

基于状态视图的高效Hilbert编码和解码算法

贾连印^{①②} 陈明鲜^① 李孟娟^③ 游进国^① 丁家满^{*①}

^①(昆明理工大学信息与自动化学院 昆明 650500)

^②(云南省计算机应用重点实验室 昆明 650500)

^③(云南师范大学图书馆技术部 昆明 650500)

摘要: Hilbert曲线是高维降到1维的重要方法, 具有较好的空间聚集和空间连续性, 在地理信息系统、空间数据库、信息检索等方面有广泛的应用。现有Hilbert编码或解码算法未考虑输入数据对编码或解码效率的影响, 因此将不同输入数据同等对待。为此, 该文通过设计高效的状态视图并结合快速置位检测算法提出高效的免计前0的Hilbert编码算法(FZF-HE)和免计前0的Hilbert解码算法(FZF-HD), 可快速识别输入数据前部为0而无需迭代计算的部分, 从而降低迭代查询次数及算法复杂度, 提高编解码效率。实验结果表明, FZF-HE算法和FZF-HD算法在数据均匀分布时效率稍高于现有算法, 而在数据偏斜分布时效率远高于现有算法。

关键词: 状态视图; 免计前0的Hilbert编码算法; 免计前0的Hilbert解码算法; Hilbert曲线

中图分类号: TN919.81

文献标识码: A

文章编号: 1009-5896(2020)06-1494-08

DOI: 10.11999/JEIT190501

State View Based Efficient Hilbert Encoding and Decoding Algorithms

JIA Lianyin^{①②} CHEN Mingxian^① LI Mengjuan^③ YOU Jinguo^① DING Jiaman^①

^①(Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China)

^②(Key Laboratory of Computer Technologies Application of Yunnan Province, Kunming 650500, China)

^③(Library of Yunnan Normal University, Kunming 650500, China)

Abstract: Hilbert curve is an important method for high-dimensional reduction to one-dimensional. It has good characteristics of spatial aggregation and spatial continuity and is widely used in geographic information system, spatial databases and information retrieval. Existing Hilbert encoding or decoding algorithms do not consider the differences between input data, thus treating them equally. To this end, an efficient Hilbert coding algorithm Front-Zero-Free Hilbert Encoding(FZF-HE) and an efficient decoding algorithm Front-Zero-Free Hilbert Decoding(FZF-HD) are proposed. FZF-HE and FZF-HD can quickly identify the 0s of the front part of input data before iterative calculation by combining efficient state views and first bit-1 detection algorithm, thus reducing the number of iterations and the complexity of the algorithm, and improving the encoding and decoding efficiency. The experimental results show that efficiencies of these two algorithms are slightly higher than existing algorithms for uniform distributed data, and are much higher than existing algorithms for skew distributed data.

Key words: State view; Front-Zero-Free Hilbert Encoding (FZF-HE); Front-Zero-Free Hilbert Decoding (FZF-HD); Hilbert curve

1 引言

空间填充曲线可将高维空间中的对象映射到

1维空间, 从而将复杂的高维问题降低为简单的1维问题, 因此其在图像处理、空间数据管理、数据分区与负载均衡^[1-6]、无线传感器网络^[7,8]、Cache局部保序^[9]等诸多领域得到了广泛的应用。

目前, 常见的空间填充曲线包括Z曲线^[10]和Hilbert曲线^[11]等, 大量理论分析与应用表明: Hilbert曲线相对其它曲线具有跳变性小, 聚合性高等优点^[12]。

收稿日期: 2019-07-05; 改回日期: 2020-02-03; 网络出版: 2020-02-27

*通信作者: 丁家满 tjom2008@126.com

基金项目: 国家自然科学基金(61562054), 国家留学基金委公派留学项目(201908530036)

Foundation Items: The National Natural Science Foundation of China (61562054), Fund of China Scholarship Council (201908530036)

尽管Hilbert曲线具有上述优点, 但因其映射规则复杂, 编码效率低等不足, 从而限制了基于Hilbert应用的进一步推广, 故许多应用转向编码效率更高的Z曲线^[13,14]。Hilbert曲线编码和解码是基于Hilbert应用的前提, 提高编解码效率具有重要的意义, 故许多研究工作都致力于解决这一问题。

经典的Hilbert编解码算法可被划分为递归和迭代两类。对递归的算法, Butz等人^[11]做出了开创性的工作, 设计了一种以字节为导向的Hilbert曲线生成算法。在Butz工作的基础上, 多种算法被提出, 但其复杂度多为 $O(n^2)$, 其中 n 为编码的阶数, 效率较低。Fisher^[15]指出多数递归的算法不适用于Hilbert编码或效率较低, 并提出了基于双查找表的迭代编码算法, 故后续的研究多基于迭代的方式展开。

