Algorithms of the $q2^r \times q2^r$-point 2-D Discrete Fourier Transform

Artyom M. Grigoryan and Sos S. Agaian

Department of Electrical and Computer Engineering
University of Texas at San Antonio

Outline

- Abstract
- Introduction
- Tensor transform of the image
  - Splitting-signals
  - Complexity of the algorithm
  - Redundancy of the tensor representation
  - Comparison with the existent algorithms

    "A fast algorithm with less operations for length-$N=q\times 2^m$ *DFTs*"

    K.Li, W.Zheng, and K.Li, IEEE SP, vol. 63, no 3, February 2015
- Conclusion
- References

Abstract

Two methods of calculation of the 2-D DFT are analyzed.

- The $q2^r \times q2^r$-point 2-D DFT can be calculated by the traditional column-row method with $2(q2^r)$ 1-D DFTs, and we also propose the fast algorithm which splits each 1-D DFT by the short transforms by means of the fast paired transforms.
- The $q2^r \times q2^r$-point 2-D DFT can be calculated by the tensor or paired representations of the image, when the image is represented as a set of 1-D signals which define the 2-D transform in the different subsets of frequency-points and they all together cover the complete set of frequencies. In this case, the splittings of the $q2^r \times q2^r$-point 2-D DFT are performed by the 2-D discrete tensor or paired transforms, respectively, which lead to the calculation with a minimum number of 1-D DFTs.

Inroduction: 2-D Transform Splitting

- In work, we use the concept of partitions revealing transforms for computing the 2-D DFT of order $q2^r \times q2^r$, where r>1 and q is a positive odd number.

  By means of such partitions, the 2-D discrete Fourier transfrom transform can be split into a number of short transforms, or 1-D M-point DFTs where $M \leq q2^r$.

  In the 1-D case, the partitions determine fast transformations that split the $q2^r$-point DFT into a set of N_k-point transforms, where $k=1:n$ and $N_1+\ldots+N_n = q2^r$, and minimizes the computational complexity of the $q2^r$-point DFT.

  In matrix form, the splitting can be written as

$$[\mathcal{F}_{q2^r}] = \text{diag}\{[\mathcal{F}_{N_1}], [\mathcal{F}_{N_2}], \ldots, [\mathcal{F}_{N_k}]\}\,[\bar{W}][\chi'_{q2^r}]$$

  where $[W]$ is a diagonal matrix with twiddle coefficients.
  We name these splitting transformations be paired χ'_{q2^r}

2D Discrete tensor and paired transforms

- 2-D DFT of the image $f_{n,m}$ of size $N\times N=q2^r\times q2^r$ is

$$F_{p,s} = \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} f_{n,m} W^{np+ms}, \quad p,s = 0:(N-1).$$

The kernel of this complex transformation $W = W_N = \exp(-2\pi j/N)$.

1. The $q2^r\times q2^r$-point 2-D DFT can be calculated by the column-row method with $2(q2^r)$ 1-D DFTs, each of which can be

- ❑ split by the short transforms, by means of the1-D paired transforms
- ❑ calculated by the scaled DFT proposed in [1] can also be used for calculating the $q2^r$-point DFT, when arithmetic complexity can be reduced to $(2N-4r)$ real multiplications

  [1] K.Li, W.Zheng, and K.Li, *IEEE SP*, 63(3), Feb. 2015.

2. Another and more effective algorithm of calculation of the $q2^r\times q2^r$-point 2-D DFT is based on the splitting by the 2-D tensor or paired transform which leads to the calculation with a minimum number of 1-D DFTs.

