IMAGE RECONSTRUCTION BY OPED ALGORITHM WITH AVERAGING

YUAN XU, OLEG TISCHENKO, AND CHRISTOPH HOESCHEN

ABSTRACT. OPED is a new image reconstruction algorithm based on orthogonal polynomial expansion on the disk. We show that the integral of the approximation function in OPED can be given explicitly and evaluated efficiently. As a consequence, the reconstructed image over a pixel can be effectively represented by its average over the pixel, instead of by its value at a single point in the pixel, which can help to reduce the aliasing caused by under sampling. Numerical examples are presented to show that the averaging process indeed improves the quality of the reconstructed images.

1. INTRODUCTION

Computed tomography (CT) offers a non-intrusive method for 2D cross-sectional or 3D imaging of an object based on x-ray data. We consider the case of 2D in this paper. Mathematically, it means reconstructing a function f from its Radon projections,

$$\mathcal{R}f(\theta,t) := \int_{I(\theta,t)} f(x,y)dxdy, \tag{1.1}$$

where $I(\theta,t)$ is a line $x\cos\theta + y\sin\theta = t$. The integral (1.1) is the 2D Radon transform. The main task is to find an approximation, say $\mathcal{A}f$, to the function f that uses a finite number of Radon projections of f. Once the $\mathcal{A}f$ is computed, the reconstructed image is shown by the values of $\mathcal{A}f$ at the pixel points.

An image on a computer screen is characteristically discrete, since the screen in made up of pixels (say 1024 × 768 pixels on a typical desktop computer), which are small squares, and the image over one pixel is represented by one value. In other words, the image shown on a computer screen is that of a step function. For reconstructed image in CT, the value on a pixel is usually taken as the evaluation of $\mathcal{A}f$ at either a corner or the center of the pixel. Such a choice, however, is not satisfactory from the approximation point of view and, in practice, it is one of the main reasons for the aliasing artifacts. The purpose of this paper is to introduce an additional procedure in the algorithm OPED that will overcome this problem. OPED is a new reconstruction algorithm based on the orthogonal polynomial expansion on the disk, it is developed recently in [9, 10, 11] and tested in [4, 12]. The algorithm is stable and can be implemented easily using fast Fourier transform

Date: March 28, 2007.

2000 *Mathematics Subject Classification.* 42A38, 42B08, 42B15.

Key words and phrases. Rreconstruction of images, Radon projections, OPED algorithtm, averaging.

The work of the first author is supported in part by the National Science Foundation under Grant DMS-0604056.

(FFT). Our main result in this paper shows that the approximation $\mathcal{A}f$ in OPED can be integrated exactly over any pixel and the result of the integration can be evaluated efficiently with an implementation that is as fast as the original OPED. As a result we can use the average of $\mathcal{A}f$ on a pixel as the value of the image on the pixel, instead of evaluation of $\mathcal{A}f$ on a point of the pixel. The result is an improved version of the algorithm, which we shall call OPED with averaging.

There are several advantages of taking averaging over the pixel. First of all, from a mathematical point of view, a good measure of the quality of the image is the error measured in L^1 norm. Taking averaging over the pixel is evidently much better than evaluation at one point on the pixel in this regard. Furthermore, aliasing artifacts appear because of under sampling, taking average over the pixel should avoid this problem altogether, as confirmed by the numerical examples in the paper. We should emphasis that OPED with averaging works because of the exact (analytic) formula for integration over pixels. This means that the integral can be carried out analytically, in contrast to approximation by using numerical integration. For other algorithms, it is often not possible to find an analytic formula for the integral over pixels; for example, it does not hold for FBP (Filtered back-projection) algorithm, the main algorithm currently used in medical imaging (see, for example, [1, 5]).

The paper is organized as follows. In the next section we describe the OPED algorithm and show how to compute the integral of the approximation function in OPED. In Section 3 we state the new algorithm OPED with averaging, which shows how the integrals of $\mathcal{A}f$ over the pixels can be implemented effectively. Results of numerical tests are given in Section 4.

2. OPED algorithm and its integration

2.1. **OPED algorithm.** The basic OPED algorithm is derived from the partial sums of the orthogonal expansion on the unit disk. Let $\mathcal{V}_n$ denote the space of orthogonal polynomials of total degree n on the unit disk $B := \{(x,y) : x^2+y^2 \le 1\}$; that is, $P \in \mathcal{V}_n$ if $P \in \Pi_n^2$ and

$$\int_B P(x,y)Q(x,y)dxdy = 0, \qquad \forall Q \in \Pi_{n-1}^2,$$

where Π_n^2 denote the space of polynomials of total degree at most n in two variables. It is well known that L^2 can be decomposed as

$$L^2(B) = \sum_{n=0}^{\infty} \bigoplus \mathcal{V}_n : \qquad f = \sum_{n=0}^{\infty} \operatorname{proj}_n f,$$

where $\operatorname{proj}_n f$ is the projection of f onto $\mathcal{V}_n$. Let $S_n f \in \Pi_n^2$ denote the partial sum of the above series. The starting point of the OPED algorithm is the following explicit expression of $S_{2m}f$ in terms of Radon projections:

$$S_{2m}f = \frac{1}{2m+1} \sum_{\nu=0}^{2m} \int_{-1}^{1} \mathcal{R}f(\phi_\nu, t)\Phi_\nu(t;x,y)dt, \tag{2.1}$$

where $\phi_\nu = \frac{2\nu\pi}{2m+1}$ and

$$\Phi_\nu(t;x,y) = \frac{1}{2m+1} \sum_{k=0}^{2m} (k+1)U_k(t)U_k(x\cos\phi_\nu + y\sin\phi_\nu). \tag{2.2}$$

Here and in the following, $U_k(t) = \sin(k+1)\theta / \sin\theta$, $t = \cos\theta$, is the Chebyshev polynomial of the second kind. The basic OPED algorithm is the following discretization of $S_{2m}f$.

