IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 28, 2021, accepted August 15, 2021, date of publication August 24, 2021, date of current version September 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3107579

A General Method for Generating Discrete Orthogonal Matrices

KA-HOU CHAN[1], WEI KE[1,2], AND SIO-KEI IM[3], (Member, IEEE)
[1]School of Applied Sciences, Macao Polytechnic Institute, Macau, China
[2]Engineering Research Centre of Applied Technology on Machine Translation and Artificial Intelligence, Ministry of Education, Macao Polytechnic Institute, Macau, China
[3]Macao Polytechnic Institute, Macau, China

Corresponding author: Ka-Hou Chan (chankahou@ipm.edu.mo)

This work was supported by The Science and Technology Development Fund, Macau, under Grant 0001/2018/AFJ.

ABSTRACT Discrete orthogonal matrices have applications in information coding and cryptography. It is often challenging to generate discrete orthogonal matrices. A common approach widely in use is to discretize continuous orthogonal functions that have been discovered. The need of such continuous functions is restrictive. Polynomials, as the simplest class of continuous functions, are widely studied for their orthogonality, to serve the purpose of generating orthogonal matrices. However, beginning with continuous orthogonal polynomials still takes much work. To overcome this complexity while improving the efficiency and flexibility, we present a general method for generating orthogonal matrices directly through the construction of certain even and odd polynomials from a set of distinct positive values, bypassing the need of continuous orthogonal functions. We present a constructive proof by induction that not only asserts the existence of such polynomials, but also tells how to iteratively construct them. Besides the derivation of the method as simple as a few nested loops, we discuss two well-known discrete transforms, the Discrete Cosine Transform and the Discrete Tchebichef Transform, about how they can be achieved using our method with the specific values, and how to embed them into the transform module of video coding. By the same token, we also give the examples for generating new orthogonal matrices from arbitrarily chosen values. The demonstrative experiments indicate that our method is not only simpler to implement, but also more efficient and flexible. It can generate orthogonal matrices of larger sizes, compared with those existing methods.

INDEX TERMS Discrete orthogonal matrices, discrete cosine transform, discrete tchebichef transform, orthogonal polynomials, invertible transformers.

I. INTRODUCTION

Orthogonal transformations have very useful properties in solving science and engineering problems. Just like the Fourier and Chebyshev series which are effective methods to project a periodic function into a series of linearly independent terms, orthogonal polynomials provide a natural way to solve the related problems, such as compression and protection in image processing [1]–[3], pattern recognition [4], [5] and feature capturing [6], [7]. It can also be applied in temporal video segmentation [8], face recognition [9], and character recognition [10]. Among various types of transformers, matrix transformers are most widely used due essentially to their simplicity and explicitness, especially for the transformations on real intervals ($\mathbb{R} \to \mathbb{R}$). Even more important, orthogonal matrices are of a special type of transformers, for they are always invertible. As a result, the source information can be recovered from the data that are transformed by an orthogonal matrix.

The invertibility of orthogonal matrices finds them a way in the applications of information coding, such as image compression. As in the above mentioned applications and techniques, image compression deals with a lot of bulky source data, for example video sequences, which often have real-time requirement. Hence, compression plays a major role in the storage and transmission. Techniques such as the Discrete Cosine Transform (DCT) [19] are typically used in video encoding for transformations from the spatial domain to the frequency domain [20], followed by coding methods such as Huffman coding. In recent years, the Discrete Tchebichef Transform (DTT) provides another transformation method using the Chebyshev moments [11], [21], which has as good energy compression properties as the DCT and works better for a certain class of images [22]. Both of the above example

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamed Elhoseny.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

Samsung Exhibit 1048, Page 1 of 12
Samsung Electronics Co., Ltd. et al. v. Genghiscomm Holdings LLC

TABLE 1. Application summary of some polynomial series.

Polynomial Series	Application	Limitation	Reference
Chebyshev	Image Compression	It is only considered on the interval $[-1, 1]$. This causes a failure of the Chebyshev approach in the problems that are naturally defined on larger domains.	[11], [12]
Hahn	Feature Extraction	Due to the high computational complexity, it is difficult to achieve feature description in real-time tasks.	[13]
Charlier	Signal Analysis	It has limitations in size when generating a discrete transform matrix.	[14]
Krawtchouk	Image Analysis	It suffers from the problem of initial value which tends to be zero as the polynomial-size increases.	[15], [16]
Gegenbauer	Character Recognition	It has poor image reconstruction capabilities.	[17]
Pseudo-Zernike	Image Description	It is not accurate in the strict sense and suffers from quantization and geometric errors.	[18]

transformations are defined upon orthogonal polynomials. The orthogonality is established over a continuous domain and approximated discretely over a certain amount of sample points. Discrete orthogonal transformations have witnessed the interplay of signal processing, semiconductor circuits, wireless networks and embedded systems to provide viable and cutting-edge technologies that are truly the state-of-the-art. The challenge lies in delivering practically realizable and economic solutions, while retaining the quality.

It is well-known that the orthogonality of two polynomials $P_i(x)$ and $P'_j(x)$, respectively having degrees i and j, is defined by extending the dot product of two vectors, the sum of the products of the corresponding components, to the integral of product $P_i(x)P'_j(x)$ over a continuous domain. Formally, when the integral becomes zero, the two polynomials are orthogonal to each other, i.e.,

$$\int P_i(x)P'_j(x)dx = 0, \quad i \neq j. \tag{1}$$

In practical applications, this definition is often approximated over a set of discrete samples $x_0, \ldots, x_{n-1}$,

$$\sum_{k=0}^{n-1} \left[P_i(x_k)P'_j(x_k)\right] = 0, \quad i \neq j. \tag{2}$$

Those satisfying this property are called discrete orthogonal polynomials [23]. Therefore, together with the degrees of polynomials ranging from 0 to $n-1$, an $n \times n$ discrete orthogonal matrix $\left[P_i(x_k)\right]$ can be constructed, where any two different row vectors are orthogonal. Discrete orthogonal matrices are commonly used in a number of orthogonal transformations over real intervals, such as the Chebyshev polynomials [24], [25], the Legendre polynomials [26], the Meixner polynomials [27], the Charlier polynomials [14], the Krawtchouk polynomials [28], the Discrete Hartley Transform [29] and the well-known Discrete Cosine Transform [19]. A comprehensive overview of these orthogonal polynomials, along with the development of their discrete matrices, is also detailed in [30]. We summarize the applications and limitations of some polynomial series in Table 1.

Once we have a set of discrete orthogonal polynomials, we are able to further obtain the orthogonal matrix by a series of substitutions. For instance, let $P_n(x) = \cos(n \arccos(x))$ be the n-degree Chebyshev polynomial. We can derive the discrete Chebyshev matrix by substituting in $P_i(x)$ the Chebyshev roots $x_j = \cos\frac{\pi(2j+1)}{2n}$, for $0 \leq i, j \leq n-1$. It can be found that for any j, the substitution of x_j in the higher degree $P_n(x_j)$ always gets zero. However, it is very hard to obtain the roots from other series of orthogonal polynomials. Our idea is to deduce the orthogonal polynomials by the given roots.

The purpose of this paper is to derive discrete orthogonal matrices directly by solving systems of linear equations, rather than to discretize existing continuous orthogonal polynomials. Our method has several advantages. It has virtually no precondition to use. Orthogonal matrices of arbitrary sizes can be generated to the need of an application. It directly follows the definition of discrete orthogonality, eliminating the need to discuss the orthogonal property over a continuous domain, such as the interval $[-1, +1]$ of the Chebyshev polynomials [25]. The method focuses on how to derive the coefficients of the polynomials that must be discretely orthogonal to each other over a set of given sample values, for example, $x_k = \cos\frac{\pi(2k+1)}{2n}$, for $k = 0, \ldots, n-1$, are the values of the $n \times n$ orthogonal matrix for the Discrete Cosine Transform (DCT) [19]. The errors in the discretization of continuous functions can also be avoided.

Generating orthogonal matrices directly from a set of values gives engineers a new way of obtaining such matrices with unlimited variations, without the need to discover and prove the properties of orthogonal polynomials mathematically in the first place. Although, by jumping to the construction directly, we sacrifice some mathematical insights and certainties, we provide a way to significantly broaden the base of discrete orthogonal matrices for engineering analyses. Our method is also simple and intuitive. It starts with the definition of discrete orthogonality, makes use of even and odd functions, inspired by the DCT and DTT, to simplify the problems, constructs the linear equation system for deriving the coefficients of the polynomials, proves that a unique solution exists, and finally inductively obtains the solution. Through the practicing of this method, we easily and effectively reproduce the orthogonal matrices for DCT and DTT in only a few simple steps. We also generate a couple of others to show the potential and flexibility.

Samsung Exhibit 1048, Page 2 of 12
Samsung Electronics Co., Ltd. et al. v. Genghiscomm Holdings LLC

Our contributions in this work can be summarized below.

- We propose a method for generating discrete orthogonal polynomials at a given $2m$ dimensional point specified by m arbitrary positive values.
- We inductively prove that this set of polynomial exists, and constructively determine all the coefficients by using the even and odd parity property.
- We obtain an efficient algorithm from the inductive proof to generate the orthogonal polynomials and the eventual discrete orthogonal matrix.
- We show that not only well-known discrete orthogonal matrices can be generated by our method, but also more arbitrary orthogonal matrices with larger sizes. Hence, our method is more general and flexible.

