

A NEW TWO-DIMENSIONAL FOURIER TRANSFORM ALGORITHM BASED ON IMAGE SPARSITY

Sheng Shi, Runkai Yang and Haihang You

Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

Email: {shisheng, yangrunkai, youhaihang}@ncic.ac.cn

ABSTRACT

With the coming age of big data, the image signals play more and more important role in our life due to the extraordinary advance of network communication technology, and the corresponding high efficiency image processing techniques are demanded urgently. The Fourier transform is an important image processing tool which is used in a wide range of applications. Traditional Fourier transform algorithm computes on the value of each point of image, regardless of their properties in frequency domain. However, most image signals possess sparsity in frequency domain. In this paper, we present a new fast two-dimensional Fourier transform based on image sparsity. With hash function including a series of procedures such as random spectrum permutation, filtering and subsampling in frequency domain, the algorithm could identify and estimate the k largest coefficients quickly. In most sparse cases, the resulting algorithm performs faster than state-of-the-art fast Fourier transform algorithm, FFTW.

Index Terms— Fourier transform, image sparsity, hash function, FFTW

1. INTRODUCTION

Advances on communication and transmission techniques over Internet have been witnessed under the background of the big data. The amount of image signals is dramatically increasing, and various applications of image processing are rapidly developed such as Magnetic Resonance Imaging (MRI) [1], Light Field Photography [2] and Radio Astronomy [3]. The discrete Fourier transform (DFT) is one of the most fundamental and important numerical algorithms which plays a central role in image processing area, including image feature extraction [4], image denoising [5], and compressed sensing [6]. The Fast Fourier Transform (FFT) [7] which computes the DFT of an n -size signal in $O(n \log n)$ time greatly simplifies the complexity of DFT, and gets a broad range of applications.

The general algorithms for computing the exact DFT must take time at least proportional to its size n . However, it is well known that most image signals possess sparsity in frequency domain [8–10]. That is, the image signals have natural-

ly sparse representations with respect to fixed Fourier basis. This property is widely used in various applications including High Efficiency Video Coding (HEVC) [11], computational learning theory [12], and compressed sensing. Therefore, for sparse image signals, the $\Omega(n)$ lower bound for the complexity of DFT no longer applies. It is crucial to study the new strategy of the Fourier transform based on image sparsity. In 2012, Hassanieh *et al* proposed sparse fast Fourier transform [13] [14] for one dimension signal which is faster than traditional DFT. However, two-dimensional image signal is more widely used, and two-dimensional sparse Fourier transform cannot simply be constructed with one-dimensional sparse Fourier transform. Therefore, in this paper, we propose a new fast two-dimensional Fourier transform that takes advantage of image sparsity (2D-SFFT).

The remainder of this paper is organized as follows. Section 2 presents the hash function which is an essential part in 2D-SFFT. Hash function includes permutation, filtering and subsampling in frequency domain. Section 3 describes the two-dimensional sparse Fourier transform algorithm. Section 4 reports the simulation results followed by the conclusion in Section 5.

2. HASH FUNCTION

Several conventions and notations are used in this thesis. An image in space domain is represented as a 2D matrix $x \in C^{n \times n}$, the 2D Fourier spectrum of the image is represented as \hat{x} . We assume that n is a integer of power of 2, the notation $[n]$ is defined as the set $\{0, 1, \dots, n-1\}$, and $[n] \times [n] = [n]^2$ to denote the $n \times n$ grid $\{(i, j) : i \in [n], j \in [n]\}$. The image support is denoted by $\text{supp}(x) \subseteq [n] \times [n]$. We use $\|x\|_0$ to denote $|\text{supp}(x)|$. All matrix indices are implicitly calculated modulo the matrix size, e.g. $x_{i,j}$ of image x is actually $x_{i \bmod n, j \bmod n}$. A set of matrix elements can be written as a matrix subscripted with a set of indices, for example $x_{I,J} = \{x_{i,j} | i \in I, j \in J\}$.

