Multi-index Mapping Theorems for Matrix Kronecker Products

Bu Zhou*

Abstract

This paper gives two theorems on the indexes of a series of matrices and their Kronecker product. Applications of the theorems in matrix algebra, spatial matrix calculations and voxel/3D array representation are discussed. R codes for computation are provided in the appendix.

Index Terms — Kronecker product, index mapping, matrix algebra, spatial matrix, block matrix, octree, quadtree, spatial array indexing.

1 Introduction

Matrix Kronecker product or tensor product plays some elementary roles in matrix algebra, multi-linear algebra, group theory, tensor analysis, quantum mechanics, systems and control as well as statistics, see [6, 7, 3, 1, 8, 10, 4, 13] and references therein. The authors in [2] investigate spectral properties of sums of certain Kronecker products. Defect calculation for Kronecker product of unitary matrices is studied in [9]. Results on matrix representations and computation of adjoint operators in spin half systems are provided based on tensor product of Pauli matrices [12].

For matrices $A_k = (a^k_{i_k j_k})_{l_k \times m_k} \in C^{l_k \times m_k}(\ k = 1, 2)$, their Kronecker product is defined by (see [7])

$$A_1 \otimes A_2 := \left(a^1_{i_1 j_1} A_2\right)_{l_1 \times m_1} \in C^{l_1 l_2 \times m_1 m_2} \tag{1}$$

The Kronecker multiplication operator $\otimes$ for matrices is associative and multiple factor Kronecker product $A_1 \otimes A_2 \otimes \cdots \otimes A_n$ can be defined by (1) recursively. Notice that the matrix $A_1 \otimes A_2 \otimes \cdots \otimes A_n$ defined in this way is expressed by sub-block matrices related to factor matrices $\{A_k\}^2_{k=2}$.

In this paper, we are concerned with problem for representing the entries $(A_1 \otimes \cdots \otimes A_n)_{\mathbb{ij}}$ of matrix $A_1 \otimes A_2 \otimes \cdots \otimes A_n$ in terms of entries $\left\{a^k_{i_k j_k}\right\}^n_{k=1}$ of factor matrices $\{A_k\}^n_{k=1}$. A multi-index transformation mapping is introduced by which a relation between indices of entries of factor matrices and index of entry of product matrix is established. It has been shown that the multi-index transformation mapping is a one to one correspondence. An algorithm is also provided to express the inverse index transformation mapping.

The organization of this paper is as follows. In Section 2, A multi-index transformation mapping is introduced and a main result is given. In Section 3, the inverse index transformation mapping and its corresponding result for matrix Kronecker product are discussed. In the last section, some applications including equivalent definition for matrix tensor products and computation of spatial matrices are given.

*School of Finance and Statistics, East China Normal University, Shanghai, 200241, P. R. China. Email: zhoubu1988@yahoo.com.cn

Throughout this paper, we use fairly standard notations listed as follows:

$\mathcal{C},\ \mathcal{C}^{m\times m}$	Complex field; The set of all complex $m\times m$ matrices.
$(A)_{ij},\ A_k(i,j),\ a^k_{i_kj_k}$	The ith row and jth column entry of a matrix A or a matrix A_k.
$\mathcal{S}_1\times\mathcal{S}_2$	The Cartesian product of two sets $\mathcal{S}_1$ and $\mathcal{S}_2$.
$A^T,\ A^\dagger$	Transpose of a matrix A; Conjugate transpose of a matrix A
$A\otimes B$	The Kroneckers product of two sets A and B
$\mathbb{N},\ \mathbb{N}_n$	Positive integer set; Set $\{1,\cdots,n\}$
$Card(\mathcal{S})$	Cardinality of a set $\mathcal{S}$.

2 The Multi-index Mapping

A multi-index transformation mapping will be introduced, which can establish a one-to-one correspondence between a multi-index and a single index. Suppose a set of matrices $\left\{A_k=(a^k_{ij})_{l_k\times m_k}\in C^{l_k\times m_k}\right\}_{k=1}^n$ are given. In terms of all the size pairs $\{l_k,\ m_k\}_{k=1}^n$, one can define some subsets of $\mathbb{N}$ as follows:

$$\mathcal{L}_k=\{1,\cdots,l_k\},\quad \mathcal{M}_i=\{1,\cdots,m_k\},\quad k=1,\cdots,n \tag{2}$$

$$\mathbb{L}=\left\{1,\cdots,\prod_{i=1}^n l_i\right\},\quad \mathbb{M}=\left\{1,\cdots,\prod_{i=1}^n m_i\right\} \tag{3}$$

Definition 1 (Multi-Index Mapping) *Let $\sigma^{\mathbb{r}}:\mathcal{L}_1\times\cdots\mathcal{L}_n\to\mathbb{L}$ and $\sigma^{\mathbb{c}}:\mathcal{M}_1\times\cdots\mathcal{M}_n\to\mathbb{M}$. $\forall\ (i_1,i_2,\cdots,i_n)\in\mathcal{L}_1\times\cdots\mathcal{L}_n$, and $(j_1,j_2,\cdots,j_n)\in\mathcal{M}_1\times\cdots\mathcal{M}_n$ their image $\mathbb{i}$ and $\mathbb{j}$ are defined respectively by*

$$\begin{aligned}\mathbb{i}&=\sigma^{\mathbb{r}}(i_1,\cdots,i_n)=(i_1-1)l_2\cdots l_n+(i_2-1)l_3\cdots l_n+\cdots+(i_{n-1}-1)l_n+i_n\\ \mathbb{j}&=\sigma^{\mathbb{c}}(j_1,\cdots,j_n)=(j_1-1)m_2\cdots m_n+(j_2-1)m_3\cdots m_n+\cdots+(j_{n-1}-1)m_n+j_n\end{aligned}$$

The first theorem shows the relationship between the elements indexes of a series of matrices and the elements indexes of the Kronecker product matrix.

Theorem 1 (Multi-index Mapping) *For any $k\in\mathbb{N}_n$, $A_k\in\mathcal{C}^{l_k\times m_k}$, the entry of A_k in slot (i_k,j_k) is denoted by $(A_k)_{i_kj_k}=(a^k_{i_kj_k})$, $(i_k,j_k)\in\mathcal{L}_k\times\mathcal{M}_k$. The entry in the $\mathbb{i}$th row and $\mathbb{j}$th column of $A_1\otimes\cdots\otimes A_n$ can be computed by*

$$(A_1\otimes\cdots\otimes A_n)_{\mathbb{i}\mathbb{j}}=a^1_{i_1j_1}\otimes\cdots\otimes a^n_{i_nj_n} \tag{4}$$

where

$$\mathbb{i}=\sigma^{\mathbb{r}}(i_1,\cdots,i_n)=(i_1-1)l_2\cdots l_n+(i_2-1)l_3\cdots l_n+\cdots+(i_{n-1}-1)l_n+i_n \tag{5}$$

$$\mathbb{j}=\sigma^{\mathbb{c}}(j_1,\cdots,j_n)=(j_1-1)m_2\cdots m_n+(j_2-1)m_3\cdots m_n+\cdots+(j_{n-1}-1)m_n+j_n \tag{6}$$

Proof 1. (Mathematical Induction):

We only need to prove the result for row index because the result for the column index can be proved with the same argument.

Case $n=1$: $i=i_1$, trivial.

Case $n=k$: for $A(k)=A_1\otimes A_2\otimes\cdots\otimes A_k$, let $\mathbb{i}(k)$ be some row index of $A(k)$, and i_t be the corresponding row index of A_t, $t=1,\cdots,k$, which satisfy (4). It is assumed that

$$\mathbb{i}(k)=(i_1-1)l_2\cdots l_k+(i_2-1)l_3\cdots l_k+\cdots+(i_{k-1}-1)l_k+i_k. \tag{7}$$

Then we consider **Case** $n = k + 1$**.** Note that

$$A(k+1) = (A_1 \otimes A_2 \otimes \cdots \otimes A_k) \otimes A_{k+1} = A(k) \otimes A_{k+1}.$$

Let $\mathbb{i}(k+1)$ be some row index of $A(k+1)$.

