LDM 地震数据体的存储格式分析与应用①

申龙斌[1], 刘　昶[1], 王贵斌[2], 于　静[3]

[1](中国石化胜利油田分公司 物探研究院, 东营 257022)

[2](中国石化胜利油田分公司 现河采油厂, 东营 257342)

[3](中国石化地球物理公司 胜利分公司, 东营 257086)

摘　要: 在油气勘探过程中, 地震数据体是最重要的基础数据之一. 地震数据体通常采用 SEGY 标准进行存储, 随着油田信息的三维可视化技术的发展, 为适应快速抽线和体渲染的需求, 出现了 Large Data Management(LDM)存储格式, 但 LDM 格式的技术细节并不公开. 本文通过定制 LDM 数据体, 分析各个数据块的存储位置, 并结合八叉树存储的基本原理, 详细分析了 LDM 的数据存储格式并给出了算法实现. 根据此算法实现的地震剖面抽取显示模块节省了存储空间, 并提高了数据抽取效率.

关键词: 地震数据体; 地震剖面; 三维可视化; 八叉树

Storage Format Analysis and Application of LDM Seismic Data Volume

SHEN Long-Bin[1], LIU Chang[1], WANG Gui-Bin[2], YU Jing[3]

[1](Shengli Geophysical Research Institute of SINOPEC, Dongying 257022, China)

[2](Xianhe Plant of Shengli Oilfield Company of SINOPEC, Dongying 257342, China)

[3](Shengli Branch of Sinopec Geophysical Company, Dongying 257086, China)

Abstract: In the process of oil and gas exploration, seismic data is one of the most important basic data. Seismic data are usually stored by the SEGY standard. With the development of 3D visualization technology in oilfield information, the Large Data Management (LDM) storage format was born to fit the needs of fast seismic line plucking and volume rendering. But the technical details of LDM format are not open. The paper, through customizing the LDM data, analyzing the storage position of each block data, analyses the storage format of LDM in detail, and gives the algorithm implementation, which combined with the basic principle of octree storage. This algorithm, under which the display module is extracted by the seismic profile, saves the storage space, and improves the efficiency of data extraction.

Key words: seismic data; seismic profile; 3D visualization; octree

① 收稿时间:2014-09-18; 收到修改稿时间:2014-10-13

在石油行业中, 地震资料解释工作是地震勘探的重要环节, 野外地震采集工作得到的原始资料, 经过室内处理后, 得到可供解释的地震剖面(三维地震数据体)和其它成果图件, 解释人员要对这些资料进行分析研究, 从而达到了解、推断地下地质情况的目的[1]. 三维地震数据体是最重要的基础数据之一, 地质研究人员需要利用解释软件中的二维剖面显示模块和三维可视化模块对地震数据体开展地质解释工作, 再结合其它资料进行综合对比分析, 从而找出有利的勘探目标. 地震数据存储通常采用 SEGY 标准[2], 在从二维地震向三维地震过渡时, 该标准进行了兼容性的更新, 将每一条测线顺序存储在一起, 就形成了三维地震数据体. 而解释软件中的地震剖面显示模块, 通常直接借用该存储标准, 此存储格式对于纵剖面的抽取显示性能没有影响, 但在显示横剖面和时间切片时带来了性能问题, 更不能适应三维可视化的显示需求. 所以一般解释软件在实现这些功能时, 会增加横线顺序和时间顺序等不同的格式来存储同一份地震数据体, 用空间来换取性能上的提升. 2002 年, Mercury Computer Systems 公司在 OpenInventor 的 VolumeViz 3.0 扩展模块中开始采用 LDM 存储技术[3], 可以快速从各个方向切片以及从大数据体中切取子体, 但由于公司的商业

利益, LDM 的技术细节并不公开, 因此研究解析 LDM 的技术可以实现地震数据体的统一存储, 既可节省大量存储空间, 还可实现多机并行存储和并行抽取, 从而进一步提升抽取剖面的性能, 在地震数据处理和解释软件中都有很广阔的应用前景.