在基于迭代方式的实现中, 基于状态视图的编解码算法因其具有简单、直观、高效、易实现等特性, 从而得到了深入地研究。李绍俊等人^[16]提出基于状态转移矩阵的、复杂度为 $O(n)$ 的迭代编码算法, 其设计了精简高效的状态转移表并采用位域共同体等优化手段来提高算法效率。考虑到基于状态视图的算法多需要针对编码和解码设计不同的状态视图, 为解决这一问题, 文献^[17]通过编码前的象限映射和解码后的象限逆映射设计了一致状态视图的编解码算法, 使设计的一个状态视图可同时适用于编解码过程。

除上述算法外, 也有一些不依赖状态视图的实现算法。文献^[18]通过计算替代查找表, 可降低查找表的空间开销, 但因其需要额外的计算开销, 所以影响算法效率的提升; Moore^[19]在Butz经典算法的基础上, 设计了基于位操作的高效非递归编码和解码算法, 在低维和高维编解码上均具有较好的效果, 因此被作为比较的基准算法^[20]并在空间范围查询等领域中得到了广泛的应用^[21]; 曹雪峰等人^[20]在考虑维度分布差异的基础上, 设计了紧致的Hilbert编码, 适用于各维度数据分布不均匀的场合, Zhang等人针对3维^[22]、Liu等人^[23,24]针对高维提出了相应的编解码算法。

考虑到状态视图的优势, 本文同样基于状态视图进行编解码算法的研究。通过对现有多种高效的编解码算法对比后发现, 现有的高效算法多未考虑输入数据对编码或解码效率的影响, 因此将不同输入数据同等对待。例如, 李绍俊等人^[16]的算法为作者所知最为高效的编码算法, 但其编码过程中需逐阶遍历, 共需 $2n$ 次查询状态视图的操作, 其复杂度为 $O(n)$ 。其次, 该算法未考虑输入数据的差异, 每

个输入数据需要等同的编码和解码时间, 故其效率有待进一步提升。

为此, 本文对状态视图深入分析后发现并证明: 无需对输入数据中前部为0的阶进行迭代编码或解码。在此基础上, 提出复杂度为 $O(m)$ 的免计前0的Hilbert编码算法(Front-Zero-Free Hilbert Encoding, FZF-HE)和免计前0的Hilbert解码算法(Front-Zero-Free Hilbert Decoding, FZF-HD), 可将查询状态视图的次数降低为 $2m$ 次, 其中 m 为需迭代的阶数, $m \leq n$ 。本文算法考虑了不同输入数据的差别, 具有输入坐标点或编码值越小, 编解码效率越高的特性, 因此能很好地适应数据的偏斜分布。扩展实验结果表明, 本文算法在均匀分布下, 效率稍高于现有最优算法, 而在数据偏斜分布下, 效率远优于现有算法。

2 Hilbert曲线

Hilbert曲线是空间填充曲线家族的一种, 可以填满整个正方形网格, 其通过每个网格中心1次且仅1次。Hilbert曲线将空间中每个网格进行编码, 1阶Hilbert曲线将正方形网格分割为4个小正方形, 小正方形的左下、左上、右下、右上分别对应着第I象限、第II象限、第III象限、第IV象限, 将小正方形的第I, II, IV, III象限中心连接起来即可构成1阶Hilbert曲线。通过在每个象限放置一个1阶曲线, 并对第I象限左旋 90° 并垂直翻转, 对第III象限右旋 90° 并垂直翻转可得到2阶曲线, 继续迭代可进而得到任意阶曲线。1, 2, 3阶Hilbert曲线的示意图如图1所示。

给定位置点 $P(X,Y)$, 其中 $X = (x_1x_2 \cdots x_n)_2$, $Y = (y_1y_2 \cdots y_n)_2$, 确定 P 在Hilbert曲线中的序号 $Z = z_1z_2 \cdots z_{2n}$ 的过程称作编码; 相应地, 给定Hilbert编码 $Z = z_1z_2 \cdots z_{2n}$, 确定其对应位置点 $P(X,Y)$ 的过程称作解码。其中, X, Y 为相应的横坐标和纵坐标, Z 为对应的编码, n 称作阶或解析度。 $x_i, y_i, z_{2i-1}z_{2i}$ 分别为第 i 阶横坐标、纵坐标和编码值, 其中 $1 \leq i \leq n$ 。

3 Hilbert曲线编码和解码

3.1 状态视图的设计

状态视图是本文Hilbert编码和解码算法的基础, 故首先对Hilbert曲线的状态进行分析。

对1阶Hilbert曲线而言, 有4种基本状态: 状态0、状态1、状态2和状态3, 分别对应曲线开口朝下、左、上和右4种情形。图2给出了4种状态下1阶Hilbert曲线对应的状态图(括号外为编码, 括号内为坐标), 不同状态下, 每个坐标对应着不同的编码值。

