanning tree of the dual graph using only three bits for each vertex in the dual graph. Since each vertex in the spanning tree has up to three sons, the three bits can indicate which sons of the current vertex exist, where their direction is implied from the traversal of the tree. Thus, the cost of representing the path is 3/4 of a bit per pixel, since it requires three bits per a vertex in the dual graph. This requires a minor change in the algorithm. Instead of encoding the path, the spanning tree is encoded as triples of bits. In the decoder, the path is reconstructed from the spanning tree structure.

Note that the cost of the representing the curve is independent of the dynamic range of the pixels and it has a fixed size. This implies the relative cost of encoding the curve is decreasing when the dynamic range of the pixels is higher. For example, in many digital medical media, like MRI or CT, a pixel dynamic range is 12 bits.

The Manhattan orientation of the graph G might force the addition of some boundary-crossing circuits to the spanning tree, as illustrated in Figure 9(a), where the addition of the small circuit which destroys the sequence of black pixels along the final curve to the spanning tree is inevitable. Note that, in this example, it is possible to have a Hamiltonian path that crosses only once from a black region to a white region and only once from white to black region. In such a case it is effective to "split" the basic problematic circuit into two parts, such that the new curve crosses the boundary from black to white once and from white to black once (see Figure 9(c)).

Therefore, when constructing the spanning tree over the dual graph and adding a vertex whose four closest neighbors are already in the tree, an additional check is required to decide whether to split the vertex horizontally or vertically. Finding the MST with that additional checking can be done in time complexity $O(N \log^* N)$. When allowing a vertex to be split into two vertices, the Manhattan orientation of the curve is destroyed. Yet, the order in which the pixels are scanned is unique, and can be recovered from the spanning tree simply by checking the number of fathers for each square. A square with two fathers is broken into two parts.

4. Autocorrelation and Redundancy

The proposal for context-based space filling curve is primarily motivated by the proposition that a curve tailored for a given image would better exploit its spatial coherence than a universal curve. To support this proposition, we compare the autocorrelation and redundancy of 1-D pixel-sequences generated by both context-based and Peano-Hilbert space filling curves. The pixel-sequences were generated for the five pi-