1 地震数据体的存储格式

1.1 SEGY 存储格式

SEGY rev0 数据交换格式是 1975 年由勘探地球物理家学会(the Society of Exploration Geophysicists, 缩写为 SEG)推出的, 该格式在地球物理界得到了广泛的应用, 随着三维数据采集技术以及高速度、大容量记录媒体的应用, 该标准已经不能满足数据采集、处理及存储的需求, 所以 2002 年, SEG 技术标准委员会(the SEG Technical Standards Committee)推出了新的格式标准 SEGY rev1.0[4]. 虽然该标准提供了扩展文件头的定义, 但石油行业仍多采用 3200 文件头和 400 字节二进制头的存储格式, 每道仍采用 240 字节的道头字, 如图 1 所示.

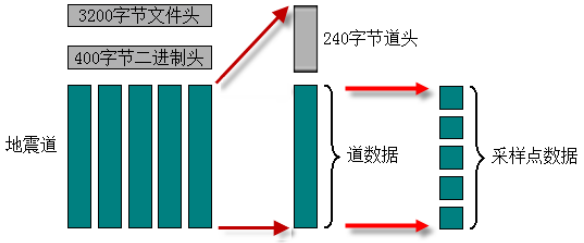

图 1 地震数据体的 SEGY 存储格式示意图

1.2 LDM 存储格式

由于 SEGY 格式重点要支持磁带介质, 所以是按照地震道按顺序存储的, 对于三维地震数据体来说, 先记录最小测线号的所有地震道, 然后依次将所有测线的地震道数据写入, 为了程序开发方便, 许多应用程序就沿用了这种存储格式. 这些程序在抽取纵测线时没有问题, 但在抽取横测线和时间切片时, 性能会受到影响, 因此很多地震剖面显示软件会增加纵线顺序、横线顺序和时间顺序 3 种方式来保证剖面抽取的效率, 随着近几年三维可视化模块的发展, 还要能够对任意一个子体进行体渲染, 因此又出现了 LDM (Large Data Management) 存储方式. LDM 是 VSG(Visualization Sciences Group) 开发的 Open Inventor 三维可视化组件中用于存储地震数据体的一种主要存储格式, LDM 用多个精细度(Level)的八叉树结构进行组织, 从而满足快速可视化交互的用户体验.

LDM 技术主要基于两项假设, 第一假设是 SEGY 数据(一种标准地震数据体记录格式)包含的空间范围较大, 但在某一时刻的用来计算的数据所在的空间位置却是局部的, 局部之外的数据对此一时刻的计算来说是无意义的. 第二假设是对局部数据在精细度方面的要求以及对不同场景使用的计算方法的需求是不一样的, 各种计算方法可以灵活处理. 为了达到这两个假设的目标, LDM 技术对数据的管理原则是: 按空间位置分块, 按精细度分层(图 2). 考虑到处理数据时的便捷性, 各块各层数据的大小相同(默认大小为 64x64x64 个数据采样点, 在 LDM 中称为 Tile, 以下统一称为砖块), 在相邻的两层数据之间, 下层数据块的精细度在 u、v、w 三个维度上都是上层的 2 倍, 数据块的数量是上层的 8 倍, 这就非常适合用八叉树结构来管理这些数据块. 这样用户在处理数据时, 只需要根据空间位置定位需要的那些数据块, 再根据计算机的内存、显存、显示器的分辨率、实时性等要求筛选出适当精细层当中的数据块进行处理. 在存储时都是最先变化 u, 再变化 v, 最后变化 w.

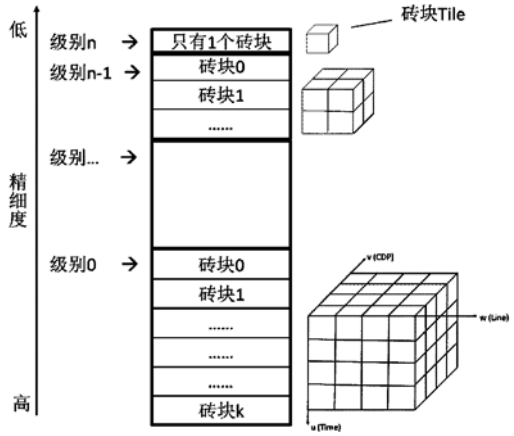

图 2 LDM 数据体的组织方式

由于地震数据体的数据量很大, 无法一次加入内存进行可视化, 尤其在 GUI 程序中, 对海量数据进行可视化会造成程序反应迟滞, 用户体验很差. 应用 LDM 技术对数据进行预处理, 可为可视化节约大量的数据检索时间和文件 IO 时间; 在预处理时为数据提供