对编码而言,设计状态视图的关键在于构建坐标和Hilbert编码值、坐标和下一阶状态之间的映射关系。根据图2和Hilbert曲线的旋转和翻转规则,易于构建编码状态视图如表1所示。

解码为编码的逆操作,同理,根据图2和Hilbert曲线的旋转和翻转规则,易于构建解码状态视图如表2所示。

需要注意的是,先前的工作^[18]将第*i*阶横纵坐标值 x_i 和 y_i 合并为 $x_i y_i$ 后执行计算及状态视图查询操作,这将导致额外的位合并开销。为避免这一开销且提高状态视图的查询效率,本文设计两个3维数组Key和Type来表示状态视图,其中Key实现某阶坐标值到该阶编码值的映射,而Type实现某阶状态到下一阶状态的映射。直接通过第*i*阶坐标值 x_i, y_i 和第*i*阶状态 T_i ,分别按照式(1)和式(2)查询状

态视图获得对应的第*i*阶编码值 $z_{2i-1} z_{2i}$ 和第*i*+1阶状态值 T_{i+1} ,从而避免位合并操作的开销

$$z_{2i-1} z_{2i} = \text{Key}[T_i][x_i][y_i] \tag{1}$$

$$T_{i+1} = \text{Type}[T_i][x_i][y_i] \tag{2}$$

对应地,通过设计两个2维数组InvKey和InvType,其中InvKey实现某阶编码值到该阶坐标值的映射,而InvType实现某阶状态到下一阶状态的映射。可根据输入的状态值 T_i 和编码值 $z_{2i-1} z_{2i}$,分别按照式(3)和式(4)查询状态视图获取第*i*阶坐标组合 $x_i y_i$ 和*i*+1阶状态值 T_{i+1}

$$x_i y_i = \text{InvKey}[T_i][z_{2i-1} z_{2i}] \tag{3}$$

$$T_{i+1} = \text{InvType}[T_i][z_{2i-1} z_{2i}] \tag{4}$$

为便于描述,本文后续编码和解码算法中,规

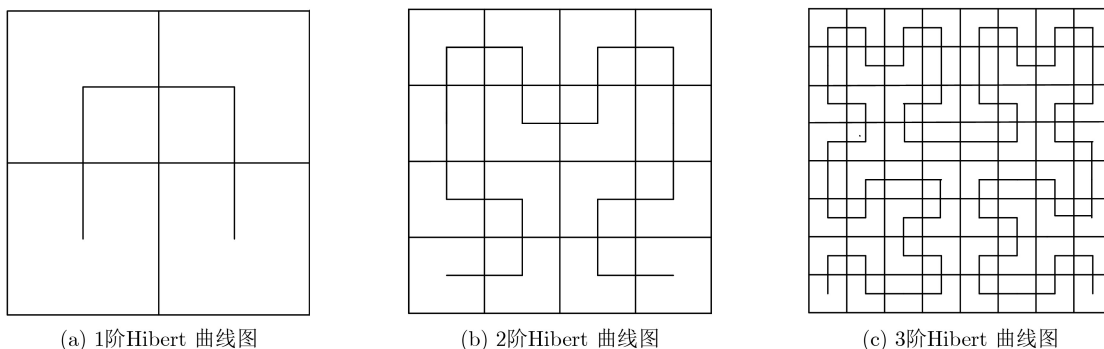


图1 1阶、2阶和3阶Hilbert曲线示意图

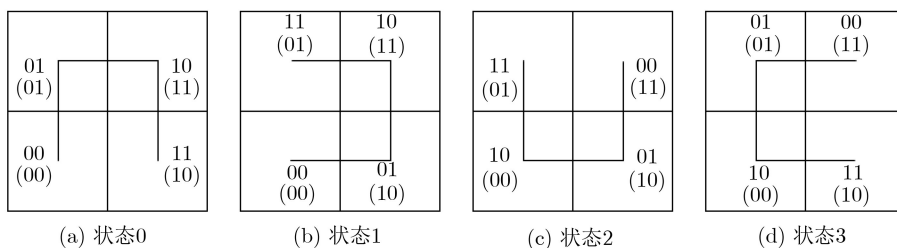


图2 1阶Hilbert曲线对应的4种状态

表1 编码状态视图

状态	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
坐标	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
编码	00	01	11	10	00	11	01	10	10	11	01	00	10	01	11	00
下阶状态	1	0	3	0	0	2	1	1	2	1	2	3	3	3	0	2

表2 解码状态视图

状态	0	0	0	0	1	1	1	1	2	2	2	2	3	3	3	3
编码	00	01	10	11	00	01	10	11	00	01	10	11	00	01	10	11
坐标	00	01	11	10	00	10	11	01	11	10	00	01	11	01	00	10
下阶状态	1	0	0	3	0	1	1	2	3	2	2	1	2	3	3	0