Recalling (1), we have that there exists an index i_{k+1} corresponding to matrix A_{k+1} such that

$$\mathbb{i}(k+1) = (\mathbb{i}(k) - 1)l_{k+1} + i_{k+1} \tag{8}$$

which can be verified in Figure 1.

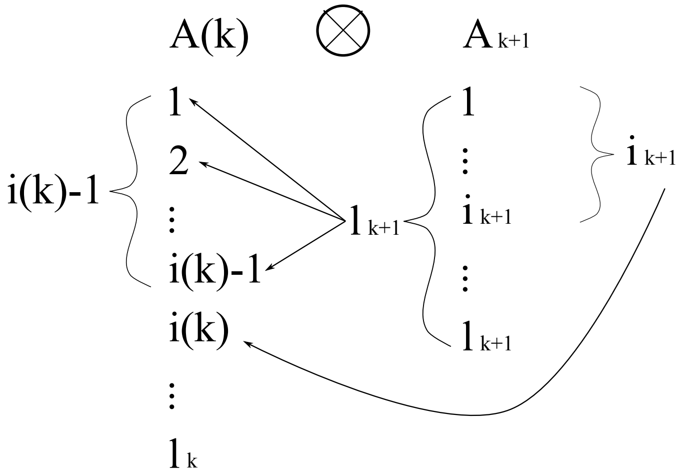

Fig.1 Relation for Row Indices $\mathbb{i}_{k+1}$ and $\mathbb{i}_k$

Substituting (7) into (8) leads to

$$\mathbb{i}(k+1) = (\mathbb{i}(k) - 1)l_{k+1} + i_{k+1} = (i_1 - 1)l_2 \cdots l_{k+1} + (i_2 - 1)l_3 \cdots l_{k+1} + \cdots + (i_k - 1)l_{k+1} + i_{k+1}$$

So the result for $n = k + 1$ holds. ■

We easily proved the result using mathematical induction. However, this proof is almost mechanical and does not touch the essence of the problem. Use the idea of trees, we can have a direct and intuitive proof which sheds more light on the results.

Proof 2 (Data Trees):

All entries of matrix $A = A_1 \otimes \cdots \otimes A_n$ can be obtained by using all entries $\{a^k_{i_k j_k}, i_k \in \mathcal{L}_k, j_k \in \mathcal{M}_k, k \in \mathbb{N}_n\}$ of matrices $A_k's$ as well as recursively using (1). Observing the all $l_1 \cdots l_n$ rows of matrix $A = A_1 \otimes \cdots \otimes A_n$, this procedure can be viewed as a process of generating a data tree with the l_1 parent branches (or roots) indexed as

$$(1), \cdots (l_1).$$

In first level, the children branches can be indexed as

$$\underbrace{\underbrace{(1,1),(1,2),\cdots,(1,l_2)}_{l_2}, \underbrace{(2,1),(2,2),\cdots,(2,l_2)}_{l_2}, \cdots\cdots, \underbrace{(l_1,1),(l_1,2),\cdots,(l_1,l_2)}_{l_2}}_{l_1}.$$

Similarly the children branches in second level are indexed as

$$\underbrace{(1,1,1),(1,1,2),\cdots,(1,1,l_3)}_{l_3},\underbrace{(1,2,1),(1,2,2),\cdots,(1,2,l_3)}_{l_3},\cdots,\underbrace{(l_1,l_2,1),(l_1,l_2,2),\cdots,(l_1,l_2,l_3)}_{l_3},$$

.

$$\vdots$$

In final level children branches (leaves) are indexed as

$$\underbrace{(1,\cdots,1,1)}_{n},\underbrace{(1,\cdots,1,2)}_{n},\cdots,\underbrace{(l_1,\cdots,l_{n-1},l_n)}_{n}.$$

An example of a three-level tree is showed in Figure 2. Now given a leaf index $(i_1, \cdots, i_n)$ in the tree, we want to know its index $\mathbb{i}$ in the product A. To find the value of $\mathbb{i}$, we only need to count the number of the leaves before the leaf indexed as $(i_1, i_2, \cdots, i_n)$ in the tree.

In the first level, $k = 1$, leaves generated from parent branches $(1), \cdots, (i_1 - 1)$, which will produce $(i_1 - 1)l_2 \cdots l_n$ leaves, comes before the leaf $(i_1, i_2, \cdots, i_n)$.

For i_1, in the second level, $k = 2$, leaves generated from branches $(2, 1), \cdots, (2, i_2 - 1)$, which will produce $(i_2 - 1)l_3 \cdots l_n$ leaves, comes before the leaf $(i_1, i_2, \cdots, i_n)$.

$\vdots$

In the final level, $k = n$, leaves $(i_1, i_2, \cdots, i_{n-1}, 1), \cdots, (i_1, i_2, \cdots, i_{n-1}, i_n - 1)$, comes before the leaf $(i_1, i_2, \cdots, i_n)$, that's $(i_n - 1)$ leaves.

So the total number of leaves before the leaf $(i_1, i_2, \cdots, i_n)$ is $(i_1 - 1)l_2 \cdots l_n + (i_2 - 1)l_3 \cdots l_n + \cdots + (i_{n-1} - 1)l_n + i_n - 1$,

that's to say, $\mathbb{i} = (i_1 - 1)l_2 \cdots l_n + (i_2 - 1)l_3 \cdots l_n + \cdots + (i_{n-1} - 1)l_n + i_n$.■

For simplicity, in the above theorem, let's call matrix A the Kronecker product (matrix), and A_k's the factor matrices.

This theorem can be used to calculate the index of a Kronecker product's element from the indexes of the factor matrices elements.

Remark 1 *The Index Mapping Theorem is a generation of Lemma 3 in* [12]*, which is the special case for a* 4×4 *matrix. This paper is part of my Master Thesis* [11].

3 The Inverse Index Mapping

Since the $a^k(i_k, j_k)$'s in (4) are unique, give a pair of index $(\mathbb{i}, \mathbb{j}) \in \mathbb{L} \times \mathbb{M}$ in the Kronecker product A, we can also find all the corresponding indexes of the factor matrices A_k by the algorithm described below.

In (2) and (3), suppose $\mathbb{i}, \mathbb{j}, l_1, \cdots, l_n, m_1, \cdots, m_n$ are known, now we can also calculate $i_1, \cdots, i_n$ and $j_1, \cdots, j_n$.

Definition 2 (Modified Module Operator) *For any* $p, q \in \mathbb{N}$*, there exist unique integers* $r, s \in \mathbb{N}\bigcup\{0\}$ *such that*

$$p = rq + s, \quad 0 \leq s \leq q - 1. \tag{9}$$

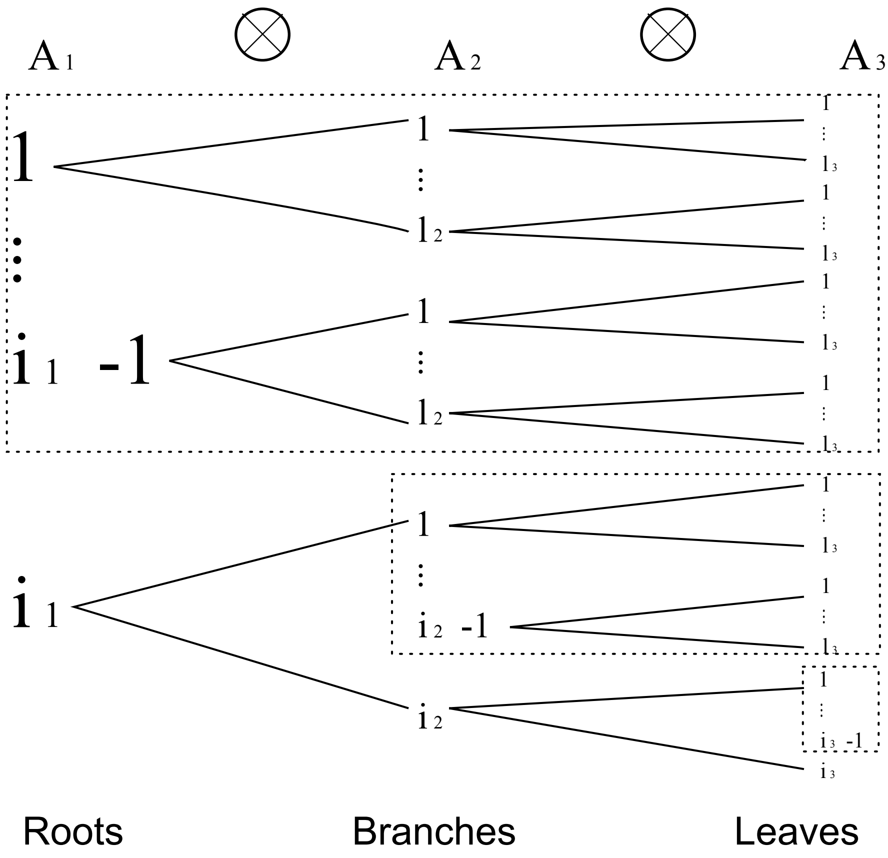

Fig.2: Three-level Tree

A modified module operator $\odot$ *is defined by*

$$\begin{cases} p \odot q = s, & 1 \leq s \leq q-1 \\ p \odot q = q, & s = 0 \end{cases} \tag{10}$$