The rest of the paper is organized as follows.

- Section II goes through the related work in generating orthogonal matrices by various means.
- Section III presents the technical details and justifications, including the design, the proof by induction and the algorithm, for orthogonal matrix generation.
- Section IV reproduces a few well-known orthogonal matrices to show the simplicity and effectiveness of the method.
- Finally, we conclude the paper in Section V.

II. RELATED WORK

In the past three decades, many researchers aimed to generalize the theory of how to construct orthogonal polynomials, from the ones of a single discrete variable, as the solution of hypergeometric type differential equations, to that of multiple variables. In early days, a method was designed in [31] that began with the three-term recurrence relation for symmetric orthogonal polynomial systems to set up a partial differential equation for the orthogonal polynomials, in case of the connection problem, or for the product of two orthogonal polynomials, in case of the linearization problem. This equation had to be solved in terms of the initial data to expand the coefficients. In [32], to make the relevant orthogonality measures continuous, the parameter domain was carefully chosen. This method focuses on a different way to obtain parameters, where the orthogonality measure becomes merely discrete that it is finitely supported on the grid points with given weights. Other discrete multi-variable extensions of hypergeometric orthogonal polynomials were considered in [33].

Later, a novel set of discrete and continuous orthogonal matrices based on orthogonal polynomials was introduced into the field of orthogonal polynomial generation [34]. In [35], several relations linking the differences between bivariate discrete orthogonal polynomials and general polynomials were given. They presented a multi-variable generalization for all the discrete families, that gave each family a hypergeometric representation and a orthogonality weight function, proving that these polynomials were orthogonal with respect to the subspace of lower degrees and biorthogonal within a given subspace [36].

Next, a systematic study of the orthogonal polynomial solutions to a second order partial differential equation with two variables of hypergeometric type was made in [37]. In the bivariate discrete case, a hypergeometric formula was also given in [38]. For an infinitely differentiable function, the formula for the expansion coefficients of a general order derivative was available for the expansions in Chebyshev polynomials [39]. Thus the generation of recurrence relations to expand the coefficients of multi-variable orthogonal polynomials is similar to that in the single variable case [40], both continuous and discrete. These results motivated the researchers interested in multidimensional mathematical physics problems to use expansions in terms of orthogonal polynomials of multiple discrete variables.

Meanwhile, there were attempts in order to expand the coefficients of an arbitrary polynomial with a discrete variable and evaluate the expanded coefficients of an orthogonal matrix. Few advantages were achieved in these problems until the recent recursive approach was presented [41], and [42] gave an alternative way to the approaches for producing classical orthogonal polynomials. [43] used recurrent equations to prove the positivity of the connection coefficients between certain instances of orthogonal polynomials. They designed a constructive algorithm which allowed us to calculate recurrently the expansion coefficients of the evaluation problem. However, this approach requires the knowledge of the differential equation of the polynomial to expand, and the recursion relation as well as the differential-difference relation must be prepared for the polynomials conforming the orthogonal set. A few years later, the approach in [44] presented a very similar algorithm for finding the recurrence relation for both the connection and linearization coefficients. Also, another algorithm was developed for solving the connection problem between the four families of classical orthogonal polynomials [45].

Recently, [46] provided a series rearrangement technique combining a connection relation with a generating function, resulting in a series with multiple sums. Then, [47] extended this technique to many generating functions to derive a generalized generating function whose coefficients were given in hypergeometric functions. To the best of our knowledge, the coefficients of polynomials are always related to the polynomials of lower degrees when they are in a series of orthogonal polynomials of the same type. It leads to that the coefficients of a higher degree polynomial can be determined by some recursion or iteration relation of the corresponding linearization. The order of summations is then rearranged, and it is often simplified to the production of a generating function whose coefficients are given in terms of general or fundamental hypergeometric functions.

III. DISCRETE ORTHOGONAL POLYNOMIALS AND MATRICES

An $n \times n$ matrix M is an orthogonal matrix if the transpose M^T equals to the inverse M^{-1}. Thus, an orthogonal matrix is always invertible. By the definition $MM^T = I$, where I is

Samsung Exhibit 1048, Page 3 of 12
Samsung Electronics Co., Ltd. et al. v. Genghiscomm Holdings LLC

the identity matrix, the rows of an orthogonal matrix form an orthonormal basis that each row vector has length one, and is perpendicular to each other rows. Formally speaking, the dot product of two row vectors $\vec{a}_i \cdot \vec{a}_j$ is 1 when $i = j$, or 0 otherwise, that is,

$$\sum_{k=0}^{n-1}(a_{ik} \times a_{jk}) = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases} \tag{3}$$

for $0 \leq i, j \leq n-1$.

We consider the construction of a type of orthogonal matrices from a set of values $x_0, x_1, \ldots, x_{n-1}$, and a set of polynomials $P_0(x), \ldots, P_{n-1}(x)$ respectively of degrees from 0 to $n-1$. We denote the coefficients of the polynomial expansions by $c_{(i,k)}$, such that

$$P_i(x) = \sum_{k=0}^{i} c_{(i,k)} x^{i-k},$$

for $0 \leq i \leq n-1$. We then construct the orthogonal matrix of the form

$$M = \left[P_i(x_k)\right], \tag{4}$$

for $0 \leq i, k \leq n-1$, by deriving the polynomials $P_0(x), \ldots, P_{n-1}(x)$ [48]. These polynomials are called the orthonormal basis of the orthogonal matrix M [23], [49], [50].

Together with the condition of orthogonal matrices in (3), we require

$$\sum_{k=0}^{n-1}\left[P_i(x_k)P_j(x_k)\right] = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases} \tag{5}$$

for $0 \leq i, j \leq n-1$. An easy way to make a summation zero is to set half of the items the opposite values of the other half, for example, when $n = 2m$, we should have

$$P_i(x_k)P_j(x_k) = -P_i(x_{k+m})P_j(x_{k+m}),$$

for $0 \leq i, j \leq 2m-1$ and $0 \leq k \leq m-1$. We can further refine this condition to

$$P_i(x_k) = -P_i(x_{k+m}) \quad \text{and} \quad P_j(x_k) = P_j(x_{k+m}). \tag{6}$$

It's clear that when $x_k = -x_{k+m}$, the condition in (6) can be fulfilled if $P_i(x)$ is an odd function and $P_j(x)$ an even function.

Based on the analysis, we narrow the range of the polynomials down to only even and odd functions, together with a set of opposite values to make use of the parity as above. Given m distinct values $y_0, \ldots, y_{m-1} > 0$, we choose $\pm y_0, \ldots, \pm y_{m-1}$ as the set of values for the matrix construction. Thus, the matrix in (4) is formulated as

$$\left[P_i(+y_0) \;\; \cdots \;\; P_i(+y_{m-1}) \;\; P_i(-y_0) \;\; \cdots \;\; P_i(-y_{m-1})\right], \tag{7}$$

for $0 \leq i \leq 2m-1$. We are going to derive the orthogonal matrix in (7) by resolving the coefficients of polynomials $P_0, \ldots, P_{2m-1}$ based on the set of values $\pm y_0, \ldots, \pm y_{m-1}$.

A. EVEN AND ODD POLYNOMIALS

Consider the expansion of an i-degree polynomial. When $i = 2t$, an even polynomial can be constructed by removing all the odd-degree terms. Thus, the expansion of such an even polynomial can be written as

$$P_{2t}(x) = \sum_{p=0}^{t}\left[c_{(2t,2p)} x^{2(t-p)}\right]. \tag{8}$$

Similarly, when $i = 2t+1$, an odd polynomial can be obtained by multiplying an x to every and each term in (8), where all the even-degree terms are removed,

$$P_{2t+1}(x) = \sum_{p=0}^{t}\left[c_{(2t+1,2p+1)} x^{2(t-p)+1}\right]. \tag{9}$$

For the parity properties of even and odd polynomials, we have

$$P_{2t}(-x) = P_{2t}(x) \;\; \text{and} \;\; P_{2t+1}(-x) = -P_{2t+1}(x). \tag{10}$$

Now, we limit the choice of the $P_i(x)$ polynomials to those of the forms in (8) and (9). The number of the unknown coefficients is reduced to $t+1$ for each of the $(2t)$- and $(2t+1)$-degree polynomials. We are going to derive these unknown coefficients based on the condition of orthogonal matrices in (5). We substitute the rows of the matrix in (7) for the rows P_i and P_j in condition (5),

$$\sum_{k=0}^{2m-1}\left[P_i(x_k)P_j(x_k)\right] = \sum_{k=0}^{m-1}\begin{bmatrix} P_i(-y_k)P_j(-y_k) \\ +P_i(+y_k)P_j(+y_k) \end{bmatrix} = \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$$

for $0 \leq i, j \leq 2m-1$, and consider the parity property in (10), we have

$$\sum_{k=0}^{2m-1}\left[P_i(x_k)P_j(x_k)\right] = \begin{cases} 0, & i \not\equiv j \pmod 2 \\ 2 \times \sum_{k=0}^{m-1}\left[P_i(y_k)P_j(y_k)\right], & i \equiv j \pmod 2. \end{cases} \tag{11}$$

To derive the coefficients for the orthogonal matrix, we focus on the case of $i \equiv j \pmod 2$, where the sum is required to be 1 when $i = j$, or 0 otherwise.