定第1阶的状态总为0, 即Hilbert曲线第1阶总为开口向下。

3.2 编码

构建编码状态视图之后, 可逐阶迭代查找状态视图, 以获取每阶的编码值和下一阶状态, 进而可生成最终的编码。

但通过这种方式, 需要遍历输入的每一阶的横纵坐标, 算法的复杂度为 $O(n)$, 且这种方式下所有输入数据需要相同的编码时间, 并未考虑不同输入数据的差别, 因此编码效率较低。故需对其进行改进, 以降低复杂度。

本文的编码算法依据定理1实现, 因此首先给出定理1如下所示:

定理1 若横坐标 $x_1x_2\cdots x_k$ 和纵坐标 $y_1y_2\cdots y_k$ 同时为0, 则前 k 阶编码为0, 第 $k+1$ 阶的状态等于0(k 为偶数)或1(k 为奇数)。

证明 $k=1$ 时, 因第1阶状态为0, x_1 和 y_1 为0, 则根据表1可知, 第1阶的编码为00, 第2阶状态为1, 故定理成立。

$k=2$ 时, 因第2阶状态1, x_2 和 y_2 为0, 则根据表1可知, 第2阶的编码同样为00, 前2阶的编码为0000, 第3阶的状态为0, 故定理成立。

$k\geq 3$ 时, 因第3阶的输入(横纵坐标值及状态)与第1阶完全相同, 故第3阶的输出(编码值和第4阶状态)与第1阶相同; 同理, 第4阶的输出与第2阶相同, 进入循环状态, 定理同样成立。

证毕

由定理1可见, 对 X, Y 的前 k 阶全为0的情形, 其前 k 阶编码值为0, 且第 $k+1$ 阶的状态可由 k 的奇偶性直接确定, 故无需对前 k 阶进行迭代编码, 从而可提高算法的效率。

如何快速检测 X 和 Y 前部同时为0的最大长度 k 是算法高效执行的关键。直观来看, 可以从 X 和 Y 的第1阶开始, 逐阶探测 x_i 和 y_i 是否同时为0。但如此仍需逐阶探测, 效率较低。

为解决这一问题, 引入高效的置位检测算法^[25], 该算法通过二分查找的思路快速检测给定整数的第1个被置1的位置 p (最右侧位的位置为0), 因此可用于快速计算 k 值。为避免对 X 和 Y 各执行1次置位检测, 本文仅对 X 和 Y 的最大值执行置位检测, 故1次编码过程, 仅需1次置位检测操作。

检测到 k 值之后, 对 $k+1$ 阶及之后的各阶, 迭代执行状态视图查询操作, 即可获取最终的Hilbert编码。

本文算法可免于计算 X, Y 前部为0的阶, 故称之为FZF-HE算法, 完整的FZF-HE算法如表3所示。

表3 FZF-HE算法

输入: X , 横坐标
Y , 纵坐标
n , 阶
输出: Hilbert编码 Z
(1) $Z \leftarrow 0$
(2) $p \leftarrow \text{msb}(\max(X, Y))$ //置位检测
(3) $t \leftarrow (n-p-1)\%2$ //计算第 $n-p$ 阶状态
(4) for i from $n-p$ to n
(5) $Z = Z << 2 \text{Key}[t][x_i][y_i]$ //查Key视图
(6) $t = \text{Type}[t][x_i][y_i]$ //查Type视图

FZF-HE算法中 msb 为置位检测算法, $n-p-1$ 即为所求 k 值, Key和Type为3.1节对应的编码状态视图。

算法复杂度分析: 算法的主要开销在于迭代查询状态视图的操作, 共需迭代执行 $m = n - k = \log_2(\max(X, Y))$ 次, 每次执行2次查表操作, 因此算法的时间复杂度为 $O(m)$, 因 $k \geq 0$, 故本文算法复杂度优于 $O(n)$ 。算法空间开销主要为2个状态视图的开销, 因Key表和Type表分别存储2位二进制表示的编码值和4种状态值, 故Key和Type均设计为char类型, 最终需要的空间为 $2 \times 4 \times 4 \times \text{sizeof}(\text{char}) = 32$ Byte, 其空间开销可基本忽略不计。

从上述分析来看, $\max(X, Y)$ 越小, FZF-HE的时间复杂度越低, 因此该算法适合数据偏斜分布, 尤其是向原点(正方形区域的左下角)偏斜的情形。在实际应用中, 数据偏斜分布是普遍存在的, 故本文算法可有效提升偏斜分布数据的编码效率¹⁾。

