Global Journal of Computer Science and Technology: G Interdisciplinary
Volume 20 Issue 6 Version 1.0 Year 2020
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Multi-Dimensional Rajan Transform

By K. Thiagarajan, Manish Prateek & Ethirajan Govindarajan
University of Petroleum and Energy Studies

Abstract- In this paper, we describe the formulation of a novel transform called Multi-Dimensional Rajan Transform, which is an extension of Rajan Transform. Basically, Rajan Transform operates on a number sequence, whose length is a power of two. It transforms any sequence of arbitrary numbers into a sequence of interrelated numbers. As regards 2D Rajan Transform, there are two methods to implement it: (i) Row- Column method and (ii) Column-Row method. The 2D Rajan Transform obtained using the first method need not be the same as that obtained using second method. Similarly, one can implement 3-D Rajan Transform using the following approaches: (i) Row-Column-Depth approach, (ii) Row-Depth- Column approach, (iii) Column-Row-Depth approach, (iv) Column-Depth-Row approach, (v) Depth-Row-Column approach and (vi) Depth-Column-Row approach. This paper explains these approaches to implement two and three dimensional Rajan Transforms.

Keywords: *discrete transforms, rajan transform, permutation invariant systems, homomorphic transforms.*

GJCST-G Classification: *F.2.1*

Strictly as per the compliance and regulations of:

Multi-Dimensional Rajan Transform

K. Thiagarajan [α], Manish Prateek [σ] & Ethirajan Govindarajan [ρ]

Abstract- In this paper, we describe the formulation of a novel transform called Multi-Dimensional Rajan Transform, which is an extension of Rajan Transform. Basically, Rajan Transform operates on a number sequence, whose length is a power of two. It transforms any sequence of arbitrary numbers into a sequence of interrelated numbers. As regards 2D Rajan Transform, there are two methods to implement it: (i) Row-Column method and (ii) Column-Row method. The 2D Rajan Transform obtained using the first method need not be the same as that obtained using second method. Similarly, one can implement 3-D Rajan Transform using the following approaches: (i) Row-Column-Depth approach, (ii) Row-Depth-Column approach, (iii) Column-Row-Depth approach, (iv) Column-Depth-Row approach, (v) Depth-Row-Column approach and (vi) Depth-Column-Row approach. This paper explains these approaches to implement two and three dimensional Rajan Transforms.

Keywords: discrete transforms, rajan transform, permutation invariant systems, homomorphic transforms.

I. Introduction

Rajan transform essentially operates on a number sequence, whose length is a power of two. It transforms any sequence of arbitrary numbers into a sequence of interrelated numbers. The resulting sequence is called 'Rajan Spectrum'. More precisely,Rajan Transform is a homomorphic map that yields the same Rajan Spectrum for an input sequence and all of its permuted versions. The definition and various outcomes of One-Dimensional Rajan Transform (1D-RT) are extended to multi-dimensional case. One-dimensional Rajan Transform is briefly explained on a need to have basis.

Definition of Rajan Transform

Rajan Transform is essentially a fast algorithm developed on the lines of Decimation-In-Frequency Fast Fourier Transform algorithms, but it is functionally different from the DIF-FFT algorithm. Given a number sequence x(n) of length N, which is a power of 2, first it is divided into the first half and the second half each consisting of (N/2) points so that the following holds good.

$$g(i) = x(i) + x(i+N/2) \; ; \; 0 \le j \le N/2; \; 0 \le i \le N/2$$

$$h(j) = |x(i) - x(i-N/2)| \; ; \; 0 \le j \le N/2; \; 0 \le i \le N/2$$

Now each (N/2) point segment is further divided into two halves each consisting of N/4 points so that the following holds good.

$$g1(k) = g(i) + g(j+(N/4)); \; 0 \le k \le (N/4); \; 0 \le j \le (N/4)$$

$$g2(k) = |g(i) - g(i-(N/4)| \; ; \; 0 \le k \le (N/4); \; (N/4) \le j \le (N/2)$$

$$h1(k) = h(j) + h(j+(N/4)); \; 0 \le k \le (N/4); \; 0 \le j \le (N/4)$$

$$h2(k) = |h(j) - h(j-N/4)| \; ; \; 0 \le k \le (N/4); \; 0 \le j \le (N/4)$$

This process is continued until no more division is possible. The total number of stages thus turns out to be $\log_2 N$. Let us denote the sum and difference operators respectively as '+' and '~'. Then the signal flow graph for the Rajan Transform of a number sequence of length 16 would be of the form shown in Fig. 1. If x(n) is a number sequence of length $N=2^k$; $k>0$, then its Rajan Transform(RT) is denoted as X(k). RT is applicable to any number sequence and it induces an isomorphism in a class of sequence, that is, it maps a domain set consisting of the cyclic and dyadic permutations of a sequence on to a range set consisting of sequence of the form X(k)E(r) where X(k) denotes the permutation invariant RT and E(r) an encryption code corresponding to an element in the domain set. This map is a one-to-one and on to correspondence and an inverse map also exists. Hence, it is viewed as a transform. Fig. 1 shows a functional block diagram of a 16-point Rajan Transform algorithm.

It is to be noted that the map $x(n) \leftrightarrow X(k)E(r)$ is an isomorphic map, and the map $x(n) \rightarrow X(k)$ is a homomorphic map. The inverse function is called Inverse Rajan Transform and is applicable to cases where Rajan Transform is considered as an isomorphic function. Homomorphic functions do not have inverse functions. The operation of Inverse Rajan Transform is not explained here in this paper for lack of space and for its irrelevance in the formulation of multi-dimensional Rajan Transforms. Apart from multi-dimensional Rajan Transforms, which are conceptual extensions of one-dimensional Rajan Transform, one can as well develop notions like "Set-Theoretic Rajan Transform" which are higher order algebraic tools that work on sequences of sets, functions and relations. These formulations are not in the scope of this paper and thus they are not presented.