了多个精细度的数据备份, 在不同条件下可动态使用适合的精细度数据, 可以为数据的可视化节约大量的时间; 把每个精细度的数据分成块, 数据可视化时按需筛选有用的数据块, 能加快可视化渲染的效率; 在数据进行可视化之前, 对数据进行分析统计, 把常用的体数据统计信息(值域范围、值的分布数据等)预存起来, 在可视化时可以直接使用, 从而提高可视化的性能.

因此 LDM 存储技术只存储一套数据体(比原来的一个 SEGY 大约多 25%), 却同时存储了多个精细度的数据, 可同时满足按块、按切片、按测线等方式的获取数据的需求, 获得了更好的用户体验. 但 LDM 存储格式出于公司的商业利益, 无从获得更详细的内部细节, 为此解析该格式的存储格式是开展相关应用程序研发的重要技术之一.

2 问题定义

已知一个地震数据体共有 W 条测线, 每条测线有 V 个 CDP, 每个 CDP 的地震道有 U 个采样点, 每个采样点用 B 字节存储, 当用 LDM 格式存储时, 每个砖块的三个维度大小都为 D, 给定一个空间点(u,v,w), 即第 w 条测线, 第 v 个 CDP, 第 u 个采样点, 求其在第 0 层数据块(最精细数据块)中的存储位置.

整个问题稍显复杂, 但由于 LDM 是按不同精细度 Level 分层存储的, 所以该问题可以分解为 4 个子问题:

问题 1: LDM 中共有几个精细度级别(Level)?

问题 2: 每个精细度级别中有多少个砖块?

问题 3: 采样点(u,v,w)对应于 LDM 的第 0 层精细度数据(最精细数据)的第几个砖块?

问题 4: 采样点(u,v,w)在第 0 层砖块的采样点编号为多少?

为了说明问题方便, 给出一个示例地震数据体, 该数据体共有 97 条测线, 每条测线有 133 个 CDP, 每个 CDP 的地震道有 2001 个采样点, 即 U=2001, V=133, W=97, 每个采样点用 4 字节浮点数存储, 即 B=4, 砖块大小为 64x64x64, 即 D=64, 每个砖块大小为 B*D3=4*64*64*64=1048576 字节.

3 LDM解析过程分析

3.1 精细度级别

由图 2 可以看出, 一个地震数据体在转换成 LDM 时会不断地每隔一个数据点抽稀采样, 直到最低精细度能够用一个砖块表示为止, 所以精细度级别的个数是由 U,V,W 中的最大值来决定的.

设最大的精细度级别为 L, 则有:

$$L=\left\lceil \log_2 \frac{\max\{U,V,W\}}{D} \right\rceil \tag{1}$$

也就是说 LDM 中共有(L+1)个精细度级别.

对于问题定义中的示例数据来说, 代入式 1 可以得到 L=5, 表示该 LDM 的最大精细度级别为 5, 也就是说共计有 6 个精细度级别, 级别 0 为最精细数据, 级别 5 为最粗糙数据.

3.2 砖块个数

数据按砖块方式分块划分后, 在 LDM 存储时由于需要统一编码, 所以三个维度上要使用统一的大小, 由于实际的数据体的 U, V, W 并不一样, 所以许多的砖块中的数据全部是 0, 这些砖块称为空砖块. 在 LDM 中并未保存这些空结点. 所以问题 2 需要回答第 i 级数据中共有多少个砖块?有多少个非空砖块?