Tensor Representation of the (N×N) Image

The tensor representation of an image $f_{n,m}$ which is the (2-D)-frequency-and-(1-D)-time representation, the image is described by a set of 1-D splitting-signals of length N each

$$\chi : \{f_{n,m}\} \to \{f_{T_{p,s}} = \{f_{p,s,t}; t = 0 : (N-1)\}\}_{(p,s)\in J_{N,N}}.$$

The components of the signals are the ray-sums of the image along the parallel lines

$$f_{p,s,t} = \sum_{(n,m)\in X} \{f_{n,m};\ np + ms = t \bmod N\}.$$

Each splitting-signals defines 2-D DFT at N frequency-points of the set

$$T_{p,s} = \{(kp \bmod N, ks \bmod N);\ k = 0 : (N-1)\}$$

on the cartesian lattice $X = X_{N,N} = \{(n,m);\ n,m = 0,1,\ldots,(N-1)\}$

$$F_{kp \bmod N, ks \bmod N} = \sum_{t=0}^{N-1} f_{p,s,t} W_N^{kt}, \quad k = 0 : (N-1).$$

Example: 768×768-point 2-D DFT

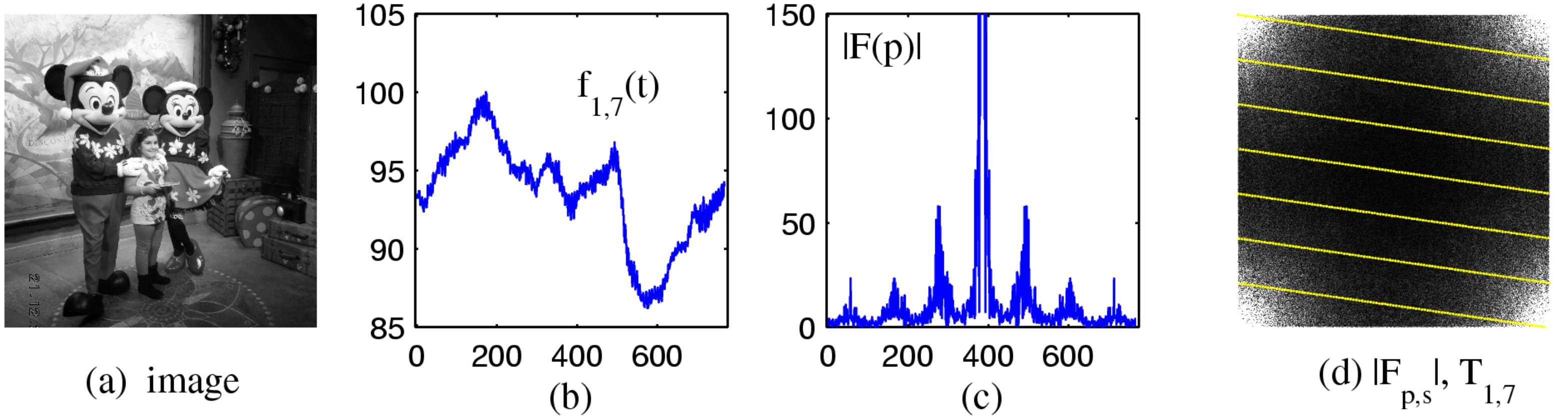


Figure 1. (a) The image, (b) splitting-signal for (1,7), (c) the magnitude of the shifted to the middle 1-D DFT of the signal, and (d) the 2-D DFT of the image with 768 frequency-points of $T_{1,7}$.