OPED Algorithm. *For $m \ge 0$, $(x,y) \in B$,*

$$\mathcal{A}_{2m}f(x,y) = \sum_{\nu=0}^{2m} \sum_{j=0}^{2m} \mathcal{R}f\,(\phi_\nu, \cos\psi_j)\, T_{j,\nu}(x,y), \tag{2.3}$$

where

$$T_{j,\nu}(x,y) = \frac{1}{(2m+1)} \sin\psi_j \Phi_\nu(\cos\psi_j; x, y), \qquad \psi_j = \frac{(2j+1)\pi}{4m+2}. \tag{2.4}$$

There are several variants of the basic OPED algorithms, see [9, 11, 12]. The derivation of the formula as stated above was carried out in [9] from scratch. It turns out, however, that there are several earlier results that can be used to derive the algorithm. In fact, the formula

$$S_n f(x,y) = \frac{1}{2\pi} \int_0^{2\pi} \int_{-1}^{1} \mathcal{R}f(\phi, t) \sum_{k=0}^{n} (k+1) U_k(t) U_k(x\cos\phi + u\sin\phi) dt d\phi \tag{2.5}$$

has appeared earlier in [2, 6], which can be used to deduce (2.1) by applying a quadrature on $[0, 2\pi]$. A natural quadrature to use is

$$\frac{1}{2\pi} \int_0^{2\pi} g(t) dt = \frac{1}{n+1} \sum_{\nu=0}^{n} g(\tfrac{2\nu\pi}{n+1}),$$

which is exact for all trigonometric polynomials of degree n [13]. We used only the discretization of $S_n f$ for $n = 2m$ in OPED algorithm, since when $n = 2m+1$ the relation $\mathcal{R}f(\pi - \phi, t) = \mathcal{R}f(\phi, -t)$ shows that there are only $m+1$ distinct directions, instead of $2m+2$ many. More directions mean better resolution in the reconstruction. The arrangement of the lines on which the Radon projections are taken is called scanning geometry. Thus, OPED algorithm is better equipped with a scanning geometry of odd number of directions. To derive the algorithm (2.3) from (2.1), we apply the Gaussian quadrature based on the zeros of the Chebyshev polynomials of the fist kind. In [7], the infinite sum that is the sum (2.5) when $n \to \infty$ was stated as the inversion of $\mathcal{R}f(\theta, t)$, but 2D algorithm and its scanning geometry were not discussed. One more reference that needs to be mentioned is [3], which also derived such an algorithm from scratch but without realizing the connection to the orthogonal partial sums. The algorithm in [3] did not specify the scanning geometry of odd directions, and it uses the quadrature based on the zeros of Chebyshev polynomials of the second kind to discretize the integral with respect to t, which results to a scanning geometry different from that of (2.3).

One important property of the OPED algorithm is that it preserves polynomials of degree $2m-1$, that is, if f is a polynomial of degree less than $2m$, then $\mathcal{A}_{2m}f = f$. In [9] it is shown that the operator norm $\|\mathcal{A}_{2m}\|_\infty$, in the uniform norm of $C(B)$, has the growth order $\|\mathcal{A}_{2m}\|_\infty \sim m \log m$ as m goes to infinity. In particular, this shows that $\mathcal{A}_{2m}f$ converges to f uniformly on B if $f \in C^2(B)$. More precisely, we have

$$\|f - \mathcal{A}_{2m}f\|_\infty \le c_f \frac{\log(m+1)}{m}, \qquad \forall f \in C^2(B), \tag{2.6}$$

where c_f is a constant that depends on the second order derivatives of f but independent of m. For practical application, it is crucial that the algorithm can be implemented with fast Fourier transform and an interpolation step [10], which makes the algorithm fast in speed. The additional interpolation step uses linear interpolation, which implies that the resulted approximation function, call it $\mathcal{AI}_{2m}$, no longer preserves polynomials. However, it is shown in [10] that the operator norm of $\mathcal{AI}_{2m}$ still satisfies $\|\mathcal{AI}_{2m}\| \sim m \log m$ and, furthermore, $\mathcal{AI}_{2m} f$ converges uniformly to f if $f \in C^4(B)$. More precisely, we have

$$\max_{(x,y)\in\Omega} |f(x,y) - \mathcal{AI}_{2m} f(x,y)| \le c_f \frac{\log(m+1)}{\sqrt{m}}, \qquad \forall f \in C^4(B), \tag{2.7}$$

for every compact set Ω in B^2, where c_f depends on the fourth order derivatives of f, but independent of m. Thus, the fast implementation of the OPED algorithm converges almost as well as the usual OPED in theory. The numerical tests ([4, 12]) have shown that OPED is accurate, has high resolution and is very much comparable with FBP algorithm.

2.2. **OPED with averaging.** The image shown on a computer screen is inside a square region. We assume that the reconstructed image by OPED is on the unit disk B, which is imbedded in the square region $[-1,1]^2$. If f represents the image, which is defined on B, then we extend f to $[-1,1]^2$ by defining its value on points in $[-1,1]^2 \setminus B$ to be zero. We divide the square $[-1,1]^2$ into $N \times N$ pixels, meaning $N \times N$ small squares of equal size. We denote these pixels by $\Box_{i,j}$, then

$$\Box_{i,j} = [x_i, x_{i+1}] \times [y_j, y_{j+1}], \qquad 0 \le i, j \le N-1,$$

where $x_i = y_i = -1 + \frac{2i}{N}$ for $0 \le i \le N$.

On each pixel, the image is represented by a single value. We have been computing the values of $\mathcal{A}_{2m}(x,y)$ on a grid of $N \times N$, in which the grid points are the centers of the pixels, and using these values to represent the image. In other words, the reconstructed image has been taken as a step function, whose value on the pixel $\Box_{i,j}$ is taken as $\mathcal{A}_{2m}(\frac{x_i+x_{i+1}}{2}, \frac{y_j+y_{j+1}}{2})$, the point evaluation at the center of the pixel. Such a choice is in common practice in image reconstruction.