Author α: Professor, Department of Mathematics, K. Ramakrishnan College of Technology, Samayapuram, Trichy, Tamil Nadu, India.
e-mail: vidhyamannan@yahoo.com
Author σ: Professor and Dean, School of Computer Science, University of Petroleum and Energy Studies, Dehradun, Uttarakhand, India.
e-mail: mprateek@ddn.upes.ac.in
Author ρ: Adjunct Professor, Department of Cybernetics, School of Computer Science, UPES, Dehradun, Uttarakhand, India.
e-mail: dr.rajaneg@gmail.com

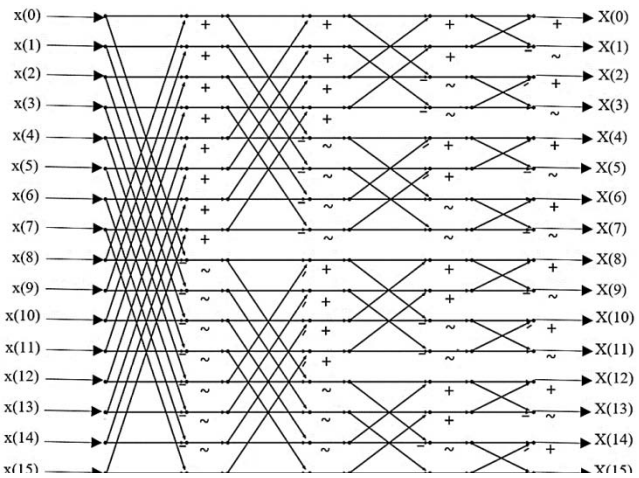


Fig. 1: Signal flow graph of Rajan Transform

II. Two Dimensional Rajan Transform and its Implementation

As outlined earlier, one can implement two-dimensional Rajan Transformin two different ways: (i) Row-Column method and (ii) Column-Row method. As stated previously, 2D Rajan Spectrum obtained using first method need not be the same as the spectrum obtained using second method. This could be verified with the help of an example. Consider a two dimensional array A = [x(m,n)].

$$A = [x(m,n)] = \begin{matrix} 0\,0\,0\,0 \\ 1\,1\,1\,0 \\ 0\,1\,0\,0 \\ 0\,1\,0\,0 \end{matrix}$$

2D-RT obtained using Row-Colum method

The RT spectra of the four rows of the array A are the four rows given in the array $[X_r(g,h)]$

$$[X_r(g,h)] = \begin{matrix} 0\,0\,0\,0 \\ 3\,1\,1\,0 \\ 1\,1\,1\,1 \\ 1\,1\,1\,1 \end{matrix}$$

Next, the RT spectra of the four columns in the array $[X_r(g,h)]$ are given in the columns of the array $[X_{r,c}(k_1,k_2)]$, which is the 2D-RT of the given array A.

$$[X_{r,c}(k_1,k_2)] = \begin{matrix} 5\,3\,3\,3 \\ 3\,1\,1\,1 \\ 3\,1\,1\,1 \\ 1\,1\,1\,1 \end{matrix}$$

2D-RT obtained using Column-Row method

The RT spectra of the four columns of the array A are the four columns given in the array $[X_c(g,h)]$

$$[X_c(g,h)] = \begin{matrix} 1\,3\,1\,0 \\ 1\,1\,1\,0 \\ 1\,1\,1\,0 \\ 1\,1\,1\,0 \end{matrix}$$

Next, the RT spectra of the four rows in the array $[X_c(g,h)]$ are given in the rows of the array $[X_{c,r}(k_1,k_2)]$, which is the 2D-RT of the given array A.

$$[X_{c,r}(k_1,k_2)] = \begin{matrix} 5\,1\,3\,3 \\ 3\,1\,1\,1 \\ 3\,1\,1\,1 \\ 3\,1\,1\,1 \end{matrix}$$

With reference to the example presented above, it is proved that $[X_{r,c}(k_1,k_2)] \neq [X_{c,r}(k_1,k_2)]$ in this case. However, one can verify that $[X_{r,c}(K_1,k_2)] = [X_{c,r}(k_1,k_2)]$ for symmetric arrays. In order to do this, let us consider Hilbert matrix H_4 and obtain 2D-RT spectra using both methods.

$$H_4 = [x(m,n)] = \begin{matrix} 1\,2\,3\,4 \\ 2\,3\,4\,5 \\ 3\,4\,5\,6 \\ 4\,5\,6\,7 \end{matrix}$$

The 2D-RT spectra of H_4 obtained using Row-Column method is $X_{r,c}(k_1,k_2)$ and it is shown below.

$$[X_{r,c}(k_1,k_2)] = \begin{matrix} 64\,08\,16\,00 \\ 08\,00\,00\,00 \\ 16\,00\,00\,00 \\ 00\,00\,00\,00 \end{matrix}$$

The 2D-RT spectra of H_4 obtained using Column-Row method is $X_{c,r}(k_1,k_2)$ and it is shown below.

$$[X_{c,r}(k_1,k_2)] = \begin{matrix} 64\,08\,16\,00 \\ 08\,00\,00\,00 \\ 16\,00\,00\,00 \\ 00\,00\,00\,00 \end{matrix}$$

In this case of symmetric matrix $[X_{r,c}(k_1,k_2)] = [X_{c,r}(K_1,k_2)]$.

2D-RT Translation invariance property

Consider the two-dimensional array A showing aT like pattern.

$$A = [x(m,n)] = \begin{matrix} 0\,0\,0\,0 \\ 1\,1\,1\,0 \\ 0\,1\,0\,0 \\ 0\,1\,0\,0 \end{matrix}$$

B is the translated version of A.

$$B = [x_t(m,n)] = \begin{matrix} 0\,1\,1\,1 \\ 0\,0\,1\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,0 \end{matrix}$$

One can verify that $[X_{c,r}(K_1,k_2)]$ for both arrays A and B remain the same. Similarly, one can verify that $[X_{r,c}(k_1,k_2)]$ remain the same for both arrays A and B. This amounts to saying that 2D-Rajan Transform is essentially a translation invariant function, which could be effectively used in pattern recognition.