3.3 解码

Hilbert解码是将编码映射到坐标的过程, 是编码的逆过程。类似地, 本文实现的FZF-HD算法依据定理2实现。

定理2 若 $z_1z_2\cdots z_{2k}$ 为0, 则前 k 阶对应的横坐标 $x_1x_2\cdots x_k$ 和纵坐标 $y_1y_2\cdots y_k$ 为0, 第 $k+1$ 阶的状态等于0(k 为偶数)或1(k 为奇数)。

证明 $k=1$ 时, 因第1阶状态为0, z_1z_2 为00, 则根据表2的状态视图可知, 第1阶的坐标 x_1 和 y_1 均为0, 第2阶状态为1, 故定理成立。

$k=2$ 时, 因第2阶状态1, z_3z_4 为00, 则根据状态表2可知, 第2阶的坐标 x_2 和 y_2 同样为0, 第3阶的状态为0, 故定理成立。

$k\geq 3$ 时, 因第3阶的输入(编码值及状态)与第1阶完全相同, 故第3阶的输出(横纵坐标和第4阶状

¹⁾对数据不向原点偏斜的情形, 可通过适当的坐标变换, 如旋转、平移等, 使其尽可能地向原点偏斜。

态)与第1阶相同;同理,第4阶的输出和与第2阶相同,进入循环状态,定理同样成立。

证毕

基于定理2, FZF-HD无需对 Z 前部编码值为0的阶进行解码。同编码算法,本文同样采用置位检测算法^[25]来检测编码 Z 中第1个被置位为1的位置 p ,并根据定理2直接确定编码值前部全为0的前 k 阶($k = n - p/2 - 1$)的坐标值和第 $k + 1$ 阶状态。对 $k + 1$ 阶及之后的各阶,迭代查询状态视图InvKey和InvType,逐阶获取该阶对应的坐标值和下一阶状态。最终实现的FZF-HD算法如表4所示。

表4 FZF-HD算法

输入: Z , Hilbert编码
n , 阶
输出: X , 横坐标
Y , 纵坐标
(1) $X, Y \leftarrow 0$
(2) $p \leftarrow \text{msb}(Z)$ //置位检测
(3) $t \leftarrow (n-p/2-1)$ //计算第 $n-p/2$ 阶状态
(4) for i from $n-p/2$ to n
(5) $\text{coord} = \text{InvKey}[t][z_{2i-1}z_{2i}]$ //查InvKey视图
(6) $Y = Y \ll 1 \text{coord} \& 0x1$
$X = X \ll 1 \text{coord} \gg 1 \& 0x1$
(7) $t = \text{InvType}[t][z_{2i-1}z_{2i}]$ //查InvType视图

算法复杂度分析:算法的主要开销在于迭代查询状态视图的操作,算法共需迭代 $m = n - k = \log_2(Z)$ 次,每次执行2次查表操作,因此算法的时间复杂度为 $O(m)$ 。算法的空间开销主要为2个状态视图的开销,因InvKey和InvType分别存储2位二进制表示的坐标值和4种状态值,故均设计为char类型,最终需要的空间同样为 $2 \times 4 \times 4 \times \text{sizeof}(\text{char}) = 32$ Byte。

4 实验结果比较及分析

4.1 实验环境及数据集

实验主要硬件平台: Intel(R) Core(TM) i5-3230m CPU @ 2.60 GHz双核,内存4 GB。软件环境: windows 10 64位, Microsoft Visual Studio C++ 2015且禁用了优化(/Od)。

本文的数据集考虑均匀分布和偏斜分布两种情况。对均匀分布,分别为4, 8, 16, 24, 32阶各随机生成 10^6 个坐标点(用于编码)及编码值(用于解码)。对偏斜分布,为考察数据偏斜程度对算法的影响,引入偏斜阶数 α 和偏斜率 β 两个参数。其中 α 表示前 α 阶取值全为0, β 则表示生成的数据中前 α 阶取值全

为0的部分占全部数据的百分比。例如, $\alpha = 10$ 和 $\beta = 40\%$,表示随机生成40%的数据的前10阶全为0。本文为每个特定的 α 和 β ,各生成1百万条32阶的随机坐标数据(用于编码)或编码数据(用于解码)作为实验数据。

为考察本文算法的有效性,将本文算法与本领域代表性的算法Li^[16], Burkardt^[18], Moore^[19], uniState^[17] 4种算法进行对比。

4.2 编码

4.2.1 均匀分布

为了考察均匀分布下阶数对不同算法编码效率的影响,给定上述生成的均匀分布的数据,比较不同算法的时间开销如图3所示。

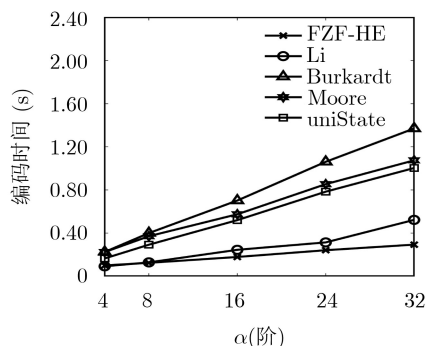


图3 均匀分布下编码效率对比

由图3可见,所有算法的编码时间均随着阶数的增加而增加。相比较而言, FZF-HE算法优于其余4种算法,且其编码时间随阶数增加的幅度最小。以32阶为例, FZF-HE编码仅需0.29 s, Li的算法需要0.52 s, 其它3种算法均需1 s以上。这些结果表明FZF-HE算法通过避免对坐标前部全为0的阶进行处理能有效提升编码效率。