B. POLYNOMIAL COEFFICIENT INDUCTION

The dot product of a row in an orthogonal matrix with itself is 1, or 0 with another row. An even polynomial $P_{2t}(x)$ in (8) has only $t+1$ coefficients to resolve. If we take the highest coefficient (with $p = 0$) out and resolve it later by the unit length condition, there are only t coefficients left,

$$d_{(2t,2p)} = \frac{c_{(2t,2p)}}{c_{(2t,0)}} \quad \text{and} \quad d_{(2t+1,2p+1)} = \frac{c_{(2t+1,2p+1)}}{c_{(2t+1,1)}},$$

Samsung Exhibit 1048, Page 4 of 12
Samsung Electronics Co., Ltd. et al. v. Genghiscomm Holdings LLC

for $1 \le p \le t$, and we have $d_{(2t,0)} = d_{(2t+1,1)} = 1$. We denote this form of $P(x)$ as $\hat{P}(x)$, i.e.,

$$\hat{P}_{2t}(x) = \frac{P_{2t}(x)}{c_{(2t,0)}} \quad \text{and} \quad \hat{P}_{2t+1}(x) = \frac{P_{2t+1}(x)}{c_{(2t+1,1)}}.$$

Obviously, we can safely replace those $P(x)$ with $\hat{P}(x)$ in the discussion of obtaining the perpendicularity between two rows in the matrix, since there is only a scalar difference. There are exactly t even polynomials, $\hat{P}_0(x), \hat{P}_2(x), \ldots, \hat{P}_{2(t-1)}(x)$ with smaller degrees in the matrix. By the condition that the t rows constructed by these smaller polynomials are perpendicular to the row from polynomial $\hat{P}_{2t}(x)$, it establishes a system of t equations. If there are solutions to the equation system, and we can find a general way to solve the coefficients $d_{(2t,2p)}$, for $1 \le p \le t$, from these equations, then we are able to obtain the coefficients of all the polynomials from $\hat{P}_0(x)$ to $\hat{P}_{2t}(x)$ inductively. The base case is trivial, that is, $\hat{P}_0(x) = 1$.

For such a matrix of size $2m \times 2m$, the equation system for the coefficients of $\hat{P}_{2t}(x)$ is straightforward, by letting the dot products with those smaller even polynomials be 0,

$$\sum_{k=0}^{m-1}\left[\hat{P}_{2i}(y_k)\hat{P}_{2t}(y_k)\right] = \sum_{k=0}^{m-1}\left[\hat{P}_{2i}(y_k)\left(y_k^{2t} + \sum_{p=1}^{t}\left(d_{(2t,2p)}y_k^{2(t-p)}\right)\right)\right] = 0,$$

for $0 \le i \le t-1$. Then, we examine the terms containing a certain coefficient $d_{(2t,2p)}$, for $1 \le p \le t$. The above equation system can be written as

$$\sum_{k=0}^{m-1}\left[\hat{P}_{2i}(y_k)y_k^{2t}\right] + \sum_{p=1}^{t}\sum_{k=0}^{m-1}\left[d_{(2t,2p)}\left(\hat{P}_{2i}(y_k)y_k^{2(t-p)}\right)\right] = 0,$$

for $0 \le i \le t-1$. Thus, we have a linear equation system for the unknown coefficients as

$$A_t D_t = -B_t, \tag{12}$$

where

$$A_t = \left[\sum_{k=0}^{m-1}\left(\hat{P}_{2i}(y_k)y_k^{2(t-p)}\right)\right]_{0\le i\le t-1, 1\le p\le t},$$

$$D_t = \left[d_{(2t,2p)}\right]_{1\le p\le t},$$

$$B_t = \left[\sum_{k=0}^{m-1}\left(\hat{P}_{2i}(y_k)y_k^{2t}\right)\right]_{0\le i\le t-1}.$$

We induct on t to prove that the determinant $\mathbf{det}\,(A_t) \neq 0$, thus (12) has a unique solution to D_t.

Proposition 1: For $1 \le t \le m-1$, $\mathbf{det}\,(A_t) \neq 0$.

Proof: The base case is trivial that $A_1 = [m]$, thus $\mathbf{det}\,(A_1) = m \neq 0$.

When $2 \le t \le m-1$, we have

$$\hat{P}_{2(t-1)}(y_k) = \left[y_k^{2(t-p)}\right]_{1\le p\le t}^T \begin{bmatrix} 1 \\ D_{t-1} \end{bmatrix},$$

for $0 \le k \le m-1$. By induction hypothesis, D_{t-1} has a unique solution, also by the Cramer's rule,

$$D_{t-1} = \left[\frac{\mathbf{det}\left(A_{t-1}\left[B_{t-1}/p\right]\right)}{\mathbf{det}\,(A_{t-1})}\right]_{1\le p\le t-1},$$

where $A[B/p]$ is the matrix formed by replacing the p-th column of A by the column vector B. Consider the matrix

$$C_t(x) = \left[\begin{array}{c|c} B_{t-1} & A_{t-1} \\ \hline \multicolumn{2}{c}{\left[x^{2(t-p)}\right]_{1\le p\le t}^T} \end{array}\right],$$

and the cofactor expansion of $\mathbf{det}\,(C_t(y_k))$ along the bottom row, we establish the following identity,

$$\mathbf{det}\,(C_t(y_k)) = \mathbf{det}\,(A_{t-1}) \times \hat{P}_{2(t-1)}(y_k),$$

for $0 \le k \le m-1$. Furthermore, if we partition A_t similarly, then we get

$$A_t = \left[\begin{array}{c|c} B_{t-1} & A_{t-1} \\ \hline \multicolumn{2}{c}{\left[\sum_{k=0}^{m-1}\left(\hat{P}_{2(t-1)}(y_k) \times y_k^{2(t-p)}\right)\right]_{1\le p\le t}^T} \end{array}\right]. \tag{13}$$

Thus, by comparing A_t with C_t, we have the following conclusion,

$$\mathbf{det}\,(A_t) = \sum_{k=0}^{m-1}\left[\hat{P}_{2(t-1)}(y_k) \times \mathbf{det}\,(C_t(y_k))\right] = \mathbf{det}\,(A_{t-1}) \times \sum_{k=0}^{m-1}\left[\hat{P}_{2(t-1)}(y_k)\right]^2 \neq 0.$$

Notice that $\hat{P}_{2(t-1)}$ is an even function, thus it has at most $t-1$ positive roots. On the other hand, we have m distinct positive y_k values and $t \le m-1$, therefore the sum of the squares above cannot be zero. □

For those odd polynomials $\hat{P}_{2t+1}(x)$ $(0 \le t \le m-1)$, the base case is $\hat{P}_1(x) = x$. We can obtain an equation system similar to (12) to solve the coefficients inductively. We denote this equation system as

$$\acute{A}_t \acute{D}_t = -\acute{B}_t, \tag{14}$$

where

$$\acute{A}_t = \left[\sum_{k=0}^{m-1}\left(\hat{P}_{2i+1}(y_k)y_k^{2(t-p)+1}\right)\right]_{0\le i\le t-1, 1\le p\le t},$$

$$\acute{D}_t = \left[d_{(2t+1,2p+1)}\right]_{1\le p\le t},$$

Samsung Exhibit 1048, Page 5 of 12
Samsung Electronics Co., Ltd. et al. v. Genghiscomm Holdings LLC

$$\acute{B}_t = \left[\sum_{k=0}^{m-1}\left(\hat{P}_{2i+1}(y_k)y_k^{2t+1}\right)\right]_{0\le i\le t-1}.$$

We can also prove that (14) has a unique solution to $\acute{D}_t$.

Proposition 2: For $1 \le t \le m-1$, $\mathbf{det}\left(\acute{A}_t\right) \neq 0$.

Proof: This proof is almost identical to the proof of Proposition 1, with a different base case $\acute{A}_1$, where

$$\acute{A}_1 = \left[\sum_{k=0}^{m-1}\left(\hat{P}_1(y_k)y_k\right)\right] = \left[\sum_{k=0}^{m-1} y_k^2\right].$$

Notice that $\hat{P}_1(x) = x$. Since all $y_k > 0$, certainly we have $\mathbf{det}\left(\acute{A}_1\right) \neq 0$. For the induction step, we substitute in the $\acute{A}$, $\acute{D}$ and $\acute{B}$ counterparts, together with

$$\acute{C}_t(x) = \left[\begin{array}{c|c} \acute{B}_{t-1} & \acute{A}_{t-1} \\ \hline \multicolumn{2}{c}{\left[x^{2(t-p)+1}\right]^T_{1\le p\le t}} \end{array}\right].$$

Also, for the number of positive roots of an odd polynomial $\hat{P}_{2(t-1)+1}$, we still have at most $t-1$, because zero is a root for any odd polynomial. □

The proofs also give us a method to derive the polynomials $\hat{P}_{2t}(x)$ and $\hat{P}_{2t+1}(x)$ inductively. We have

$$\hat{P}_{2t}(x) = \left[x^{2(t-p)}\right]^T_{0\le p\le t}\begin{bmatrix}1\\ D_t\end{bmatrix} \quad \text{and}$$

$$\hat{P}_{2t+1}(x) = \left[x^{2(t-p)+1}\right]^T_{0\le p\le t}\begin{bmatrix}1\\ \acute{D}_t\end{bmatrix}, \tag{15}$$

for $0 \le t \le m-1$, where $D_t = A_t^{-1}B_t$ and $\acute{D}_t = \acute{A}_t^{-1}\acute{B}_t$ respectively.