III. Three-Dimensional Rajan Transform

The definition and various properties associated with two dimensional RT is extended to the three-dimensional case also. Especially, the basic methods for implementing2D Rajan Transform like Row-Column method and Column-Row method could further be generalized to implement 3D Rajan Transform. One can implement 3-D Rajan Transform using any one of the following six methods: (i) Row-Column-Depth method, (ii) Row-Depth-Column method, (iii) Column-Row-Depth method, (iv) Column-Depth-Row method, (v) Depth-Row-Column method and (vi) Depth-Column-Row method. Let us consider the three dimensional array of matrix size 4x4x4 as shown in Fig. 2.

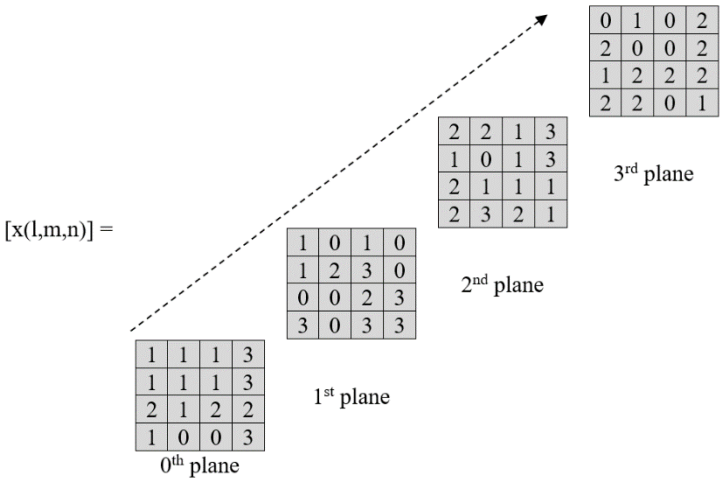


Fig. 2: Sample three dimensional array

The above 3-D matrix is represented in a linear array of 2-D planes for easy understanding.

$$x(l, m, n) = \left[\begin{array}{cccc|cccc|cccc|cccc} 1&1&1&3 & 1&0&1&0 & 2&2&1&3 & 0&1&0&2 \\ 1&1&1&3 & 1&2&3&0 & 1&0&1&3 & 2&0&0&2 \\ 2&1&2&2 & 0&0&2&3 & 2&1&1&1 & 1&2&2&2 \\ 1&0&0&3 & 3&0&3&3 & 2&3&2&1 & 2&2&0&1 \end{array}\right]$$

0^{th} plane 1^{st} plane 2^{nd} plane 3^{rd} plane

3D-RT using Row-Column-Depth (RCD) Approach

The 1D-RT of the four rows of the 0^{th} plane of the matrix X are {1,1,1,3}→{6,2,2,2}, {1,1,1,3}→{6,2,2,2}, {2,1,2,2}→{7,1,1,1}, {1,0,0,3}→{4,2,4,2}. The 1D-RT of the four rows of the 1^{st} plane are {1,0,1,0}→{2,2,0,0}, {1,2,3,0}→{6,2,4,0}, {0,0,2,3}→{5,1,5,1}, {3,0,3,3}→{9,3,3,3}. The 1D-RT of the four rows of the 2^{nd} plane are {2,2,1,0}→{8,2,2,0}, {1,0,1,3}→{5,1,3,3}, {2,1,1,1}→{5,1,1,1}, {2,3,2,1}→{8,0,2,2}. The 1D-RT of the final plane of four rows are {0,1,0,2}→{3,3,1,1}, {2,0,0,2}→{4,0,4,0}, {1,2,2,2}→{7,1,1,1}, {2,2,0,1}→{5,1,3,1}.

Now the 3D array of the Rajan Transform computed row wise is given by

$$X_r(k_1,k_2,k_3) = \left[\begin{array}{cccc|cccc|cccc|cccc} 6&2&2&2 & 2&2&0&0 & 8&2&2&0 & 3&3&1&1 \\ 6&2&2&2 & 6&2&4&0 & 5&1&3&3 & 4&0&4&0 \\ 7&1&1&1 & 5&1&5&1 & 5&1&1&1 & 7&1&1&1 \\ 4&2&4&2 & 9&3&3&3 & 8&0&2&2 & 5&1&3&1 \end{array}\right]$$

0^{th} plane 1^{st} plane 2^{nd} plane 3^{rd} plane

The 1D-RT of the four columns of the 0^{th} plane of the matrix $X_r(k_1,k_2,k_3)$ are {6,6,7,4}→{23,3,3,1}, {2,2,1,2}→{7,1,1,1}, {2,2,1,4}→{9,3,3,1}, {2,2,1,2}→{7,1,1,1}. The 1D-RT of the four columns of the 1^{st} plane are {2,6,5,9}→{22,8,6,0}, {2,2,1,3}→{8,2,2,0}, {0,4,5,3}→{12,2,6,4}, {0,0,1,3}→{4,2,4,2}. The 1D-RT of the four columns of the 2^{nd} plane are {8,5,5,8}→{26,0,6,0}, {2,1,1,0}→{4,2,2,0}, {2,3,1,2}→{8,2,2,0}, {0,3,1,2}→{6,4,2,0}. The 1D-RTs of the final plane of four columns are {3,4,7,5}→{19,1,5,3}, {3,0,1,1}→{5,3,3,1}, {1,4,1,3}→{9,5,1,1}, {1,0,1,1}→{3,1,1,1}.Now the 3D array of the Rajan Transform Row-Column wise is given by

$$X_{r,c}(k_1,k_2,k_3) = \begin{bmatrix} 23 & 7 & 9 & 7 & 22 & 8 & 12 & 4 & 26 & 4 & 8 & 6 & 19 & 5 & 9 & 3 \\ 3 & 1 & 3 & 1 & 8 & 2 & 2 & 2 & 0 & 2 & 2 & 4 & 1 & 3 & 5 & 1 \\ 3 & 1 & 3 & 1 & 6 & 2 & 6 & 4 & 6 & 2 & 2 & 2 & 5 & 3 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 4 & 2 & 0 & 0 & 0 & 0 & 3 & 1 & 1 & 1 \end{bmatrix}$$
$$0^{th}\text{ plane} \qquad 1^{st}\text{ plane} \qquad 2^{nd}\text{ plane} \qquad 3^{rd}\text{ plane}$$