As an alternative to the above representation of the image, we propose to use the average of $\mathcal{A}_{2m}(x,y)$ over a pixel as the value for the image on that pixel; that is, we propose to use the average

$$\begin{aligned} \mathcal{A}_{i,j} f := \mathcal{A}_{i,j}^{(m)} f &= \frac{1}{\operatorname{meas}(\Box_{i,j})} \int_{\Box_{i,j}} \mathcal{A}_{2m} f(x,y)\,dxdy \\ &= \frac{N^2}{4} \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} \mathcal{A}_{2m} f(x,y)\,dxdy \end{aligned} \tag{2.8}$$

as the value of the reconstructed image on the pixel $\Box_{i,j}$. From the point of view of approximation, using average over the pixel is better than using point evaluation at one point. If g is a function and $\bar{g}(x,y)$ is the step function whose value on each pixel is the average of g on that pixel, then the integral of $\bar{g}$ is equal to the integral of g. The same is clearly not true for the step function defined by point evaluation. From a practical point of view, the point evaluation on the pixel could lead to aliasing artifact due to under sampling, the average over the pixel avoids the problem of under sampling altogether.

How to represent the image, by average over the pixel or by point evaluation, appears to be independent of the reconstruction method. However, taking average turns out particularly simple for OPED, since the integral of $\mathcal{A}_{2m}(x,y)$ over a pixel can be explicitly given and effectively computed. This is certainly not the case for FBP, for which the integral cannot be computed exactly. In the following we derive an explicit formula for $\mathcal{A}_{i,j}f$ in (2.8).

By (2.3) and (2.4), the essential step in evaluating $F_{i,j}$ is to compute

$$I_{i,j}(k,\phi) := \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} U_k(x\cos\phi + y\sin\phi)dxdy.$$

Proposition 2.1. *For $0 \le i,j \le N-1$, we have $I_{i,j}(0,\phi) = 4N^{-2}$; for $k \ge 1$,*

$$I_{i,j}(k,0) = \frac{2N^{-1}}{k+1}\left[T_{k+1}(x_{i+1}) - T_{k+1}(x_i)\right],$$

and, if $\sin 2\phi \ne 0$,

$$I_{i,j}(k,\phi) = \frac{1}{(k+1)\sin(2\phi)}\left[G_{i,j}(k+2,\phi) - G_{i,j}(k,\phi)\right],$$

where

$$G_{i,j}(k,\phi) := \frac{1}{k}\left[T_k(\phi;x_{i+1},y_{j+1}) - T_k(\phi;x_{i+1},y_j) - T_k(\phi;x_i,y_{j+1}) + T_k(\phi;x_i,y_j)\right], \tag{2.9}$$

in which $T_m(t) = \cos m(\arccos t)$ is the Chebyshev polynomial of the first kind and we define $T_m(\phi;x,y) = T_m(x\cos\phi + y\sin\phi)$.

Proof. The case $k=0$, that is, $I_{i,j}(0,\phi) = 4N^{-2}$, is evident. For $k \ge 1$, we start with the relation that $T'_{k+1}(x) = (k+1)U_k(x)$, so that we can use integration by parts to get, assuming $\cos\phi\sin\phi \ne 0$,

$$\begin{aligned} I_{i,j}(k,\phi) &= \int_{x_i}^{x_{i+1}} \int_{y_j}^{y_{j+1}} \frac{1}{(k+1)\cos\phi}\frac{d}{dx}[T_{k+1}(x\cos\phi_+ y\sin\phi)]dxdy \\ &= \frac{1}{(k+1)\cos\phi}\int_{y_j}^{y_{j+1}} \left[T_{k+1}(\phi;x_{i+1},y) - T_{k+1}(\phi;x_i,y)\right]dy. \end{aligned}$$

Next we define a sequence of new functions:

$$S_0(t) = T_1(t), \quad S_1(t) = \frac{1}{4}T_2(t), \qquad S_k(t) := \frac{1}{2}\left[\frac{T_{k+1}(t)}{k+1} - \frac{T_{k-1}(t)}{k-1}\right], \quad k \ge 2.$$

It is easy to see that $S'_k(t) = T_k(t)$. Indeed, this can be verified as follows:

$$S'_k(t) = \frac{1}{2}\left[U_k(t) - U_{k-2}(t)\right] = \frac{1}{2}\left[\frac{\sin(k+1)\theta - \sin(k-1)\theta}{\sin\theta}\right] = \cos k\theta = T_k(t).$$

This relation allows us to do once more integration by parts. The result is

$$\begin{aligned} I_{i,j}(k,\phi) &= \frac{1}{(k+1)\cos\phi\sin\phi}\int_{y_j}^{y_{j+1}} \frac{d}{dy}[S_{k+1}(x_{i+1}\cos\phi + y\sin\phi) \\ &\qquad - S_{k+1}(x_i\cos\phi + y\sin\phi)]dy \\ &= \frac{1}{(k+1)\cos\phi\sin\phi}\left[S_{k+1}(\phi;x_{i+1},y_{j+1}) - S_{k+1}(\phi;x_{i+1},y_j) \right.\\ &\qquad \left. - S_{k+1}(\phi;x_i,y_{j+1}) + S_{k+1}(\phi;x_i,y_j)\right]. \end{aligned}$$

If $\phi = 0$, then we need to do integration by parts only once, which gives us

$$I_{i,j}(k,0) = 2N^{-1}\int_{x_i}^{x_{i+1}} U_k(x)dx = \frac{2N^{-1}}{(k+1)}\left[T_{k+1}(x_{i+1}) - T_{k+1}(x_i)\right].$$

This completes the proof. □

The explicit formula for the average $\mathcal{A}_{i,j}$ in (2.8) is given in the following:

Proposition 2.2. *For $m \ge 0$, $0 \le i \le j \le N-1$,*

$$\mathcal{A}_{i,j}f = \frac{N^2}{4}\int_{\square_{i,j}} \mathcal{A}_{2m}f(x,y)dxdy = \sum_{\nu=0}^{2m}\sum_{\lambda=0}^{2m}\mathcal{R}_{\phi_\nu}\left(f;\cos\psi_\lambda\right)T_{i,j}(\lambda,\nu)$$

where

$$T_{i,j}(\lambda,\mu) = \frac{N^2}{4(2m+1)^2}\sin\psi_\lambda\Phi_{i,j}(\cos\psi_\lambda,\phi_\nu), \qquad \psi_\lambda = \frac{(2\lambda+1)\pi}{4m+2},$$

with, for $\sin(2\phi) \ne 0$,

$$\begin{aligned}\Phi_{i,j}(t,\phi) =& \frac{1}{\sin(2\phi)}\Bigg(-2\sum_{k=2}^{2m}T_k(t)G_{i,j}(k,\phi)\\ &\qquad +U_{2m-1}(t)G_{i,j}(2m+1,\phi) + U_{2m}(t)G_{i,j}(2m+2,\phi)\Bigg)\end{aligned}$$

whiles for $\phi = 0$,

$$\Phi_{i,j}(t,0) = 2N^{-1}\sum_{k=0}^{2m}U_k(t)\left[T_{k+1}(x_{i+1}) - T_{k+1}(x_i)\right].$$

Proof. From (2.2) and the previous proposition, we immediately have

$$\begin{aligned}\Phi_{i,j}(t,\phi_\nu) &= \int_{\square_{i,j}}\Phi_\nu(t;x,y)dxdy\\ &= I_{i,j}(0,\phi_\nu) + \sum_{k=1}^{2m}(k+1)U_k(t)I_{i,j}(k,\phi_\nu)\\ &= N^{-2} + \frac{1}{\sin(2\phi)}\sum_{k=1}^{2m}U_k(t)\left(G_{i,j}(k+2,\phi) - G_{i,j}(k,\phi)\right),\end{aligned}$$

where we write $\phi = \phi_\nu$. Changing the summation order and using the fact that $U_{k-2}(t) - U_k(t) = -2T_k(t)$, $U_1(t) = 2\cos t = 2T_1(t)$ and $U_2(t) = 1 + 2\cos 2t =$

$1+2T_2(t)$, we get

$$\begin{aligned}
&\sum_{k=1}^{2m} U_k(t)\left(G_{i,j}(k+2,\phi)-G_{i,j}(k,\phi)\right)\\
&\quad=\sum_{k=3}^{2m+2} U_{k-2}(t)G_{i,j}(k,\phi)-\sum_{k=1}^{2m} U_k(t)G_{i,j}(k,\phi)\\
&\quad=-U_1(t)G_{i,j}(1,\phi)-U_2(t)G_{i,j}(2,\phi)+\sum_{k=3}^{2m}(U_{k-2}(t)-U_k(t))G_{i,j}(k,\phi)\\
&\qquad+U_{2m-1}(t)G_{i,j}(2m+1,\phi)+U_{2m}(t)G_{i,j}(2m+2,\phi)\\
&\quad=-G_{i,j}(2,\phi)-2\sum_{k=1}^{2m} T_k(t)G_{i,j}(k,\phi)\\
&\qquad\quad+U_{2m-1}(t)G_{i,j}(2m+1,\phi)+U_{2m}(t)G_{i,j}(2m+2,\phi).
\end{aligned}$$

Using the fact that $T_1(x)=x$, it follows readily that $G_{i,j}(1,\phi)=0$ for all i,j. Furthermore, using the fact that $T_2(x)=2x^2-1$, it is easy to see that $G_{i,j}(2,t)=N^{-2}\sin(2\phi)$. Hence the term $G_{i,j}(2,t)$ upon dividing by $\sin(2\phi)$ cancels the term N^{-2}. This proves the formula for $\Phi_{i,j}(t,\phi_\nu)$ for the case $\sin\phi\neq 0$. The case $\phi_\nu=0$ is easier. We only need to note that $T_1(x_{i+1})-T_1(x_i)=2N^{-1}$, the rest is straightforward. □

We can also write the formula for $\mathcal{A}_{i,j}f$ in a more compact form. Let us introduce the function $W_\phi(k)$ as follows:

If $\phi=0$, then for $0\le k\le 2m$,

$$W_0(k)=\frac{N}{2(2m+1)^2}\sum_{\lambda=0}^{2m}\mathcal{R}_\phi\left(f;\cos\psi_\lambda\right)\sin(k+1)\psi_\lambda. \tag{2.10}$$

For $1\le k\le 2m$, define

$$W_\phi(k)=-\frac{N^2}{2(2m+1)^2}\sum_{\lambda=0}^{2m}\sin\psi_\lambda\mathcal{R}_\phi\left(f;\cos\psi_\lambda\right)T_k(\cos\psi_\lambda); \tag{2.11}$$

for $k=2m+1$ and $k=2m+2$,

$$W_\phi(k)=\frac{N^2}{4(2m+1)^2}\sum_{\lambda=0}^{2m}\sin\psi_\lambda\mathcal{R}_\phi\left(f;\cos\psi_\lambda\right)U_{k-2}(\cos\psi_\lambda). \tag{2.12}$$

Then the formula for $\mathcal{A}_{i,j}f$ in the previous proposition can be rewritten in a compact form:

Proposition 2.3. *For $m\ge 0$, $0\le i,j\le N-1$,*

$$\begin{aligned}
\mathcal{A}_{i,j}f=&\sum_{k=0}^{2m} W_0(k)\left[T_{k+1}(x_{i+1})-T_{k+1}(x_i)\right]\\
&\qquad+\sum_{\nu=1}^{2m}\frac{1}{\sin(2\phi_\nu)}\sum_{k=2}^{2m+2} W_{\phi_\nu}(k)G_{i,j}(k,\phi_\nu).
\end{aligned} \tag{2.13}$$

The proof is straightforward, we only need to change the summation order. The advantage of the last formula lies in the numerical implementation, as we shall see in the following section.