Remark 2 *It should be pointed out that the modified module operator* $\odot$ *is different from the common mod operator because* $p \odot q = q \neq 0$ *if* $p|q$*. Apart from the difference, they are almost the same.*

Lemma 1 *For any* $p, q \in \mathbb{N}$*, there exist unique* $s \in \mathbb{N}$ *such that*

$$p \odot q = s$$

This lemma follows readily by the definition of $p \odot q$ and the uniqueness of s in (9).

Definition 3 (Inverse Multi-Index Mapping) *The inverse index mapping* $\sigma^{-\mathbb{r}} : \mathbb{L} \to \mathcal{L}_1 \times \cdots \mathcal{L}_n$ *and* $\sigma^{-\mathbb{c}} : \mathbb{M} \to \mathcal{M}_1 \times \cdots \mathcal{M}_n$ *can be defined as follows.* $\forall\, \mathbb{i} \in \mathbb{L}$ *and* $\mathbb{j} \in \mathbb{M}$*, their image* $(i_1, i_2, \cdots, i_n) \in$

$\mathcal{L}_1 \times \cdots \mathcal{L}_n$ *and* $(j_1, j_2, \cdots, j_n) \in \mathcal{M}_1 \times \cdots \mathcal{M}_n$ *are defined respectively by*

$$\begin{cases} i_n = \mathbb{i} \odot l_n \\ i_{n-1} = [(\mathbb{i} - i_n)/l_n + 1] \odot l_{n-1} \\ i_{n-2} = \{[(\mathbb{i} - i_n - (i_{n-1} - 1)l_n)]/[l_{n-1}l_n] + 1\} \odot l_{n-2} \\ \vdots \\ i_2 = \{[\mathbb{i} - i_n - (i_{n-1} - 1)l_n - \cdots - (i_3 - 1)(l_4 \cdots l_n)]/[l_3 \cdots l_n] + 1\} \odot l_2 \\ i_1 = \{[\mathbb{i} - i_n - (i_{n-1} - 1)l_n - \cdots - (i_2 - 1)(l_3 \cdots l_n]/[l_2 \cdots l_n]\} + 1 \end{cases}$$

$$\begin{cases} j_n = \mathbb{j} \odot m_n \\ j_{n-1} = \{[(\mathbb{j} - j_n)/m_n] + 1\} \odot m_{n-1} \\ j_{n-2} = \{[(\mathbb{j} - j_n - (j_{n-1} - 1)m_n)]/[m_{n-1}m_n] + 1\} \odot m_{n-2} \\ \vdots \\ j_2 = \{[\mathbb{j} - j_n - (j_{n-1} - 1)m_n - \cdots - (j_3 - 1)(m_4 \cdots m_n)]/[m_3 \cdots m_n] + 1\} \odot m_2 \\ j_1 = \{[\mathbb{j} - j_n - (j_{n-1} - 1)m_n - \cdots - (j_2 - 1)(m_3 \cdots m_n]/[m_2 \cdots m_n]\} + 1 \end{cases}$$

Theorem 2 (One to One Correspondence) *The mappings $\sigma^{\mathbb{r}}$ and $\sigma^{\mathbb{C}}$ defined in Definition 1 as well as expressed by* (5) *and* (6) *are one to one correspondences, their inverse mappings are $\sigma^{-\mathbb{r}}$ and $\sigma^{-\mathbb{C}}$ defined in Definition 3 as well as expressed by* (15) *and* (16)*.*

Proof . Only consider $\sigma^{\mathbb{r}}$.

Step 1: Using pair $(\mathbb{i}, l_n)$ to determine i_n.

Recall

$$\begin{aligned} \mathbb{i} &= \sigma^{\mathbb{r}}(i_1, \cdots, i_n) = (i_1 - 1)l_2 \cdots l_n + (i_2 - 1)l_3 \cdots l_n + \cdots + (i_{n-1} - 1)l_n + i_n \\ &= [(i_1 - 1)l_2 \cdots l_{n-1} + (i_2 - 1)l_3 \cdots l_{n-1} + \cdots + (i_{n-1} - 1)]\, l_n + i_n \end{aligned} \tag{11}$$

By noticing that $1 \le i_n \le l_n$ and using the modification module operator $\odot$, one has that

$$i_n = \mathbb{i} \odot l_n.$$

Based on lemma 1, the positive integer i_n is uniquely determined by $\mathbb{i}$ and l_n.

Step 2: Using $(\mathbb{i},\ i_n, l_n,\ l_{n-1})$ to determine i_{n-1}. It follows from (11) that

$$[(\mathbb{i} - i_n)/l_n] + 1 = [(i_1 - 1)l_2 \cdots l_{n-2} + \cdots + (i_{n-2} - 1)l_3 \cdots l_{n-2}]l_{n-1} + i_{n-1} \tag{12}$$

Similarly, one has

$$i_{n-1} = [(\mathbb{i} - i_n)/l_n + 1] \odot l_{n-1}$$

Step 3–Step $n - 1$**:** We can follow the above procedure to determine $(i_{n-2}, \cdots, i_2)$. Suppose that in the kth $(2 \le k \le n-3)$ Step, the $k+1$-tuple $(i_n, \cdots, i_{n-k})$ are obtained. Formula (11) gives

$$\begin{aligned} &[\mathbb{i} - i_n - (i_{n-1} - 1)l_n - \cdots - (i_{n-k} - 1)l_{n-k+1} \cdots l_n]/[l_{n-k} \cdots l_n] + 1 = \\ &[(i_1 - 1)l_2 \cdots l_{n-k-2} + \cdots + (i_{n-k-2} - 1)]l_{n-(k+1)} + i_{n-(k+1)} \end{aligned}$$

Recalling $1 \leq i_{n-(k+1)} \leq l_{n-(k+1)}$ leads to

$$i_{n-(k+1)} = \{[\mathbb{i} - i_n - (i_{n-1}-1)l_n - \cdots - (i_{n-k}-1)l_{n-k+1}\cdots l_n]/[l_{n-k}\cdots l_n] + 1\} \odot l_{n-(k+1)}$$

Step n**:** In this step $(i_n, \cdots, i_2)$ are obtained. The index i_1 is obtained readily by using (11) and $(\mathbb{i}, i_n, \cdots, i_2)$.

We thus conclude that $\sigma^{-\mathbb{r}}$ is an inverse mapping of $\sigma^{\mathbb{r}}$. The mapping $\sigma^{-\mathbb{c}}$ is an inverse mapping of $\sigma^{\mathbb{c}}$ can be proved by the similar arguments.