The 1D-RT of the depth wise of four corresponding elements in four planes of the matrix $X_{r,c}(k_1,k_2,k_3)$, (RT can compute starting with any element in 0^{th} plane, in this example the RTs computed starting with row wise for easy understanding) are {23,22, 26,19}→{90,8,6,0}, {7,8,4,5}→{24,2,6,0}, {9,12,8,9}→{38,4,4,2}, {7,4,6,3}→{20,6,2,0}, {3,8,0,1}→{22,6,10,4}, {1,2,2,3}→{8,2,2,0}, {3,2,2,5}→{12,2,4,2}, {1,2,4,1}→{8,2,4,2}, {3,6,6,5}→{20,2,4,2}, {1,2,2,3}→{8,2,2,0}, {3,6,2,1}→{12,2,6,4}, {1,4,2,1}→{8,2,4,2}, {1,0,0,3}→{4,2,4,2}, {1,0,0,1}→{2,0,2,0}, {1,4,0,1}→{6,4,4,2} and {1,2,0,1}→{4,2,2,0}. Now the 3D array of the Rajan Transform Row-Column-Depth wise is given by

$$X_{r,c,d}(k_1,k_2,k_3) = \begin{bmatrix} 90 & 24 & 38 & 20 & 8 & 2 & 4 & 6 & 6 & 6 & 4 & 2 & 0 & 0 & 2 & 0 \\ 12 & 8 & 12 & 8 & 6 & 2 & 2 & 2 & 10 & 2 & 4 & 4 & 4 & 0 & 2 & 2 \\ 20 & 8 & 12 & 8 & 2 & 2 & 2 & 2 & 4 & 2 & 6 & 4 & 2 & 0 & 4 & 2 \\ 4 & 2 & 6 & 4 & 2 & 0 & 4 & 2 & 4 & 2 & 4 & 2 & 2 & 0 & 2 & 0 \end{bmatrix}$$
$$0^{th}\text{ plane} \qquad 1^{st}\text{ plane} \qquad 2^{nd}\text{ plane} \qquad 3^{rd}\text{ plane}$$

3D-RT using Row-Depth-Column (RDC) Approach
The 3D matrix using Row-Depth-Column Rajan Transform is

$$X_{r,d,c}(k_1,k_2,k_3) = \begin{bmatrix} 90 & 24 & 38 & 20 & 18 & 6 & 12 & 12 & 24 & 12 & 18 & 12 & 8 & 0 & 10 & 4 \\ 4 & 2 & 12 & 6 & 2 & 2 & 0 & 4 & 4 & 4 & 2 & 0 & 4 & 0 & 2 & 0 \\ 10 & 6 & 4 & 4 & 10 & 2 & 4 & 4 & 8 & 4 & 2 & 4 & 4 & 0 & 2 & 4 \\ 0 & 4 & 2 & 2 & 2 & 2 & 0 & 4 & 4 & 4 & 2 & 0 & 0 & 0 & 2 & 0 \end{bmatrix}$$
$$0^{th}\text{ plane} \qquad 1^{st}\text{ plane} \qquad 2^{nd}\text{ plane} \qquad 3^{rd}\text{ plane}$$

Note that $X_{r,c,d}(k_1,k_2,k_3) \neq X_{r,d,c}(k_1,k_2,k_3)$, but the CPI is same as 90.
3D-RT using Column-Row-Depth (CRD) Approach
The 3D matrix using Column-Row-DepthRajan Transform is

$$X_{c,r,d}(k_1,k_2,k_3) = \begin{bmatrix} 90 & 18 & 26 & 8 & 8 & 4 & 0 & 6 & 6 & 8 & 8 & 8 & 0 & 2 & 2 & 6 \\ 24 & 12 & 4 & 4 & 6 & 2 & 2 & 2 & 6 & 2 & 2 & 2 & 4 & 0 & 0 & 0 \\ 34 & 12 & 10 & 4 & 4 & 2 & 8 & 2 & 10 & 8 & 4 & 2 & 0 & 2 & 2 & 0 \\ 16 & 2 & 6 & 6 & 2 & 0 & 0 & 0 & 2 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \end{bmatrix}$$
$$0^{th}\text{ plane} \qquad 1^{st}\text{ plane} \qquad 2^{nd}\text{ plane} \qquad 3^{rd}\text{ plane}$$

Note that $X_{c,r,d}(k_1,k_2,k_3)$ is difference from others, but the CPI is same as 90.
3D-RT using Column-Depth-Row (CDR) Approach
The 3D matrix using Column-Depth-RowRajan Transform is

$$X_{c,d,r}(k_1,k_2,k_3) = \begin{bmatrix} 90 & 6 & 18 & 14 & 12 & 4 & 4 & 4 & 20 & 0 & 8 & 4 & 11 & 7 & 7 & 3 \\ 20 & 16 & 8 & 4 & 6 & 2 & 2 & 2 & 8 & 0 & 0 & 0 & 2 & 2 & 2 & 2 \\ 34 & 10 & 2 & 2 & 4 & 0 & 4 & 0 & 16 & 8 & 4 & 4 & 6 & 2 & 2 & 2 \\ 16 & 0 & 4 & 4 & 2 & 2 & 2 & 2 & 6 & 2 & 2 & 2 & 0 & 0 & 0 & 0 \end{bmatrix}$$
$$0^{th}\text{ plane} \qquad 1^{st}\text{ plane} \qquad 2^{nd}\text{ plane} \qquad 3^{rd}\text{ plane}$$

