LETTER

A Method of Making Lookup Tables for Hilbert Scans*

Sei-ichiro KAMATA†, Michiharu NIIMI††, *and* Eiji KAWAGUCHI†, *Members*

SUMMARY Recently applications of Hilbert curves are studied in the area of image processing, image compression, computer hologram, etc. We have proposed a fast Hilbert scanning algorithm using lookup tables in N dimensional space. However, this scan is different from the one of previously proposed scanning algorithms. Making the lookup tables is a problem for the generation of several Hilbert scans. In this note, we describe a method of making lookup tables from a given Hilbert scan which is obtained by other scanning methods.

key words: *Hilbert curves, lookup tables, image compression, N dimensional space*

1. Introduction

Hilbert curve published in 1891 is one of space-filling curves which are often called *Peano curves*. Several applications of Hilbert curves are studied in the area of image analysis [1], [2], image compression [3], etc. A Hilbert scan which goes along a Hilbert curve in N dimensional space is computed by a recursive algorithm [4] or an analytical method [5]. Recently, a method using lookup tables is proposed for a fast computation and a simple hardware implementation [6]. However, these three metheds for $N \geqq 3$ generate different Hilbert scans. Making the lookup tables in our method is a problem for the generation of these Hilbert scans. In this note, if a Hilbert scan is given using the other methods, we show a method of making the corresponding lookup tables to generate this Hilbert scan.

2. Hilbert Scans

2.1 Address Alignment and Hilbert Scans

An address alignment in N dimensional space is often used for image expression such as quadtree or octree in the following. There is an original hypercube with the side 2^M in N dimensional space. Let us split the original hypercube into 2^{mN} equal subhypercubes ($1 \leqq m \leqq M$) as shown in Fig. 1 (ex. $N=2$). Assigning N dimensional coordinates as $X_1, X_2, \cdots, X_N$, the range of the subhypercube for X_i $(i = 1, 2, \cdots, N)$ is

$$[2^{M-m}\mathbf{x}_i, 2^{M-m}(\mathbf{x}_i+1)-1],$$

where $\mathbf{x}_i$ is expressed with m-bits binary digits

$$\mathbf{x}_i = x_{m,i}\cdots x_{3,i}x_{2,i}x_{1,i}.$$

Any subhypercube with the side 2^{M-m} is expressed in mN-bits binary digits $\rho = \rho^m \cdots \rho^2\rho^1 =$

$$\underbrace{\overbrace{x_{m,1}x_{m,2}\cdots x_{m,N}}^{N\text{bits}}\cdots\overbrace{x_{2,1}\cdots x_{2,N}}^{N\text{bits}}\overbrace{x_{1,1}\cdots x_{1,N}}^{N\text{bits}}}_{mN\text{bits}},$$

where $\rho^m = x_{m,1}x_{m,2}\cdots x_{m,N}$ [6]. ρ is called *an address*.

For example, any quadrant in 2 dimensional space is expressed as a bit-sequence as shown in Fig. 1. After one split as shown in Fig. 1 (a), the addresses of each quadrant are set to 00 (Lower Left), 01 (Upper Left), 10 (Lower Right) and 11 (Upper Right). After M splits, we obtaint the smallest quadrants which are called *pixels* in general.

A Hilbert scan always moves a hypercube to a $2N$-neighbor hypercube. When ith address in a Hilbert scan is denoted as ρ_i, the Hilbert scan is written by (ρ_1, ρ_2, $\cdots$, $\rho_{2^{mN}}$). For example, on 2×2 quadrants in 2 dimensional space (cf. Fig. 1 (a)), the Hilbert scan is expressed as (00, 01, 11, 10). With 4×4 quadrants (cf. Fig. 1 (b)), the Hilbert scan is (0000, 0010, 0011, 0001, 0100, 0101, 0111, 0110, 1100, 1101, 1111, 1110, 1011, 1001, 1000, 1010). The Hilbert scan in Fig. 1(c) is (000000, 000001, 000011, 000010, $\cdots$, 101010).

2.2 Lookup Tables for Hilbert Scans

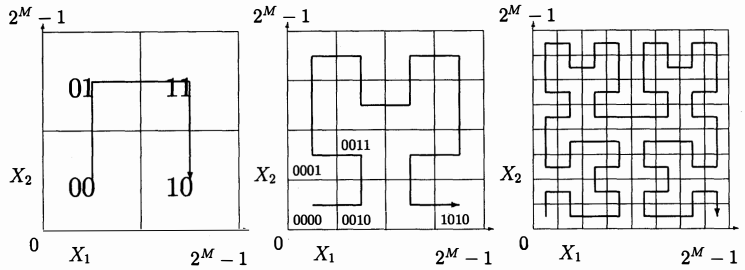
In the computational method of Ref. [6], there are two kinds of lookup tables such as induction table and terminal table, corresponding to induction rules and terminal rules. If we make the terminal table, the induction table can be generated using this terminal table. The terminal table stores address sequences of Hilbert scans $\{\mathbf{a}_k, k=1,2,\cdots,N2^{N-1}\}$ in the case of $M=1$, where the number of scan types is $N2^{N-1}$ and each scan generates 2^N addresses. For instance of the scan type, the scan type of three scans in Fig. 1 is all 1 (this moves from Lower Left quadrant to Lower Right quadrant).