C. OBTAINING UNIT VECTORS

To make each row vector of (7) having the unit length, we refer to the condition in (11),

$$2 \times \sum_{k=0}^{m-1}[P_i(y_k)]^2 = 1.$$

Thus, together with the fact $c_{(i,0)}\hat{P}_i(x) = P_i(x)$, we have

$$2 \times c_{(i,0)}^2 \sum_{k=0}^{m-1}\left[\hat{P}_i(y_k)\right]^2 = 1,$$

hence,

$$c_{(i,0)} = \pm\left(2 \times \sum_{k=0}^{m-1}\left[\hat{P}_i(y_k)\right]^2\right)^{-\frac{1}{2}}. \tag{16}$$

As a result, we have derived the method to obtain a $2m \times 2m$ orthogonal matrix based on any set of m distinct positive values. Algorithm 1 presents the overall procedure.

The procedure strictly follows the inductive proofs in Section III-B. The first loop, #1–4, initializes $\hat{P}_0$ and $\hat{P}_1$, corresponding to the base cases of the induction. According to (6), half of the $2m$ polynomial evaluations can be derived from the other half because of the parity property, we thus only need to set the first m values for each polynomial. The second loop, #5–20, computes the higher degree polynomials iteratively by constructing and solving the equation systems (12) and (14). We use $2t$ and $2t+1$ to step over the even and odd degrees pair by pair, ranging over $2, 3, \ldots, 2m-2, 2m-1$. According to (13), we use i to range over the rows and p over the columns to build $A, \acute{A}, B, \acute{B}$ progressively, from lower to higher dimensions. Then, we solve the equation systems and obtain the polnomials by (15). In the final loop, #21–30, we compute P from $\hat{P}$ by the unit length normalization in (16), and fill-in the orthogonal matrix as the result.

Algorithm 1 Orthogonal Matrix Generation

Input: m distinct positive values $y_0, \ldots, y_{m-1}$

Output: a $2m \times 2m$ orthogonal matrix M

1: **for** $k \leftarrow 0, \ldots, m-1$ **do**
2: $\quad \hat{P}_{0,k} \leftarrow 1$
3: $\quad \hat{P}_{1,k} \leftarrow y_k$
4: **end for**
5: **for** $t \leftarrow 1, \ldots, m-1$ **do**
6: $\quad$ **for** $i \leftarrow 0, \ldots, t-1$ **do**
7: $\quad\quad$ **for** $p \leftarrow 1, \ldots, t$ **do**
8: $\quad\quad\quad A_{i,p-1} \leftarrow \sum_{k=0}^{m-1}\left[\hat{P}_{2i,k} \times y_k^{2(t-p)}\right]$
9: $\quad\quad\quad \acute{A}_{i,p-1} \leftarrow \sum_{k=0}^{m-1}\left[\hat{P}_{2i+1,k} \times y_k^{2(t-p)+1}\right]$
10: $\quad\quad$ **end for**
11: $\quad\quad B_i \leftarrow \sum_{k=0}^{m-1}\left[\hat{P}_{2i,k} \times y_k^{2t}\right]$
12: $\quad\quad \acute{B}_i \leftarrow \sum_{k=0}^{m-1}\left[\hat{P}_{2i+1,k} \times y_k^{2t+1}\right]$
13: $\quad$ **end for**
14: $\quad D \leftarrow A^{-1}(-B)$
15: $\quad \acute{D} \leftarrow \acute{A}^{-1}\left(-\acute{B}\right)$
16: $\quad$ **for** $k \leftarrow 0, \ldots, m-1$ **do**
17: $\quad\quad \hat{P}_{2t,k} \leftarrow y_k^{2t} + \sum_{p=1}^{t}\left[y_k^{2(t-p)} \times D_{p-1}\right]$
18: $\quad\quad \hat{P}_{2t+1,k} \leftarrow y_k^{2t+1} + \sum_{p=1}^{t}\left[y_k^{2(t-p)+1} \times \acute{D}_{p-1}\right]$
19: $\quad$ **end for**
20: **end for**
21: **for** $t \leftarrow 0, \ldots, m-1$ **do**
22: $\quad c \leftarrow \pm\left(2\sum_{k=0}^{m-1}\hat{P}_{2t,k}^2\right)^{-\frac{1}{2}}$
23: $\quad \acute{c} \leftarrow \pm\left(2\sum_{k=0}^{m-1}\hat{P}_{2t+1,k}^2\right)^{-\frac{1}{2}}$
24: $\quad$ **for** $k \leftarrow 0, \ldots, m-1$ **do**
25: $\quad\quad M_{2t,k} \leftarrow c \times \hat{P}_{2t,k}$
26: $\quad\quad M_{2t+1,k} \leftarrow -\acute{c} \times \hat{P}_{2t+1,k}$
27: $\quad\quad M_{2t,k+m} \leftarrow c \times \hat{P}_{2t,k}$
28: $\quad\quad M_{2t+1,k+m} \leftarrow \acute{c} \times \hat{P}_{2t+1,k}$
29: $\quad$ **end for**
30: **end for**

IV. GENERATING SAMPLE ORTHOGONAL MATRICES

In order to practice our method in real world scenarios, we apply the procedure to the solutions found in the classical

Samsung Exhibit 1048, Page 6 of 12
Samsung Electronics Co., Ltd. et al. v. Genghiscomm Holdings LLC

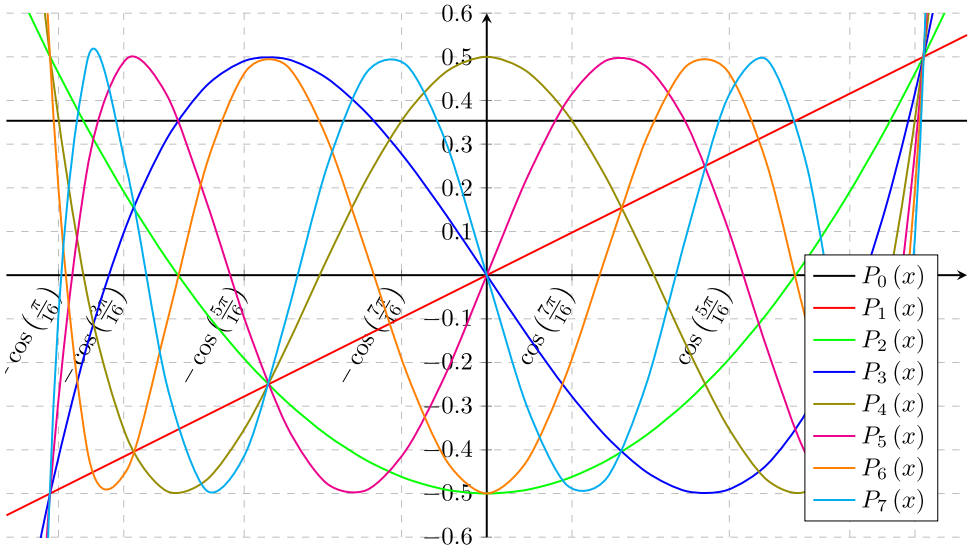

FIGURE 1. The first 8 polynomials of the DCT matrix in domain $x \in (-1, +1)$, the corresponding 8 roots as $\left\{\pm\cos\left(\frac{\pi}{16}\right), \pm\cos\left(\frac{3\pi}{16}\right), \pm\cos\left(\frac{5\pi}{16}\right), \pm\cos\left(\frac{7\pi}{16}\right)\right\}$.

expansions and reproduce those orthogonal matrices currently in wide use, as samples. As described in Section III, to generate an $n \times n$ orthogonal matrix, $n = 2m$ must be an even number. This requirement is in fact less restrictive than that of most other generating methods, where n must be a power of 2. Therefore, all the sample matrices can be generated by our method without any problem in their dimensions.