The above expression of $\mathcal{A}_{i,j}f$ is the essential formula for the OPED algorithm with averaging. For a positive integer m, let $\mathcal{M}_m f$ denote the function that represents the reconstructed image. We define $\mathcal{M}_m$ as follows

$$\mathcal{M}_m f(x,y) := \mathcal{A}_{i,j}^{(m)} \quad \text{for } (x,y) \in \square_{i,j}, \quad \text{whenever} \quad \square_{i,j} \cap B = \square_{i,j}, \tag{2.14}$$

and $\mathcal{M}_m f(x,y) = 0$ for all other $\square_{i,j}$. In other words, we define $\mathcal{M}_m f(x,y) = \mathcal{A}_{i,j}^{(m)}$ when the pixel $\square_{i,j}$ lies entirely in B and define $\mathcal{M}_m f(x,y) = 0$ for all other pixel $\square_{i,j}$. An explanation is needed for the above definition. In practice, the object being investigated lies inside the region of interest (ROI) and we are interested only in the image. In our situation, ROI is B which is imbedded in the square $[-1,1]^2$. So (2.14) defines the approximate value of the reconstructed image when the pixel $\square_{i,j}$ is entirely inside B. Recall that we assume f to be zero outside B. Thus, it is natural to assume that $\mathcal{M}_m f(x,y) = 0$ on pixels that lie outside B. Furthermore, if a pixel $\square_{i,j}$ is partly inside B and partly out, then $A_{i,j}f$ will not be zero. In this case, we also define the value of $\mathcal{M}f(x,y)$ to be zero.

OPED Algorithm with Averaging. *For all i,j for which $\square_{i,j} \subset B$, $0 \le i,j \le N$, evaluate $A_{i,j}f$ in* (2.14).

The algorithm produces the step function $\mathcal{M}_m f(x,y)$. It represents the image on the union of all pixels inside B, that is, on

$$\Omega_m = \cup\{\square_{i,j} : \square_{i,j} \subset B,\ 0 \le i,j \le N\}.$$

In practice, we can always set the scale so that the object being investigated lies inside Ω_m in B.

The OPED algorithm with averaging converges under the same conditions as the OPED algorithm. We state this formally as:

Theorem 2.4. *If $f \in C^2(B)$, then $\mathcal{M}_m f$ converges uniformly to f on every compact subset inside B. More precisely,*

$$\max_{(x,y)\in\Omega_m} |\mathcal{M}_m f(x,y) - f(x,y)| \le c_f \frac{\log(m+1)}{m}$$

where c_f is a constant depend on f but independent of m.

Proof. Let $f_{i,j}$ denote the average of $f(x,y)$ on the pixel $\square_{i,j}$. For $(x,y) \in \square_{i,j} \subset \Omega_m$, the triangle inequality immediately implies that

$$|f(x,y) - \mathcal{M}_m f(x,y)| \le |f(x,y) - f_{i,j}(x,y)| + |f_{i,j}(x,y) - \mathcal{M}f(x,y)|.$$

Since $f \in C(B^2)$, the mean value theorem shows that the first term in is bounded by $(\|\partial_1 f\|_\infty + \|\partial_2 f\|_\infty)/(2m+1)$, where ∂_i denote the partial derivatives of f. Moreover,

$$|f_{i,j}(x,y) - \mathcal{M}f(x,y)| \le \frac{N^2}{4} \int_{\square_{i,j}} |f(u,v) - \mathcal{A}_{2m} f(u,v)| du dv \le c_f \frac{\log(m+1)}{m}$$

by (2.6). Putting these two estimates together completes the proof. □

3. Fast implementation of OPED algorithm with averaging

The structure of the expression $\mathcal{A}_{i,j}$ given in (2.13) allows us to use FFT to compute $W_\phi(k)$. The main task for computation, however, lies in the evaluation of the double sum in (2.13). Let $M := 2m+1$. The evaluation of the matrix $W_{\phi_\mu}(k)$, $0 \le k \le 2m$ and $0 \le \mu \le 2m$, costs $\mathcal{O}(M^2 \log M)$ operations (flops). The evaluation of the double sums for each $\mathcal{A}_{i,j} f$ costs $\mathcal{O}(M^2)$ and the total calculation for all $\mathcal{A}_{i,j} f$ costs $\mathcal{O}(N^2 M^2)$, which is much more than $\mathcal{O}(M^2 \log M)$ when $N \sim M$, meaning that N and M are in the same order. The same structure holds for the original OPED. In [10], a fast implementation of OPED is introduced by using a linear interpolation step, which reduces the total cost of the reconstruction to $\mathcal{O}(N^3)$, assuming $N \sim M$, which is in the same order as the algorithm. A similar method can be used to implement OPED with averaging, which we now describe.

First let us consider the evaluation of $W_{\phi_\nu}(k)$. Let

$$y_{\nu,k} := \sum_{\lambda=0}^{2m} \mathcal{R}_{\phi_\nu}(f; \cos\psi_\lambda) \sin(k+1)\psi_\lambda, \qquad 0 \le k \le 2m.$$

Using the fact that $2\sin\psi\cos k\psi = \sin(k+1)\psi - \sin(k-1)\psi$, we can write $W_{\phi_\nu}(k)$ in terms of $y_{\nu,k}$. For example, $W_{\phi_\nu}(k) = N^2(y_{\nu,k-2} - y_{\nu,k})/((2m+1)^2)$ for $2 \le k \le 2m$. For each ν, the numbers $y_{\nu,k}$, $0 \le k \le 2m$, can be computed by the FFT for the discrete sine transform, so are $W_{\phi_\nu}(k)$.