$\square$

Theorem 3 (Inverse Multi-Index Mapping) *For any $k \in \mathbb{N}_n$, $A_k \in \mathcal{C}^{l_k \times m_k}$, the entry of A_k in slot (i_k, j_k) is denoted by $(A_k)_{i_k j_k} = (a^k_{i_k j_k})$, $(i_k, j_k) \in \mathcal{L}_k \times \mathcal{M}_k$. Given the entry $(A_1 \otimes \cdots \otimes A_n)_{\mathbb{i}\mathbb{j}}$ in the $\mathbb{i}$th row and $\mathbb{j}$th column of a tensor product $A_1 \otimes \cdots \otimes A_n$, there exist n factors $\left\{a^k_{i_k j_k}\right\}_{k=1}^n$ in factor matrices $\{A_k\}_{k=1}^n$ such that*

$$a^1_{i_1 j_1} \otimes \cdots \otimes a^n_{i_n j_n} = (A_1 \otimes \cdots \otimes A_n)_{\mathbb{i}\mathbb{j}} \tag{13}$$

where

$$(i_1, \cdots, i_n) = \sigma^{-\mathbb{r}}(\mathbb{i}), \qquad (j_1, \cdots, j_n) = \sigma^{-\mathbb{c}}(\mathbb{j}) \tag{14}$$

which can be determined by

$$\begin{cases} i_n = \mathbb{i} \odot l_n \\ i_{n-1} = [(\mathbb{i} - i_n)/l_n + 1] \odot l_{n-1} \\ i_{n-2} = \{[(\mathbb{i} - i_n - (i_{n-1}-1)l_n)]/[l_{n-1}l_n] + 1\} \odot l_{n-2} \\ \vdots \\ i_2 = \{[\mathbb{i} - i_n - (i_{n-1}-1)l_n - \cdots - (i_3-1)(l_4\cdots l_n)]/[l_3\cdots l_n] + 1\} \odot l_2 \\ i_1 = \{[\mathbb{i} - i_n - (i_{n-1}-1)l_n - \cdots - (i_2-1)(l_3\cdots l_n]/[l_2\cdots l_n]\} + 1 \end{cases} \tag{15}$$

$$\begin{cases} j_n = \mathbb{j} \odot m_n \\ j_{n-1} = \{[(\mathbb{j} - j_n)/m_n] + 1\} \odot m_{n-1} \\ j_{n-2} = \{[(\mathbb{j} - j_n - (j_{n-1}-1)m_n)]/[m_{n-1}m_n] + 1\} \odot m_{n-2} \\ \vdots \\ j_2 = \{[\mathbb{j} - j_n - (j_{n-1}-1)m_n - \cdots - (j_3-1)(m_4\cdots m_n)]/[m_3\cdots m_n] + 1\} \odot m_2 \\ j_1 = \{[\mathbb{j} - j_n - (j_{n-1}-1)m_n - \cdots - (j_2-1)(m_3\cdots m_n]/[m_2\cdots m_n]\} + 1 \end{cases} \tag{16}$$

This Theorem can be proved directly by Theorem 2. $\blacksquare$

4 Some Applications

4.1 Equivalent Definition for Matrix Tensor Products

Based on Theorem 1 and 2, an equivalent definition of matrix tensor product can be obtained as the following corollary.

Corollary 1 (Tensor Product Equivalent Definition) *For any $k \in \mathbb{N}_n$, $A_k \in \mathcal{C}^{l_k \times m_k}$, the entry of A_k in slot (i_k, j_k) is given by $(A_k)_{i_k j_k} = (a^k_{i_k j_k})$, $(i_k, j_k) \in \mathcal{L}_k \times \mathcal{M}_k$. The tensor product $A_1 \otimes \cdots \otimes A_n$ can be defined as a matrix in $\mathcal{C}^{l_1 \cdots l_n \times m_1 \cdots m_n}$ whose $(\mathbb{i}, \mathbb{j})$ entry $(A_1 \otimes \cdots \otimes A_n)_{\mathbb{i}\mathbb{j}}$ is defined by*

$$(A_1 \otimes \cdots \otimes A_n)_{\mathbb{i}\mathbb{j}} = a^1_{i_1 j_1} \otimes \cdots \otimes a^n_{i_n j_n} \tag{17}$$

where the factor indices $(i_1, \cdots, i_n) = \sigma^{-\mathrm{r}}(\mathbb{i})$ and $(j_1, \cdots, j_n) = \sigma^{-\mathrm{c}}(\mathbb{j})$ can be determined by (15) and (16).

Note that this equivalent definition actually gives an algorithm for calculating Kronecker products. And this entry-oriented definition is more convenient for studying entries properties in Kronecker products.

4.1.1 Some Properties of Matrix Tensor products

All the properties can be proved by this equivalent definition of matrix tensor product. We only list a few of them.

Corollary 2

$$(A_1 B_1) \otimes (A_2 B_2) = (A_1 \otimes A_2)(B_1 \otimes B_2) \tag{18}$$

Proof

$$\begin{aligned}
&[(A_1 \otimes A_2)(B_1 \otimes B_2)]_{\mathbb{i}\mathbb{j}} \\
&= \sum_{\mathbb{k}} (A_1 \otimes A_2)_{\mathbb{i}\mathbb{k}} (B_1 \otimes B_2)_{\mathbb{k}\mathbb{j}} = \sum_{k_1} \sum_{k_2} a^1_{i_1 k_1} a^2_{i_2 k_2} b^1_{k_1 j_1} b^2_{k_2 j_2} = \sum_{k_1} a^1_{i_1 k_1} b^1_{k_1 j_1} \sum_{k_2} a^2_{i_2 k_2} b^2_{k_2 j_2} \\
&= (A_1 B_1)_{i_1 j_1} (A_2 B_2)_{i_2 j_2} = [(A_1 B_1) \otimes (A_2 B_2)]_{\mathbb{i}\mathbb{j}}
\end{aligned}$$

□

Corollary 3

$$(A_1 \otimes A_2)^\dagger = A_1^\dagger \otimes A_2^\dagger \tag{19}$$

Proof

$$\left([A_1 \otimes A_2]^\dagger\right)_{\mathbb{i}\mathbb{j}} = \overline{(A_1 \otimes A_2)_{\mathbb{j}\mathbb{i}}} = \overline{a^1_{j_1 i_1} a^2_{j_2 i_2}} = \overline{a^1_{j_1 i_1}}\ \overline{a^2_{j_2 i_2}} = \left(A_1^\dagger\right)_{i_1 j_1} \left(A_2^\dagger\right)_{i_2 j_2} = \left(A_1^\dagger \otimes A_2^\dagger\right)_{\mathbb{i}\mathbb{j}}$$

□

Corollary 4

$$A_1 \otimes A_2 \otimes A_3 \otimes A_4 = (A_1 \otimes A_2) \otimes (A_3 \otimes A_4) \tag{20}$$

Proof Let $A = A_1 \otimes A_2 \otimes A_3 \otimes A_4$, note that $A_{ij} = a^1_{i_1 j_1} a^2_{i_2 j_2} a^3_{i_3 j_3} a^4_{i_4 j_4} = (a^1_{i_1 j_1} a^2_{i_2 j_2})(a^3_{i_3 j_3} a^4_{i_4 j_4})$,
where $i = \sigma^r(i_1, i_2, i_3, i_4) = (i_1 - 1) l_2 l_3 l_4 + (i_2 - 1) l_3 l_4 + (i_3 - 1) l_4 + i_4$
$= [(i_1 - 1) l_2 + i_2 - 1] l_3 l_4 + [(i_3 - 1) l_4 + i_4] = (\sigma^r(i_1, i_2) - 1) l_3 l_4 + \sigma^r(i_3, i_4) = \sigma^r[\sigma^r(i_1, i_2), \sigma^r(i_3, i_4)]$,
$j = \sigma^c(j_1, j_2, j_3, j_4) = \sigma^c[\sigma^c(j_1, j_2), \sigma^c(j_3, j_4)]$.■

4.2 Conversion of Number with Different Bases

Observe the results (5), (6) and (15), (16), there is a similarity between the indexes and number. Note that the dimension of matrix l_i, m_i has the similiar property of the base of number and the indexes of matrix entries in each row or column is similiar to the digitals of a number. The only differences is that indexes begin with 1, but digitals begin with 0. That difference explains the -1 term in the results (5), (6) and (15), (16).

4.2.1 Example: A number of base 10

A number of base 10 - $123456 = 1 \times 10^5 + 2 \times 10^4 + 3 \times 10^3 + 4 \times 10^2 + 5 \times 10 + 6 = \sigma^r(2, 3, 4, 5, 6, 6)$, where $l_1 = l_2 = \cdots = l_6 = 10$.