Manuscript received October 11, 1995.

†The authors are with the Faculty of Engineering, Kyushu Institute of Technology, Kitakyushu-shi, 804 Japan.

††The author is with Computer Science and Control Engineering Course, Nagasaki Institute of Applied Science, Nagasaki-shi, 851–01 Japan.

*A part of this paper was presented at IAPR Scandinavian Conf. on Image Analysis, Uppsala, 1995.



(a) 2 × 2 (b) 4 × 4 (c) 8 × 8

Fig. 1 Space-filling curves proposed by D. Hilbert in 1891.

3. Making Lookup Tables

3.1 Tree Expression of a Hilbert Scan

Looking at a Hilbert scan $(\rho_1, \rho_2, \cdots, \rho_{2^{mN}})$, we notice that

$$
\begin{aligned}
&\rho_1^m = \rho_2^m = \cdots = \rho^m_{2^{(m-1)N}},\\
&\rho^m_{2^{(m-1)N}+1} = \rho^m_{2^{(m-1)N}+2} = \cdots = \rho^m_{2\cdot 2^{(m-1)N}},\\
&\vdots\\
&\rho^m_{(2^N-1)2^{(m-1)N}+1} = \cdots = \rho^m_{2^N\cdot 2^{(m-1)N}},
\end{aligned}
$$

and

$$
\begin{aligned}
&\rho_1^{m-1} = \rho_2^{m-1} = \cdots = \rho^{m-1}_{2^{(m-2)N}},\\
&\rho^{m-1}_{2^{(m-2)N}+1} = \rho^{m-1}_{2^{(m-2)N}+2} = \cdots = \rho^{m-1}_{2\cdot 2^{(m-2)N}},\\
&\vdots\\
&\rho^{m-1}_{(2^{2N}-1)2^{(m-2)N}+1} = \cdots = \rho^{m-1}_{2^{2N}\cdot 2^{(m-2)N}},
\end{aligned}
$$

and so forth. A Hilbert scan can be expressed by a tree structure representation which we call 2^N-tree. Figure 2 shows the tree structure representation in the case of $M=3$ and $N=2$. Each nonterminal node has an N-bits address and a scan type, while each terminal node has only an N-bits address. Here ϕ means a root node. For example, in the case of $m=3$, the address 00 corresponds to ρ_1^3 $(=\rho_2^3=\cdots=\rho_{16}^3)$, and the address 01 corresponds to ρ_{17}^3 $(=\cdots=\rho_{32}^3)$, and so forth.

3.2 Property of Hilbert Scans and Algorithm of Making Terminal Tables

Property 1: If a Hilbert scan is given in the case of $M=3$ in N dimensional space, it is possible to make the lookup tables corresponding to its scan.

We have confirmed that this is true for $N \leqq 8$ in the practical use. The drawback of our method is that the huge storage requires as the dimension N is large. For example of $N=9$, the storage of the lookup tables requires about 3 Mbytes. This is a problem to be solved in the future.

Using the above property, an algorithm of making terminal tables is shown in the following.

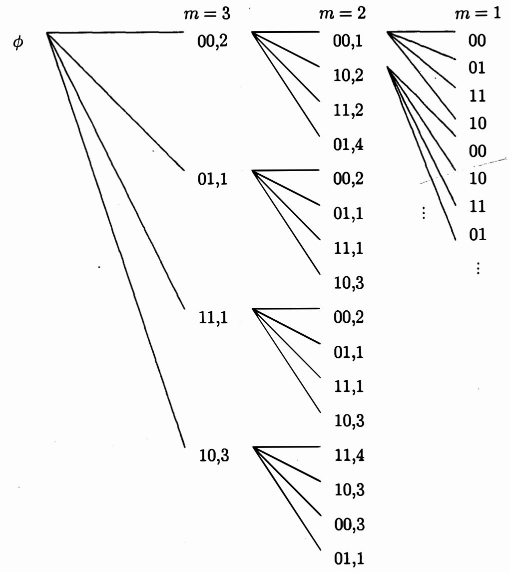


Fig. 2 Tree structure representation ($M=3, N=2$).

Algorithm 1:

(1) Generate an address sequence $(\rho_1, \rho_2, \cdots, \rho_{2^{3N}})$ in the case of $M=3$ (each address ρ_i consists of $3N$-bits).