Note that $X_{c,r,d}(k_1,k_2,k_3) \neq X_{c,d,r}(k_1,k_2,k_3)$, but the CPI is same as 90.
3D-RT using Depth-Row-Column (DRC) Approach
The 3D matrix using Depth-Row-ColumnRajan Transform is

$$X_{d,r,c}(k_1,k_2,k_3) = \begin{bmatrix} 90 & 6 & 22 & 10 & 26 & 10 & 12 & 4 & 28 & 6 & 8 & 4 & 16 & 2 & 10 & 4 \\ 4 & 4 & 0 & 0 & 4 & 0 & 2 & 2 & 2 & 0 & 6 & 2 & 2 & 0 & 0 & 2 \\ 10 & 6 & 2 & 6 & 6 & 2 & 2 & 4 & 8 & 2 & 2 & 2 & 6 & 2 & 2 & 2 \\ 0 & 4 & 0 & 4 & 0 & 0 & 0 & 2 & 6 & 0 & 0 & 0 & 4 & 0 & 0 & 0 \end{bmatrix}$$
$$0^{th}\text{ plane} \qquad 1^{st}\text{ plane} \qquad 2^{nd}\text{ plane} \qquad 3^{rd}\text{ plane}$$

Note that $X_{d,r,c}(k_1,k_2,k_3)$ is difference from others, but the CPI is same as 90.
3D-RT using Depth-Column-Row (DCR) Approach
The 3D matrix using Depth-Column-RowRajan Transform is

$$X_{d,\,c,\,r}(k_1,k_2,k_3) = \left[\begin{array}{cccc|cccc|cccc|cccc} 90 & 6 & 18 & 14 & 26 & 2 & 10 & 2 & 38 & 6 & 6 & 2 & 16 & 0 & 4 & 4 \\ 4 & 4 & 4 & 4 & 4 & 0 & 4 & 0 & 12 & 0 & 8 & 4 & 2 & 2 & 2 & 2 \\ 10 & 6 & 2 & 2 & 10 & 6 & 6 & 2 & 8 & 0 & 0 & 0 & 10 & 2 & 2 & 2 \\ 8 & 4 & 4 & 0 & 4 & 4 & 0 & 0 & 6 & 2 & 2 & 2 & 4 & 0 & 4 & 0 \end{array}\right]$$

0^{th} plane 1^{st} plane 2^{nd} plane 3^{rd} plane

Note that $X_{d,r,c}(k_1,k2,k3) \neq X_{d,c,r}(k_1,k_2,k_3)$, but the CPI is same as 90.

3D-RT Translation invariance property

Let us consider a solid 3-D object of 3x3x3 size placed in top left corner of the lattice grid as shown in Fig. 3. Now Rajan transform is applied to this 8x8x8 matrix Row-wise (ref Fig. 4), Column-wise (ref Fig. 5) and Depth-wise (ref Fig. 6). The resultant 3-D matrix is shown in Fig. 6. Next, the 3-D object subimage is translated in the 8x8x8 matrix as shown in Fig. 7. Subsequently RT is applied to this 8x8x8 translated matrix Row-wise (ref Fig. 8), Column-wise (ref Fig. 9) and Depth-wise (ref Fig. 10) and resultant matrix is shown in Fig. 10. Fig. 11 shows translation invariance property demonstrated by 3D-RT