4.2.2 Example: Numbers of base 2^n

Numbers of base 2^n are popular in computer science. The result of Corollary (20) can be directly used to convert numbers of base 2^n.

For example, given a number of base 2: $a_8a_7a_6a_5a_4a_3a_2a_1$, $a_i \in \{0, 1\}$, $l_i = 2$. Note that

$$l_1l_2 = l_3l_4 = l_5l_6 = l_7l_8 = 2^2 = 4$$

so

$$(a_8a_7a_6a_5a_4a_3a_2a_1)_2 = (a_8a_7)_4(a_6a_5)_4(a_4a_3)_4(a_2a_1)_4 = (a_{87})_4(a_{65})_4(a_{43})_4(a_{21})_4$$

where $a_{87} = a_8 \times 2 + a_7$, $a_{65} = a_6 \times 2 + a_5$, $a_{43} = a_4 \times 2 + a_3$, $a_{21} = a_2 \times 2 + a_1$, and $a_{87}, a_{65,} a_{43}, a_{21} \in \{0, 1, 2, 3\}$. Similarly, because $l_1l_2l_3 = l_4l_5l_6 = l_7l_8l_9 = 2^3 = 8$, we have

$$(a_8a_7a_6a_5a_4a_3a_2a_1)_2 = (0a_8a_7)_8(a_6a_5a_4)_8(a_3a_2a_1)_8$$

and $l_1l_2l_3l_4 = l_5l_6l_7l_8 = 2^4 = 16$, so

$$(a_8a_7a_6a_5a_4a_3a_2a_1)_2 = (a_8a_7a_6a_5)_{16}(a_4a_3a_2a_1)_{16}.$$

The index mapping theorems gives a general way for convertations between numbers of different bases - we can firstly convert number of all bases using the index mapping theorem into a intermediary base like 10, then use the inverse index mapping theorem to convert the number into the target base. But for numbers of base like 2^n, Corollary (20) gives a much easier way.

For convenience, we usually use a single base - $l_1 = l_2 = \cdots$. However, it's obvious that we can easily break this convention. Use the index mapping theorems, we can generate numbers of a single base to numbers of variable bases - $l_1 \neq l_2 \neq \cdots$.

4.3 Spatial Matrices

Let A_k be a $l_k \times m_k$ spatial matrix, $k = 1, \cdots, n$. Suppose the index sets $\mathcal{S}_k$ for non-zero elements of A_k are defined by

$$\mathcal{S}_k = \Big\{(i_k, j_k)| \quad a^k_{i_kj_k} \neq 0, \quad \forall\ (i_k, j_k)\ \in \mathcal{L}_k \times \mathcal{M}_k\Big\} \tag{21}$$

Remark 3 *A_k being a $l_k \times m_k$ spatial matrix means not only that $\mathcal{S}_k$ is a real subset of $\mathcal{L}_k \times \mathcal{M}_k$ but also that*

$$Card(\mathcal{S}_k) \prec\prec l_k \times m_k$$

where Card denotes the number of index pairs in $\mathcal{S}_k$ and $\prec\prec$ implies that the left-hand term is sufficiently strictly smaller than the right one.

According to the **The Multi-index Mapping Theorem**, the non-zero elements of $A = A_1 \otimes A_2 \otimes \cdots \otimes A_n$ are determined by the non-zero elements of A_k, $k = 1, \cdots, n$, and we can find all the non-zero elements of A using the non-zero elements of A_ks by (5) and (6). Non-zero elements of the Kronecker product are the elements whose indexes have nothing to do with indexes of zero elements in the factor matrices. The non-zero elements of A can be listed by substituting ordered indices pair $\{(i_k, j_k) \in \mathcal{S}_k\}$ into (5) and (6), and can be calculated by multiplying corresponding elements in A_k's.

Corollary 5 *For any $k \in \mathbb{N}_n$, consider a $l_k \times m_k$ spatial matrix A_k. Assume that the index set $\mathcal{S}_k$ of the non-zero elements of A_k is given by (21). Then the non-zero elements of tensor product $A_1 \otimes \cdots \otimes A_n$ can be calculated by $a^1_{i_1 j_1} \otimes \cdots \otimes a^n_{i_n j_n}$ in which for any $k \in \mathbb{N}_n$, the ordered index pairs $(i_k, j_k) \in \mathcal{S}_k$. The index set $\mathcal{S}$ of non-zero term $a^1_{i_1 j_1} \otimes \cdots \otimes a^n_{i_n}$ is given*

$$\mathcal{S} = \{(\mathbb{i}_k, \mathbb{j}_k)|\ \ \mathbb{i} = \sigma^{\mathbb{r}}(i_1, \cdots, i_n),\ \ \mathbb{j} = \sigma^{\mathbb{c}}(j_1, \cdots, j_n) \text{ with } (i_k, j_k)\ \in \mathcal{S}_k\}$$

■

4.3.1 An Illustrative Example

Let $A_1 \in \mathcal{C}^{4\times 5}, A_2 \in \mathcal{C}^{5\times 6}, A_3 \in \mathcal{C}^{9\times 8}$ be three spatial matrices. Suppose the non-zero elements of A_1, A_2, A_3 are given as follows:

$$A_1(1,1) = 1,\ \ A_1(1,2) = 2, A_1(3,4) = 3; \quad A_2(2,2) = 4,\ \ A_2(3,5) = 5; \quad A_3(7,7) = 6.$$

It is clear that

$$\begin{aligned}
\mathcal{S}_1 &= \left\{\begin{pmatrix} i_1 \\ j_1 \end{pmatrix}\right\} = \left\{\begin{pmatrix} 1 \\ 1 \end{pmatrix}\begin{pmatrix} 1 \\ 2 \end{pmatrix}\begin{pmatrix} 3 \\ 4 \end{pmatrix}\right\} \\
\mathcal{S}_2 &= \left\{\begin{pmatrix} i_2 \\ j_2 \end{pmatrix}\right\} = \left\{\begin{pmatrix} 2 \\ 2 \end{pmatrix}\begin{pmatrix} 3 \\ 5 \end{pmatrix}\right\}, \quad \mathcal{S}_3 = \left\{\begin{pmatrix} i_3 \\ j_3 \end{pmatrix}\right\} = \left\{\begin{pmatrix} 7 \\ 7 \end{pmatrix}\right\}.
\end{aligned}$$

One can define a multi-index set which is Cartesian product $\mathcal{S}_1 \times \mathcal{S}_2 \times \mathcal{S}_3$. The set is given by

$$\mathcal{S}_1 \times \mathcal{S}_2 \times \mathcal{S}_3 = \left\{\begin{pmatrix} i_1 & i_2 & i_3 \\ j_1 & j_2 & j_3 \end{pmatrix}\right\} =$$

$$\left\{\begin{pmatrix} 1 & 2 & 7 \\ 1 & 2 & 7 \end{pmatrix} \begin{pmatrix} 1 & 3 & 7 \\ 1 & 5 & 7 \end{pmatrix} \begin{pmatrix} 1 & 2 & 7 \\ 2 & 2 & 7 \end{pmatrix} \begin{pmatrix} 1 & 3 & 7 \\ 2 & 5 & 7 \end{pmatrix} \begin{pmatrix} 3 & 2 & 7 \\ 4 & 2 & 7 \end{pmatrix} \begin{pmatrix} 3 & 3 & 7 \\ 4 & 5 & 7 \end{pmatrix}\right\}$$

Though $A = A_1 \otimes A_2 \otimes A_3$ is a $(4 \cdot 5 \cdot 9) \times (5 \cdot 6 \cdot 8) = 180 \times 240$ matrix, the number of non-zero elements of A is equal to $Card(\mathcal{S}_1 \times \mathcal{S}_2 \times \mathcal{S}_3) = 3 \times 2 \times 1 = 6$. These non-zero elements of A are listed as follows.