(2) Extracting the N-bits from each address, getting $\{\rho_i^3, \rho_i^2, \rho_i^1, i=1,2,\cdots,2^{3N}\}$, we devide those into equal blocks, $\{\mathbf{b}_j, j=1,2,\cdots,2^{2N}+2^N+1\}$, where

$$
\begin{aligned}
&\mathbf{b}_1 \leftarrow (\rho_1^3, \rho^3_{2^{2N}+1}, \cdots, \rho^3_{(2^N-1)\cdot 2^{2N}+1}),\\
&\mathbf{b}_2 \leftarrow (\rho_1^2, \rho^2_{2^N+1}, \cdots, \rho^2_{(2^N-1)\cdot 2^N+1}),\\
&\mathbf{b}_3 \leftarrow (\rho^2_{2^{2N}+1}, \cdots, \rho^2_{2^{2N}+(2^N-1)\cdot 2^N+1}),\\
&\vdots\\
&\mathbf{b}_{2^N+1} \leftarrow (\rho^2_{(2^N-1)\cdot 2^{2N}+1}, \cdots, \rho^2_{(2^N-1)2^N(2^N+1)+1}),\\
&\mathbf{b}_{2^N+2} \leftarrow (\rho_1^1, \rho_2^1, \cdots, \rho^1_{2^N}),\\
&\mathbf{b}_{2^N+3} \leftarrow (\rho^1_{2^N+1}, \rho^1_{2^N+2}, \cdots, \rho^1_{2^{N+1}}),\\
&\vdots\\
&\mathbf{b}_{2^{2N}+2^N+1} \leftarrow (\rho^1_{(2^N-1)2^N(2^N+1)+1}, \cdots, \rho^1_{2^{3N}}).
\end{aligned}
$$

We obtain $\{\mathbf{b}_j\}$ by traversing the 2^N-tree using a breadth first search.

(3) From $\{\mathbf{b}_j,\ j = 1, 2, \cdots, 2^{2N} + 2^N + 1\}$ we extract the terminal rules $\{\mathbf{a}_k,\ k = 1, 2, \cdots, N2^{N-1}\}$ as follows.

  (a) $\mathbf{a}_1 \leftarrow \mathbf{b}_1$; $k = 1$;

  (b) for $j = 2, \cdots, 2^{2N} + 2^N + 1$

  If $\mathbf{b}_j$ does not exist in $\{\mathbf{a}_1, \cdots, \mathbf{a}_k\}$,

  then $\mathbf{a}_{k+1} \leftarrow \mathbf{b}_j$, $k \leftarrow k + 1$.

(4) If $k \neq N2^{N-1}$, then output "missing some rules".

(5) Stop.

For example, in the case of $M = 3$ and $N = 2$ as shown in Fig. 1 (c), all terminal rules are obtained as $\mathbf{a}_1$ = (00, 01, 11, 10), $\mathbf{a}_2$ = (00, 10, 11, 01), $\mathbf{a}_3$ = (11, 10, 00, 01) and $\mathbf{a}_4$ = (11, 01, 00, 10) using the above algorithm. The induction tables can be generated from the terminal tables as stated before in 2.2.

Using an address sequence obtained by Butz algorithm [5], we have confirmed that the algorithm of making lookup tables works well in the case of $N \leqq 8$.

4. Conclusion

We have discussed a method of making lookup tables from a given Hilbert scan which is obtained by other scanning methods. Our method can generate several kinds of Hilbert scans from this result. Problems to be solved are the proof of the property in 3.2, and an hardware implementation of the N dimensional Hilbert scanning algorithm using the lookup tables. We believe that the property is true for any N, because the average occurrence for each rule, that is $(2^{2N}+2^N+1)/(N2^{N-1})$, increases as the dimension N becomes large.

Acknowledgement

This work has been supported partly by a grant from the Ministry of Education, Science and Culture (Project No. 07750437).

References

[1] S. Kamata, M. Niimi, and E. Kawaguchi, "A method of an interactive analysis for multi-dimensional images using a Hilbert curve," IEICE Trans., vol.J77-D-II, no.7, pp.1255–1264, July 1994. (in Japanese)

[2] R.J. Stevens, A.F. Lehar, and F.H. Preston, "Manipulation and presentation of multi-dimensional image data using the peano scan," IEEE Trans.Pattern Anal. & Mach. Intell., vol.PAMI-5, no.5, pp.520–526, Sept. 1983.

[3] W. Skarbek, T. Agui, and M. Nakajima, "Compression of dithered binary images using Hilbert scan," IEICE Trans., vol.E72, no.11, pp.1235–1242, Nov. 1989.

[4] J. Quinqueton and M. Berthod, "A locally adaptive Peano scanning algorithm," IEEE Trans. Pattern Anal. & Mach. Intell., vol.PAMI-3, no.4, pp.520–526, July 1983.

[5] A.R. Butz, "Convergence with Hilbert's space filling curve," J. Comput. & Syst. Sci., vol.3, pp.128–146, 1969.

[6] S. Kamata, A. Perez, and E. Kawaguchi, "A computation of Hilbert's space-filling curves in N dimensional space," IEICE Trans., vol.J76-D-II, no.3, pp.797–801, March 1993. (In Japanese)