从第 0 层开始分析, u,v,w 三个方向上 Tile 的最大维度为$\lceil \max(U,V,W)/D \rceil$, 总共有$\lceil \max(U,V,W)/D \rceil^3$个结点, 非空结点个数为$\lceil U/D \rceil * \lceil V/D \rceil * \lceil W/D \rceil$.

第 i 层由于抽稀采样, 所以总共有$\lceil \max(U,V,W)/D^{i+1} \rceil^3$个结点, 非空结点数为$\lceil U/D^{i+1} \rceil * \lceil V/D^{i+1} \rceil * \lceil W/D^{i+1} \rceil$.

仍以前面的数据体为例, 每层的砖块数如表 1 所示.

表 1 每层精细度数据体的砖块个数

精细度	最大维度方向上的砖块个数	总砖块数	非空砖块个数	起始偏移量($B*D^3$)
5	1	1=1*1*1	1=1*1*1	0
4	2	$8=2^3$	2=2*1*1	1
3	4	$64=4^3$	4=4*1*1	3
2	8	$512=8^3$	8=8*1*1	7
1	16	$4096=16^3$	32=16*2*1	15
0	32	$32768=32^3$	192=32*3*2	47
合计		37449	239	

LDM 中只存储了非空砖块, 所以累计各层的非空砖块数, 就可以定位到第 0 层精细度数据所在的位置, 对于示例数据体, 第 0 层精细度的数据块之前共有 47 块(即 1+2+4+8+32), 即偏移量字节数为 49283072(即 47*64*64*64*4).

3.3 砖块索引号

由于 LDM 中数据存放的顺序并没有公开的资料可查，所以这里采用制作特殊 SEG-Y 文件转换成 LDM，通过认真对比每个数据块在转换前后位置的变化，可得到存储位置关系.

主要步骤:

1)以问题定义中的 SEG-Y 为例，每个采样点的数值都经过特殊设计，可以采用 V = line * 1000 + cdp + time * 0.0001，例如：在采样点(line:24，cdp:120，time:346)上的数值为 24120.0346.

2)用 open inventor 9.0 自带的 LDMConverter.exe 程序将这个 SEG-Y 转换为 LDM，输出的 LDM 默认是按 4 字节 IEEE 浮点来存储，Tile 大小采用默认的 64x64x64.

转换过程中程序给出几条提示信息，这三行信息正好与前 2 个问题相对应.

octree level: 5

octree #tiles: 37449

octree #not empty tiles: 239

转换成功后，会生成 2 个文件，一个扩展名为 LDM，一个扩展名为 DAT.

扩展名为 LDM 的文件用 XML 描述 LDM 中各个维度的大小、存储格式、数据值分布情况等信息，扩展名为 DAT 的文件用来存储地震数据体的振幅值.

3)编写程序读取最高精细度数据中的每个 Tile 的第一个数据项，根据 1)中的公式反向计算出 line,cdp,time，这样就可以逐步将所有的采样点的排列顺序确定出来. 经过分析，该排列符合八叉树中的排列规律.

仍以前面的数据体为例，总共有 239 个非空 Tile，最高精细度的数据有 192 个非空 Tile，这 192 个非空 Tile 是用三维 Morton 编码的顺序来存放的.

为了理解方便，先来了解二维 Morton 编码. 一名叫 Morton 的加拿大学者于 1966 年提出了一种编码，用于将二维坐标转换为一维的坐标值来记录，后人称为莫顿码(Morton Codes)，也称为 z-order. 图 3 说明了二维莫顿码的编码方式，它定义了一种映射关系: (X,Y)-->M，其中 X,Y 是从 0 到 7 的整数，M 是从 0 到 63 的整数. M 是由 X,Y 的二进制数交织而成的，对于 x=1, Y=2, x 的二进制为 001, Y 的二进制为 010, X 和 Y 的二进制码交织排列后为 000110，即十进制的 6. 再举一个例子，对于 x=6, Y=7, x 的二进制为 110, Y 的二进制为 111, X 和 Y 的二进制码交织排列后为 111101，即十进制的 61.