Next, assuming that all values of $W_{\phi_\nu}(k)$ are already computed. Let

$$\mathcal{B}_{i,j} := \sum_{\nu=1}^{2m} \sum_{k=1}^{2m+2} \frac{1}{k} W_{\phi_\nu}(k) T_k(\phi_\nu; x_i, y_j), \quad 0 \le i,j \le N.$$

By (2.9), we can write $\mathcal{A}_{i,j}$ in (2.13) as

$$\mathcal{A}_{i,j} f = \sum_{k=0}^{2m} W_0(k) \left[T_{k+1}(x_{i+1}) - T_{k+1}(x_i)\right] + \mathcal{B}_{i,j} - \mathcal{B}_{i+1,j} - \mathcal{B}_{i,j+1} + \mathcal{B}_{i+1,j+1}.$$

The first term is a single sum, which can be evaluated using the usual method of evaluating the Chebyshev series. To evaluate the other terms, we essentially need to compute $\mathcal{B}_{i,j}$ for $0 \le i,j \le N$, for which we introduce our interpolation step. Let us define

$$\alpha_\nu(\theta) := \sum_{k=1}^{2m+2} \frac{1}{k} W_{\phi_\nu}(k) \cos k\theta,$$

which is the inner sum of $\mathcal{B}_{i,j}$ when $\theta = x_i \cos\phi_\nu + y_j \sin\phi_\nu$. The FFT for discrete cosine transform can be used to evaluate the numbers

$$\alpha_{l,\nu} := \alpha_\nu(\xi_l), \qquad \xi_l := \frac{(l+1/2)\pi}{2m+2}, \qquad l = 0, 1, \ldots, 2m+1,$$

effectively. For each (i,j) and ν, we then choose an integer l by

$$l = \left\lfloor \frac{2m+2}{\pi} \arccos t_{i,j}^\nu - \frac{1}{2} \right\rfloor, \qquad \text{where} \quad t_{i,j}^\nu = x_i \cos\phi_\nu + y_j \sin\phi_\nu,$$

which means that $t_{i,j}^\mu$ lies between ξ_l and ξ_{l+1}, so that we can use the value of the linear interpolation between $\alpha_{l,\nu}$ and $\alpha_{l+1,\nu}$ as an approximation to the inner sum of $\mathcal{B}_{i,j}$. The linear interpolation is given by

$$\ell_\nu(\theta) = u_\nu(\theta)\alpha_{l+1,\nu} + (1 - u_\nu(\theta))\alpha_{l,\nu}, \qquad u_\nu(\theta) := \frac{\theta - \xi_l}{\xi_{l+1} - \xi_l}, \tag{3.1}$$

where $\xi_l \le \theta \le \xi_{l+1}$, and the inner sum of $\mathcal{B}_{i,j}$ is approximated by $\ell_\nu(t^\nu_{i,j})$. Then the fast implementation of OPED with averaging is given as follows:

Fast OPED algorithm with averaging: Let m be a positive integer,

Step 1. For each $\nu = 0, \ldots, 2m$, use FFT to compute for each $k = 0, \ldots, 2m$,

$$y_{\nu,k} = \sum_{j=0}^{2m} \mathcal{R}_{\phi_\nu} f(\cos\psi_\lambda)\sin(k+1)\psi_j.$$

Step 2. Set $\Delta_0 := N/(2m+1)^2$ and $\Delta_\nu := N\Delta_0$. Compute $W_0(k) = \Delta_0 y_{0,k}$ for $0 \le k \le 2m$, and for $1 \le \nu \le 2m$ compute

$$W_{\phi_\nu}(k) = \Delta_\nu(y_{\nu,k-2} - y_{\nu,k}), \qquad 2 \le k \le 2m,$$

and also compute

$$W_{\phi_\nu}(2m+1) = \Delta_{2m-1} y_{\nu,2m-1}, \qquad W_{\phi_\nu}(2m+2) = \Delta_{2m} y_{\nu,2m}.$$

Step 3. For each $l = 0, 1, \ldots, 2m+1$, use FFT to compute

$$\alpha_{l,\nu} := \sum_{k=1}^{2m+1} \frac{1}{k+1} W_{\phi_\nu}(k) \cos\left(k\frac{(l+1/2)\pi}{2m+2}\right).$$

Step 4. For each pair (i,j) such that $\Box_{i,j} \subset \Omega_m$, determine integers l such that

$$l = \left\lfloor \frac{2m+2}{\pi} \arccos t^\nu_{i,j} - \frac{1}{2} \right\rfloor, \qquad \textit{where} \quad t^\nu_{i,j} = x_i \cos\phi_\nu + y_j \sin\phi_\nu,$$

and evaluate

$$\mathcal{B}_{i,j} = \sum_{\nu=1}^{2m} \frac{1}{\sin(2\phi_\nu)} \left[(1-u_\nu)\alpha_{l,\nu} + u_\nu \alpha_{l+1,\nu}\right],$$

where $u_\nu = (2m+1)t^\nu_{i,j}/\pi - (l+1/2)$.

Step 5. For each pair (i,j) such that $\Box_{i,j} \subset \Omega_m$, evaluate

$$\mathcal{A}_{i,j} = \sum_{k=0}^{2m} W_0(k)\left[T_{k+1}(x_{i+1}) - T_{k+1}(x_i)\right] + \mathcal{B}_{i,j} - \mathcal{B}_{i+1,j} - \mathcal{B}_{i,j+1} + \mathcal{B}_{i+1,j+1}.$$

The algorithm uses FFT twice, the main step in Step 4 is a single sum whose evaluation costs $\mathcal{O}(M)$ operations. Hence, the cost of evaluations on an $M \times M$ grid with $M \approx N$ is $O(N^3)$, which is in the same order of magnitude as the fast implementation of OPED [10], as well as FBP algorithm ([5]).

