Orthogonal discrete periodic Radon transform.
Part II: applications

Daniel P.K. Lun*, T.C. Hsung, T.W. Shen
Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Received 14 September 2001; received in revised form 28 November 2002

Abstract

In this paper, we study the properties and possible applications of the newly proposed orthogonal discrete periodic Radon transform (ODPRT). Similar to its previous version, the new ODPRT also possesses the useful properties such as the discrete Fourier slice theorem and the circular convolution property. They enable us to convert a 2-D application into some 1-D ones such that the computational complexity is greatly reduced. Two examples of using ODPRT in the realization of 2-D circular convolution and blind image resolution are illustrated. With the fast ODPRT algorithm, efficient realization of 2-D circular convolution is achieved. For the realization of blind image restoration, we convert the 2-D problem into some 1-D ones that reduces the computation time and memory requirement. Besides, ODPRT adds more constraints to the restoration problem in the transform domain that makes the restoration solution better. Significant improvement is obtained in each case when comparing with the traditional approaches in terms of quality and computation complexity. They illustrate the potentially widespread applications of the proposed technique.
© 2003 Elsevier Science B.V. All rights reserved.

1. Introduction

The orthogonal discrete periodic Radon transform (ODPRT) is proposed recently [8]. Similar to its predecessor, namely discrete periodic Radon transform (DPRT), ODPRT also possesses the important properties, such as the discrete Fourier slice theorem and the circular convolution property that allow a 2-D application to be computed using some 1-D approaches. In addition, due to the orthogonal structure of the ODPRT projections and fast algorithms for the forward and inverse transforms [8], the computational complexity of the applications using ODPRT is further reduced. In this paper, we study the useful properties of ODPRT and its possible applications in the computation of 2-D circular convolution and blind image restoration.

For the computation of 2-D convolution, efficient techniques have been studied for long due to its widespread applications in digital image processing. A popular approach for computing 2-D convolution is to convert the original 2-D convolution into the 2-D cyclic convolution, and utilize the fast algorithms for computing the 2-D cyclic convolution to resolve the problem. There have been many fast algorithms for the computation of 2-D circular convolution. The most traditional one is to use the convolution theorem of DFT which computes the 2-D circular convolution...
in the frequency domain as simple multiplications. Further improvement is achieved by using the polynomial transform proposed by Nussbaumer [13], who suggested to map a 2-D convolution into 1-D convolutions and polynomial multiplications. Based on this idea, Truong et al. [18] showed that a combination of a fast polynomial transform (FPT) and the Chinese remainder theorem (CRT) can be used to efficiently compute a 2-D convolution of two \( N_1 \times N_2 \) arrays, where \( N_1 = 2^m \) and \( N_2 = 2^{m-r+1} \) for \( 1 \leq r \leq m \).

Other alternatives for the computation of 2-D circular convolution include those that make use of number theoretical transform [13]. Fast algorithms that make use of a combination of number theoretical transform and the polynomial transform were proposed [15] for the computation of 2-D integer circular convolution. A common point for the above approaches is that they are all algebraic means for solving the problem of 2-D circular convolution. The transform domain of, for instance, the polynomial transform or the number theoretical transform has least, if any, physical meaning. Recently, DPRT was proposed for the computation of 2-D circular convolution based on its circular convolution property [9]. While there is no significant improvement in computational complexity over the previous approaches, it gives a geometrical point of view to the whole problem, which allows a better understanding to the computing structure of the algorithm to facilitate efficient realization.

While the ODPRT is useful in computing the 2-D convolution of two functions, it is also useful in estimating the original functions if the 2-D convolution result is available. The later problem is known as blind deconvolution, which is a well-known ill-posed problem. One of the major applications of 2-D blind deconvolution is on blind image restoration, in which the original image is to be recovered from its blurred version. One of the important classes of approach for solving the blind image restoration problem is based on the ARMA parameter estimation method. This kind of method models the true image as a 2-D autoregressive (AR) process and the point spread function (PSF) as a 2-D moving average (MA) process. Based on these models, the resulting blurred image is represented as an autoregressive moving average (ARMA) process. Identifying the ARMA parameters allows us to identify the true image and PSF. The maximum-likelihood (ML) estimation [6] and generalized cross-validation (GCV) approaches [16] are two of the most popular approaches in this category. Nevertheless, a major limitation of the ARMA approaches is that their performance can deteriorate significantly when the number of estimated parameters is large (for example, due to large PSF). While the phase information of the image is preserved in some types of blurring process [7,17], there is another class of much simple restoration techniques that make use of the Fourier phase for image restoration. It is achieved by alternatively switching the blurred image between the spatial and Fourier domain and applying appropriate constraints in both domains [3,4,7].

In this paper, the application of ODPRT in the computation of circular convolution and blind image restoration are studied. Due to the circular convolution property of ODPRT, a 2-D circular convolution is converted into some 1-D ones that reduce the computational complexity to a great extent. Further reduction is achieved by ODPRT due to its orthogonal projections and the fast algorithms for the computation of the forward and reverse transform. As far as the blind image restoration is concerned, traditional Fourier phase approaches [3,4,7] make use of 2-D FFT and inverse 2-D FFT to transform the estimate of the blurred image between time and frequency domains alternatively. By using the discrete Fourier slice theorem of ODPRT, the 2-D FFT and the inverse 2-D FFT can be converted to become some 1-D FFTs and inverse 1-D FFTs that greatly simplifies the problem. Furthermore, since the phase information of an image is preserved in the ODPRT domain, the same constraints as in the previous approaches can be applied. In addition, we make use of the convolution property of the ODPRT to impose further constraint on the restoration process such that the rate of convergence is increased and the quality of the estimated image is also improved as compared to the traditional approaches [3,4,7].