4.2.2 偏斜分布

为考察偏斜分布下不同算法的效率,分别从以下两个方面进行实验对比: (1)固定 $\alpha = 16$, 设置 β 分别为25%(对应均匀分布), 50%, 75%, 100%; (2)固定 $\beta = 50\%$, 设置 α 分别为8, 12, 16, 20, 24, 实验结果分别如图4和图5所示。

由图4可见,在 $\alpha = 16$ 时, FZF-HE的编码时间大幅低于其他算法,且随着 β 的增大而明显降低,而其它算法则不随 β 的变化而明显变化。例如,在 β 为100%时, FZF-HE的编码时间为0.17 s, 仅为另外4种算法中最优的Li的算法时间的47.6%。分析可知对FZF-HE而言,随着 β 的增大,前 α 阶为0的数据量逐步增大,需迭代编码的部分降低,故算法效率逐步提高。

由图5可见, FZF-HE算法的编码时间最低,

且随着 α 增大而减小，而其他算法则不随 α 的变化明显变化。这是因为随着 α 增大，数据中无需编码的阶增加，所以算法效率逐步提高。

4.3 解码

4.3.1 均匀分布

为了考察均匀分布下阶数对不同算法解码效率的影响，给定上述生成的均匀分布的数据，比较不同算法的时间开销如图6所示。

由图6可见，解码结果与编码结果类似，总体来看，FZF-HD算法仍为最优的算法，同样表明FZF-HD算法通过避免对编码值前部全为0的阶进行解码，从而能有效提升解码效率。

4.3.2 偏斜分布

为考察偏斜分布对不同算法的影响，分别从以下两个方面进行实验对比：(1)固定 $\alpha = 16$ ，设置

β 分别为25%(对应均匀分布)，50%，75%，100%；(2)固定 $\beta = 50%$ ，设置 α 分别为8, 12, 16, 20, 24，实验结果分别如图7和图8示。

由图7可见，解码的实验结果与编码类似，FZF-HD的解码时间最低，且随着 β 的增大而降低，其它算法则不随 β 的变化而明显变化。当 β 为100%时，FZF-HD解码 10^6 个32阶数据仅需0.14 s，效率同样远优于其它算法。对FZF-HD而言，随着 β 的增大，前 α 阶为0的数据量逐步增大，故需迭代解码的部分降低，算法效率逐步提高。

由图8可见，FZF-HD算法的解码时间随着 α 增大而减小，其它算法则不随 α 变化而明显变化；因为随着 α 的增大，数据中无需迭代解码的阶增加，故算法效率逐步提高。

5 结束语

本文在设计高效状态视图并结合快速置位检测技术的基础上，提出高效的编码算法FZF-HE和解码算法FZF-HD，可避免对输入数据前部全为0的阶进行编码或解码，从而可降低查询状态视图的次数，提高编解码的效率。扩展实验表明，本文算法优于现有算法且更适合数据偏斜分布的场合。下一步工作拟将FZF-HE和FZF-HD算法应用于先前采用Z曲线解决的问题或领域^[13,14]，以扩展Hilbert编解码算法的应用范围；此外，通过并行化等手段研究进一步提高Hilbert编解码算法效率的方法。