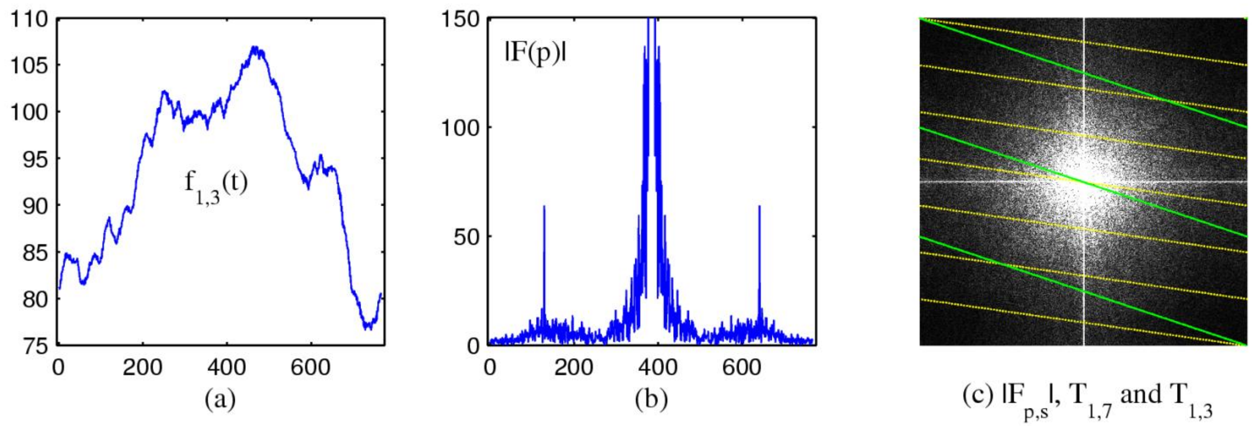


Figure 2. (a) The splitting-signal for (1,3), (b) 1-D DFT of the signal, and (c) the 2-D DFT of the image with the frequency-points of the sets $T_{1,3}$ and $T_{1,7}$.

Set of generators (p,s) in TR of Images

- Let The set $J_{N,N}$ of frequency-points (p,s), or generators, of the splitting-signals is selected in a way that covers the Cartesian lattice $X_{N,N} = \{(p,s);\ p,s = 0{:}(N-1)\}$ with a minimum number of subsets $T_{p,s}$. In other words, an irreducible covering of the Cartesian lattice is used for a certain set of generators $J_{N,N}$ in $X_{N,N}$.

$$\sigma = \sigma_{N,N} = \left(T_{p,s}\right)_{(p,s)\in J_{N,N}}$$

Example: $N=20=5(2^2)$ when $q=5$ and $r=2$, Figure 3 shows the incomplete covering of the lattice $X_{20,20}$ by *21* sets Tp,s in part a.

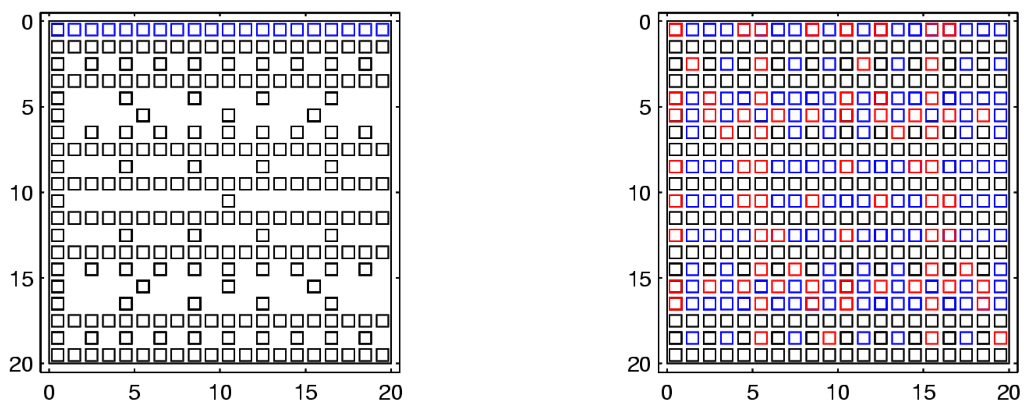


Figure 3. The set of 21 subsets of the covering of the lattice 24.