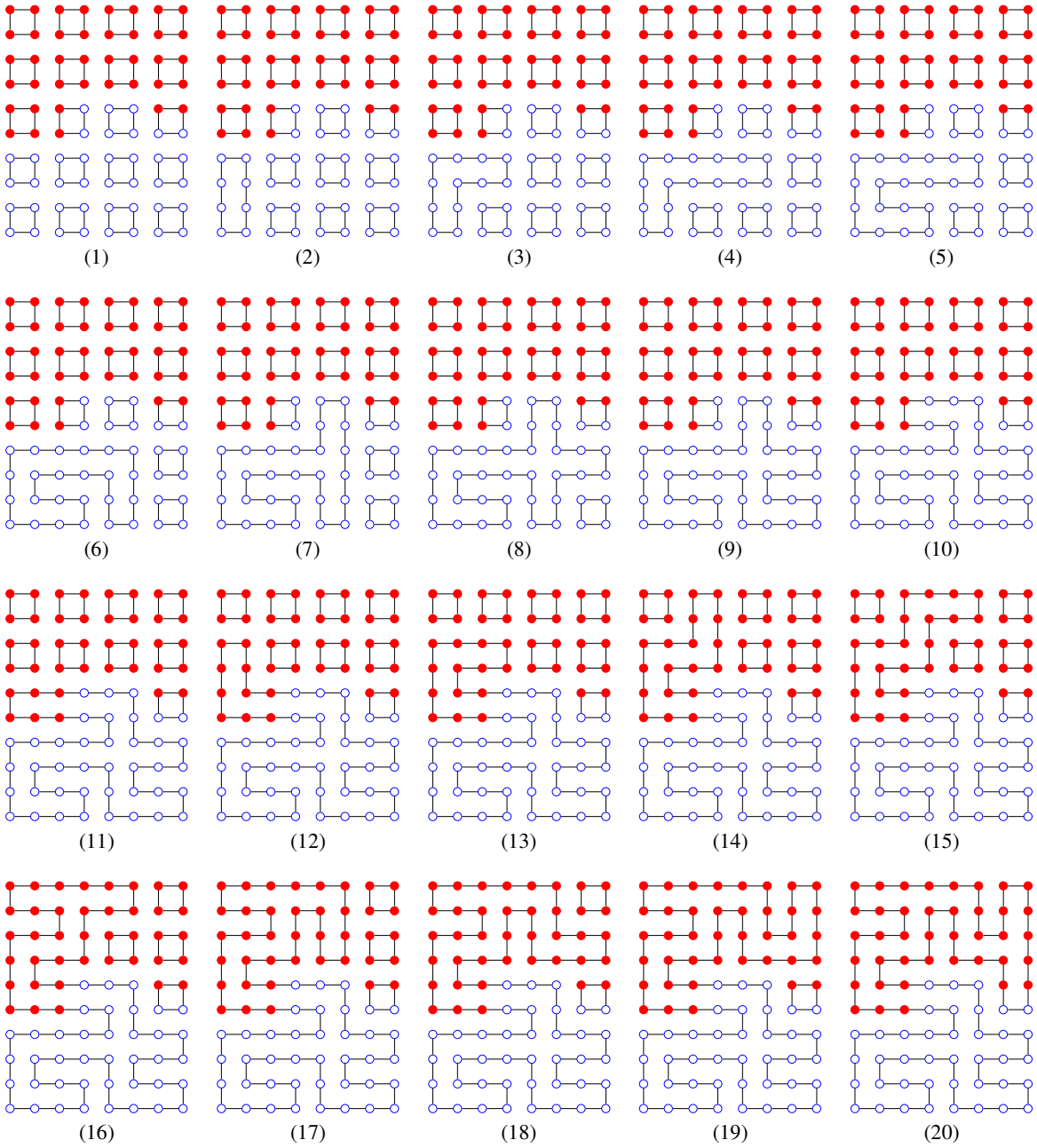

Figure 8: *Generating a context-based SFC by merging small circuits.*

cures in Figure 11, and their average autocorrelation is displayed.

As expected, the autocorrelation of context-based scans is better than the one for the Peano-Hilbet scans. For comparison, we also display the average autocorrelation of scan-line, which is inferior to both.

As a complementary approach to evaluate the redundancy of the pixel-sequences, we use a Lempel-Ziv encoder (LZW) [15, 14], whose performance is dependent on the redundancy of the sequence. The pixel-sequences of 2D images were positioned using line scan to form pictures (as illustrated in Figure 4), and were compressed using GIF encoder, which effectively executes an LZW encoding to the sequences. The results, depicted in Table 1, show an increased redundancy at the range of 5-10% over Peano-Hilbert scans. Similar im-

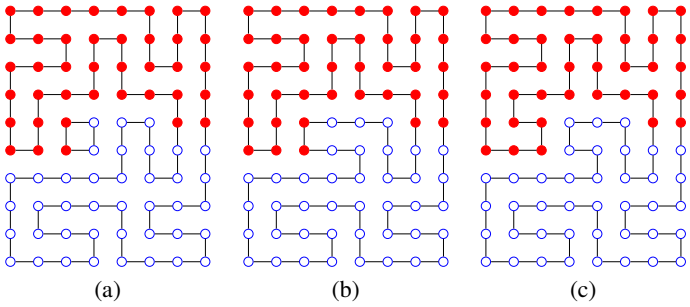

Figure 9: *Splitting a circuit into two.*

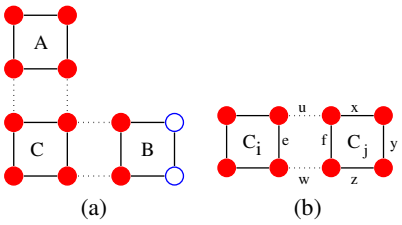

Figure 10: *The improved weights.*

provements were observed for other Lempel-Ziv encoders (e.g. Unix gzip).

To evaluate the concrete advantage, one has to also consider the cost incurred by the representation of the associated curve. When adding the size of the encoded curve, the gain over the Peano scans is lost: the advantage of 5-10% over the Peano-Hilbert curve becomes a disadvantage of 4-7%. However, for a sequence of correlated images, such as video sequences or volume slices, taking the XOR of two context-based curves corresponding to successive slices can reduce the cost of the curve. Since context-based SFCs of successive slices are coherent, their pixel-to-pixel XOR would result in many 0's and few 1's, which has a low entropy. To apply the compression on a sequence of frames/slices, the first frame is compressed along its context-based curve. Then for each of the other frames, the corresponding curve is computed and its XOR with that of the previous frame is stored compressed, along with the compressed sequence of pixels. Compression results of sequences of video frames show an improvement of 1-2% over the line-scan compression. Two frames of two video-sequences, 32 frames each, frame size 240×360, are shown in Figure 12. It should be emphasized again that the compression is lossless.