We define $\omega = e^{-2\pi i/n}$ to be a primitive n -th root of unity. In the following sections, we will use the following

definition of the 2D-DFT without the constant scaling factor:

$$\hat{x}_{i,j} = \sum_{u \in [n]} \sum_{v \in [n]} x_{u,v} \omega^{iu+jv}, \quad (u,v) \in \Omega_n$$

$$\Omega_n = \{(u,v) | 0 \leq i \leq n-1, 0 \leq j \leq n-1\} \quad (1)$$

This makes some proofs easier, but is not relevant in practical implementations.

The 2D-SFFT algorithm firstly uses a hash function to extract useful image information. Hash function includes random spectrum permutation, filtering and subsampling in frequency domain.

2.1. RANDOM SPECTRUM PERMUTATION

The first important step for the 2D-SFFT is spectrum permutation as defined in Definition 1:

Definition 1 Let σ_1 and σ_2 be invertible modulo n , i.e. $\gcd(\sigma_1, n) = 1$, $\gcd(\sigma_2, n) = 1$, and $\tau_1 \in [n]$, $\tau_2 \in [n]$. Then, $i \rightarrow \sigma_1 i + \tau_1 \bmod n$ and $j \rightarrow \sigma_2 j + \tau_2 \bmod n$ are permutations on $[n]$. The associated permutation $P_{\sigma_1, \sigma_2, \tau_1, \tau_2}$ on a matrix x is then given by

$$(P_{\sigma_1, \sigma_2, \tau_1, \tau_2} x)_{i,j} = x_{\sigma_1 i + \tau_1, \sigma_2 j + \tau_2} \quad (2)$$

When a permutation is applied to an image x in space domain, the image's frequency domain \hat{x} is also permuted. This interesting property is derived in Lemma 1.

Lemma 1 Let $P_{\sigma_1, \sigma_2, \tau_1, \tau_2}$ be a permutation and x be an two-dimensional vector. Then

$$(P_{\sigma_1, \sigma_2, \tau_1, \tau_2} x)_{\sigma_1 i, \sigma_2 j} = \hat{x}_{i,j} \omega^{-(\tau_1 i + \tau_2 j)}, \quad (i,j) \in \Omega_n \quad (3)$$

Proof. For $(i,j) \in \Omega_n$,

$$(P_{\sigma_1, \sigma_2, \tau_1, \tau_2} x)_{i,j} = \sum_{u \in [n]} \sum_{v \in [n]} x_{\sigma_1 u + \tau_1, \sigma_2 v + \tau_2} \omega^{iu+jv} \quad (4)$$

with $a_1 = \sigma_1 u + \tau_1$, $a_2 = \sigma_2 v + \tau_2$

$$\begin{aligned} & (P_{\sigma_1, \sigma_2, \tau_1, \tau_2} x)_{i,j} \\ &= \sum_{a_1 \in [n]} \sum_{a_2 \in [n]} x_{a_1, a_2} \omega^{\frac{(a_1 - \tau_1)}{\sigma_1} i + \frac{(a_2 - \tau_2)}{\sigma_2} j} \\ &= \omega^{-\left(\frac{\tau_1}{\sigma_1} i + \frac{\tau_2}{\sigma_2} j\right)} \sum_{a_1 \in [n]} \sum_{a_2 \in [n]} x_{a_1, a_2} \omega^{\left(\frac{\tau_1}{\sigma_1} i + \frac{\tau_2}{\sigma_2} j\right)} \\ &= \omega^{-(\tau_1 \sigma_1^{-1} i + \tau_2 \sigma_2^{-1} j)} \hat{x}_{\sigma_1^{-1} i, \sigma_2^{-1} j} \end{aligned} \quad (5)$$

This Lemma follows by substituting $i = \sigma_1 i$, $j = \sigma_2 j$. Note that $\omega^{-(\tau_1 i + \tau_2 j)}$ changes the phase, but does not change the magnitude of $\hat{x}_{i,j}$.

We do not have access to the input image's Fourier spectrum since that would involve performing a DFT. The permutation in the 2D-SFFT algorithm allows to permute the image's Fourier spectrum by modifying the image's space-domain x .