Set of generators (p,s) in TR of Images

- Case 1: $q=1$ when $N=2^r$. The set of generators contains $3N/2$ elements and can be defined as

$$J_{N,N} = \{(1,s); s=0:(N-1)\} \cup \{(2p,1); p=0:(N/2-1)\}.$$

- Case 2: q is prime and $r=0$, $N=q2^r=q$. The set of generators can be defined as

$$J_{N,N} = \{(1,s); s=0:(N-1)\} \cup \{(0,1)\}.$$

- General case: q is prime and $r\geq 0$, $N=q2^r$.
  The irreducible covering (Tps) of the Cartesian lattice $X_{N,N}$ has the cardinality

$$c(N) = card\,\sigma_{N,N} = 2N - \varphi(N) + \sum_{p\in B_N} \beta(p).$$

$$J_{N,N} = \bigcup_{p_2=0}^{N-1} (1,p_2) \bigcup \Big(\bigcup_{p_1\in B_N\cup 0} (p_1,1) \Big) \cap \Big(\bigcup_{g.c.d.(p_1,p_2)=1,\; p_1p_2\leq N} \{(p_1,p_2);\; p_1,p_2 \in B_N\} \Big).$$

Here we denote by $\varphi(N)$ the Euler function, i.e., the number of the positive integers which are smaller than N and coprime with N. B_N is the set $B_N = \{n \in X_N;\; g.c.d.(n,N) > 1\}$, β(p) is the number of the elements s in B_N, that are coprime with p and such that $ps<N$.

Set of generators (p,s) in TR of Images

- Case 3: r=*1* when *N=2q*. The set of generators contains can be defined as

$$J_{2q,2q} = \bigcup_{p_2=0}^{2q-1} (1, p_2) \cup \Big(\bigcup_{g.c.d.(p_1,2q)>1,\, p_1=0} (p_1, 1) \Big) \cup \{(2, q), (q, 2)\}.$$

To calculate the N×N-point discrete Fourier transform, it is sufficient to fulfill

$$c(2q) = 2q + (q + 1) + 2 = 3(q + 1)$$

2q-point 1D DFTs of splitting-signals $\{f_{p,s,t};\ t=0:(2q-1)\}$ generated by the set $J_{2q,2q}$

Example 2: q = 131. The 262 × 262-point DFT uses 396 262-point DFTs, and the column-row method uses 524 such 1-D DFTs. The tensor representation allows for reducing the number of the 1-D DFTs by 524−396=128.

$q = 5$	10×10-point DFT	18 (versus 20) 10-point DFTs
$q = 7$	14×14-point DFT	24 (versus 28) 14-point DFTs
$q = 9$	18×18-point DFT	30 (versus 36) 18-point DFTs
$q = 13$	26×26-point DFT	36 (versus 52) 26-point DFTs
$q = 17$	34×34-point DFT	54 (versus 68) 34-point DFTs
$q = 21$	42×42-point DFT	66 (versus 84) 42-point DFTs

The 1-D $q2^r$-point DFT

- We consider the paired algorithm for computing the N-point DFT,

$$F_p = (\mathcal{F}_N \circ f)_p = \sum_{n=0}^{N-1} f_n W^{np} = \sum_{t=0}^{N-1} f_{p,t} W^t, \quad p = 0 : (N-1),$$

$$f_{p,t} = \sum_n \{f_n;\ \overline{np} = t\} = \sum_n \{f_n;\ np = t \bmod N\}.$$

Paired representation (*L>1 a factor of N*):

$$p: \ \{f_n;\ n = 0 : (N-1)\} \rightarrow \{f'_{p,t};\ t = 0 : (N/L - 1)\}$$

$$f'_{p,t} = f'_{p,t;L} = \chi'_{p,t;L} \circ f = \sum_{k=0}^{L-1} f_{p,t+kN/L} W_L^k,$$

$$F_{\overline{(Lm+1)p}} = \sum_{t=0}^{N/L-1} (f'_{p,t} W^t) W_{N/L}^{mt}, \quad m = 0 : (N/L - 1).$$

$$T' = T'_p = T'_{p;L} = \{(Lm+1)p \bmod N;\ m = 0 : (N/L - 1)\}.$$

We need compose a partition $\sigma'_N = (T')$ of the set $X_N = \{0, 1, \ldots, N-1\}$ to obtain a splitting of the N-point DFT by small 1-D DFTs over the splitting-signals.