$$\begin{aligned}
A(\mathbb{i}_1, \mathbb{j}_1) &= A_1(1,1) \times A_2(2,2) \times A_3(7,7) = 1 \times 4 \times 6 = 24, \\
\mathbb{i}_1 &= \sigma^{\mathbb{r}}(1,2,7) = (1-1) \times 5 \times 9 + (2-1) \times 9 + 7 = 16, \\
\mathbb{j}_1 &= \sigma^{\mathbb{c}}(1,2,7) = (1-1) \times 6 \times 8 + (2-1) \times 8 + 7 = 15
\end{aligned}$$

$$\begin{aligned}
A(\mathbb{i}_2, \mathbb{j}_2) &= A_1(1,1) \times A_2(3,5) \times A_3(7,7) = 1 \times 5 \times 6 = 30, \\
\mathbb{i}_2 &= \sigma^{\mathbb{r}}(1,3,7) = (1-1) \times 5 \times 9 + (3-1) \times 9 + 7 = 25, \\
\mathbb{j}_2 &= \sigma^{\mathbb{c}}(1,5,7) = (1-1) \times 6 \times 8 + (5-1) \times 8 + 7 = 39
\end{aligned}$$

$$\begin{aligned}
A(\mathbb{i}_3, \mathbb{j}_3) &= A_1(1,2) \times A_2(2,2) \times A_3(7,7) = 2 \times 4 \times 6 = 48, \\
\mathbb{i}_3 &= \sigma^{\mathbb{r}}(1,2,7) = 16, \\
\mathbb{j}_3 &= \sigma^{\mathbb{c}}(2,2,7) = ((2-1) \times 6 \times 8 + (2-1) \times 8 + 7 = 63
\end{aligned}$$

$$\begin{aligned}
A(\mathbb{i}_4, \mathbb{j}_4) &= A_1(1,2) \times A_2(3,5) \times A_3(7,7) = 2 \times 5 \times 6 = 60, \\
\mathbb{i}_4 &= \sigma^{\mathbb{r}}(1,3,7) = 25, \\
\mathbb{j}_4 &= \sigma^{\mathbb{c}}(2,5,7) = (2-1) \times 6 \times 8 + (5-1) \times 8 + 7 = 87
\end{aligned}$$

$$\begin{aligned}
A(\mathbb{i}_5, \mathbb{j}_5) &= A_1(3,4) \times A_2(2,2) \times A_3(7,7) = 3 \times 4 \times 6 = 72, \\
\mathbb{i}_5 &= \sigma^{\mathbb{r}}(3,2,7) = (3-1) \times 5 \times 9 + (2-1) \times 9 + 7 = 106, \\
\mathbb{j}_5 &= \sigma^{\mathbb{c}}(4,2,7) = (4-1) \times 6 \times 8 + (2-1) \times 8 + 7 = 159
\end{aligned}$$

$$\begin{aligned}
A(\mathbb{i}_6, \mathbb{j}_6) &= A_1(3,4) \times A_2(3,5) \times A_3(7,7) = 3 \times 5 \times 6 = 90, \\
\mathbb{i}_6 &= \sigma^{\mathbb{r}}(3,3,7) = (3-1) \times 5 \times 9 + (3-1) \times 9 + 7 = 115, \\
\mathbb{j}_6 &= \sigma^{\mathbb{c}}(4,5,7) = (4-1) \times 6 \times 8 + (5-1) \times 8 + 7 = 183
\end{aligned}$$

4.4 Block Matrix

Let B_n be a $(l_1 l_2 \cdots l_n) \times (m_1 m_2 \cdots m_n)$ block matrix with $l_n \times m_n$ blocks, and suppose its (i_k, j_k)-th block is a $(l_1 l_2 \cdots l_{k-1}) \times (m_1 m_2 \cdots m_{k-1})$ block matrix with $l_{k-1} \times m_{k-1}$ blocks, $k = 1, \cdots n$. Denote $[(i_1, \cdots, i_n), (j_1, \cdots, j_n)]$ as the index for the (i_k, j_k)-th element in the (i_{k-1}, j_{k-1})-th block of B_k, $k = 1, \cdots, n$. Figure 3 gives an illustration of the block structure.

Let

$$[\mathit{Col\text{-}Vec}(A)]^T = \begin{bmatrix} a_{11} & \cdots & a_{\mathbb{l}1}, & a_{12} & \cdots & a_{\mathbb{l}2}, & \cdots, & a_{1\mathbb{m}} & \cdots & a_{\mathbb{l}\mathbb{m}} \end{bmatrix}$$

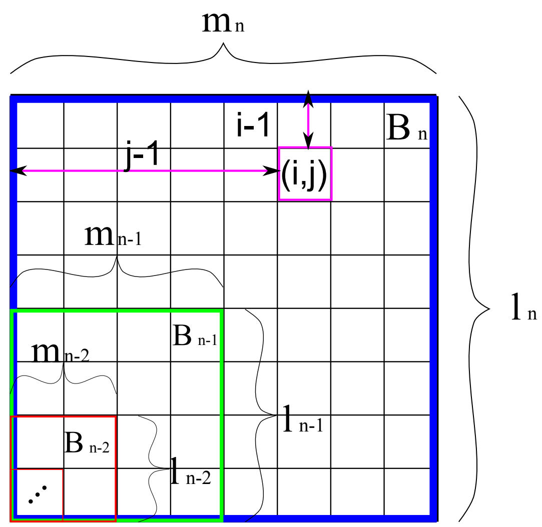

Fig.3: Block Matrix Structure

where

$$\mathbb{l} = \prod_{k=1}^{n} l_k, \quad \mathbb{m} = \prod_{k=1}^{n} m_k$$

For an element b in B_n, let its index in B_n be $ind(n) = (i, j)$, and let its index in $Col - Vec(B_n)$ be $t(n)$. Suppose b's indexes in the blocks are $[(i_1, \cdots, i_n), (j_1, \cdots, j_n)]$.

Now we can find the relationship between $ind(n)$ and $t(n)$ easily using The Index Mapping Theorem.

It is easy to see that B_n can be decomposed as $B_n = B(1) \otimes B(2) \otimes \cdots \otimes B(n)$, where $B(k)$ is a $l_k \times m_k$ matrix, $k = 1, \cdots, n$.

According to The Index Mapping Theorem,

$i = (i_1 - 1)l_2 \cdots l_n + (i_2 - 1)l_3 \cdots l_n + \cdots + (i_{n-1} - 1)l_n + i_n$, and $j = (j_1 - 1)m_2 \cdots m_n + (j_2 - 1)m_3 \cdots m_n + \cdots + (j_{n-1} - 1)m_n + j_n$.

Let $m = m_1 \times m_2 \times \cdots \times m_n$, since $t(n) - 1$ is the number of elements before b in $Vec(B_n)$,

we have $t(n) = (j - 1)m + i$.$\blacksquare$

Use the similar idea of index mapping in Kronecker product, we can give an entry-oriented definition of vectorization.

Corollary 6 (Vectorization Equivalent Definition) *For any $A \in C^{l \times m}$, the entry of A in slot (i, j) is given by $A_{i,j} = (a_{i,j})$, $(i, j) \in L \times M$. The vectorization of A can be defined as a vector in $C^{lm \times 1}$ whose (k)th entry is a_{ij}, where $i = k \odot l$ and $j = (k - i)/l + 1$ (or $k = (j - 1) \times l + i$).*

4.5 Octree

The applications of the index mapping theorems are not limited to matrices and are not even limited to 2D. Example of 4.4 can also be extended to higher dimensions. Results in 3D, where the block matrix structure will be replaced by an octree (Figure 4), can be used to derive the algorithms for octree traversal and voxel searching.