5. Conclusions

We have presented a context-based scanning technique for images which exploits the spatial coherence of the images better than the conventional line scan, and the well studied Peano-Hilbert space filling curve. There are many applications that operate on an image by first projecting the image into a 1-D sequence, and then processing it sequentially. Space filling curves are standard means for such projection; they translate the 2-D spatial coherence in the image into a 1-D autocorrelation in the sequence. The use of Peano-Hilbert curve in such applications was studied extensively, due to their favorable universal property. The increased autocorrelation of context-based space filling curves is of potential advantage to those applications. We learned that using context-based scan for 2D-image compression can improve the compression ratio of the pixel-sequence alone. However, in 3D case the cost of the curve is amortized over a number of frames and the overall compression can be better than conventional lossless compressions.

In some applications, where the curve encoding can be given implicitly or by other means, the compression can be effective. For other applications such as halftoning, pattern recognition, or texture analysis, the cost of the curve is irrelevant, and the potential advantage is under study.

The balance between the improved quality of the 1-D sequence and the cost of the context-based curve introduces an interesting trade-off. One can compute more limited context-based scans, which only partially exploits the spatial coherence, and which on the other hand incur reduced cost. For instance, a curve, which would use mostly rigid patterns (e.g., straight line), except when it is sufficiently advantageous to change its course, can be represented more concisely. We are now studying the use of such curves, which we call *biased* context-based scans, and the tradeoff obtained. See for example the biased space filling curve in Figure 13.

Another promising direction of research underway in-

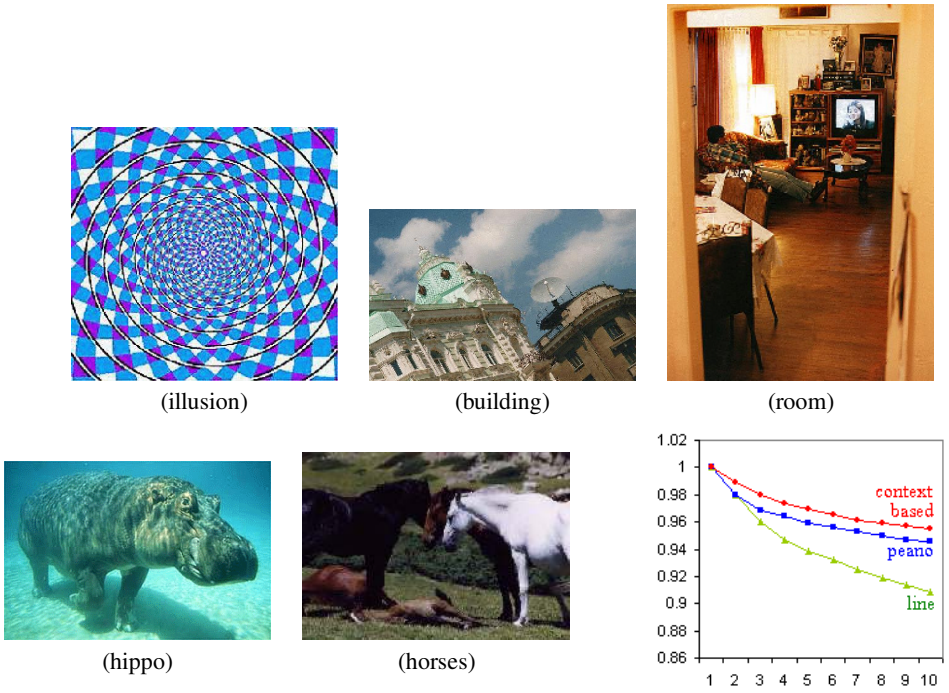

Figure 11: *The average autocorrelation of the five figures above. Each image was scanned using line scan, Peano scan and context-based scan. The average autocorrelation of each kind of scan is the average autocorrelation of the red blue and green components of all the images scanned that way. The context-based scanned images have the best autocorrelation.*

Table 1: GIF compression results of the line scanned, Peano scanned and context-based scanned 2D-images.

figure	original	scan-line	peano	context-based
illusion	328440	94147	93167	85054
room	790560	149958	142081	128735
building	633696	149086	143182	136536
hippo	180000	44271	44039	41916
horses	339600	80277	80690	76263

cludes extending the context-based scanning technique to 3D data. A true 3D context-based curve would exploit the 3D spatial coherency which may be quite better than the 2D spatial coherency of each volume slice.

