VOL. 11, NO. 1, JANUARY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

FAST ALGORITHM FOR SOLVING THE FLAT PROBLEM OF COMPUTED TOMOGRAPHY BY DIRECT FOURIER METHOD FOR THE FUNCTIONS F(X, Y) = F(Y, X) USING INTERLINATION OF FUNCTIONS

Vitaliy Mezhuyev[1], Oleg M. Lytvyn[2] and Kristina Babenko[2]
[1]University Malaysia Pahang, Gambang, Malaysia
[2]Ukrainian Engineering and Pedagogical Academy, Kharkiv, Ukraine
E-Mail: mejuev@ukr.net

ABSTRACT

The paper proposes new algorithm for solving the flat problem of computed tomography by direct method of Fourier for the symmetric functions using spline-interlination. The algorithm reduces the number of operations needed to compute the Fourier series asymptotically twice.

Keywords: direct Fourier transform, interlination of functions, algorithm, computed tomography.

INTRODUCTION

In computed tomography, is often arose the problem of calculation of sums of Fourier series for the functions of one and many variables on a given system of points (e.g., on a line segment or on a plane). The most widely used mathematical apparatus to solve this problem is a Fast Fourier Transform (FFT) [1; 2], which allows us significantly reduce the time of computations. For example, direct calculation of the vector $X, X_p = \sum_{k=1}^{n} x_k e^{i2\pi(k-1)(p-1)/n},\ p=\overline{1,n}$ of the length n needs $O(n^2)$ Arithmetic Operations (AO), and when using FFT - only $O(n\log_2 n)$ AO. Widely used for scientific research MATLAB system defines two-dimensional FFT - *fft2(X)* - with the help of one-dimensional FFT - *fft(X)* - by the formula

fft2(X) = (fft (fft(X))')'[1]

For the matrix *X* of dimension $n \times n$ the number of operations, needed to find the sum

$$F_{p,q} = \\ = \sum_{k=1}^{n}\sum_{l=1}^{n} e^{i2\pi((k-1)(p-1)+(q-1)(l-1))/n} \cdot h_{k,l}, \\ p,q=\overline{1,n}$$

by the known matrix h is $2n^2\log_2 n$ AO. Without use of FFT to compute this matrix n^4 of AO is needed.

[1] The function *fft* in MATLAB computes the discrete sums of Fourier. The symbol $'$ is the operation of matrix transposition.

Despite of such obvious effectiveness of two-dimensional FFT, analysis of continuously increasing massifs of data requires further optimization of algorithms for the fast Fourier transform. One of the promising directions for this optimization is a combination of FFT with the method of spline-interlination of functions [3] for computing values of an array of two-dimensional Fourier sums.

Another direction is a selection of subclasses of problems (e.g. symmetric, axisymmetric, plane, etc.), which is also widely used when solving three-dimensional problems in mathematics. This approach allows us significantly reduce the number of needed arithmetic operations and, in some cases, obtain a precise solution. This paper combines two mentioned directions and proposes an algorithm that reduces the number of AO, needed to compute the Fourier sums asymptotically twice.

MATERIAL AND METHODS

In this paper we consider the functions of the form *f(x,y)=f(y,x)*, which graphs are symmetric with respect to the bisectors of the 1 and 3 quadrants. Let us analyse the possible directions of reducing the computation time of the double Fourier sum of *f(x,y)* on the system of points $x=\frac{p}{n}, y=\frac{q}{n},\ p,q=\overline{1,n}$ using FFT and spline-interlination of functions.

Suppose, we need compute the double sum of Fourier series with the matrix of coefficients $h_{kl},\ k=\overline{-\frac{n}{2},\frac{n}{2}},\ l=\overline{-\frac{n}{2},\frac{n}{2}}$ on the system of points $x=\frac{p}{n}, y=\frac{q}{n},\ p,q=\overline{1,n}$. We will have

ARPN Journal of Engineering and Applied Sciences

$$F_{p,q} = \sum_{k=-\frac{n}{2}}^{\frac{n}{2}} \sum_{l=-\frac{n}{2}}^{\frac{n}{2}} h_{kl} e^{-i2\pi(kp+lq)/n} =$$

$$= \sum_{l=-\frac{n}{2}}^{\frac{n}{2}} \left(\sum_{k=-\frac{n}{2}}^{\frac{n}{2}} h_{kl} e^{-i2\pi kp/n} \right) e^{-i2\pi lq/n} =$$

$$= \sum_{l=-\frac{n}{2}}^{\frac{n}{2}} W_{p,\,l} e^{-i2\pi lq/n}$$

where $W_{p,\,l} = \sum_{k=-\frac{n}{2}}^{\frac{n}{2}} h_{kl} e^{-i2\pi kp/n}$.