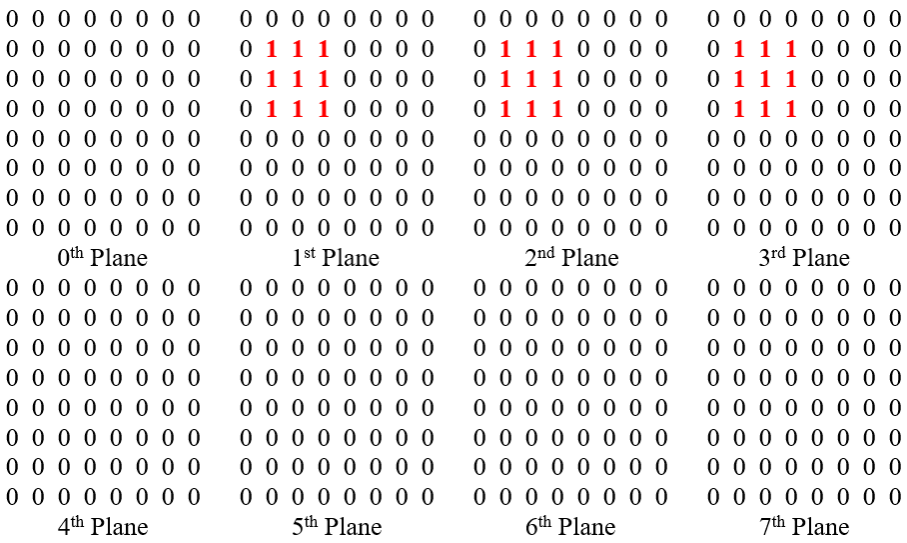


Fig. 3: 3-D Object of size 3x3x3 with in the 8x8x8 grid

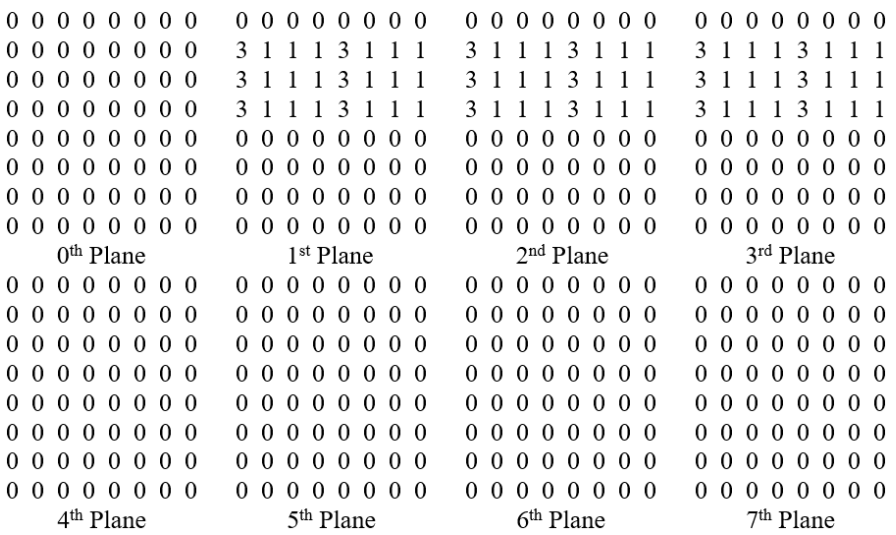


Fig. 4: Resultant 3-D array after applying row wise RT to Fig. 3

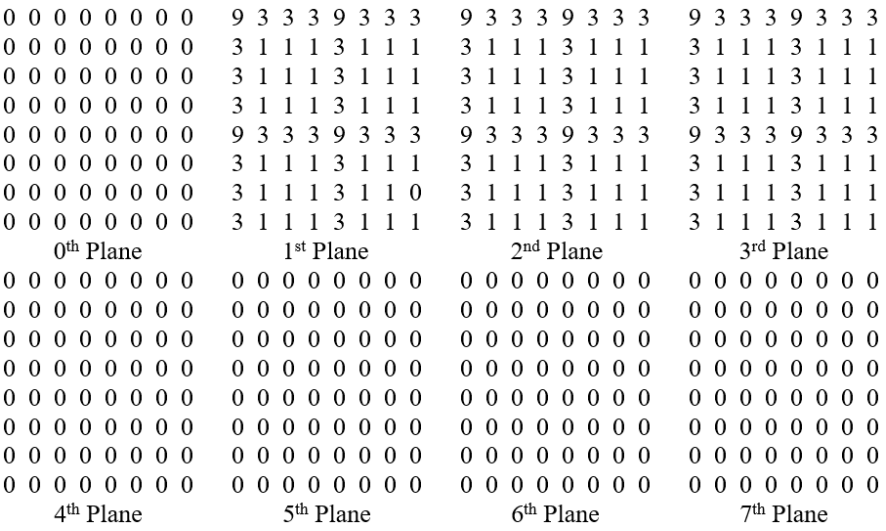


Fig. 5: Resultant 3-D array after applying column wise RT to Fig. 4

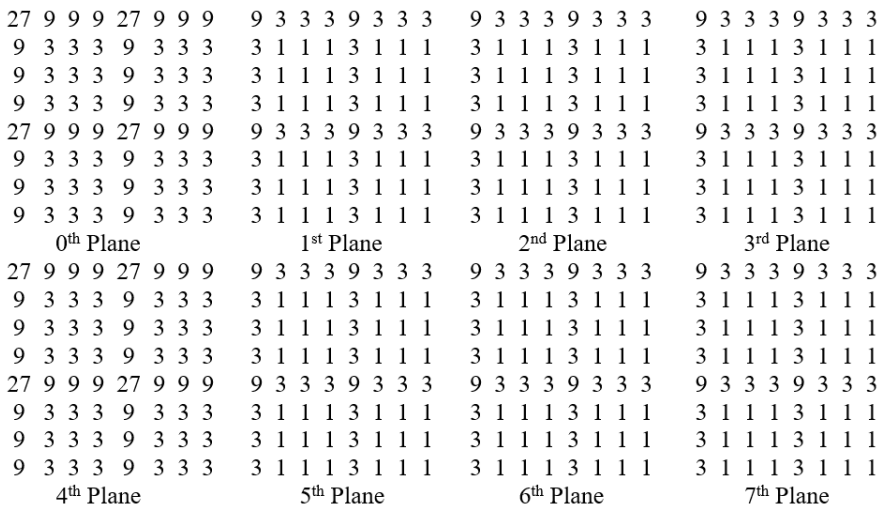


Fig. 6: Resultant 3-D array after applying depth wise RT to Fig. 5

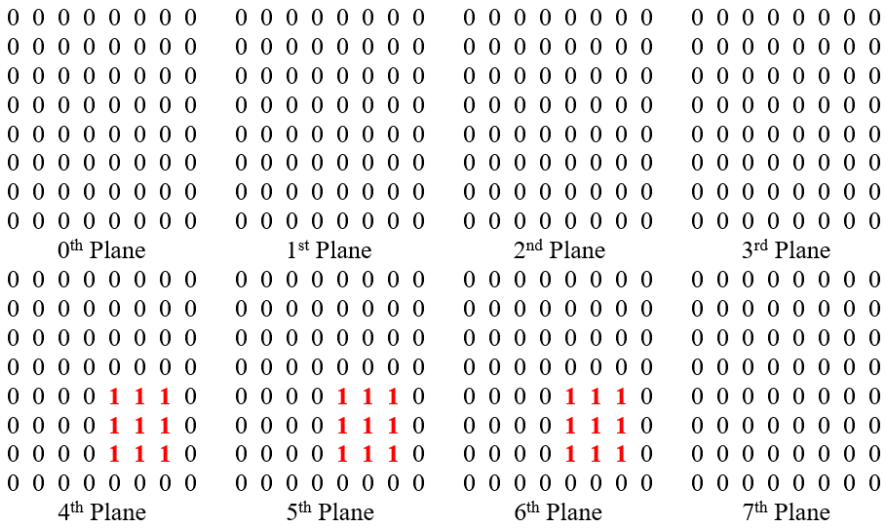


Fig. 7: 3-D Object is translated with in the 8x8x8 grid

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0^{th} Plane								1^{st} Plane								2^{nd} Plane								3^{rd} Plane							
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	0	0	0	0	0	0	0	0
3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	0	0	0	0	0	0	0	0
3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4^{th} Plane								5^{th} Plane								6^{th} Plane								7^{th} Plane							

Fig. 8: Resultant 3-D array after applying row wise RT to Fig. 7

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0^{th} Plane								1^{st} Plane								2^{nd} Plane								3^{rd} Plane							
9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3	0	0	0	0	0	0	0	0
3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	0	0	0	0	0	0	0	0
3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	0	0	0	0	0	0	0	0
3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	0	0	0	0	0	0	0	0
9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3	0	0	0	0	0	0	0	0
3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	0	0	0	0	0	0	0	0
3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	0	0	0	0	0	0	0	0
3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	0	0	0	0	0	0	0	0
4^{th} Plane								5^{th} Plane								6^{th} Plane								7^{th} Plane							