In 3D voxel rendering, the simplest way to represent voxel data is to use a 3D array - VoxelData[x][y][z], where the index (x, y, z) is the position of the voxel in 3D space and the value of the array member is the color for that voxel. However, for spatial voxel data, i.e., large part of the voxel data is empty space, 3D array is not efficient because it takes more computer memory than needed. A popular structure for

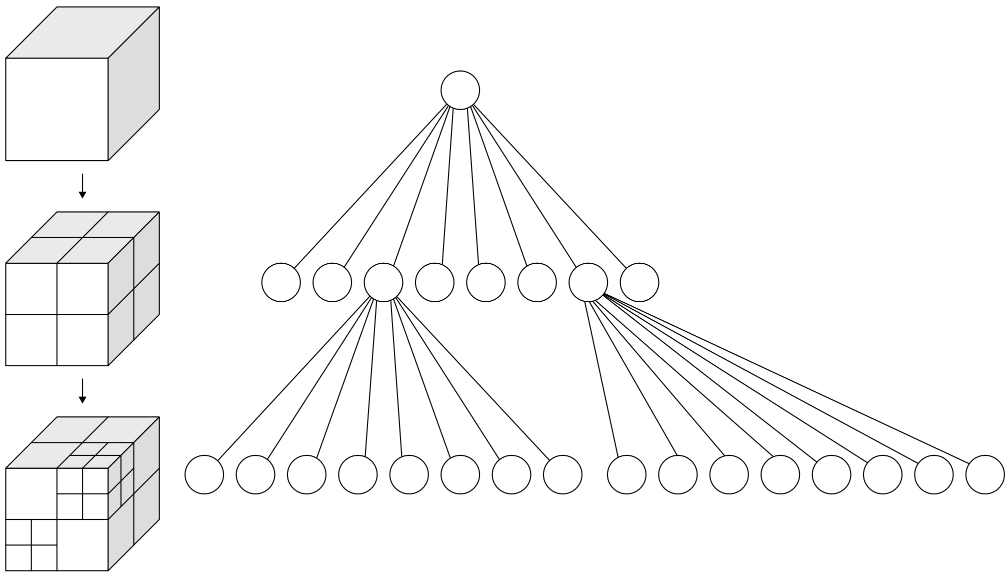

Fig.4: Octree (Courtesy of WhiteTimberwolf, licensed with Cc-by-sa-3.0-migrated,GFDL)

representing spatial voxel data is the octree. Octrees can be used to partition a 3D space. In an octree, the 3D space is recursively divided into 8 cubes. It can save the computer memory because in an octree, we can use less levels or fewer branches for representing empty spaces.

To encode a 3D array into an octree, we should give an one-to-one mapping of the 3D array indexes and the octree indexes. Note that for an octree of n levels, we have $2^n \times 2^n \times 2^n$ elements. For each dimension, X or Y or Z, we have 2^n leaves. The octree structure can be viewed as the Kronecker product of n vectors or levels $A_1, \cdots A_n$, where A_i is a 1×2 vector. So in each dimension, the index in the 3D array can be mapped into the tree structure using The Inverse Index Mapping Theorem. Further more, since A_i is a 1×2 vector, if we use the number 0 to indicate that a voxel is in the left branch of the level, and number 1 to indicate that a voxel is in the right branch of the level, we immediately got a mapping from number of base 10 (the index in the 3D array, or equivalently, the position of the voxel in 3D space), to the number of base 2 (the indexes of the levels in the octree). This will be extremely useful because a fast algorithm for the mapping can be obtained using bitwise operation.

For example, in Figure 5, the shaded cube can be encoded as in the following table:

Dimension	3D Position	Level 0	Level 1	Level 2	Level 3
X	6	0	1	1	0
Y	7	0	1	1	1
Z	7	0	1	1	1

Note that the mapping is can be calculated using bitwise operators:

$6 = 0 << 3|1 << 2|1 << 1|0$,

$0_{6level3} = 6 >> 0\&2$,

$1_{6level2} = 6 >> 1\&2$,

$1_{6level1} = 6 >> 2\&2$,

$0_{6level0} = 6 >> 3\&2$;

$7 = 0 << 3|1 << 2|1 << 1|1$,

$1_{7level3} = 7 >> 0\&2$,

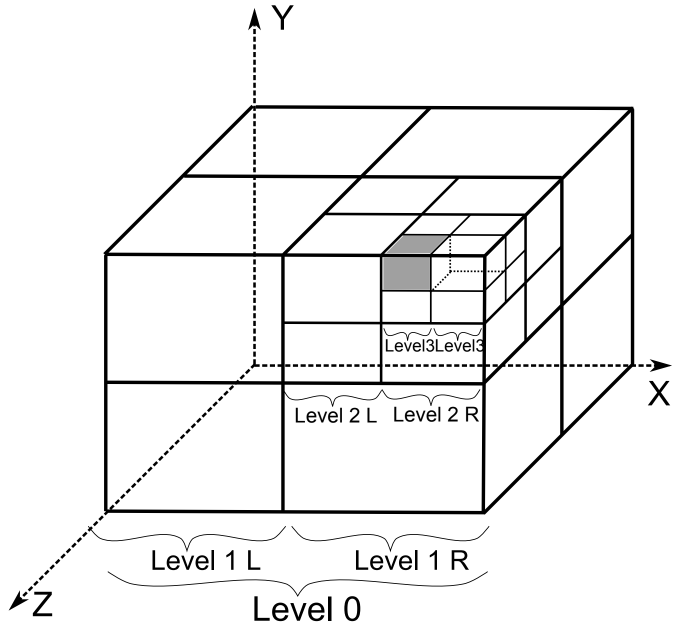

Fig.5: A 3-level Octree

$1_{7level2} = 7 >> 1\&2$,
$1_{7level1} = 7 >> 2\&2$,
$1_{7level0} = 7 >> 3\&2$;

where $A << B$ denotes bitwise left shift A by B, $A >> B$ denotes bitwise right shift A by B, $|$ denotes bitwise OR operator, & denotes bitwise AND operator.

Another popular structure is quadtree. We can view a quadtree as an octree projected to the 2D XY plane - the 2D plane is recursively divided into 4 squares. A quadtree can also be seen as the special case of example 4.4, when the blocks are all 2×2. Similar to octree's usage in voxel representation and spatial 3D array indexing, quadtree can be used for image representation and spatial 2D array indexing. As we have seen, octree, quadtree can be used to partition a space into smaller parts. The index mapping theorem can be used for encoding the positions in tree-based space partiation method. And the result is not limited to 2D and 3D. We can easily extend it to 4D, 5D, $\cdots$, because we treat each dimension independently.

4.6 2D Image and 3D voxel Scaling

In computer graphics, an image can be stored in a 2D array. Equivalently, we can use a $l_1 \times m_1$ matrix A_1 to represent an image, where l_1 is the height of the image, m_1 is the width of the image, and the entries of A_1 are the pixel values of the image.

In image scaling, an enlarged image can be viewed as a Kronecker product of the original image matrix A_1 and a $l_2 \times m_2$ scaling matrix A_2, where l_2 is the scaling factor of height and m_2 is the scaling factor of width. The simplest form of A_2 is a matrix whose entries are all 1. After the scaling transformation, every pixel in the original image will be enlarged into $l_2 \times m_2$ pixels, this is the so-called "nearest-neighbor interpolation". The index mapping theorem can be used to design a scaling algorithm. To get the new image matrix $A(2) = A_1 \otimes A_2$, all we need is to fill the entries of the new $lp \times mq$ image matrix $A(2)$ using the index mapping theorem - for each entry (i,j) in $A(2)$, use the theorem to find the entry's index in A_1, then set the entry's value as the same value in the corresponding entry in A_1. More generally, given a sequence of scale matrix $A_2, \cdots, A_n$, we can get the scaled image $A(n) = A_1 \otimes A_2 \otimes \cdots \otimes A_n$.

On the other hand, to narrow an image, just factor the image matrix A into $A_1 \otimes \cdots \otimes A_n$, view $A_2, \cdots, A_n$ as the scaling matrices, use the inverse index mapping theorem, fill the entries of A_1 by picking a value from entries of same root level indexes in A.

The advantage of the index mapping theorems here is that when enlarging an image, we don't need to calculate the Kronecker product, we calculate the indexes only; and when narrowing an image, we have a clear guide for sampling the original image. What's more, because the index mapping theorems are matrix entry-oriented, the algorithm is pixel independent - given any position pair (x,y), we can find the pixel value there. In a scaling sequence, this gives us more flexibility - we don't need to wait for calculating the whole image with the first enlargement applied if we only want to find a specified pixel value after two consequent enlargement.

The result can be extended into higher dimensions, for example, in 3D, what we do is voxel data scaling.

We see that the Kronecker product can be used for LOD (level of detail) control of 2D images or 3D spaces, and the index mapping theorems are suitable for designing algorithm related to Kronecker product.