In the above algorithm we evaluated α_ν at $\xi_l = (l+1/2)\pi/(2m+2)$. The points $t_l = \cos\xi_i$ are the zeros of the Chebyshev polynomial $T_{2m+2}(t)$, which are more clustered toward the end of the interval $[-1,1]$ and coarser in the middle. When we use them to do interpolation in Step 4, it could happen that all four corners of a pixel lie between the same two interpolation points, which would cause a problem in the difference $\mathcal{B}_{i+1,j+1} - \mathcal{B}_{i+1,j} - \mathcal{B}_{i,j+1} + \mathcal{B}_{i,j}$. One way to overcome this potential problem is to use finer interpolation points, say, $\xi_l = l\pi/(2M)$ for $0 \le l \le 2M$, or choose ξ_l such that $t_l = \cos\xi_l$ are equal spaced points in $[-1,1]$, for example, $t_l = l\pi/(2m+1)$, $0 \le l \le 2m+1)$. In either of these alternatives, we will not be able to use FFT to compute $\alpha_\nu(\xi_l)$. However, since the computation of $\alpha_\nu(\xi_l)$ costs at most $\mathcal{O}(M^3)$ flops, the total cost of the implementation remains at $\mathcal{O}(N^3)$ if $M \approx N$.

4. Implementation and Result

Our numerical example is conducted on the Shepp-Logan head phantom [8] (see Fig. 1), which is an analytic phantom, defined by a step function. The function is not continuous and has jumps at the boundary of every ellipse in the image. We reconstruct the image with the OPED algorithm and the OPED algorithm with averaging, both with fast implementation that includes the interpolation step. We use the FFT for discrete sine/cosine transform in the package FFTW (http://www.fftw.org/).

The reconstruction is carried out on a CELSIUS R610 computer with two Intel Xeon(TM) CPU, each 3065 MHz, and 4 GB RAM. The code is written in C language. When the reconstruction was carried out with $m = 255$ and evaluated on a 512×512 pixels, it took 3 seconds with OPED algorithm, and 13 seconds with OPED algorithm with averaging. However, the OPED algorithm is optimized, while the OPED algorithm with averaging has not yet been optimized.

Figure 1 depicts the images reconstructed by OPED and OPED with averaging. The top one is the image reconstructed by OPED, the bottom one is the one reconstructed by OPED with averaging. Both images are constructed with $m = 505$ and the size of the images is 256×256 pixels.

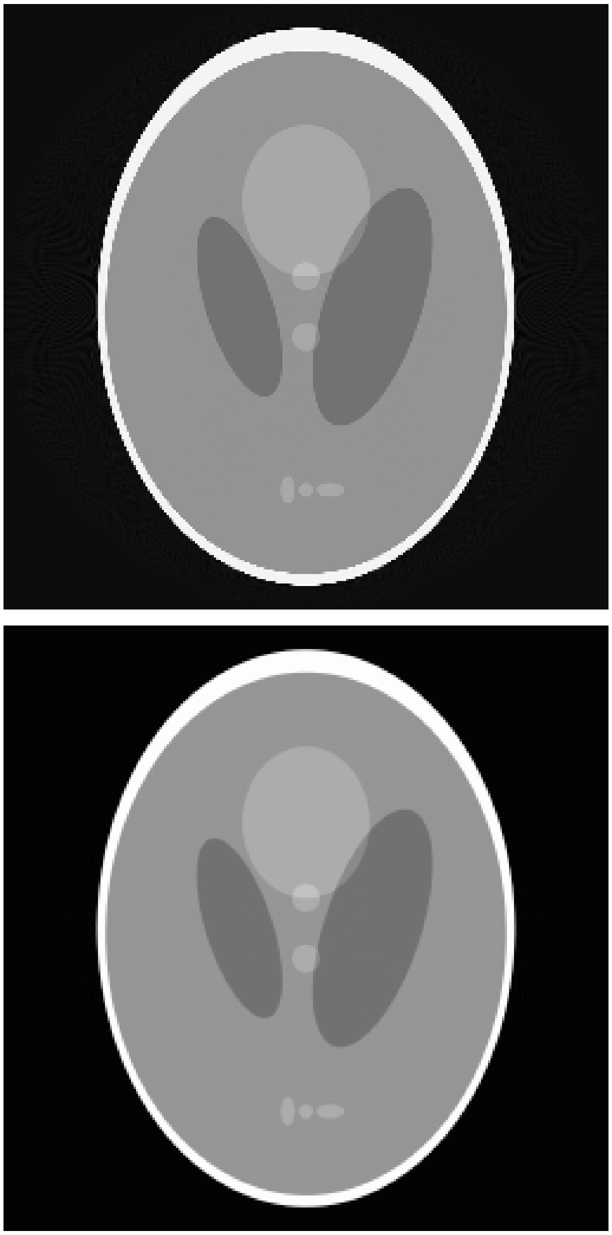

Figure 1. Top: reconstruction by OPED algorithm with $m = 505$. Bottom: reconstruction by OPED algorithm with averaging with $m = 505$.

These images show clearly that OPED with averaging produces images that have smoother edges. In Figure 2, the differences between the original Shepp-Logan phantom and the two reconstructed images in Figure 1 are depicted, in which we multiplied the error by 1000 to increase the visuality. These images show clearly that images produced by OPED with averaging have less noises and less aliasing artifacts.