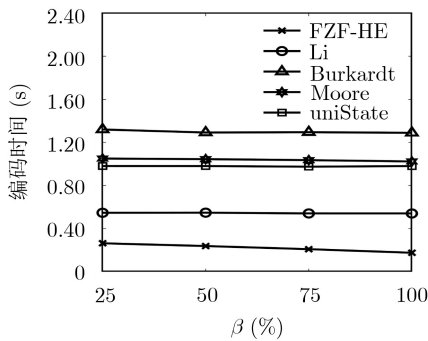


图4 β 对编码效率的影响

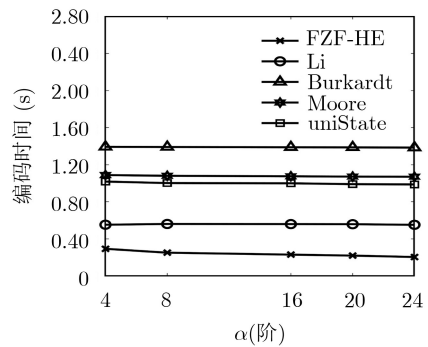


图5 α 对编码效率的影响

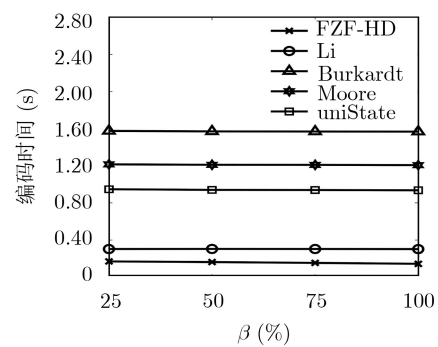


图7 β 对解码效率的影响

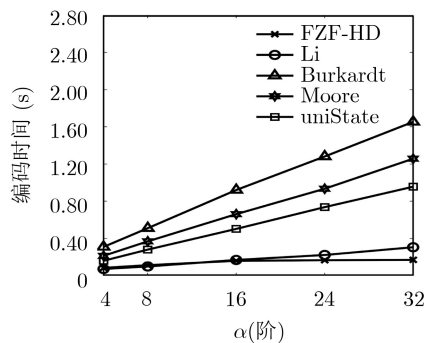


图6 均匀分布下解码效率对比

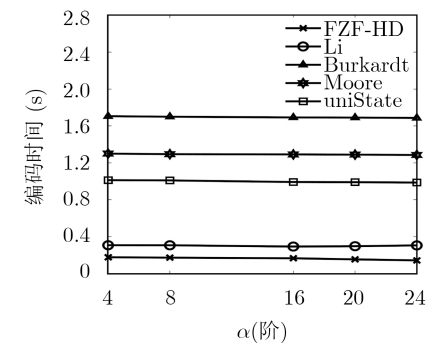


图8 α 对解码效率的影响

参考文献

- [1] GUNTHER S and STOCCO L. Generation of spatial orders and space-filling curves[J]. *IEEE Transactions on Image Processing*, 2015, 24(6): 1791–1800. doi: [10.1109/TIP.2015.2409571](https://doi.org/10.1109/TIP.2015.2409571).
- [2] 周映虹, 马争鸣. 基于层次模型的重要性编码算法[J]. *电子与信息学报*, 2009, 31(6): 1497–1500.
ZHOU Yinghong and MA Zhengming. A significance coding algorithm based on layer model[J]. *Journal of Electronics & Information Technology*, 2009, 31(6): 1497–1500.
- [3] CHEN Lu, GAO Yunjun, LI Xinhan, *et al.* Efficient metric indexing for similarity search and similarity joins[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2017, 29(3): 556–571. doi: [10.1109/TKDE.2015.2506556](https://doi.org/10.1109/TKDE.2015.2506556).
- [4] SINGH H and BAWA S. A survey of traditional and mapReduceBased spatial query processing approaches[J]. *ACM SIGMOD Record*, 2017, 46(2): 18–29. doi: [10.1145/3137586.3137590](https://doi.org/10.1145/3137586.3137590).
- [5] ABDULJABBAR M, MARKOMANOLIS G S, IBEID H, *et al.* Communication reducing algorithms for distributed hierarchical N-Body problems with boundary distributions[C]. The 32nd International Supercomputing Conference, Frankfurt, Germany, 2017: 79–96.
- [6] LIU Hui, WANG Kun, YANG Bo, *et al.* Dynamic load balancing using hilbert space-filling curves for parallel reservoir simulations[C]. The SPE Reservoir Simulation Conference, Montgomery, USA, 2017. doi: [10.2118/182613-MS](https://doi.org/10.2118/182613-MS).
- [7] HAN Guangjie, XUAN Xuan, LIU Li, *et al.* A disaster management-oriented path planning for mobile anchor node-based localization in wireless sensor networks[J]. *IEEE Transactions on Emerging Topics in Computing*, 2020, 8(1): 115–125. doi: [10.1109/TETC.2017.2687319](https://doi.org/10.1109/TETC.2017.2687319).
- [8] KONG Lingping, PAN J S, SUNG T W, *et al.* An energy balancing strategy based on hilbert curve and genetic algorithm for wireless sensor networks[J]. *Wireless Communications and Mobile Computing*, 2017, 2017: 5720659.
- [9] BOHM C, PERDACHER M, and PLANT C. A novel hilbert curve for cache-locality preserving loops[J]. *IEEE Transactions on Big Data*, To be published. doi: [10.1109/TBDATA.2018.2830378](https://doi.org/10.1109/TBDATA.2018.2830378).
- [10] MORTON G M. A computer oriented geodetic data base and a new technique in file sequencing[EB/OL]. <https://domino.research.ibm.com/library/cyberdig.nsf/0/0dabf9473b9c86d48525779800566a39?OpenDocument>, 1966.
- [11] BUTZ A R. Alternative algorithm for Hilbert's space-filling curve[J]. *IEEE Transactions on Computers*, 1971, C-20(4): 424–426. doi: [10.1109/T-C.1971.223258](https://doi.org/10.1109/T-C.1971.223258).