Fig. 9: Resultant 3-D array after applying column wise RT to Fig. 8

27	9	9	9	27	9	9	9	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3
9	3	3	3	9	3	3	3	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1
9	3	3	3	9	3	3	3	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1
9	3	3	3	9	3	3	3	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1
27	9	9	9	27	9	9	9	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3
9	3	3	3	9	3	3	3	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1
9	3	3	3	9	3	3	3	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1
9	3	3	3	9	3	3	3	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1
0^{th} Plane								1^{st} Plane								2^{nd} Plane								3^{rd} Plane							
27	9	9	9	27	9	9	9	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3
9	3	3	3	9	3	3	3	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1
9	3	3	3	9	3	3	3	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1
9	3	3	3	9	3	3	3	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1
27	9	9	9	27	9	9	9	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3	9	3	3	3
9	3	3	3	9	3	3	3	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1
9	3	3	3	9	3	3	3	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1
9	3	3	3	9	3	3	3	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1	3	1	1	1
4^{th} Plane								5^{th} Plane								6^{th} Plane								7^{th} Plane							

Fig. 10: Resultant 3-D array after applying depth wise RT to Fig. 9

Fig. 11 given in the next page, shows that 3D-RT of the original 3-D matrix (Fig. 6) and the 3D-RT of the translated 3-D matrix (Fig. 10) are the same. So, the invariance property of 3D-RT could be seen to be applicable to 3D images also. This amounts to saying that 1D-RT, 2D-RT and 3D-RT could be reliably used in real world applications.

Global Journal of Computer Science and Technology (G) Volume XX Issue VI Version I Year 2020

Cyclic Shift Invariance Property

Consider the sequence x(n)=3,8,5,6,0,2,9,6. Now, seven more cyclic shifted versions such as $x_{c1}(n)$=6,3,8,5,6,0,2,9; $x_{c2}(n)$=9,6,3,8,5,6,0,2; $x_{c3}(n)$=2,9,6,3,8,5,6,0; $x_{c4}(n)$=0,2,9,6,3,8,5,6; $x_{c5}(n)$=6,0,2,9,6,3,8,5; $x_{c6}(n)$=5,6,0,2,9,6,3,8 and $x_{c7}(n)$=8,5,6,0,2,9,6,3 could be generated from x(n). All the eight sequences have the same X(k)=39,5,13,9,13,1,7,5, meaning Rajan Transform of a given sequence of length N would remain invariant for N cyclically permuted sequences. This property is satisfied by higher order RT, that is, 2D-RT and 3D-RT also.

Graphical Inverse Invariance Property

Consider x(n)=3,8,5,6,0,2,9,6. Its graphical inverse is $x^{-1}(n)$=6,9,2,0,6,5,8,3. Now, one can generate seven more cyclic shifted versions such as $x_{c1}^{-1}(n)$=3,6,9,2,0,6,5,8; $x_{c2}^{-1}(n)$=8,3,6,9,2,0,6,5; $x_{c3}^{-1}(n)$=5,8,3,6,9,2,0,6, $x_{c4}^{-1}(n)$=6,5,8,3,6,9,2,0; $x_{c5}^{-1}(n)$=0,6,5,8,3,6,9,2; $x_{c6}^{-1}(n)$=2,0,6,5,8,3,6,9 and$x_{c7}^{-1}(n)$=9,2,0,6,5,8,3,6. One can easily verify that all these eight sequences have the same X(k)=39,5,13, 9,13,1,7,5, meaning Rajan Transform of a given sequence of length N would remain invariant for N graphically inverted sequence and its cyclically permuted sequences. This property is satisfied by higher order RT, that is, 2D-RT and 3D-RT also.

Dyadic Shift Invariance Property

Consider x(n)=3,8,5,6,0,2,9,6 and transpose its first half with the second half. The resulting sequence $Td^{(2)}[x(n)]$=0,2,9,6,3,8,5,6 is the 2-block dyadic shifted version of x(n). The symbol $Td^{(2)}$ denotes the 2-block dyadic shift operator. In the same manner, one would obtain $Td^{(4)}[Td^{(2)}[x(n)]]$=9,6,0,2,5,6,3,8 and $Td^{(8)}[Td^{(4)}[Td^{(2)}[x(n)]]]$=6,9,2,0,6,5,8,3. Note that the graphical inverse of x(n) is $x^{-1}(n)$=(6,9,2,0,6,5,8,3) and it is the same as $Td^{(8)}[Td^{(4)}[Td^{(2)}[x(n)]]]$=6,9,2,0,6,5,8,3. One can easily verify that all these dyadic shifted sequences have the same X(k), that is, the sequence 39,5,13,9,13,7,5. There is yet another way of dyadic shifting the input sequence x(n) to $Td^{(2)}[Td^{(4)}[Td^{(8)}[x(n)]]]$. Consider the sequence x(n)=3,8,5,6,0,2,9,6 and one can obtain $Td^{(8)}[x(n)]$=8,3,6,5,2,0,6,9; $Td^{(4)}[Td^{(8)}[x(n)]]$=6,5,8,3,6,9, 2,0 and $Td^{(2)}[Td^{(4)}[Td^{(8)}[x(n)]]]$=6,9,2,0,6,5,8,3 as dyadic shifts. Note that $Td^{(2)}[Td^{(4)}[Td^{(8)}[x(n)]]] = Td^{(8)}[Td^{(4)}[Td^{(2)}[x(n)]]]$. This property is satisfied by higher order RT, that is, 2D-RT and 3D-RT also.

