Peano Scanning of Arbitrary Size Images

A. Pérez

Centro de Inteligencia Artificial
ITESM
Monterrey, México 64849

S. Kamata and E. Kawaguchi

Department of Computer Engineering
Kyushu Institute of Technology
Kitakyushu, Japan 804

Abstract

Discrete space–filling curves are not uniquely defined. In addition to the condition that the curve must pass for all the points of the array only once, continuously, it is necessary to add some critera to select the best curves. Our aim is to preserve two–dimensional continuity as much as possible. The weighted sum of the distances of the points in the curve is minimized, where the weights are inversely proportional to the spatial distance between the points. However, the minimum is not unique. Particularly, space–filling curves always come on symmetric pairs. The generation of a near optimal space–filling curve is done hierarchically.

1 Introduction

In the estimation of local parameters of a digital image two major considerations are noise tolerance and speed of computation. Usually, a image of a real scene is characterized by high correlation among neighboring pixels. However, pixel by pixel processing is unrealiable because of inhomogeneoues pixels representing areas with different parameters. It is thus necessary to use a processing window around the pixel of interest and to take into account previously calculated parameters. In order to take advantage of the correlation between neighboring pixels and at the same time keep the computational cost low, it is convenient to define a scanning procedure that minimizes the distance between neighbors.

The Peano scan is a general technique for continuous scanning of multidimensional data by a space–filling curve [5]. A space–filling curve is a mapping, $h_n : \Re \rightarrow U_n$, that maps a unit interval onto an n–dimensional unit hypercube continuously [2]. In the case of discrete sampled data, a multidimensional space region is partitioned into a finite number of elementary regions, which are assigned to the nodes of discrete hypercubes iteratively. Peano scanning does not favor one direction over the others.

Hilbert [3] presented a simplified version of Peano curves in terms of binary divisions, *i.e., Hilbert scanning*. Hilbert scanning has wide applications. In data compression, for example, an adaptive predictive coder based on the Peano scan can better estimate the local probability distributions of the gray–levels in an image and experimental evaluation shows Hilbert scanning to be consistently better than raster methods [4]. Hilbert scanning is well understood when the dimensions of the array are equal on all directions and powers of two, *e.g.*, $2^m \times 2^m$ images [4, 6, 8]. Many image data sets, however, do not meet this restriction, *e.g.*, LANDSAT data consists of 400 x 512 images.

We presents a generalized algorithm for images of arbitrary size. Agui *et al.* [1] presented a generalized Peano scan algorithm. However, their algorithm is defined in terms of a global iterative optimization procedure. Ours is a hierarchical algorithm, which is based on the self–replicating properties of Peano curves. Once the curves are generated, the algorithm can be implemented in terms of a set of tables. The use of tables for the generation of Hilbert curves is an efficient approach amenable to parallelization and hardware implementation [4, 6, 8].

For the sake of clarity, let us begin by describing a Hilbert scanning algorithm and then generalize it to the case of images of arbitrary size.

2 Hilbert Scanning

Any point (x, y) on a $2^m \times 2^m$ discrete plane can be represented by two m–bit binary numbers, $x = x_0 \cdots x_i \cdots x_{m-1}$ and $y = y_0 \cdots y_i \cdots y_{m-1}$, or a single $2m$–bit binary address by concatenating x and y, as follows

$$x_0 y_0 \cdots x_i y_i \cdots x_{m-1} y_{m-1}. \qquad (1)$$

Then,

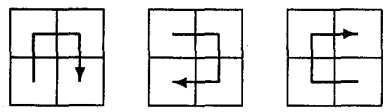
$$x_0 y_0 \cdots x_i y_i, \quad (i = 0, 1, 2, \cdots, m-1), \qquad (2)$$

can be consider the address of a $2^{m-1-i} \times 2^{m-1-i}$ subimage. At the highest level, $i = 0$, an image is divided into four quadrants. This quadrants can be joined together by a continuous curve that corresponds to Gray–code sequence in the quadrant labels (see Fig. 1). The two–dimensional Hilbert canonical form is uniquely defined under the operations of rotation and sequence reversal.