Matrix $W_{p,\,l}$, $p = \overline{-\frac{n}{2}, \frac{n}{2}}$ is one-dimensional Fourier transform of the *l-th* row of the matrix h_{kl} .

To compute $F_{p,q}$ we need use all of the elements of the matrix h . In other words, it is necessary to perform one-dimensional Fourier transform for all rows and then for all columns of the matrix h_{kl} . Note, using FFT algorithm it is impossible to find a single value of the matrix F .

The idea of proposed approach is to compute two-dimensional Fourier transform not for each row of the matrix h_{kl} , but only for

$$l = -\frac{n}{2}, -\frac{n}{2} + \sqrt{n}, -\frac{n}{2} + 2\sqrt{n} ..., 0, ... \frac{n}{2},$$

that is, on the mesh, sparse along the rows matrix h_{kl} with the step $\sqrt{n}$.

Restore values between rows of the resulting matrix we propose using spline-interlination (blended interpolation) of functions [4], which has high accuracy.

Let we have matrix h_{kl} of dimensionality $n \times n$. In the traditional approach to compute two-dimensional FFT, it is necessary to perform $2n$ one-dimensional fast transformations: n for each column and n for each rows of the matrix.

When calculating on a sparse along the rows of the matrix h_{kl} mesh, it is necessary to perform n FFT for each column and $\sqrt{n}$ FFT to each row of obtained matrix. Similarly, when calculating on sparse along the columns of the matrix h_{kl} mesh, it is necessary to perform n FFT for each row and $\sqrt{n}$ FFT for each column of obtained matrix.

Statement 1: If *f(x,y)=f(y,x)*, then $h_{k,l} = h_{l,k}$, $k, l = \overline{1, n}$.

Proof: If the sum $\sum_{p=-\frac{n}{2}}^{\frac{n}{2}} \sum_{q=-\frac{n}{2}}^{\frac{n}{2}} h_{k,l} e^{i2\pi(px+qy)}$ is the sum of the Fourier of the function *f(x,y),* then

$$h_{k,l} = \int_0^1\int_0^1 f(x, y) e^{i2\pi(kx+ly)} dxdy =$$

$$= \int_0^1\int_0^1 f(y, x) e^{i2\pi(ly+kx)} dydx = h_{l,k} .$$

Statement 1 is proved.

Statement 2: If $h_{k,l} = h_{l,k}$ then *fft(* h *)=fft(* h^T *).*

Statement 3: If $h = h^T$, *V=fft(h)', W=fft(V), V1=fft(h')', W1=fft(V1)*, then *W1 = W*.

Proof: Follows from the obvious equality *V=V1*.

Theorem 1: Suppose we need find the value of $F_{p,q}, p = 1, \sqrt{n}, 2\sqrt{n}, ..., n; q = \overline{1, n}$ using FFT in the computation of the following discrete Fourier transform

$$F_{p,q} = \sum_{k=1}^{n} \sum_{l=1}^{n} e^{2\pi i(kp+ql)/n} h_{kl} =$$

$$= \sum_{k=1}^{n} \left(\sum_{l=1}^{n} e^{2\pi i(kp+ql)/n} h_{kl} \right), \quad .$$

$$p, q = \overline{1, n}$$

To compute it, lets use the following algorithm

Step 1: Find $U = fft(h)$. In MATLAB this operation means application of *fft* to each of the columns of the matrix *h* separately.

Step 2: Find $V = fft(U')$. However, we do not find all values of V , but only those, which satisfy $F_{p,q}, p = 1, \sqrt{n}, 2\sqrt{n}, ..., n; q = \overline{1, n}$.