The 1-D $q2^r$-point DFT

- The following partition of the set of all frequency-points takes place

$$\sigma'_N = (T'_{1;2}, T'_{2;2}, T'_{4;2}, T'_{8;2}, \ldots, T'_{2^r;2})$$

The N-point DFT can be reduced to transforms $\{F_{N/2}, F_{N/4}, \ldots, F_{N/2^r}, F_q\}$,

$$[\mathcal{F}_N] = \left(\bigoplus_{n=0}^{r-1} [\mathcal{F}_{L_n}] \oplus [\mathcal{F}_q]\right)[\overline{W}][\chi'_N], \quad L_n = N/2^{n+1}$$

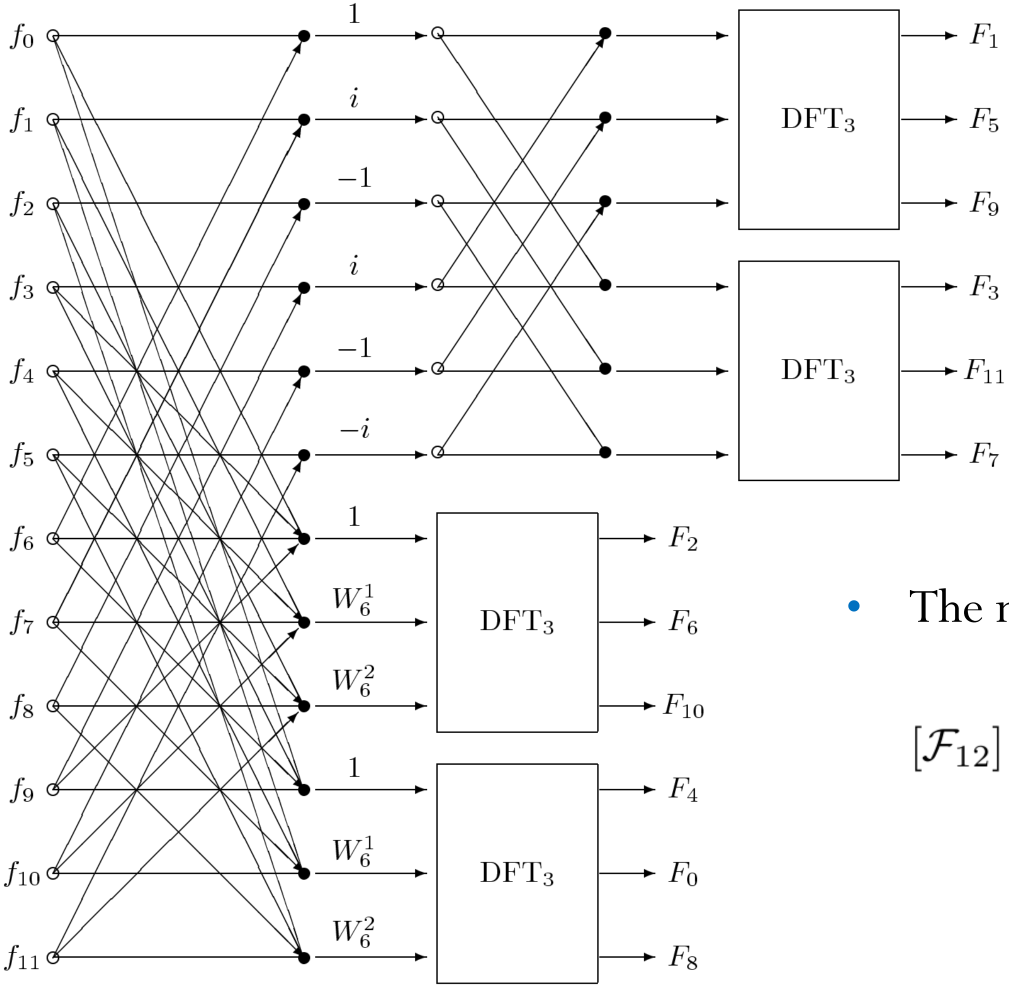
$$[\overline{W}] = \bigoplus_{n=0}^{r} \operatorname{diag}\left\{1, W^1_{2L_n}, W^2_{2L_n}, \ldots, W^{L_n-1}_{2L_n}\right\}, \quad L_r = q.$$