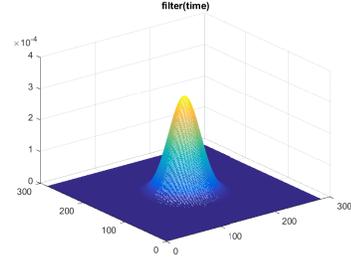


Fig. 1. 2D Gaussian window function

2.2. WINDOW FUNCTIONS

In order to achieve a sub-linear runtime, 2D-SFFT only uses a part of the input image for computations. The standard window function acts like a filter, allowing us to focus on a subset of the Fourier coefficients. Ideally, however, we would like the pass region of our filter to be as flat as possible to avoid spectral leakage. Therefore, two-dimension flat Gaussian window functions are used in 2D-SFFT.

The 2D flat Gaussian window function can be obtained from a 2D Gaussian standard window function which is shown in Fig. 1 by convolving it with a two-dimension ‘‘box car’’ window function which can be presented as:

$$r(i,j) = \begin{cases} 1, & (i,j) \in D \\ 0, & (i,j) \in D' \end{cases} \quad (6)$$

where $D = \{(i,j) | -\frac{b}{2} \leq i \leq \frac{b}{2}, -\frac{b}{2} \leq j \leq \frac{b}{2}\}$. The 2D Gaussian window function is presented as:

$$f(i,j) = A \exp\left[-\left(\frac{(i-i_0)^2}{2\sigma_i^2} + \frac{(j-j_0)^2}{2\sigma_j^2}\right)\right] \quad (7)$$

By the convolution of (6) and (7), we can get the 2D Gaussian flat window function G . Fig. 2 shows it in time domain and frequency domain.

Using 2D Gaussian flat window function G , a part of size $|supp(G)|$ can be extracted out of $P_{\sigma_1, \sigma_2, \tau_1, \tau_2} x$ by multiplying G and x and neglecting the coefficients with value zero. According to the convolution theorem, the multiplication is equivalent to a convolution of \hat{G} and \hat{x} . Filter process can expand the area of non-zero coefficients, in preparation for the subsequent sub-sampling and reverse steps, and further increase the probability of detecting non-zero coefficients.

2.3. FAST SUBSAMPLING AND DFT

Lemma 2 Let $B \in N$ divide n , x be an $N \times N$ two-dimensional matrix and y be a $B \times B$ two-dimensional matrix with $y_{i,j} = \sum_{u \in [n]} \sum_{v \in [n]} x_{i+Bu, j+Bv}$ for $i = 1, \dots, B$, $j = 1, \dots, B$. Then, $\hat{y}_{i,j} = \hat{x}_{i(n/B), j(n/B)}$

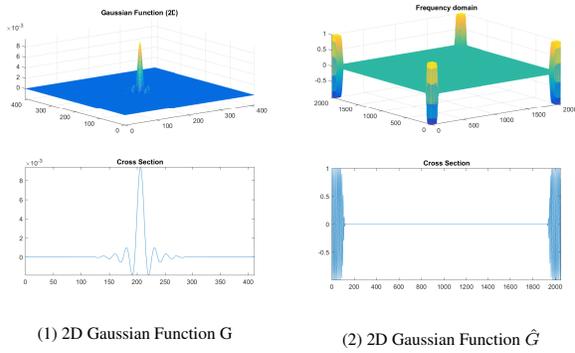


Fig. 2. The 2D Gaussian flat window function in time domain and frequency domain

Proof.