ARPN Journal of Engineering and Applied Sciences

Corresponding software implementation of this algorithm is given below.

Step 3: Find $F = V^T$.

Given that for the class of the functions *f(x,y) = f(y,x)* transposition is not necessary, the step 3 may be skipped.

Then to compute $F_{p,q}, p = 1, \sqrt{n}, 2\sqrt{n}, ..., n; q = \overline{1,n}$ we need the following number of AO:

$$Q = O\left(n^2 \log_2 n + n^{\frac{3}{2}} \log_2 n \right)$$

Proof: To find the matrix U we need perform n times FFT for each column of the matrix h separately. To do this, we use $nO(n\log_2 n) = O(n^2 \log_2 n)$ AO.

To compute the matrix V, given that we have to find only the values in $\sqrt{n}$ rows, we need $\sqrt{n} \cdot n\log_2 n = O\left(n^{\frac{3}{2}} \log_2 n \right)$ AO. Thus, in the proposed method, to obtain information on n columns and $\sqrt{n}$ rows, we need perform only $Q = O\left(n^2 \log_2 n + n^{\frac{3}{2}} \log_2 n \right)$ AO.

Theorem 1 is proven.

Result 1: Given the properties of the functions *f(x,y)=f(y,x)* and the interpolation formula, based on interlination of functions, it is enough to make calculations similar to Theorem 1, since the values of the rows are equal to the values of the transposed columns $F_{p,q} = F_{q,p}$.

Result 2: To find this information using MATLAB subroutine *fft2(X)* we need $Q_1 = 2n^2 \log_2 n$ AO. Thus, reduced number of arithmetic operations is asymptotically equal to

$$\frac{2n^2 \log_2 n}{n^2 \log_2 n + n^{\frac{3}{2}} \log_2 n} =$$

$$= \frac{2\sqrt{n}}{\sqrt{n}+1}$$

$$\lim_{n \to \infty} \frac{2\sqrt{n}}{\sqrt{n}+1} = 2$$

Thus, to obtain information we use asymptotically twice fewer amount of AO. I.e., the efficiency of the algorithm increases comparatively with two-dimensional FFT algorithm asymptotically two times. This reduction is due to that algorithm computes not the entire matrix, but only a part of it.

Note, that the standard MATLAB-function to compute FFT of two variables is ***fft2(X) = fft(fft(X)')'***. Thus, the proposed algorithm even is more effective, since the given value of increasing efficiency does not consider two matrix transpositions, performed in the MATLAB function ***fft2***.

To compare effectiveness of the algorithms we rewrite both algorithms in C++. We implement the traditional two-dimensional FFT (a function *fft2*) and optimized algorithm for the two-dimensional FFT (a function *Myfft2*).

Both these programs use one-dimensional Fast Fourier Transform (function *fft*). The only difference is the organization of loops: if *fft2* performs *fft* for each row, and then *fft* for all columns of the input matrix, then *Myfft2* performs *fft* for each row, and then *fft* for each $\sqrt{n}$ -th column of the input matrix.

The size of the matrix is defined by the global constant n, where $n = 2^k$, $k = 1,2,3...$

In the process of computing experiment we managed to implement calculations for the maximum value *n=4096*, wherein the number of matrix elements is equal to *4096* x *4096=16 777 216* for the real numbers and the same amount of the matrix of complex numbers. Thus, the total amount of operating memory used is *16 777 216* x *2* x *4 byte = 134 Mb*. Table-1 shows results of computation experiment.

VOL. 11, NO. 1, JANUARY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Table-1. Comparison of theoretical and experimental values of computation time.

N	32	64	128
t_1, c (fft2)	0	0,010	0,045
t_2, c (Myfft2)	0	0,006	0,030
t_1/t_2(theoretical) $\frac{2\sqrt{n}}{\sqrt{n}+1}$	1,7	1,778	1,838
t_1/t_2 (experimental)	–	1,666	1,500

N	256	512	1024
t_1, c (fft2)	0,236	0,999	4,155
t_2, c (Myfft2)	0,155	0,650	2,660
t_1/t_2(theoretical) $\frac{2\sqrt{n}}{\sqrt{n}+1}$	1,882	1,915	1,939
t_1/t_2 (experimental)	1,523	1,537	1,562

N	2048	4096
t_1, c (fft2)	17,10	721,111
t_2, c (Myfft2)	10,87	416,526
t_1/t_2(theoretical) $\frac{2\sqrt{n}}{\sqrt{n}+1}$	1,957	1,969
t_1/t_2 (experimental)	1,57	1,731

Thus, the results of computation experiment confirm theoretically found trend.