The number of multiplications required to compute the N-point DFT is less than

$$M_{q2^r} = 2^r(M_q - 1) + q(r-1)2^{r-1} + 2q,$$

where M_q stands for the number of multiplications in the q-point DFT.

Example (q=3, r=2): The 1-D *12*-point DFT



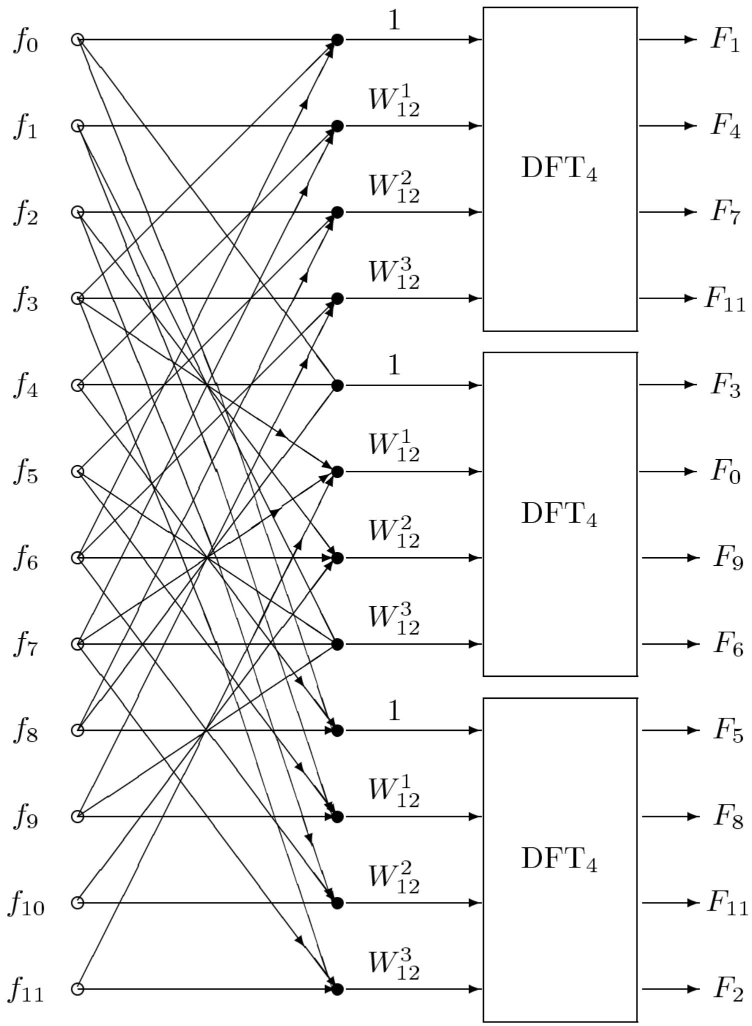
$$W_6^1 = 1/2 - i\sqrt{3}/2$$

$$W_6^2 = -1/2 + i\sqrt{3}/2$$

- The matrix of the 12-point DFT

$$[\mathcal{F}_{12}] = \left(\bigoplus_{1}^{4} [\mathcal{F}_3]\right)[\chi_{12}^2][\overline{W}^3][\chi'_{12}]$$