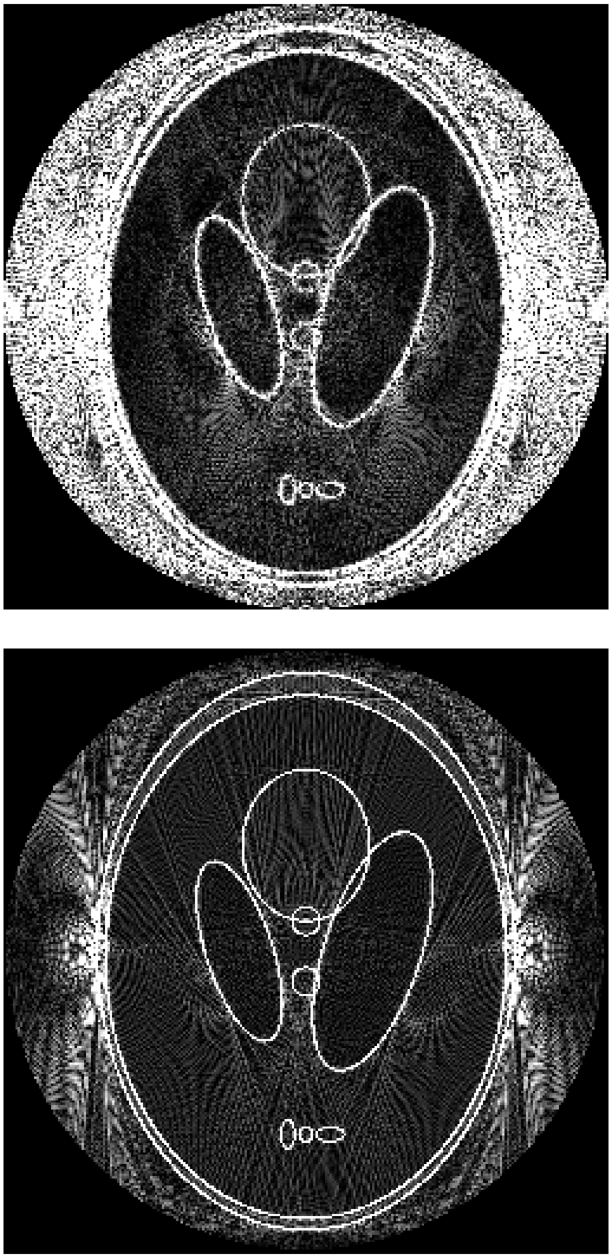

Figure 2. The difference between the original image and the reconstruction by OPED (top) and the reconstruction by OPED with average (bottom).

To measure the image quality, we compute the errors between the reconstruction and the original image. Let X denote an image, represented by its pixel values, so that $X = \{X_i : 1 \le i \le N\}$, where N is the number of pixels in the image. Let X^R denote the reconstructed image, $X^R = \{X_i^R : 1 \le i \le N\}$. The relative least square error (RLSE) between X and X^R is defined by

$$Q_{RSE} = \frac{\left(\sum_i (X_i^R - X_i)^2\right)^{1/2}}{\left(\sum_i^N (X_i^R)^2\right)^{1/2}},$$

and the mean error (ME) between X and X^R is defined by

$$Q_{ME} = \frac{1}{N} \sum_{i=1}^{N} |X_i - X_i^R|.$$

In our case $N = 256 \times 256 = 256^2$. For fast OPED, the reconstruction is evaluated at the center points of the pixels, we take X_i also as the value at those points; in other words, the error on the pixel is computed as the difference of the value of the original image on the center point and the reconstructed value. For OPED with averaging, we take X_i^R as the averaging of the original image over the pixel; in other words, the error on the pixel is computed as the difference of the average of the original image over the pixel and the reconstructed value. The result of the reconstruction in Figure 1 is reported in Table 1, in which ORIG stands for the original Shepp-Logan phantom, OPED and OPEDa stand for reconstructed image by OPED and by OPED with averaging, respectively. For example, ORIG vs OPED means the error between the original phantom and the reconstruction by OPED.

	ORIG vs OPED	ORIG vs OPEDa	
RLSE	0.0516492	0.0032618	
ME	0.00781484	0.00133138	

TABLE 1. Error Estimates of OPED and OPED with averaging

The results in the table show that OPED with averaging is better in both relative least square error and mean error.

In conclusion, our numerical test shows that the image reconstructed by OPED with averaging has better visual appearance and smaller errors. Furthermore, it also has less aliasing artifacts, which concurs with our expectation.

REFERENCES

[1] H. H. Barret and K. J. Myers, *Foundations of Image Science*, John Wiley & Sons, Inc., Hoboken, New Jersey, 2004.

[2] B. Bojanov and I. K. Georgieva, Interpolation by bivariate polynomials based on Radon projections, *Studia Math*, **162** (2004), 141 - 160.

[3] T. Bortfeld and U. Oelfke, Fast and exact 2D image reconstruction by means of Chebyshev decomposition and backprojection, *Phys. Med. Biol.*. **44** (1999), 1105-1120.

[4] H. de las Heras, O. Tischenko, Yuan Xu and C. Hoeschen, Comparison of the interpolation functions to improve a rebinning-free CT-reconstruction algorithm, *submitted*, 2006.

[5] F. Natterer, *The mathematics of computerized tomography*, Reprint of the 1986 original. Classics in Applied Mathematics, 32. SIAM, Philadelphia, PA, 2001.

[6] P. Petrushev, Approximation by ridge functions and neural networks, *SIAM J. Math. Anal.* **30** (1999), 155–189.

[7] M. Reimer, *Constructive Theory of Multivariate Functions*, Mannheim: BI-Wiss.-verlag, 1990.

[8] L. Shepp and B. Logan, The Fourier reconstruction of a head section, *IEEE Trans. Nucl. Sci.*, **NS-21**, 1974, 21-43.

[9] Yuan Xu, A direct approach for reconstruction of images from Radon projections, *Adv. in Applied Math.*, **36** (2006), 388-420.

[10] Yuan Xu and O. Tischenko, Fast OPED algorithm for reconstruction of images from Radon data, $s$ubmitted, 2006. ArXiv:math.NA/0703617

[11] Yuan Xu, O. Tischenko and C. Hoeschen, Approximation and Reconstruction from Attenuated Radon Projections, *SIAM J. Numer. Anal.*, **45** (2007), 108-132.

[12] Yuan Xu, O. Tischenko and C. Hoeschen, A new reconstruction algorithm for Radon Data, *Proc. SPIE, Medical Imaging 2006: Physics of Medical Imaging*, **vol. 6142**, p. 791-798.

[13] A. Zygmund, *Trigonometric Series*, Cambridge Univ. Press, Cambridge, 1959.

Department of Mathematics, University of Oregon, Eugene, Oregon 97403-1222.
E-mail address: yuan@math.uoregon.edu

Institute of Radiation Protection, GSF - National Research Center for Environment and Health, D-85764 Neuherberg, Germany
E-mail address: oleg.tischenko@gsf.de
E-mail address: christoph.hoeschen@gsf.de