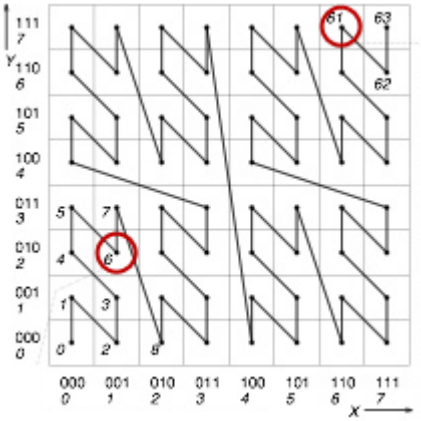

图 3 二维莫顿码编码示意图

很容易将此推广到三维空间，它定义了一种映射关系: (u,v,w)-->M，其中 u,v,w 分别是从 0 到 3 的整数，M 是从 0 到 63 的整数. 如图 4 所示，对于 u=3,v=0,w=1，二进制交织后为 001101，即十进制的 13. 对于 u=2,v=1,w=3，二进制交织后为 101110，即十进制的 46.

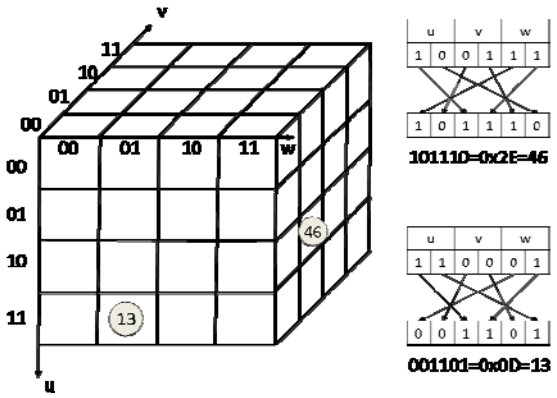

图 4 三维莫顿编码示意图

三维莫顿码的 C 语言源程序为:

```
/// 三维 morton 编码，
/// x,y,z 都是不超过 10 位二进制的整数，即从 0 到 1023
/// 返回的 morton 编码为 30 位二进制的整数
/// 这里用了一些位运算技巧来提高计算速度
public static int MortonCode(int x, int y, int z)
{
    x = (x | (x << 16)) & 0x030000FF;
    x = (x | (x << 8)) & 0x0300F00F;
```

```
    x = (x | (x << 4)) & 0x030C30C3;
    x = (x | (x << 2)) & 0x09249249;
    y = (y | (y << 16)) & 0x030000FF;
    y = (y | (y << 8)) & 0x0300F00F;
    y = (y | (y << 4)) & 0x030C30C3;
    y = (y | (y << 2)) & 0x09249249;
    z = (z | (z << 16)) & 0x030000FF;
    z = (z | (z << 8)) & 0x0300F00F;
    z = (z | (z << 4)) & 0x030C30C3;
    z = (z | (z << 2)) & 0x09249249;
    return x | (y << 1) | (z << 2);
}
```

到现在，问题还没有解决，这个编码方案是对所有 Tile 进行编号的，但 LDM 只存储了非空的 Tile，因此需要将莫顿码再映射到非空 Tile 编号，才是最终的存储位置.

以二维莫顿码为例，当编码中有空结点时，最后将对所有非空结点排序，图 5 显示了一个示例.

y \ x	0	1	2	3	4	5	6	7
0	0	1	4	5	16	x	x	x
1	2	3	6	7	18	x	x	x
2	8	9	12	13	24	x	x	x
3	10	11	14	15	26	x	x	x
4	32	33	36	37	48	x	x	x
5	34	35	38	39	50	x	x	x
6	40	41	44	45	56	x	x	x
7	x	x	x	x	x	x	x	x

初始morton编码

⇒

y \ x	0	1	2	3	4	5	6	7
0	0	1	4	5	16	x	x	x
1	2	3	6	7	17	x	x	x
2	8	9	12	13	18	x	x	x
3	10	11	14	15	19	x	x	x
4	20	21	24	25	32	x	x	x
5	22	23	26	27	33	x	x	x
6	28	29	30	31	34	x	x	x
7	x	x	x	x	x	x	x	x