Example (q=3, r=2): The 1-D *12*-point DFT



- The matrix of the 12-point DFT

$$[\mathcal{F}_{12}] = \left(\bigoplus_{n=1}^{3}[\mathcal{F}_4]\right)\left(\bigoplus_{1}^{3}\operatorname{diag}\left\{1, W_{12}^1, W_{12}^2, i\right\}\right)[\chi'_{12;3}]$$

$$\begin{bmatrix}
1 & 0 & 0 & 0 & W & 0 & 0 & 0 & W^2 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & W & 0 & 0 & 0 & W^2 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & W & 0 & 0 & 0 & W^2 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & W & 0 & 0 & 0 & W^2 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & W^2 & 0 & 0 & 0 & W^2 & 0 & 0 & 0 & W^2 \\
0 & 0 & W & 0 & 0 & 0 & W & 0 & 0 & 0 & W & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & W^2 & 0 & 0 & 0 & W & 0 & 0 & 0 \\
0 & W & 0 & 0 & 0 & W^2 & 0 & 0 & 0 & W & 0 & 0 \\
0 & 0 & W^2 & 0 & 0 & 0 & W & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & W^2 & 0 & 0 & 0 & W
\end{bmatrix}$$

$$W = W_3 = -1/2 + i\sqrt{3}/2$$

$$W^2 = -1/2 - i\sqrt{3}/2$$

Complexity: Multiplications for the 2-D DFT

The number of operations of multiplication for $q2^r \times q2^r$-point DFT can be estimated as

- 1. The column-row method:

$$M'_{q2^r,q2^r} = 2(q2^r) \times M'_{q2^r}$$

- 2. The tensor transform-based method:

$$M'_{q2^r,q2^r} = c(q2^r) \times M'_{q2^r} \qquad c(q2^r) < q2^{r+1}.$$

The number of operations of multiplications for the $q2^r$-point DFT:

$$M'_{2^r} = 2^{r-1}(r-3)+2. \qquad M'_{2^r3} \leq 2^{r-1}(3r-3)+6$$

$$M'_{2^r5} \leq 2^{r-1}(5r+13)+10 \qquad M'_{2^r7} \leq 2^{r-1}(7r+23)+14.$$

Conclusion

We presented the concept of partitions revealing transforms for computing the 2-D DFT of order $q2^r \times q2^r$, where $r>1$ and q is odd number greater than 1.

- When the 2-D $q2^r \times q2^r$-point DFT is calculated by the column-rowmethod with $2(q2^r)$ 1-D DFTs, the fast algorithms of the 1-D DFTs of order $q2^r$ are required. We propose the fast algorithms which splits each 1-D DFT by the short transforms by using the fast 1-D paired transforms.
- The 2-D $q2^r \times q2^r$-point DFT can also be calculated by using the tensor or paired representations of the image, when the image is represented as a set of 1-D signals which define the 2-D transform in the different subsets of frequency-points and they all together cover the complete set of frequencies.

References

- K. Li, W. Zheng, and K. Li, "A fast algorithm with less operations for length-$N=q\times 2^m$ DFTs," *Signal Processing, IEEE Transactions on*, vol. 63, no. 3, pp. 673–683, Feb.1, 2015.
- S. Winograd, "A new method of computing DFT," Proc. *IEEE Int. Conf. Acoust., Speech and Signal Proc*,vol. 1, pp. 366-368, 1977.
- A.M. Grigoryan and S.S. Agaian, *Multidimensional Discrete Unitary Transforms: Representation, Partitioning, and Algorithms*, New York: Marcel Dekker, 2003.
- A.M. Grigoryan, "An algorithmfor computing the discrete Fourier transform with arbitrary orders," *Journal Vichislitelnoi Matematiki i Matematicheskoi Fiziki, AS USSR*. vol. 30, no. 10, pp. 1576–1581, Moscow 1991.(translated in http://fasttransforms.com/Art-USSR-papers/Art–JVMATofASUSSR1991.pdf).

THANK YOU VERY MUCH!

QUESTIONS, PLEASE?