The subdivision of an image in quadrants can be carried out iteratively. However, the assignment of Hilbert curves to the subquadrants must be done in such a way that the global curve is continuous. We refer to this assignment rule as an induction rule. In general, the following induction rule holds [6]:

$$s \rightarrow (s\mathbf{ri})\, s\, s\, (s\mathbf{li}), \qquad (3)$$

0-8186-2920-7/92 $3.00 © 1992 IEEE



00_2	01_2	10_2
01_2	11_2	00_2
11_2	10_2	01_2
10_2	00_2	11_2
(a)	(b)	(c)

Figure 1: Two dimensional Hilbert curves: (a) canonical orientation and corresponding quadrant binary address sequence; (b) right rotation of canonical curve; (c) left rotation of canonical curve.

where **r** indicates a right rotation, **i** a sequence reversal, and **l** a left rotation.

3 General Case

Let us look at the case of arrays of size $p^m \times 2^m$, where p is a prime integer greater than 2. Similarly to the 2×2 case, a point (r, y) on a $p^m \times 2^m$ discrete plane can be represented by a m–digit base p number, $r = r_0 \cdots r_i \cdots r_{m-1}$ and a m–bit binary number, $y = y_0 \cdots y_i \cdots y_{m-1}$. Hence,

$$r_0 y_0 \cdots r_i y_i, \quad (i = 0, 1, 2, \cdots, m-1), \qquad (4)$$

can be consider the address of a $p^{m-1-i} \times 2^{m-1-i}$ subimage. These subimages can be joined together by a continuous curve that corresponds to a generalized Gray–code sequence where successive quadrant labels differ in a single base p digit or a single bit. A continuous space–filling curve on a $p \times 2$ array is completely defined by its beginning and ending points. Using the labels defined in Fig. 2(a) for the case $p = 2$ we can write expressions for the curves and their induction rules. The basic curves are of the form depicted in Figs. 2(b) and 2(c). The overall shape of the curves is independent of the value of p. Using the notation s_{ab} for the curve in Figs. 2(b) and s_{ad} for the curve in Fig. 2(c), the induction rules can be written as

$$s_{ab} \rightarrow s_{ad} s_{bc} \cdots s_{ad} s_{bc} s_{ab} s_{ba} s_{da} s_{cb} \cdots s_{da} s_{cb},$$

and

$$s_{ad} \rightarrow s_{ab} s_{ad} s_{ba} s_{bc} s_{ab} s_{ad} \cdots s_{ba} s_{bc} s_{ab} s_{ad}.$$

This set of curves is self–contained in terms of the induction rules. However, when defining induction rules for $p \times q$ curves, with p and q both prime integers greater than 2, in terms of $p \times 2$ curves, then it might be necessary to use curves of the type depicted in Figs. 2(e) and 2(f), as illustrated in Fig. 3(c). Note, however, that since p is by definition an odd number, valid curves of the form ac do not exist.

In the case of arrays of size $p^m \times q^m$, with p and q both prime integers greater than 2, a point (r, c) can be represented by a m–digit base p number, $r = r_0 \cdots r_i \cdots r_{m-1}$ and a m–digit base q number, $c = c_0 \cdots c_i \cdots c_{m-1}$. Hence,

$$r_0 c_0 \cdots r_i c_i, \quad (i = 0, 1, 2, \cdots, m-1), \qquad (5)$$

can be consider the address of a $p^{m-1-i} \times q^{m-1-i}$ subimage. These subimages can be joined together by a continuous curve that corresponds to a generalized Gray–code sequence where successive quadrant labels differ in a single digit. $p \times q$ space–filling curves are not uniquely defined. In addition to the condition that the curve must pass for all the points of the array only once, continuously, it is necessary to add some criterium to select the best curves. Since our aim is to preserve two–dimensional continuity as much as possible, we should select a cost function that penalizes a curve when it takes neighboring points far apart. That is, we want the weighted sum of the distances of the points in the curve to be a minimum, where the weights are inversely proportional to the spatial distance between the points, *i.e.,* we want a curve such that, given an initial and final point,

$$\min \sum_{i=0}^{p \times q - 1} \sum_{j>i} \frac{j-i}{d(\mathbf{x}_i, \mathbf{x}_j)} \qquad (6)$$

holds, where i and j are indices for the points covered by the curve and $d(\mathbf{x}_i, \mathbf{x}_j)$ is a function of the spatial separation between $\mathbf{x}_i$ and $\mathbf{x}_j$, such as, the euclidean or the city–block distance. For example, the cost of the curve in Fig. 2(b), using the city–block distance as the function d, is 82. The minimum is not unique. Particularly, space–filling curves always come on symmetric pairs, and therefore, there are at least two solutions for any given case.