Note, that the time of computations increases at *n=4096*. This is due to the fact that when *n=2048* an array is fully located in operating memory, but at *n=4096* the program start using dump on the hard drive, having much slower speed of file operations. Meanwhile, the ratio of the time $\mathbf{t_1/t_2}$ remaining true.

USING INTERLINATION OF FUNCTIONS

Let us discuss the method to restore the internal structure of an object using spline interlination of functions. We will reconstruct the structure of an object by its projections by development of the method, which uses both direct Fourier method and interpolation operators, built on the basis of interlination.

Let's investigate the function $f(x,y)$, $\operatorname{supp} f = [-c;c]\times[-c;c]$. For it, the Radon transform $p(r,\theta)$ has compact support in $[-d,d]\times[0,\pi]$, where $d \le c\sqrt{2}$.

Direct Radon transform of the function $f(x,y)$ is determined by the formula [5]

$$[Rf](r,\theta) = p(r,\theta) = \\ = \int_{-\infty}^{\infty} f(r\cos\theta - s\sin\theta, r\cos\theta + s\sin\theta)ds, \quad (1)$$

where θ is an angle,
and the reverse - by the formula

$$f(x,y) = R^{-1} f(x,y) := \\ := \frac{1}{2\pi^2}\int_0^{\pi}\int_{-\infty}^{\infty} \frac{\frac{\partial p(r,\theta)}{\partial r}}{x\cos\theta + y\sin\theta - r} dr d\theta. \quad (2)$$

Double Fourier transform $\tilde{f}(\xi,\eta)$ of the function $f(x,y)$ is calculated by the formula

VOL. 11, NO. 1, JANUARY 2016 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

$$\tilde{f}(\xi,\eta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-2i\pi(\xi x+\eta y)}dxdy = \\ = \int_{-c}^{c}\int_{-c}^{c} f(x,y)e^{-2i\pi(\xi x+\eta y)}dxdy \tag{3}$$

The proposed direct Fourier method is as follows:

a) we get $\tilde{p}(\rho,\theta)$ from $p(r,\theta)$ by the formula

$$\tilde{p}(\rho,\theta) = \\ = \int_{-d}^{d} p(r,\theta)e^{-2i\pi\rho r}dr \tag{4}$$

i.e. $p(r,\theta) = \frac{1}{2d}\sum_{m=-\infty}^{\infty}\hat{p}_m(\theta)e^{i\pi mr/d}$,

$$\hat{p}_m(\theta) = \\ = \frac{1}{2d}\int_{-d}^{d} p(r,\theta)e^{-i\pi mr/d}dr.$$

b) obtain $\tilde{f}(\xi,\eta) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)e^{-i\pi(\xi x+\eta y)}dxdy$ using so-called slice theorem about central section by the formulas

$$\tilde{p}(\rho,\theta) = \tilde{f}(\rho\cos\theta, \rho\sin\theta), \\ -\infty < \rho < \infty, 0 \le \theta < \pi \\ \tilde{f}(\xi,\eta) = \tilde{p}\left(\sqrt{\xi^2+\eta^2}, arctg\left(\tfrac{\eta}{\xi}\right)\right), \\ -\infty < \xi,\eta < \infty \tag{5}$$

b) find an unknown function $f(x,y)$ using Fourier series

$$f(x,y) = \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty}\hat{f}_{k,l}e^{i\pi(kx+ly)/c} \tag{6}$$

where $\hat{f}_{k,l} = \frac{1}{4c^2}\int_{-c}^{c}\int_{-c}^{c} f(x,y)e^{-i\pi(kx+ly)/c}dxdy$ are Fourier coefficients of the function $f(x,y)$.