重新排序后的编码

图 5　非空砖块的重新排序

在程序实现的初始化时要根据 u,v,w 的大小，判断出有哪些空结点，然后形成一个映射表，每次计算得到莫顿码后，再进行一次查表，就可以得到 LDM 数据体中的存储位置，该算法的伪代码:

```
// 这个数组用于记录非空结点的莫顿码
List mortonList;
// 加入所有非空 Tile 的莫顿编码
    mortonList ← mortonCode(u,v,w), 其中:
0 ≤ u < ⌈U / D⌉,  0 ≤ v < ⌈V / D⌉, 0 ≤ w < ⌈W / D⌉
// 排序后所处的位置就是在 LDM 中的存储位置
mortonList.Sort();
// 用一个字典把这种对应关系记录下来
Dict<int,int> dict;
dict <-- (mortonList[i], i)  其中:
0≤i < mortonList.Count
```

有了这个 dict 之后，想查找某个 Tile 在 LDM 中的位置就非常容易了，以图 5 为例，莫顿码为 35 的砖块对应的排序号为 23，则表示在 LDM 中直接跳到第 23 个块中就可以直接读出这个数据块了.

3.4 采样点编号

一个数据块得到后，它包含着 64x64x64 个数据(以 D=64 为例)，在这个数据块中，也是先变化 u，再变化 v，最后变化 w，所以(u,v,w)所在相对位置用取模运算和一些乘法就可以得出.

```
public float GetSample(float []tileData, int u, int v, int w)
{
    int modU = u % D;
    int modV = v % D;
    int modW = w % D;
    int index = modU + D * (modV + D * modW);
    return tileData[index];
}
```

上面叙述了从 LDM 中取出任意一个采样点的过程，但实际应用中通常是从 LDM 中取出一个切片的数据，所以只需根据以上的代码把取出的数据放到准确的数组位置上即可，这里不再赘述.

4　应用效果

胜利油田的勘探决策支持系统的应用在 2007 年全面推广应用，其中的地震数据体仍采用纵线顺序、横线顺序和时间顺序 3 套 SEGY 方式存储，2012 年发布了三维可视化模块，又增加了 LDM (Large Data Management)存储方式，这样同一个地震数据体，共存储了 4 套(图 6)，既浪费了存储空间，也造成了管理和维护的困难，也难以实现并行抽线算法.

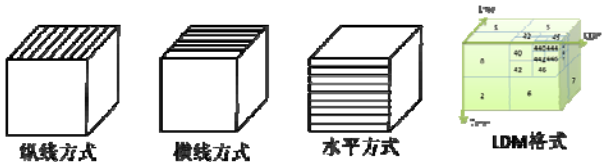

图 6　地震数据体的 4 种存储方式

在彻底分析了 LDM 存储格式的基础上，当前已经将 SEGY 开始全部转换为 LDM 存储，同时满足地震剖面模块中抽线需求和三维可视化模块中的地震体渲染功能，当前正在研究并行化的抽线算法，进一步提高算法的执行效率.

5 结语

本文全面分析了 LDM 的内部存储格式, 并给出了相应的算法实现, 可以得到以下认识:

1) 传统的 SEGY 在抽线上效率不高, 采用多精细度的 LDM 技术可提高抽线效率;

2) LDM 存储多个精细度, 所以会多占用约 25%的空间;

3) LDM 内部是用八叉树存储各个地震数据砖块的, 采用三维莫顿码不寻址;

4) LDM 文件中只存储非空砖块, 算法中要构建非空砖块序号到全部砖块序号的映射表;

5) LDM 适合于采用并行算法来进一步提高效率.

参考文献

1 陆基孟.地震勘探原理(下).北京:石油大学出版社,1993.

2 王秀文,姚立平,赖德伦等.地震数据交换标准.地震地磁观测与研究,1994,(2).

3 王家华,陈雨馨.基于 VolumeViz 的储层可视化研究与实现.软件导刊,2013,(12).

4 王侠.SEGY 数据新标准: SEG Y rev 1 . 0. 油气地球物理, 2006,(4):50–52.