$$\begin{aligned}
\hat{x}_{i(\frac{N}{B}),j(\frac{N}{B})} &= \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} x_{u,v} \omega_N^{(iu+jv)(\frac{N}{B})} \\
&= \sum_{a_1=0}^{B-1} \sum_{u=0}^{\frac{N}{B}-1} \sum_{a_2=0}^{B-1} \sum_{v=0}^{\frac{N}{B}-1} x_{Bu+a_1, Bv+a_2} \omega_N^{(i(Bu+a_1)+j(Bv+a_2))(\frac{N}{B})} \\
&= \sum_{a_1=0}^{B-1} \sum_{u=0}^{\frac{N}{B}-1} \sum_{a_2=0}^{B-1} \sum_{v=0}^{\frac{N}{B}-1} x_{Bu+a_1, Bv+a_2} \omega_N^{(ia_1+ja_2)(\frac{N}{B})} \\
&= \sum_{a_1=0}^{B-1} \sum_{a_2=0}^{B-1} y_{i,j} \omega_N^{(ia_1+ja_2)(\frac{N}{B})} \quad (8)
\end{aligned}$$

Note that $\omega_N^{(ia_1+ja_2)N} = (e^{-2\pi b/N})^{(ui+vj)N} = 1$ and $\omega_n^{n/B} = (e^{-2\pi b/N})^{N/B} = \omega_B$. Thus, it follows $\hat{y}_{i,j} = \hat{x}_{i(n/B),j(n/B)}$.

Lemma 2 effectively reduces dimension by subsampling image in time domain and summing up the result. As the image is sparse in frequency domain, dimension reduction can reduce the complexity of searching position and amplitude of non-zero elements.

Random spectrum permutation, filtering and subsampling describe a hash function $h_{\sigma_1, \sigma_2} : [N \times N] \rightarrow [B \times B]$

$$h_{\sigma_1, \sigma_2}(i, j) = \text{round}(\sigma_1 \sigma_2 i j \frac{B^2}{N^2}), i \in [N], j \in [N] \quad (9)$$

Hash function h_{σ_1, σ_2} maps each of the $N \times N$ coordinates of the input image to one of $B \times B$ bins.

An example of image in frequency domain ($k = 2$) is shown in Fig 3(1). The process of hash function are shown in Fig. 3. Filter process can expand the area of non-zero coefficients to increase the detection probability. Subsampling can effectively reduces complexity.

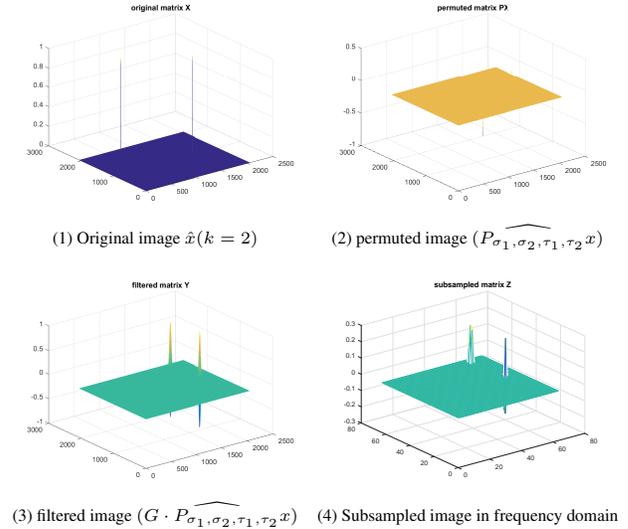


Fig. 3. The process of permutation, filtering and subsampling on image ($k = 2$)

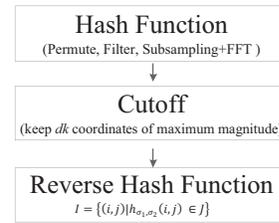


Fig. 4. The process of seek location

3. TWO-DIMENSION SPARSE FOURIER TRANSFORM ALGORITHM

The 2D-SFFT consists of multiple executions of two kinds of operations: location seeking and coefficient estimation. Location seeking is to generate a list of candidate coordinates which have a certain probability of being indices of nonzero coefficients in frequency domain. While coefficient estimation is used to precisely determine the frequency coefficients.

3.1. LOCATION SEEKING

The process of location seeking is shown in Fig. 4. By running multiple iterations of the location seeking, we can find candidate coordinates with high probability of being of the k nonzero coordinates.