The experiments are carried out as follows. We first determine $n = 2m$ distinct values for the targeted sample matrix. In fact, among the $2m$ values, half of them are the opposites of the other half, thus only m distinct positive values are required. As discussed in Section III-B–III-C, the entire procedure can be separated into two batches, that are, (i) the *even*-numbered polynomials $P_0(x), P_2(x), \ldots, P_{n-2}(x)$, and (ii) the *odd*-numbered polynomials $P_1(x), P_3(x), \ldots, P_{n-1}(x)$, iteratively and respectively from the base cases $P_0(x)$ and $P_1(x)$. In particular, we choose only the arithmetic square roots in Algorithm 1 to simplify the results. At the end of the section, we illustrate that, by using arbitrary distinct values, we are also able to produce new and unique orthogonal matrices, not just the special values of those discovered matrices.[1]

A. 8×8 DISCRETE COSINE TRANSFORM MATRIX

To generate the $n \times n$ ($n = 8$) DCT matrix, we must first confirm the n distinct values. Since the DCTs are also closely related to the Chebyshev polynomials [19], where the coefficients of $P_1(x)$ are the roots of the n-th Chebyshev polynomial $P_n(\cos(x)) = \cos(nx)$, that is,

$$P_n(x) = \cos(n \arccos(x)) = 0. \tag{17}$$

Solving (17), we have the n roots to be

$$x_i = \cos\left(\frac{i + \frac{1}{2}}{n}\pi\right) \qquad i = 0, 1, \ldots, n-1.$$

Also by Algorithm 1, #3, we notice that the coefficients of $P_1(x)$ are also the set of n distinct values we are using to generate the matrix. Thus for the 8×8 orthogonal matrix, we have the 8 values as the roots of (17), $\pm\cos\left(\frac{\pi}{16}\right), \pm\cos\left(\frac{3\pi}{16}\right), \pm\cos\left(\frac{5\pi}{16}\right), \pm\cos\left(\frac{7\pi}{16}\right)$. By taking these values, Algorithm 1 produces an 8×8 DCT matrix as shown in Appendix A-A. In fact, the Appendix A-A can be multiplied by $64\sqrt{8}$ and rounded to Appendix A-B, which is the widely used DCT-II matrix [51] that has been employed in the video coding standard. Various sizes of DCT-II have also been embedded in the next generation video coding standard, Versatile Video Coding [52], where those DCT matrics were called `DEFINE_DCT2_PX_MATRIX` that can be found in the reference software VVC Test Model (VTM). In the transform module, all transform matrices are also projected into the integer domain to increase the computational performance. Depending on the size n of the coding block, the n^2 matrix elements are up-scaled by $64\sqrt{n}$, and then be approximated and rounded to integers, where 64 is the maximum Quantization Parameter (QP) values. Figure 1 gives the corresponding plot of the first 8 Chebyshev polynomials. The plot of the polynomials shows some features of cosine functions. The roots are also in the range of $[-1, +1]$, and all of them are the cosine values of the radians in an arithmetic sequence.

[1] We have implemented the procedures to generate the sample orthogonal matrices in: `github.com/ChanKaHou/DiscreteOrthogonalMatrices`

Samsung Exhibit 1048, Page 7 of 12
Samsung Electronics Co., Ltd. et al. v. Genghiscomm Holdings LLC

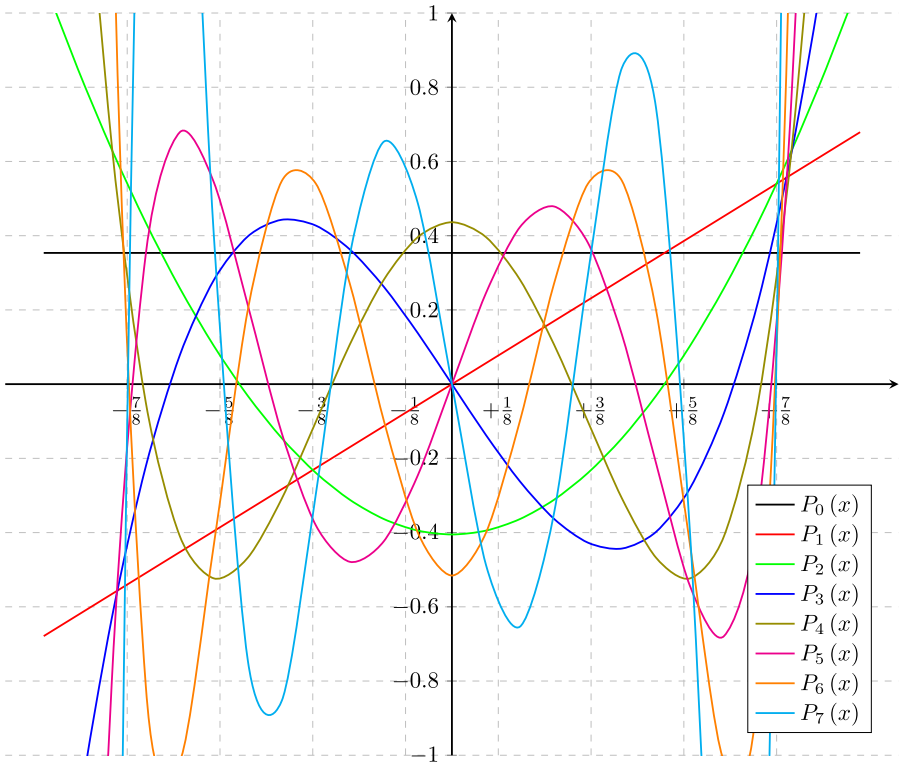

FIGURE 2. The first 8 polynomials of the DTT matrix in domain $x \in (-1, +1)$, the corresponding 8 roots as $\left\{\pm\frac{1}{8}, \pm\frac{3}{8}, \pm\frac{5}{8}, \pm\frac{7}{8}\right\}$.

B. 8×8 DISCRETE TCHEBICHEF TRANSFORM MATRIX

The Discrete Tchebichef Transform (DTT) is another widely used transform method by using the Chebyshev polynomials [11], which has as good energy compaction properties as those of the DCT, and works better for a certain class of 2D information. Because the Chebyshev polynomials are too complex, unlike in the DCT case, the roots of the n-th polynomial $P_n(x) = 0$ are difficult to obtain for setting the values to generate the matrix. However, as discussed in the DCT case, the discovered orthogonal matrices can help us determine the coefficients of polynomial $P_1(x)$, thus the values for our generation method. For example, a 4×4 DTT matrix has been discussed in [53] where the coefficients of $P_1(x)$ are

$$-\frac{3\sqrt{5}}{10}, \quad -\frac{\sqrt{5}}{10}, \quad +\frac{\sqrt{5}}{10}, \quad +\frac{3\sqrt{5}}{10},$$

which form an arithmetic sequence. We can use these values to generate the orthogonal matrix. Furthermore, consider the loop of Algorithm 1, #21–30 to normalize each row to a unit vector, the values for generating the matrix can be scaled arbitrarily. Therefore, we can use a better distributed arithmetic sequence

$$-\frac{3}{4}, \quad -\frac{1}{4}, \quad +\frac{1}{4}, \quad +\frac{3}{4}$$

in the range of $[-1, 1]$ as the generating values. As a result, we obtain the same matrix as in Appendix B-A by using Algorithm 1 with these values. Similarly, in order to generate an 8×8 DTT matrix, we must determine the 8 generating values first. Following the same principle, we use the evenly distributed arithmetic sequence of 8 values in the range of $[-1, 1]$,

$$\pm\frac{1}{8}, \quad \pm\frac{3}{8}, \quad \pm\frac{5}{8}, \quad \pm\frac{7}{8}.$$

We are able to obtain the 8×8 DTT matrix as in Appendix B-C, which is identical to the one describe in [11]. Similar to Appendix A-B, Appendix B-A and B-C can also be scaled and rounded to an integer matrix like Appendix B-B and B-D, which can be used in coding applications. Figure 2 shows the corresponding plot of the first 8 DTT polynomials. Different from the DCT case, the values to generate a DTT matrix themselves form an arithmetic sequence. Thus, for the generation of a general $2m \times 2m$ DTT matrix we should set the arithmetic sequence

$$\pm\frac{1}{2m}, \quad \pm\frac{3}{2m}, \ldots, \pm\frac{2m-1}{2m}$$

as the generating values in our method. This enables us to generate DTT matrices of arbitrarily large sizes.

C. FURTHER DISCUSSION

It may not always be possible to come up with natural discretizations as in these examples. Switching to our method, we need only to determine the generating values, then the

Samsung Exhibit 1048, Page 8 of 12
Samsung Electronics Co., Ltd. et al. v. Genghiscomm Holdings LLC

corresponding matrix can be produced accordingly. As indicated in the DCT and DTT cases, their generating values have certain patterns, of which we can make use to produce larger orthogonal matrices of the same class. In fact, our method has the advantage to accept any real numbers as the generating values, the DCT and DTT are only two well-known cases serving as the evidence of success in our practice. There are other potential sequences such as triangular numbers $\pm 1, \pm 3, \pm 6, \pm 10, \ldots$, prime numbers $\pm 2, \pm 3, \pm 5, \pm 7, \ldots$ and Fibonacci numbers $\pm 1, \pm 2, \pm 3, \pm 5, \ldots$ that can be examined further for application. The respective 8×8 orthogonal matrices from these sequences are shown in Appendix C, D and E. In the same way like Appendix A-B, all of them can be multiplied by $64\sqrt{8}$ and rounded to integer matrices for applicable applications. It is worth noting that for polynomials of high degrees, some definitions near $[-1, 1]$ classify a large percentage of the determined values as potential polynomial roots. We can scale the polynomials for optimal root arrangement. However, significantly decreasing the scaling factor will increase the energy of polynomials, always resulting in very large $c_{(k,0)}$. Although this in turn affects the weight function when dealt with in the continuous form, it is out of the scope of this work.

On the other hand, there is a weak point of our method that we only present these polynomials in the form of approximate coefficients. Although all the polynomials for the orthogonal matrices can be formulated in the accurate form like those 0 and 1-degree polynomials, it will be too complex to read and implement for the higher degree polynomials. When we have to approximate the coefficients iteratively, rounding errors must be taken into consideration. Further, our method may not support the class of discrete polynomials that are orthogonal on non-uniform lattice, such as the rotation matrix in a 3D transformer. Because a rotation matrix is not limited to $2m \times 2m$ in size and its determinant must meet an additional condition, i.e., equal to ± 1. In real world applications, the DCT and DTT have been widely use in image compression. Also, our method has the potential to apply to cryptography. For example, the user can arbitrary determine a set of values as the encryption key, and go through our algorithm to generate a unique orthogonal matrix to map the plain text into the cipher text. Using the inverse matrix will be able to decrypt the cipher text back to the original.