4.7 Fractal and Self-similarity Geometry

The block matrix, octree and quadtree all have a self-similarity geometry. The self-similarity property of Kronecker product make it suitable for representation of self-similarity geometries.

4.7.1 Example: The Cantor Set

The Cantor set is defined as repeatedly divide a line segments into 3 parts and remove the middle one.

The mathematical expression of the set is $C = [0,1] \setminus \bigcup_{m=1}^{\infty} \bigcup_{k=0}^{3^{m-1}-1} \left(\frac{3k+1}{3^m}, \frac{3k+2}{3^m}\right)$. A five iterations of the set's generating process is showed in Fig.4.7.1.

Fig.6: The Cantor set

We only study a finite level, $m = n$, of the Cantor set. A usual way to calculate the length removed from $[0,1]$ is

$$\sum_{k=0}^{n} \frac{2^k}{3^{k+1}} = \frac{1}{3} + \frac{2}{9} + \frac{4}{27} + \frac{8}{81} + \cdots + \frac{2^n}{3^{n+1}} = \frac{1}{3}\left(\frac{1-(\frac{2}{3})^n}{1-\frac{2}{3}}\right) = 1 - (\frac{2}{3})^n.$$

And so the length left is $(\frac{2}{3})^n$. Now we will show how to use Kronecker Product and the index mapping theorem to give the same result.

Let $A = \begin{pmatrix} \frac{1}{3} \\ 0 \\ \frac{1}{3} \end{pmatrix}$, then the Cantor set can be expressed by $A \otimes A \otimes A \otimes \cdots$, and let $A^{(n)} = \underbrace{A \otimes \cdots \otimes A}_{n}$, use the index mapping theorem, or apply the result of spatial matrix, we have

$$A(i_1, \cdots, i_{n-1}, i_n) = \begin{cases} 0 & other \\ \frac{1}{3^n} & i_j \neq 2, j = 1, \cdots, n \end{cases}$$

The length left is $(1_{3^n})^T A^{(n)} = Card\{S(i_1, \cdots, i_{n-1}, i_n), i_j \in \{1,3\}\} \times \frac{1}{3^n}$, where 1_{3^n} is a $3^n \times 1$ vector of 1. It is easy to see that $Card\{S(i_1, \cdots, i_{n-1}, i_n), i_j \in \{1,3\}\} = 2 \times \cdots \times 2 = 2^n$, so the length left is $(\frac{2}{3})^n$.

5 Appendix

5.1 An implementation of the theorems in R language

The implementation of Equation 5 is simple. For Equation 15, a concise form is
$i_n = i \odot l_n$,
$i_c = ((i - i_n - \Sigma_{v=c+1}^{n-1}[(i_v - 1)\Pi_{w=v+1}^{n} l_w])/\Pi_{v=c+1}^{n} l_v) \odot l_c + 1$, for $c = 2, \cdots n-1$,
$i_1 = ((i - i_n - \Sigma_{v=c+1}^{n-1}[(i_v - 1)\Pi_{w=v+1}^{n} l_w])/\Pi_{v=2}^{n} l_v) + 1$.
We can set $A(c) = \Sigma_{v=c+1}^{n-1}[(i_v - 1)\Pi_{w=v+1}^{n} l_w]$, $B(c) = \Pi_{v=c+1}^{n} l_v$.
The R code for the two equations is as follows.

```
#This software is released under the MIT License
#<http://www.opensource.org/licenses/mit-license.php>
#*Copyright (c) <2011> <Bu Zhou>

#simple function to calculate product of the elements in a vector l
mul <- function(l)
  {
    len <- length(l);
    returnVar <- 1;
    for(i in 1:len)
      {
        returnVar <- returnVar*l[i];
      }
    return (returnVar);
  }#end of mul

#the modified module function
mmod <- function(p,q)
  {
    returnVar <- p%%q;
    if(returnVar==0)returnVar=p;
    return (returnVar);
  }#end of mmod

#The Index Mapping Function
```

```
#i: the row/column indexes of the factor matrices
#l: the length vector of the factor matrices
#return: the row/column index of kronecker product matrix
kronecker_index <- function(i,l)
  {
    returnVar <- 0;
    len <- length(i);
    for(j in 1:(len-1))
      {
        returnVar <- returnVar + (i[j]-1)*mul(l[(j+1):len]);
      }
    returnVar <- returnVar+i[len];
    return (returnVar);
  }#end of kronecker_index

#The Inverse Index Mapping Function
#c: the row/column index of kronecker product matrix
#l: the length vector of the factor matrices
#return: the row/column indexes of the factor matrices
kronecker_inv_index <- function(c,l)
  {
    n <- length(l);
    i <- rep(0,n);
    i[n] <- mmod(c,l[n]);

    if(n-1>0)
    for(j in (n-1):1)
      {
        A <- 0;
        if(j+1<n)
        for(v in (j+1):(n-1))
          {
            A <- A+(i[v]-1)*mul(l[(v+1):n]);
          }

        B <- mul(l[(j+1):n]);

        if(j>1)
        {
          i[j] <- mmod((c-i[n]-A)/B,l[j])+1;
        }
        else
        {
          i[j] <- ((c-i[n]-A)/B)+1;
```

```
        }

      }#end of if&for
    return (i);
  }#end of kronecker_inv_index
```

5.2 R code for verifying the result in Example 4.3.1

```
A1<-matrix(0,4,5);
A2<-matrix(0,5,6);
A3<-matrix(0,9,8);
A1[1,1]<-1;
A1[1,2]<-2;
A1[3,4]<-3;
A2[2,2]<-4;
A2[3,5]<-5;
A3[7,7]<-6;
A<-kronecker(A1,A2);
A<-kronecker(A,A3);
> A[16,15]
[1] 24
> A[16,63]
[1] 48
> A[106,159]
[1] 72
> A[25,39]
[1] 30
> A[25,87]
[1] 60
> A[115,183]
[1] 90
library(Matrix);
> nnzero(A);#print the number of non-zero elements in A
[1] 6
```

References

[1] I. Beju, E.Soós and P.P. Teodorescu. " Euclidean tensor calculus with applications," *Abacus Press, England*, 1983.

[2] Jun-e Feng, James Lam and Yimin Wei, "Spectral properties of sums of certain Kronecker products,"*Linear Algebra and its Applications* 431(9) (2009) 1691-1701.

[3] Hall B. C., "Lie groups Lie algebras and represesntations: an elementary introduction," Springer-Verlag, New York (2003)

[4] R. van Handel, J. K. Stockton and H. Mabuchi, “Feedback Control of Quantum State Reduction,” *IEEE Transactions on Automatic Control* 50 (6) (2005) 768-780.

[5] Horn R. A., Johnson C. R.: Topics in matrix analysis. Cambridge University Press, (2007)

[6] H.V. Henderson, F. Pukelsheim, S.R. Searle, “On the history of the Kronecker product”, Linear and Multilinear Algebra 14 (2) (1983) 113-120

[7] R. A. Horn and C. R. Johnson, ”Topics in Matrix Analysis,” *Cambridge University Press*, 2007.

[8] Shafarevich I. R.: Basic notions of algebra. Springer-Verlag, Berlin Heidelberg (2005)

[9] Wojciech Tadej, “Defect of a Kronecker product of unitary matrices,” *Linear Algebra and its Applications,* In Press, Corrected Proof, Available online 21 October 2011.

[10] Yanagisawa M. and Kimura H., “Transfer function approach to quantum control—Part I: Dynamics of quantum feedback systems”, *IEEE Transactions on Automatic Control*, 48(12) (2003) 2107-2120.

[11] Bu Zhou,”An orthogonality-based estimation of moments for nonparametric linear mixed models (manuscript),” *Master Thesis*, 2011

[12] Shaosheng Zhou and Shizhou Fu, ”Matrix representations for adjoint and anti-adjoint operators in multi-spin 1/2 systems,” *Quantum Information Processing*, 10(3) (2011) 379-394.

[13] Shaosheng Zhou, Wenyan Ren and James Lam, “Stabilization for T-S Model Based Uncertain Stochastic Systems”, *Information Sciences*, 181(4) (2011) 779-791.