Since it is always possible to define *ab*, *cd*. *ac*, and *bd* curves (see Fig. 3), the generation of induction rules for the $p \times q$ case is carried out as a minimization of the total curve cost as defined in Eq. (6) in terms of these kind of curves. Furthermore, as long as sub–blocks dimensions satisfy the condition $p+q$ even, the induction rule are independent of the block size. For example, the same rule can be used to split a 5×5 array into 5×5, 5×3, and 2×2 sub–arrays. However, all the blocks at a given level must be of the same size. It is possible to split a $p \times q$ array into $p' \times 2$ sub–arrays, $p' > 3$ but a special rule must be used (see Fig. 3(c)). Thus, for an arbitrary size image, the image dimensions can be decomposed in terms of their prime factors as

$$(p'_0{}^{m_0} p'_1{}^{m_1} \cdots p'_k{}^{m_k}) \times (q'_0{}^{n_0} q'_1{}^{n_1} \cdots q'_l{}^{n_l}), \qquad (7)$$

where

$$1 < p'_0 < p'_1 < \cdots < p'_k,$$

and

$$1 < q'_0 < q'_1 < \cdots < q'_l,$$

are all prime numbers. Assuming, without lost of generality, that

$$\sum m_i \geq \sum n_j.$$

Then, Eq. (7) can be rewritten as

$$(p_0'^{m_0} p_1'^{m_1} \cdots p_k'^{m_k}) \times (1^{\sum m_i - \sum n_j} q_0'^{n_0} q_1'^{n_1} \cdots q_l'^{n_l}), \quad (8)$$

or, more succintly, as

$$(p_0 p_1 \cdots p_t) \times (q_0 q_1 \cdots q_t), \quad (9)$$

where p_i and q_j are the prime factors in Eq. (8), including 1, if necessary. This factorization is used to represent a point (x, y) by t digits in bases p_i, $x_{p_0} \cdots x_{p_i} \cdots x_{p_{m-1}}$, and t digits in bases q_j, $y_{q_0} \cdots y_{q_i} \cdots y_{q_{m-1}}$. Hence,

$$x_{p_0} y_{q_0} \cdots x_{p_i} y_{q_i}, \quad (i = 0, 1, 2, \cdots, m-1), \quad (10)$$

can be consider the address of a $(p_0 \cdots p_{m-1-i}) \times (q_0 \cdots q_{m-1-i})$ subimage. This description is used for the generation of generalized Gray-code sequences. The generation of an near optimal space-filling curve is done hierarchically, staring form $p_0 \times q_0$ and then on down. There is always a valid induction rule assignment and any size array can be mapped into a space-filling curve. Curves at any given level in the hierarchy are optimal. Hence, the overall curve should be near optimal. However, since the induction rule is a function of the ending points of the sub-blocks, it is necessary to carry out a second optimization step in terms of total cost for the generation of the induction rules. Nevertheless, the factors are sorted from smaller values on up. For example, only 2×2 and 3×2 blocks are ever divided into 3×2 sub-blocks. Therefore, is expected that the actual rules will be independent of the sub-block size.

4 Example

As an example, let us look at the case of LANDSAT images. LANDSAT images are of the form $(1^3 2^4 5^2) \times (2^9)$. Therefore there is a $1^3 \times 2^3$, $2^4 2^4$, $5^2 2^2$ hierarchy in the space-filling mapping. This can be used for either hierarchical or predictive coding. The Peano scanning of a multispectral image can be viewed as a random vector sequence [7]. The elements of the random vector can be decorrelated by the Karhunen-Loève Transform (KLT). The explicit use of the KLT in combination with adaptive arithmetic coding presents some computational problems since the Joint Cumulative Probability Density Function would need to be recalculated for every new point in the scan. However, the decorrelation properties of the KLT can be approximated by the Discrete Cosine Transform (DCT). It can be shown that for a first-order Markov source model, the KLT basis functions converge to the DCT basis functions as the adjacent pixel correlation approaches unity. In order to apply a fast DCT algorithm to the processing of LANDSAT images the channels need to be expanded to 8. However this does not require an expansion on the coding since the expanded channel is known to be zero and therefore the DCT coefficients are linearly dependent and only 7 need to be coded. Since the DCT coefficients are less correlated than the original multispectral image, each coefficient is independently coded using DPCM.