Substituting (6) into (3) we obtain

$$\tilde{f}(\xi,\eta) = \\ = \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty}\hat{f}_{k,l}\int_{-c}^{c}\int_{-c}^{c} e^{i\pi(kx+ly)/c}e^{-2i\pi(\xi x+\eta y)/c}dxdy = \\ = \sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty}\hat{f}_{k,l}\sin c(\pi(k-2c\xi))\sin c(\pi(l-2c\eta))$$

From (5) we obtain

$$\sum_{l=-\infty}^{\infty}\hat{p}_m(\theta)\sin c(\pi(m-2d\rho)) = \\ = \sum_{j=-\infty}^{\infty}\sum_{n=-\infty}^{\infty}\hat{f}_{jn}\sin c(\pi(j-2c\rho\cos\theta))\sin c(\pi(n-2c\rho\sin\theta)) \tag{7}$$

$$\sin c(t) = \frac{\sin(t)}{t}$$

In the formula (6) for each pair (*k, l*) we choose (θ,ρ) in a way that

$$k = 2c\rho\cos\theta,\ l = 2c\rho\sin\theta \tag{8}$$

Such selection (θ,ρ) leaves only one set of $\hat{f}_{k,l}$ in the right side of (6). Given that $-d \le \rho \le d,\quad 0 \le \theta < \pi$, using these formulas we obtain (9).

$$f_{k,l} = \sum_{m=-\infty}^{\infty}\hat{p}_m\left(arctg\left(\tfrac{l}{k}\right)+\pi s_0(kl)\right) \\ \sin c\left(\pi m - \pi s_1(l)\frac{d}{c}\sqrt{k^2+l^2}\right), \tag{9}$$

Note, that formula (9) is an exact formula for the calculation of the Fourier coefficients. However, in practice, it is necessary to replace the infinite sum by the finite one. In addition, when calculating $\hat{p}_m\left(arctg\left(\tfrac{l}{k}\right)+\pi s_0(kl)\right)$ using the formula $\hat{p}_m(\theta) = \frac{1}{2d}\int_{-d}^{d} p(r,\theta)e^{-i\pi mr/d}dr$ we need use the appropriate quadrature formula.

In this case, the function $f(x,y)$ will be calculated by the formula

$$f(x_k, y_l) = \frac{1}{4c^2} \sum_{k=-N/2}^{N/2} \sum_{l=-M/2}^{M/2} \tilde{h}_{k,l} e^{-2\pi i (x_k k + y_l l)/c}$$

Note, that using proposed method it becomes possible to calculate value of functions $f(x,y)$ at points $f(x_k, y_l)$ using standard form for the Fourier sums, rather than the formula

$$f(x_k, y_l) = \frac{1}{4c^2} \sum_{k=-N/2}^{N/2-1} \sum_{l=-M/2}^{M/2-1} \tilde{h}_{k,l} e^{-2\pi i (x_k k + y_l l)/c}$$

CONCLUSIONS

The paper proposes a new effective method, which can be used to find an internal structure of a body by its projections on a given system of planes. The method uses Fast Fourier Transform and is adopted for the needs of computed tomography. For the rendering a tomography image, the method uses operators of interpolation built on the base of spline interlination of functions. It allows us for the functions *f(x, y)=f(y,x)* to reduce the number of AO asymptotically twice.

REFERENCES

[1] Gottlieb D., Gustaffson B., Forssén P. 2000. On the Direct Fourier Method for Computer Tomography. IEEE Transactions on Medical Imaging. 19(3).

[2] Babenko K., Lytvyn O., Mezhuyev V. 2003. Mathematical model of computer tomography based on direct Fourier method and interlination of functions. Dopovidi natsional noyi akademiyi nauk Ukrayiny. (10): 70-74.

[3] Oleg M. 2002. Lytvyn Interlination of functions and some its applications. Kharkiv: Osnova. p. 544.

[4] Lytvyn O.N. 2004. Interlineation and interflation functions of many variables (blending function interpolation) and economical algorithms in the approximation theory. The International Conference of Computational Mathematics, December, 15-17, 2004, Singapore. pp. 1100-1109.

[5] Helgason Sigurdur. 2011. Integral Geometry and Radon Transforms. Springer. p. 301.