As we can see, with our generating method, the result polynomials and matrix are determined by the initial given values. An intuitive question is how the properties of the generated polynomials and matrix are related to the distribution of the given values. These properties can be further divided into whether they are general or application oriented. For example, when the generated orthogonal matrix is used in image compression, the property of coding efficiency is of interest. It can be found in the experiments that the distribution of the initial given values has impact on the applicability of the generated transform matrix to signals of certain frequency distribution, such as DCT fitting better for lower frequency areas. A thorough study of the relation between initial given values and the properties of outcome is a research direction of future studies, both with and without the contexts of applications.

V. CONCLUSION

In this paper, we present a general method for generating discrete orthogonal matrices of arbitrary even numbered sizes, from user determined sets of distinct positive real numbers. We give the complete induction procedure which also leads to the formal justification and the algorithm. Our method is able to generate a class of discrete polynomials that are orthogonal on uniform lattices. We have reproduced the well-known DCT and DTT matrices in terms of the corresponding positive values without using the continuous polynomials. Our method provides a shortcut to the development of undiscovered orthogonal transforms for potential applications. Invertible transformers can be generated more efficiently that are effective for sample data testing and evaluation of new ideas. The application of this method can help eliminating the need of heavy mathematics for using certain class of orthogonal matrices. The result of our practice shows the power and flexibility of this generating method compared with other methods for discrete orthogonal transformers. In addition, we show that the generated matrices have the potential to facilitate other applications and analysis.

APPENDIX A
DCT MATRIX
A. 8×8 DCT MATRIX

See (A.1), as shown at the bottom of the next page.

B. 8×8 DCT MATRIX IN INTEGER, NAMELY DEFINE_DCT2_P8_MATRIX

$$ROUND\left(\text{Appendix } A - A \times 64\sqrt{8}\right) = \begin{bmatrix} 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\ 89 & 75 & 50 & 18 & -18 & -50 & -75 & -89 \\ 84 & 35 & -35 & -84 & -84 & -35 & 35 & 84 \\ 75 & -18 & -89 & -50 & 50 & 89 & 18 & -75 \\ 64 & -64 & -64 & 64 & 64 & -64 & -64 & 64 \\ 50 & -89 & 18 & 75 & -75 & -18 & 89 & -50 \\ 35 & -84 & 84 & -35 & -35 & 84 & -84 & 35 \\ 18 & -50 & 75 & -89 & 89 & -75 & 50 & -18 \end{bmatrix} \tag{A.2}$$

APPENDIX B
DTT MATRIX
A. 4×4 DTT MATRIX

$$\begin{bmatrix} 0.5000000 & 0.5000000 & 0.5000000 & 0.5000000 \\ 0.6708204 & 0.2236068 & -0.2236068 & -0.6708204 \\ 0.5000000 & -0.5000000 & -0.5000000 & 0.5000000 \\ 0.2236068 & -0.6708204 & 0.6708204 & -0.2236068 \end{bmatrix} \tag{B.1}$$

Samsung Exhibit 1048, Page 9 of 12
Samsung Electronics Co., Ltd. et al. v. Genghiscomm Holdings LLC

B. 4×4 DTT MATRIX IN INTEGER

$$ROUND\left(\text{Appendix } B - A \times 64\sqrt{4}\right) = \begin{bmatrix} 64 & 64 & 64 & 64 \\ 86 & 29 & -29 & -86 \\ 64 & -64 & -64 & 64 \\ 29 & -86 & 86 & -29 \end{bmatrix} \tag{B.2}$$

C. 8×8 DTT MATRIX

See (B.3), as shown at the bottom of the page.

D. 8×8 DTT MATRIX IN INTEGER

$$ROUND\left(\text{Appendix } B - C \times 64\sqrt{8}\right) = \begin{bmatrix} 64 & 64 & 64 & 64 & 64 & 64 & 64 & 64 \\ 98 & 70 & 42 & 14 & -14 & -42 & -70 & -98 \\ 98 & 14 & -42 & -70 & -70 & -42 & 14 & 98 \\ 78 & -56 & -78 & -33 & 33 & 78 & 56 & -78 \\ 51 & -95 & -22 & 66 & 66 & -22 & -95 & 51 \\ 27 & -89 & 66 & 58 & -58 & -66 & 89 & -27 \\ 11 & -56 & 100 & -56 & -56 & 100 & -56 & 11 \\ 3 & -22 & 65 & -108 & 108 & -65 & 22 & -3 \end{bmatrix} \tag{B.4}$$

$$\begin{bmatrix} 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 \\ 0.49039264 & 0.41573481 & 0.27778512 & 0.09754516 & -0.09754516 & -0.27778512 & -0.41573481 & -0.49039264 \\ 0.46193977 & 0.19134172 & -0.19134172 & -0.46193977 & -0.46193977 & -0.19134172 & 0.19134172 & 0.46193977 \\ 0.41573481 & -0.09754516 & -0.49039264 & -0.27778512 & 0.27778512 & 0.49039264 & 0.09754516 & -0.41573481 \\ 0.35355339 & -0.35355339 & -0.35355339 & 0.35355339 & 0.35355339 & -0.35355339 & -0.35355339 & 0.35355339 \\ 0.27778512 & -0.49039264 & 0.09754516 & 0.41573481 & -0.41573481 & -0.09754516 & 0.49039264 & -0.27778512 \\ 0.19134172 & -0.46193977 & 0.46193977 & -0.19134172 & -0.19134172 & 0.46193977 & -0.46193977 & 0.19134172 \\ 0.09754516 & -0.27778512 & 0.41573481 & -0.49039264 & 0.49039264 & -0.41573481 & 0.27778512 & -0.09754516 \end{bmatrix} \tag{A.1}$$

$$\begin{bmatrix} 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 \\ 0.54006172 & 0.38575837 & 0.23145502 & 0.07715167 & -0.07715167 & -0.23145502 & -0.38575837 & -0.54006172 \\ 0.54006172 & 0.07715167 & -0.23145502 & -0.38575837 & -0.38575837 & -0.23145502 & 0.07715167 & 0.54006172 \\ 0.43082022 & -0.30772873 & -0.43082022 & -0.18463724 & 0.18463724 & 0.43082022 & 0.30772873 & -0.43082022 \\ 0.28203804 & -0.52378493 & -0.12087344 & 0.36262033 & 0.36262033 & -0.12087344 & -0.52378493 & 0.28203804 \\ 0.14978617 & -0.49215457 & 0.36376642 & 0.32097037 & -0.32097037 & -0.36376642 & 0.49215457 & -0.14978617 \\ 0.06154575 & -0.30772873 & 0.55391171 & -0.30772873 & -0.30772873 & 0.55391171 & -0.30772873 & 0.06154575 \\ 0.01706972 & -0.11948803 & 0.35846409 & -0.59744015 & 0.59744015 & -0.35846409 & 0.11948803 & -0.01706972 \end{bmatrix} \tag{B.3}$$

$$\begin{bmatrix} 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 \\ 0.58520574 & 0.35112344 & 0.17556172 & 0.05852057 & -0.05852057 & -0.17556172 & -0.35112344 & -0.58520574 \\ 0.57732044 & -0.00454583 & -0.25002066 & -0.32275395 & -0.32275395 & -0.25002066 & -0.00454583 & 0.57732044 \\ 0.38929157 & -0.44380688 & -0.36478867 & -0.13570834 & 0.13570834 & 0.36478867 & 0.44380688 & -0.38929157 \\ 0.20332262 & -0.58495156 & 0.04304563 & 0.33858331 & 0.33858331 & 0.04304563 & -0.58495156 & 0.20332262 \\ 0.07731044 & -0.42115272 & 0.49603067 & 0.26571989 & -0.26571989 & -0.49603067 & 0.42115272 & -0.07731044 \\ 0.01900050 & -0.18113808 & 0.55734795 & -0.39521037 & -0.39521037 & 0.55734795 & -0.18113808 & 0.01900050 \\ 0.00306922 & -0.04876649 & 0.30010145 & -0.63839763 & 0.63839763 & -0.30010145 & 0.04876649 & -0.00306922 \end{bmatrix} \tag{C.1}$$

$$\begin{bmatrix} 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 \\ 0.53066863 & 0.37904902 & 0.22742941 & 0.15161961 & -0.15161961 & -0.22742941 & -0.37904902 & -0.53066863 \\ 0.54924560 & 0.06550636 & -0.25698647 & -0.35776548 & -0.35776548 & -0.25698647 & 0.06550636 & 0.54924560 \\ 0.43765512 & -0.25996056 & -0.38500488 & -0.3043842 & 0.3043842 & 0.38500488 & 0.25996056 & -0.43765512 \\ 0.26765175 & -0.56758752 & -0.02499814 & 0.32493391 & 0.32493391 & -0.02499814 & -0.56758752 & 0.26765175 \\ 0.16311855 & -0.52457442 & 0.24653144 & 0.37072397 & -0.37072397 & -0.24653144 & 0.52457442 & -0.16311855 \\ 0.04113166 & -0.22034819 & 0.55527745 & -0.37606092 & -0.37606092 & 0.55527745 & -0.22034819 & 0.04113166 \\ 0.01552862 & -0.11646468 & 0.48915166 & -0.49691598 & 0.49691598 & -0.48915166 & 0.11646468 & -0.01552862 \end{bmatrix} \tag{D.1}$$