- [12] MOON B, JAGADISH H V, FALOUTSOS C, *et al.* Analysis of the clustering properties of the hilbert space-filling curve[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2001, 13(1): 124–141. doi: [10.1109/69.908985](https://doi.org/10.1109/69.908985).
- [13] NAGARKAR P, CANDAN K S, and BHAT A. Compressed spatial hierarchical bitmap (cSHB) indexes for efficiently processing spatial range query workloads[J]. *Proceedings of The VLDB Endowment*, 2015, 8(12): 1382–1393. doi: [10.14778/2824032.2824038](https://doi.org/10.14778/2824032.2824038).
- [14] CHRISTOFORAKI M, HE Jinru, DIMOPOULOS C, *et al.* Text vs. space: Efficient geo-search query processing[C]. The 20th ACM International Conference on Information and Knowledge Management, Glasgow, UK, 2011: 423–432.
- [15] FISHER A J. A new algorithm for generating Hilbert curves[J]. *Practice and Experience*, 1986, 16(1): 5–12. doi: [10.1002/spe.4380160103](https://doi.org/10.1002/spe.4380160103).
- [16] 李绍俊, 钟耳顺, 王少华, 等. 基于状态转移矩阵的Hilbert码快速生成算法[J]. *地球信息科学学报*, 2014, 16(6): 846–851.
LI Shaojun, ZHONG Ershun, WANG Shaohua, *et al.* An algorithm for Hilbert ordering code based on state-transition matrix[J]. *Journal of Geo-Information Science*, 2014, 16(6): 846–851.
- [17] XCTORRES. Hilbert_spatial_index[EB/OL]. https://github.com/xcTorres/hilbert_spatial_index, 2017.
- [18] BURKARDT J. HILBERT_CURVE convert between 1D and 2D coordinates of Hilbert curve[EB/OL]. http://people.math.sc.edu/Burkardt/c_src/hilbert_curve/hilbert_curve.html, 2015.
- [19] MOORE D. Fast Hilbert curve generation, sorting, and range queries[EB/OL]. <https://github.com/Cheedoong/hilbert>, 2016.
- [20] 曹雪峰, 万刚, 张宗佩. 紧致的Hilbert曲线Gray码索引算法[J]. *测绘学报*, 2016, 45(S1): 90–98. doi: [10.11947/j.AGCS.2016.F010](https://doi.org/10.11947/j.AGCS.2016.F010).
CAO Xuefeng, WAN Gang, and ZHANG Zongpei. Compact Hilbert curve index algorithm based on Gray code[J]. *Acta Geodaetica et Cartographica Sinica*, 2016, 45(S1): 90–98. doi: [10.11947/j.AGCS.2016.F010](https://doi.org/10.11947/j.AGCS.2016.F010).
- [21] LI R. Hilbert range query decomosition[EB/OL]. https://github.com/cloudray8580/Hilbert_RangeQuery_Decomosition, 2019.
- [22] ZHANG Jian and KAMATA S I. A generalized 3-D Hilbert scan using look-up tables[J]. *Journal of Visual Communication and Image Representation*, 2012, 23(3): 418–425. doi: [10.1016/j.jvcir.2011.12.005](https://doi.org/10.1016/j.jvcir.2011.12.005).

- [23] LIU Hui, CUI Tao, LENG Wei, *et al.* Encoding and decoding algorithms for arbitrary dimensional Hilbert order[J]. 2016, arXiv: 1601.01274.
- [24] 刘辉, 冷伟, 崔涛. 高维Hilbert曲线的编码与解码算法设计[J]. 数值计算与计算机应用, 2015, 36(1): 42-58.
LIU Hui, LENG Wei, and CUI Tao. Development of encoding and decoding algorithms for high dimensional Hilbert curves[J]. *Journal on Numerical Methods and Computer Applications*, 2015, 36(1): 42-58.
- [25] ROSETTA. Find first and last set bit of a long integer[EB/OL]. http://rosettacode.org/wiki/Find_first_and_last_set_bit_of_a_long_integer, 2019.
- 贾连印: 男, 1978年生, 副教授, 研究方向为数据库、信息检索、并行计算.
- 陈明鲜: 男, 1994年生, 硕士生, 研究方向为图像处理、信息检索和自然语言处理.
- 李孟娟: 女, 1983年生, 馆员, 研究方向为并行计算和信息检索.
- 游进国: 男, 1978年生, 副教授, 研究方向为数据库和数据仓库.
- 丁家满: 男, 1974年生, 副教授, 研究方向为数据库和数据挖掘.