3.2. COEFFICIENT ESTIMATION

The implementation of coefficient estimation also uses hash function. Given a set of coordinates I , $\hat{x}_{i,j}$ can be estimated

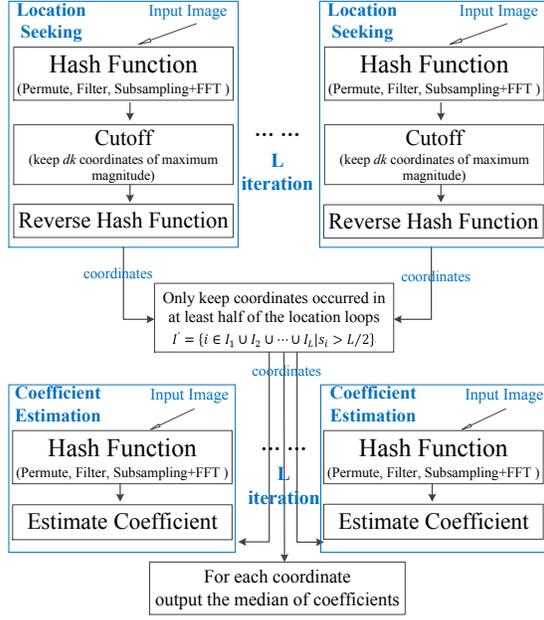


Fig. 5. A simplified flow diagram of 2D-SFFT

as

$$\hat{x}'_{i,j} = \hat{Z}_{h_{\sigma_1, \sigma_2}(i,j)} \omega^{\tau_1 i + \tau_2 j} / \hat{G}_{o_{\sigma_1, \sigma_2}(i,j)} \quad (10)$$

which basically removes the phase change due to the permutation and the effect of the filter.

A simplified flow chart of 2D-SFFT is shown in Fig. 5. After running multiple iterations of the location seeking we only need to keep coordinates occurred in at least half of the location loops. For the coordinates I' , the median of the corresponding outputs of L coefficient estimation is the frequency coefficient.

4. NUMERICAL EXPERIMENTS

In this section, we compare 2D-SFFT with 2D-FFTW in FFTW algorithms library [15].

Experiment 1: The sparsity k is fixed to a constant ($k = 50, 100$), image size ranges from $2^8 \times 2^8$ to $2^{13} \times 2^{13}$, and the runtime of the compared algorithms are shown in Fig. 6. As expected, the runtimes of them are approximately linear in the log scale. However, the slope of the line for 2D-SFFT is less than 2D-FFTW, indicating that 2D-SFFT has the faster runtime than 2D-FFTW for a large range of image size.

Experiment 2: The image size N is fixed to a constant ($N = 2^{11} \times 2^{11}, 2^{12} \times 2^{12}$) and the runtime vs. image sparsity k ($k = 50, 100, 200, 500, 1000$) are shown in Fig. 7. Fig. 7(1) shows 2D-SFFT is faster than 2D-FFTW when sparsity $k < 2500$, while 2D-SFFT presents disadvantage when sparsity $k > 2500$. In addition, we find that when sparsity

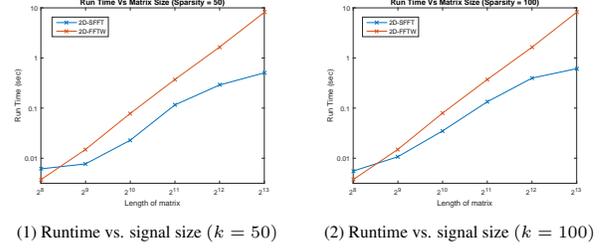


Fig. 6. The running time comparison on different image sizes

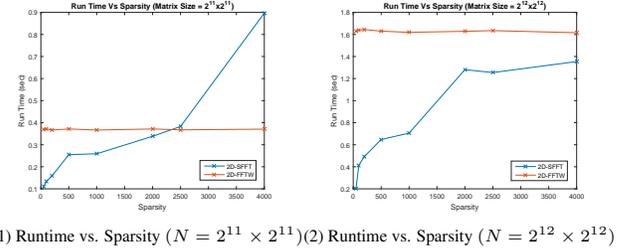


Fig. 7. The running time comparison on different sparsity degrees

k is increasing, the runtime of 2D-SFFT is also increasing, while the runtime of 2D-FFTW is essentially constant, which depends on image size not sparsity k .