This work is a first step in exploring the utility of context-based space filling curves, as alternative to well studied context-free space filling curves such as the Peano-Hilbert curves. Future research will reveal to what applications and to what degree the context-based scanning technique can be advantageous.

References

1. V.V Alexandrov, A.I. Alexeev, and N.D. Gorsky. A recursive algorithm for pattern recognition. *IEEE Intl. Conf. Pattern Recognition*, pages 431–433, October 1982.

2. A. Ansari and A. Fineberg. Image data compression and ordering using peano scan and lot. *IEEE Transactions on Consumer Electronics*, 38(3):436–445, August 1992.

3. T. Cormen, C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. McGraw-Hill book Company, 1990.

Figure 12: *Two frames taken from two video sequences.*

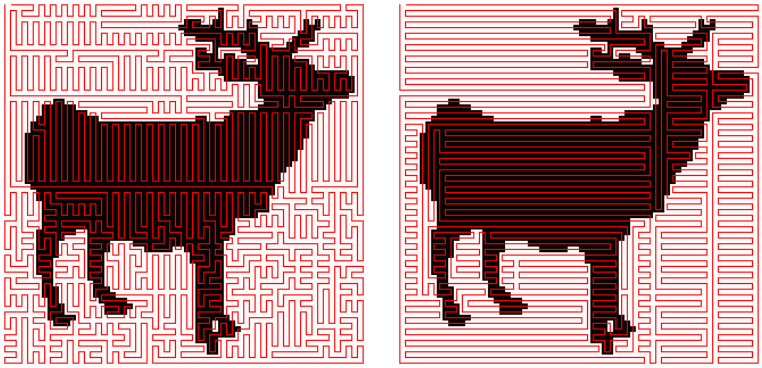

Figure 13: *Scanning a deer image using a context-based scan (left) and a biased context-based scan (right)*

4. G. Wyvill et al. Three plus five makes eight: A simplified approach to halftoning. *Proceedings of CGI '91 Scientific Visualization of Physical Phenomena*, pages 379–392, 1991.
5. Velho et al. Digital halftoning with space filling curves. *Computer Graphics*, 25(4):81–90, July 1991.
6. Witten et al. Using peano curves for bilevel display of continuous-tone images. *IEEE Computer Graphics and Applications*, 2(5):47–52, 1982.
7. M.L. Fredman and R.E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3):596–615, 1987.
8. A. Itai, C.H. Papadimitriou, and J.L. Szwarcfiter. Hamilton paths in grid graphs. *Siam J. Comput*, 1982.
9. S. Kamata, R.O.Eason, and E. Kawaguchi. An implementation of the hilbert scanning algorithm and its application to data compression. *IEICE Transaction information and systems*, E-76(4):420–427, April 1993.
10. K.S.Thyagarajan and S. Chatterjee. Fractal scanning for image compression. *Conference Record of the Twenty-Fifth Asilomar Conference on Signals,Systems and Computers*, pages 467–471, June 1992.
11. Y. Matias and A. Shamir. A video scrambling technique based on space filling curves. *Proceedings of Advances in Cryptology - CRYPTO'87, Springer LNCS 293*, pages 398–417, 1987.
12. N. Max. Visualizing hilbert curves. *IEEE Visualization'98*, pages 447–450, 1998.
13. P.T. Nguyen and J. Quinqueton. Space filling curves and texture analysis. *IEEE Intl. Conf. Pattern Recognition*, pages 282–285, October 1982.
14. T.A. Welch. A technique for high-performance data compression. *Computer*, 17:8–19, June 1984.
15. J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE Transactions on Information Theory*, IT-24(5):530–536, September 1978.
16. J. Ziv and A. Lempel. Compression of two-dimensional data. *IEEE Trans. on Information Theory*, 32(1):2–8, January 1986.