Samsung Exhibit 1048, Page 10 of 12
Samsung Electronics Co., Ltd. et al. v. Genghiscomm Holdings LLC

$$\begin{bmatrix} 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 & 0.35355339 \\ 0.56613852 & 0.33968311 & 0.22645541 & 0.1132277 & -0.1132277 & -0.22645541 & -0.33968311 & -0.56613852 \\ 0.58246001 & -0.02864557 & -0.21961607 & -0.33419837 & -0.33419837 & -0.21961607 & -0.02864557 & 0.58246001 \\ 0.41600392 & -0.36846062 & -0.37440353 & -0.2258307 & 0.2258307 & 0.37440353 & 0.36846062 & -0.41600392 \\ 0.18773874 & -0.54375769 & -0.0518894 & 0.40790836 & 0.40790836 & -0.0518894 & -0.54375769 & 0.18773874 \\ 0.07984321 & -0.47485699 & 0.30256374 & 0.42022742 & -0.42022742 & -0.30256374 & 0.47485699 & -0.07984321 \\ 0.02223738 & -0.28019101 & 0.56927697 & -0.31132334 & -0.31132334 & 0.56927697 & -0.28019101 & 0.02223738 \\ 0.00727855 & -0.1528496 & 0.46582736 & -0.50949867 & 0.50949867 & -0.46582736 & 0.1528496 & -0.00727855 \end{bmatrix} \tag{E.1}$$

APPENDIX C
8×8 DISCRETE TRIANGULAR MATRIX

See (C.1), as shown at the bottom of the previous page.

APPENDIX D
8×8 DISCRETE PRIME MATRIX

See (D.1), as shown at the bottom of the previous page.

APPENDIX E
8×8 DISCRETE FIBONACCI MATRIX

See (E.1), as shown at the top of the page.

REFERENCES

[1] A. Al-Haj, "Combined DWT-DCT digital image watermarking," *J. Comput. Sci.*, vol. 3, no. 9, pp. 740–746, Sep. 2007, doi: 10.3844/jcssp.2007.740.746.

[2] F. Arman, A. Hsu, and M. Chiu, "Image processing on compressed data for large video databases," in *Proc. 1st ACM Int. Conf. Multimedia*, J. J. Garcia-Luna-Aceves and P. V. Rangan, Eds. Anaheim, CA, USA: ACM Press, Aug. 1993, pp. 267–272, doi: 10.1145/166266.166297.

[3] P. Garg, L. Dodeja, Priyanka, and M. Dave, "Hybrid color image watermarking algorithm based on DSWT-DCT-SVD and Arnold transform," in *Advances in Signal Processing and Communication* (Lecture Notes in Electrical Engineering). Singapore: Springer, Nov. 2018, pp. 327–336, doi: 10.1007/978-981-13-2553-3_31.

[4] D. M. Monro, S. Rakshit, and D. Zhang, "DCT-based iris recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 586–595, Apr. 2007, doi: 10.1109/TPAMI.2007.1002.

[5] S. S. Harakannanavar, C. R. Prashanth, S. Patil, and K. B. Raja, "Face recognition based on SWT, DCT and LTP," in *Integrated Intelligent Computing, Communication and Security*. Singapore: Springer, Sep. 2018, pp. 565–573, doi: 10.1007/978-981-10-8797-4_57.

[6] Z. He, W. Lu, W. Sun, and J. Huang, "Digital image splicing detection based on Markov features in DCT and DWT domain," *Pattern Recognit.*, vol. 45, no. 12, pp. 4292–4299, 2012, doi: 10.1016/j.patcog.2012.05.014.

[7] S. V. Dhavale, "DWT and DCT based robust iris feature extraction and recognition algorithm for biometric personal identification," *Int. J. Comput. Appl.*, vol. 40, no. 7, pp. 33–37, Feb. 2012, doi: 10.5120/4978-7235.

[8] S. H. Abdulhussain, S. A. R. Al-Haddad, M. I. Saripan, B. M. Mahmmod, and A. Hussien, "Fast temporal video segmentation based on Krawtchouk-Tchebichef moments," *IEEE Access*, vol. 8, pp. 72347–72359, 2020, doi: 10.1109/ACCESS.2020.2987870.

[9] W. A. Jassim and P. Raveendran, "Face recognition using discrete Tchebichef-Krawtchouk transform," in *Proc. IEEE Int. Symp. Multimedia*, Irvine, CA, USA, Dec. 2012, pp. 120–127, doi: 10.1109/ISM.2012.31.

[10] S. H. Abdulhussain, B. M. Mahmmod, M. A. Naser, M. Q. Alsabah, R. Ali, and S. A. R. Al-Haddad, "A robust handwritten numeral recognition using hybrid orthogonal polynomials and moments," *Sensors*, vol. 21, no. 6, p. 1999, Mar. 2021, doi: 10.3390/s21061999.

[11] R. Mukundan, S. H. Ong, and P. A. Lee, "Image analysis by Tchebichef moments," *IEEE Trans. Image Process.*, vol. 10, no. 9, pp. 1357–1364, Sep. 2001, doi: 10.1109/83.941859.

[12] S. H. Abdulhussain, A. R. Ramli, S. A. R. Al-Haddad, B. M. Mahmmod, and W. A. Jassim, "On computational aspects of Tchebichef polynomials for higher polynomial order," *IEEE Access*, vol. 5, pp. 2470–2478, 2017, doi: 10.1109/ACCESS.2017.2669218.

[13] P. T. Yap, R. Paramesran, and S. H. Ong, "Image analysis using Hahn moments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 11, pp. 2057–2062, Nov. 2007, doi: 10.1109/TPAMI.2007.70709.

[14] A. M. Abdul-Hadi, S. H. Abdulhussain, and B. M. Mahmmod, "On the computational aspects of Charlier polynomials," *Cogent Eng.*, vol. 7, no. 1, Jan. 2020, Art. no. 1763553, doi: 10.1080/23311916.2020.1763553.

[15] P. T. Yap, R. Paramesran, and S.-H. Ong, "Image analysis by Krawtchouk moments," *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1367–1377, Nov. 2003, doi: 10.1109/TIP.2003.818019.

[16] S. H. Abdulhussain, A. R. Ramli, S. A. R. Al-Haddad, B. M. Mahmmod, and W. A. Jassim, "Fast recursive computation of Krawtchouk polynomials," *J. Math. Imag. Vis.*, vol. 60, no. 3, pp. 285–303, 2018, doi: 10.1007/s10851-017-0758-9.

[17] S. Liao, A. Chiang, Q. Lu, and M. Pawlak, "Chinese character recognition via Gegenbauer moments," in *Proc. Object Recognit. Supported User Interact. Service Robots*, Quebec, QC, Canada, 2002, pp. 485–488, doi: 10.1109/ICPR.2002.1047982.

[18] T. Xia, H. Zhu, H. Shu, P. Haigron, and L. Luo, "Image description with generalized pseudo-Zernike moments," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 24, no. 1, pp. 50–59, Jan. 2007. [Online]. Available: http://josaa.osa.org/abstract.cfm?URI=josaa-24-1-50

[19] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. 23, no. 1, pp. 90–93, Jan. 1974, doi: 10.1109/T-C.1974.223784.

[20] J. Song, Z. Xiong, X. Liu, and Y. Liu, "PVH-3DDCT: An algorithm for layered video coding and transmission," in *Proc. 4th Int. Conf./Exhib. High Perform. Comput. Asia–Pacific Region*, vol. 2, 2000, pp. 700–703, doi: 10.1109/HPC.2000.843529.

[21] R. Mukundan, "Improving image reconstruction accuracy using discrete orthonormal moments," in *Proc. Int. Conf. Imag. Sci., Syst. Technol. (CISST)*, vol. 1, H. R. Arabnia and Y. Mun, Eds. Las Vegas, NV, USA: CSREA Press, Jun. 2003, pp. 287–291.

[22] K. W. See, K. S. Loke, P. A. Lee, and K. F. Loe, "Image reconstruction using various discrete orthogonal polynomials in comparison with DCT," *Appl. Math. Comput.*, vol. 193, no. 2, pp. 346–359, Nov. 2007, doi: 10.1016/j.amc.2007.03.080.

[23] M. Abramowitz, I. A. Stegun, and R. H. Romer, "Handbook of mathematical functions with formulas, graphs, and mathematical tables," *Amer. J. Phys.*, vol. 56, no. 10, p. 958, Oct. 1988, doi: 10.1119/1.15378.

[24] K. Xu, "The Chebyshev points of the first kind," *Appl. Numer. Math.*, vol. 102, pp. 17–30, Apr. 2016, doi: 10.1016/j.apnum.2015.12.002.

[25] J. C. Mason and D. C. Handscomb, *Chebyshev Polynomials*. Boca Raton, FL, USA: CRC Press, 2003. [Online]. Available: http://www.crcnetbase.com/isbn/9780849303555

[26] W. Schweizer, "Legendre polynomials and Legendre functions," in *Special Functions in Physics with MATLAB*. Springer, 2021, pp. 33–63, doi: 10.1007/978-3-030-64232-7_3.