5. CONCLUSION

FFT is an important image processing algorithm. Consider most images possess sparsity in frequency domain, we propose a novel algorithm 2D-SFFT which takes advantage of the sparsity characteristics of images based on existing research. Experimental results show that 2D-SFFT presents a substantial advantage than traditional 2D-FFTW.

As this paper is exploratory, there are many intriguing questions that future work should consider. The 2D-SFFT is a probabilistic algorithm which needs multiple loops to increase the accuracy. Signal sparsity and selection of filter play an important role to successful algorithm implementation and application.

6. ACKNOWLEDGMENT

This work is supported by the National Key Research and Development Program of China (2016YFE0100300, 2016YFB0200803), the 100 Talents Program of the Chinese Academy of Sciences (2920154070) and the Knowledge Innovation Project of the Chinese Academy of Sciences (5120146040).

7. REFERENCES

- [1] C. F. Beckmann and S. M. Smith, "Probabilistic independent component analysis for functional magnetic resonance imaging," *IEEE Transactions on Medical Imaging*, vol. 23, no. 2, pp. 137–152, 2004.
- [2] Shota Taki, Fumihiko Sakaue, and Jun Sato, "High resolution light field photography from split ray imaging and coded aperture," in *VISAPP 2014 - Proceedings of the 9th International Conference on Computer Vision Theory and Applications, Volume 2, Lisbon, Portugal, 5-8 January, 2014*, pp. 605–612.
- [3] Rik Jongerius, Stefan J. Wijnholds, Ronald Nijboer, and Henk Corporaal, "An end-to-end computing model for the square kilometre array," *IEEE Computer*, vol. 47, no. 9, pp. 48–54, 2014.
- [4] Jon Atli Benediktsson, Martino Pesaresi, and Kolbeinn Amason, "Classification and feature extraction for remote sensing images from urban areas based on morphological transformations," *IEEE Trans. Geoscience and Remote Sensing*, vol. 41, no. 9, pp. 1940–1949, 2003.
- [5] S. Grace Chang, Bin Yu, and Martin Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Trans. Image Processing*, vol. 9, no. 9, pp. 1532–1546, 2000.
- [6] S. Shi, R. Xiong, S. Ma, X. Fan, and W. Gao, "Image compressive sensing using overlapped block projection and reconstruction," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2015, pp. 1670–1673.
- [7] B. Srinivasa Reddy and Biswanath N. Chatterji, "An fft-based technique for translation, rotation, and scale-invariant image registration," *IEEE Trans. Image Processing*, vol. 5, no. 8, pp. 1266–1271, 1996.
- [8] X. Zhang, R. Xiong, X. Fan, S. Ma, and W. Gao, "Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4613–4626, Dec 2013.
- [9] X. Zhang, R. Xiong, W. Lin, S. Ma, J. Liu, and W. Gao, "Video compression artifact reduction via a spatio-temporal multi-hypothesis prediction," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 6048–6061, Dec 2015.
- [10] X. Zhang, W. Lin, R. Xiong, X. Liu, S. Ma, and W. Gao, "Low-rank decomposition-based restoration of compressed images via adaptive noise estimation," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4158–4171, 2016.
- [11] G. J. Sullivan, J. R. Ohm, W. J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.
- [12] Abdu Rahiman V, Rohit U., and S. N. George, "Modified dictionary learning method for sparsity based single image super-resolution," in *2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, 2016, pp. 473–477.
- [13] Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price, "Simple and practical algorithm for sparse fourier transform," in *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, 2012, pp. 1183–1194.
- [14] Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price, "Nearly optimal sparse fourier transform," in *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, 2012, pp. 563–578.
- [15] Matteo Frigo and Steven G. Johnson, "FFTW: an adaptive software architecture for the FFT," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98, Seattle, Washington, USA, May 12-15, 1998*, 1998, pp. 1381–1384.