Acknowledgements

This work was supported in part by the Japan Society for the Promotion of Science Postdoctoral Fellowship for Foreign Researchers program. Thanks are due to Mrs. Yamashita for her assistance in the editing of the paper.

References

[1] Agui, T., T. Nagae, and M. Nakajima, "Generalized Peano Scans for Arbitrarily-Sized Arrays", *IEICE Transactions,* Vol. E74, No. 5, pp. 1337–1342, May 1991.

[2] Butz, A. R.,"Alternative Algorithm for Hilbert's Space-Filling Curve," *IEEE Trans. on Computers*, vol. C-20, pp. 424–426, April 1971.

[3] Hilbert, D., "Über die stetige Abbildung einer Linie auf ein Flächenstück", *Mathematische Annalen,* Vol. 38, pp. 459–460, 1891.

[4] Kamata, S., A. Pérez, and E. Kawaguchi, "A Method of Computing Hilbert Curves in Two and Three Dimensional Spaces", *The Transactions of the Institute of Electronics, Information, and Communication Engineers,* Vol. J74-D-II, No. 9, pp. 1217–1226, Sep. 1991.

[5] Peano, G., "Sur une courbe, qui remplit toute une aire plane", *Mathematische Annalen,* Vol. 36, pp. 157–160, 1890.

[6] Pérez, A., S. Kamata, and E. Kawaguchi, "N-Dimensional Hilbert Scanning and its Application to Data Compression", *Proceedings of the SPIE/SPSE Conference on Image Processing Algorithms and Techniques II,* 1991 SPIE/SPSE Symposium on Electronic Technology: Science and Technology, Feb. 1991, San Jose, California.

[7] Pérez, A., S. Kamata, and E. Kawaguchi, "Hilbert Scanning Arithmetic Coding for Multispectral Image Compression", *Proceedings of the SPIE Conference on Applications of Digital Image Processing XIV,* July 21–26, 1991, San Diego, California.

[8] Skarbek, W., "Software tool for Hilbert scan of large image," *Trans. IEICE*, vol. E72, num. 5, pp. 561–564, May 1989.

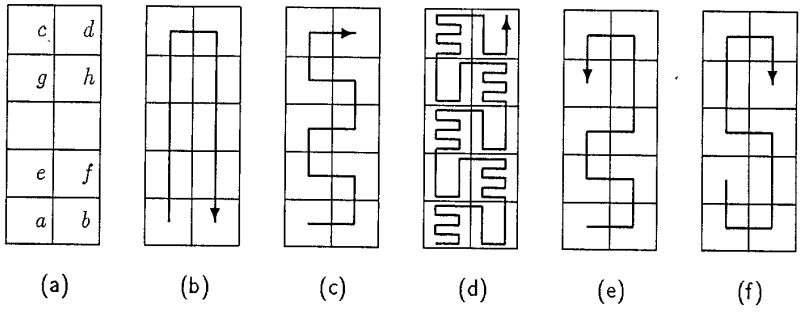


Figure 2: Typical space–filling curves for 5×2 arrays: (a) labels for corner quadrants; (b) ab curve (c) ad curve; (d) ad first–order curve; (e) ag curve; (f) eh curve.

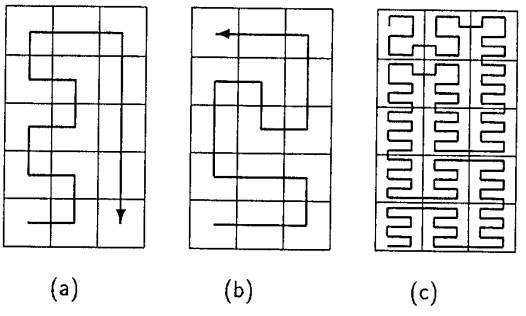


Figure 3: Space–filling curves for 5×3 arrays: (a) ab curve; (b) ac curve; (c) ac curve expanded into 5×2 arrays.