[27] S. H. Abdulhussain and B. M. Mahmmod, "Fast and efficient recursive algorithm of Meixner polynomials," *J. Real-Time Image Process.*, to be published, doi: 10.1007/s11554-021-01093-z.

[28] B. M. Mahmmod, A. M. Abdul-Hadi, S. H. Abdulhussain, and A. Hussien, "On computational aspects of Krawtchouk polynomials for high orders," *J. Imag.*, vol. 6, no. 8, p. 81, Aug. 2020, doi: 10.3390/jimaging6080081.

Samsung Exhibit 1048, Page 11 of 12
Samsung Electronics Co., Ltd. et al. v. Genghiscomm Holdings LLC

[29] K. Jones, "The discrete Hartley transform," in *The Regularized Fast Hartley Transform*. Amsterda, The Netherlands: Springer, Dec. 2009, pp. 27–40, doi: 10.1007/978-90-481-3917-0_3.

[30] R. Roy, *Orthogonal Polynomials*. Cambridge, U.K.: Cambridge Univ. Press, 2011, ch. 28, pp. 574–598, doi: 10.1017/cbo9780511844195.029.

[31] C. Markett, "Linearization of the product of symmetric orthogonal polynomials," *Constructive Approximation*, vol. 10, no. 3, pp. 317–338, Sep. 1994, doi: 10.1007/bf01212564.

[32] J. F. van Diejen, "Properties of some families of hypergeometric orthogonal polynomials in several variables," *Trans. Amer. Math. Soc.*, vol. 351, no. 1, pp. 233–270, 1999, doi: 10.1090/s0002-9947-99-02000-0.

[33] Y. Xu, "On discrete orthogonal polynomials of several variables," *Adv. Appl. Math.*, vol. 33, no. 3, pp. 615–632, Oct. 2004, doi: 10.1016/j.aam.2004.03.002.

[34] Y. Xu, "Second-order difference equations and discrete orthogonal polynomials of two variables," *Int. Math. Res. Notices*, vol. 2005, no. 8, pp. 449–475, 2005, doi: 10.1155/IMRN.2005.449.

[35] J. Rodal, I. Area, and E. Godoy, "Orthogonal polynomials of two discrete variables on the simplex," *Integral Transforms Special Functions*, vol. 16, no. 3, pp. 263–280, Apr. 2005, doi: 10.1080/1065246042000272036.

[36] R. Koekoek, P. A. Lesky, and R. F. Swarttouw, *Hypergeometric Orthogonal Polynomials and Their Q-Analogues*. Berlin, Germany: Springer, 2010, doi: 10.1007/978-3-642-05014-5.

[37] J. Rodal, I. Area, and E. Godoy, "Structure relations for monic orthogonal polynomials in two discrete variables," *J. Math. Anal. Appl.*, vol. 340, no. 2, pp. 825–844, Apr. 2008, doi: 10.1016/j.jmaa.2007.09.003.

[38] J. Rodal, I. Area, and E. Godoy, "Linear partial difference equations of hypergeometric type: Orthogonal polynomial solutions in two discrete variables," *J. Comput. Appl. Math.*, vol. 200, no. 2, pp. 722–748, Mar. 2007, doi: 10.1016/j.cam.2006.01.027.

[39] E. H. Doha and H. M. Ahmed, "Recurrences and explicit formulae for the expansion and connection coefficients in series of classical discrete orthogonal polynomials," *Integral Transforms Special Functions*, vol. 17, no. 5, pp. 329–353, May 2006, doi: 10.1080/10652460500422270.

[40] H. M. Ahmed, "Recurrence relation approach for expansion and connection coefficients in series of classical discrete orthogonal polynomials," *Integral Transforms Special Functions*, vol. 20, no. 1, pp. 23–34, Jan. 2009, doi: 10.1080/10652460801936747.

[41] E. Godoy, A. Ronveaux, A. Zarzo, and I. Area, "Minimal recurrence relations for connection coefficients between classical orthogonal polynomials: Continuous case," *J. Comput. Appl. Math.*, vol. 84, no. 2, pp. 257–275, Oct. 1997, doi: 10.1016/s0377-0427(97)00137-4.

[42] A. Zarzo, I. Area, E. Godoy, and A. Ronveaux, "Results for some inversion problems for classical continuous and discrete orthogonal polynomials," *J. Phys. A, Math. Gen.*, vol. 30, no. 3, pp. L35–L40, Feb. 1997, doi: 10.1088/0305-4470/30/3/002.

[43] A. Ronveaux, A. Zarzo, and E. Godoy, "Recurrence relations for connection coefficients between two families of orthogonal polynomials," *J. Comput. Appl. Math.*, vol. 62, no. 1, pp. 67–73, Aug. 1995, doi: 10.1016/0377-0427(94)00079-8.

[44] R. Álvarez-Nodarse, J. Arvesú, and R. J. Yáñez, "On the connection and linearization problem for discrete hypergeometric Q-polynomials," *J. Math. Anal. Appl.*, vol. 257, no. 1, pp. 52–78, May 2001, doi: 10.1006/jmaa.2000.7302.

[45] P. Woźny, "Recurrence relations for the coefficients of expansions in classical orthogonal polynomials of a discrete variable," *Applicationes Mathematicae*, vol. 30, no. 1, pp. 89–107, 2003, doi: 10.4064/am30-1-6.

[46] H. S. Cohl, "On a generalization of the generating function for Gegenbauer polynomials," *Integral Transforms Special Functions*, vol. 24, no. 10, pp. 807–816, Oct. 2013, doi: 10.1080/10652469.2012.761613.

[47] H. S. Cohl, C. MacKenzie, and H. Volkmer, "Generalizations of generating functions for hypergeometric orthogonal polynomials with definite integrals," *J. Math. Anal. Appl.*, vol. 407, no. 2, pp. 211–225, Nov. 2013, doi: 10.1016/j.jmaa.2013.04.067.

[48] D. Day and L. Romero, "Roots of polynomials expressed in terms of orthogonal polynomials," *SIAM J. Numer. Anal.*, vol. 43, no. 5, pp. 1969–1987, Jan. 2005, doi: 10.1137/040609847.

[49] T. S. Chihara, *An Introduction to Orthogonal Polynomials* (Dover Books on Mathematics). Mineola, NY, USA: Dover, 2011.

[50] A. F. Nikiforov, V. B. Uvarov, and S. K. Suslov, "Classical orthogonal polynomials of a discrete variable," in *Classical Orthogonal Polynomials of a Discrete Variable*. Berlin, Germany: Springer, 1991, pp. 18–54, doi: 10.1007/978-3-642-74748-9_2.

[51] X. Shao and S. G. Johnson, "Type-II/III DCT/DST algorithms with reduced number of arithmetic operations," *Signal Process.*, vol. 88, no. 6, pp. 1553–1564, 2008, doi: 10.1016/j.sigpro.2008.01.004.

[52] M. Wien and B. Bross, "Versatile video coding–algorithms and specification," in *Proc. IEEE Int. Conf. Vis. Commun. Image Process. (VCIP)*, Macau, China, Dec. 2020, pp. 1–3, doi: 10.1109/VCIP49819.2020.9301820.

[53] K. Nakagaki and R. Mukundan, "A fast 4×4 forward discrete Tchebichef transform algorithm," *IEEE Signal Process. Lett.*, vol. 14, no. 10, pp. 684–687, Oct. 2007, doi: 10.1109/LSP.2007.898331.

KA-HOU CHAN received the master's degree from the Faculty of Science and Technology, University of Macau, in 2015, and the bachelor's degree in computer science from Macau University of Science and Technology, in 2009. He is currently pursuing the Ph.D. degree with the School of Applied Sciences, Macao Polytechnic Institute. His research interests include algorithm analysis and optimization, parallel computing, video coding, neural networks, and computer graphics.

WEI KE received the Ph.D. degree from the School of Computer Science and Engineering, Beihang University. He is currently an Associate Professor with the Computing Program, Macao Polytechnic Institute. His research interests include programming languages, image processing, computer graphics, tool support for object-oriented, and component-based engineering and systems. His research interests also include the design and implementation of open platforms for applications of computer graphics and pattern recognition, including programming tools, environments, and frameworks.

SIO-KEI IM (Member, IEEE) received the bachelor's degree in computer science and the master's degree in enterprise information system from the King's College, University of London, U.K., in 1998 and 1999, respectively, and the Ph.D. degree in electronic engineering from the Queen Mary University of London (QMUL), U.K., in 2007. He gained the position of a Lecturer within the Computing Program, Macao Polytechnic Institute (MPI), in 2001. In 2005, he became the Operations Manager of the MPI–QMUL Information Systems Research Center jointly operated by MPI and QMUL, where he carried out signal processing work. He was promoted to a Professor with Macao Polytechnic Institute, in 2015. He was a Visiting Scholar with the School of Engineering, University of California at Los Angeles (UCLA), Los Angeles, and an Honorary Professor at The Open University of Hong Kong.

Samsung Exhibit 1048, Page 12 of 12
Samsung Electronics Co., Ltd. et al. v. Genghiscomm Holdings LLC