Dual Class Invariance Property

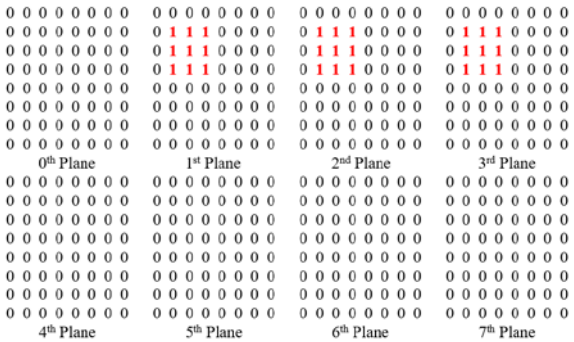
Given a sequence x(n), one can construct another sequence y(n) consisting of at least one number which is not present in x(n) such that X(k)=Y(k). In such a case, y(n) is called the 'dual' of x(n). Consider two sequences x(n)=2,4,2,2 and y(n)=3,1,3,3. Then X(k)=Y(k)=10,2,2,2. An underlying theorem to characterize a sequence of length $N=2^n$ to pair up with a dual sequence is "*A sequence is said to have a dual if and only if its CPI is an even number and is divisible by N/2*". This theorem advocates a necessary condition but not a sufficient condition. For example, consider the sequence x(n)=6,8,2,0. This indeed satisfies the theorem. That is, its CPI is 16 and the value of CPI/(N/2) is 8. Now its dual is computed as y(n)=2,0,6,8, which is not a dual of x(n) as per the definition. In such cases, they are called 'self dual' pairs. Some of the properties of dual sequences are: (i) if y(n) is a dual of x(n), then x(n) is also called the dual of y(n). Hence the pair<x(n),y(n)> is called 'dual pair'; (ii) dual of a sequence, say y(n) will necessarily exhibit geometric symmetry together with the original sequence x(n); (iii) each dual pair has a value called 'Differential Mean' (DM), which is equal to $(|x(i)\sim y(i)|)/2; 0\leq i\leq(N-1)$ about which the dual sequences are 'flip' symmetric. DM could be a real number. This property is satisfied by higher order RT, that is, 2D-RT and 3D-RT also.

Scalar Property

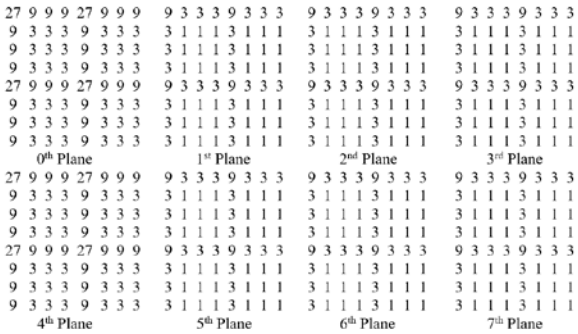
Let x(n) be a number sequence and λ be a scalar. Then the RT of λx(n) will be λX(k), where X(k) is the RT of x(n). For example, let us consider a sequence x(n)=1,3,1,2 and a scalar λ of value 2. Now the RT X(k) of x(n) is 7,3,1,1. RT of λx(n)=2,6,2,4 is 14,6,2,2 which is nothing but λX(k).This property is satisfied by higher order RT, that is, 2D-RT and 3D-RT also.

Linearity Property

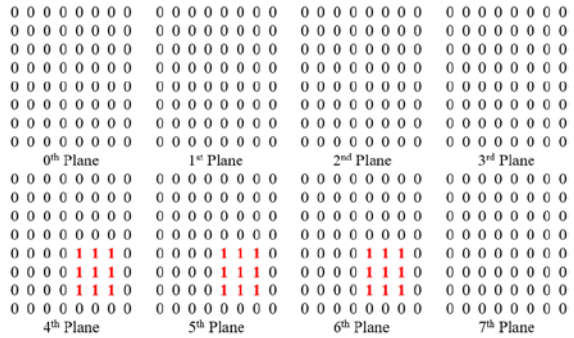
In general, RT does not satisfy the linearity property for all sequences. It was observed that for a pair x(n) and y(n) which are number sequences either in the increasing order or in the decreasing order, the linearity property works. That is, for λx(n)+my(n) where λ and m are scalars and x(n) and y(n) are two number sequences either in the increasing or decreasing order, the RT will be λX(k)+mY(k), where X(k) and Y(k) are respectively the RTs of x(n) and y(n). This property is satisfied by higher order RT, that is, 2D-RT and 3D-RT also.



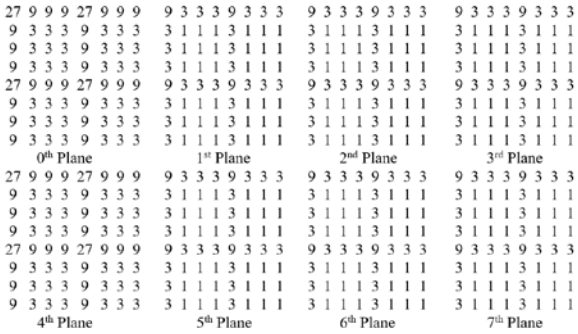
3-D Object of size 3x3x3 with in the 8x8x8 grid



Corresponding 3D Rajan Transform as shown in Fig. 6



3-D Object is translated with in the 8x8x8 grid



Corresponding3D Rajan Transform as shown in Fig. 10

Fig. 11: Translation invariance property demonstrated by 3D-RT

IV. Conclusions

This paper introduces the formulation of a novel transform called Multi-Dimensional Rajan Transform, which is an extension of One-Dimensional Rajan Transform (1D-RT). 2D and 3D Rajan Transforms are presented and the translation invariant property of 3D Rajan Transform demonstrated with the help of an example. One can explore the algebraic properties of Multi-Dimensional Rajan Transform as future work.

Acknowledgement

The first author expresses his gratitude to the second and third authors for introducing the fundamental notion of Rajan Transform for carrying out his Post-Doctoral research work.

References Références Referencias

1. K. Thiagarajan, "Multi-Dimensional Rajan Transforms and Their Applications", Post Doctoral Research Report, Fellow No. PRC/Admn/PDF-Thiagarajan/2019-1, Pattern Recognition Division, Pentagram Research Centre, Hyderabad, India, August 10, 2020.