This paper is organized as follows. In Section 2, the definition and various properties of ODPRT are reviewed. In Section 3, we suggest the application of ODPRT in the computation of 2-D circular convolution. Furthermore, we investigate the use of ODPRT to the problem of blind image restoration in Section 4. The paper is then concluded in Section 5.
2. Orthogonal discrete periodic Radon transform

Denote subsets of integers \(\{0, 1, 2, \ldots, N-1\}\) as \(Z_N, Z_N \times Z_N = Z_N^2\). Let \(I^p(Z_N^2)\) be a set of measurable and square-summable functions over \(Z_N^2\), and the residue of \(a\) modulo \(P\) as \(\langle a \rangle_P\). The ODPRT on \(Z_N^2\), where \(n\) is any positive integer, is described in definition 1 below.

**Definition 1** (ODPRT). The ODPRT of \(f(x, y) \in I^p(Z_N^2)\) (ODPRT2: \(I^p(Z_N^2) \rightarrow I^p((Z_N \times Z_{2^n-1}) \cup Z_N \cup Z_{2^n-1})\)), consists of three components: projections \(f_m^p, f_s^q\), and spatially aliased function \(f^1\), which are defined as follows;

\[
\begin{align*}
  f_m^p(d) &= f_m^p(d) - f_s^q(d + N/2), \\
  f_s^q(d) &= f_s^q(d) - f_s^q(d + N/2),
\end{align*}
\]

where \(m \in Z_N, d, s \in Z_{N/2}\), and,

\[
\begin{align*}
  f^1(x, y) &= f(x, y) + f(x, y + N/2) \\
  &+ f(x + N/2, y) \\
  &+ f(x + N/2, y + N/2),
\end{align*}
\]

where \(x, y \in Z_{N/2}\) and the DPRT projections is defined as

\[
\begin{align*}
  f_s^q(d) &= \sum_{y=0}^{N-1} f(\langle d + 2sy \rangle_N, y), \\
  f_m^p(d) &= \sum_{x=0}^{N-1} f(x, \langle d + mx \rangle_N),
\end{align*}
\]

where \(d, m \in Z_N\) and \(s \in Z_{N/2}\).

The reconstruction of ODPRT is much simpler than its predecessor, DPRT. It implements a “back-projection” from \(f_m^p, f_s^q\).

**Proposition 1.** The function \(f(x, y) \in I^p(Z_N^2)\) can be reconstructed from its ODPRT by

\[
\begin{align*}
  f(x, y) &= \frac{1}{4} f^1(\langle x \rangle_{N/2}, \langle y \rangle_{N/2}) \\
  &+ \frac{1}{2N} \sum_{m=0}^{N-1} f_m^p(\langle y - mx \rangle_N) \\
  &+ \frac{1}{2N} \sum_{x=0}^{N/2-1} f_s^q(\langle x - 2sy \rangle_N),
\end{align*}
\]

where \(x, y \in Z_N, N = 2^n\).

The proof can be found in [8]. In the following sections, we shall investigate the useful properties of ODPRT.

2.1. Discrete Fourier slice theorem of ODPRT

We are given a 2-D function \(f(x, y)\) which has the DFT spectrum \(F(u, v)\), where \(x, y, u, v \in Z_N\). From [5], we know that the DFT spectrum of a 2-D function is related to the DFT spectrum of its DPRT projections through the discrete Fourier slice theorem:

\[
\begin{align*}
  F(u, \langle -2su \rangle_N) &= \sum_{d=0}^{N-1} f_s^q(d) \exp(-j\pi ud/N), \\
  F(\langle -mv \rangle_N, v) &= \sum_{d=0}^{N-1} f_m^p(d) \exp(-j\pi vd/N),
\end{align*}
\]

where \(f_m^p(d)\) and \(f_s^q(d)\) are the DPRT of \(f; m, d \in Z_N; s \in Z_{N/2}\).

Considering only the odd frequencies, Eqs. (6) and (7) can be rewritten as follows:

\[
\begin{align*}
  F(2u' + 1, \langle -2s(2u' + 1) \rangle_N) &= \sum_{d=0}^{N-1} f_s^q(d) \exp(-j\pi ud/2) \exp(-j\pi ud/2) \\
  &= \sum_{d=0}^{N-1} f_s^q(d) \exp(-j\pi ud(2u' + 1)/2) \exp(-j\pi ud(2u' + 1)/2),
\end{align*}
\]

\[
\begin{align*}
  F(\langle -m(2v' + 1) \rangle_N, 2v' + 1) &= \sum_{d=0}^{N-1} f_m^p(d) \exp(-j\pi vd(2v' + 1)/2) \exp(-j\pi vd(2v' + 1)/2),
\end{align*}
\]

where \(u', v' \in Z_{N/2}\). Hence,

\[
\begin{align*}
  F(2u' + 1, \langle -2s(2u' + 1) \rangle_N) &= \sum_{d=0}^{N/2-1} (f_s^q(d) - f_s^q(d + N/2)) \\
  &\times \exp(-j\pi ud(2u' + 1)/2) \exp(-j\pi ud(2u' + 1)/2),
\end{align*}
\]

\[
\begin{align*}
  F(\langle -m(2v' + 1) \rangle_N, 2v' + 1) &= \sum_{d=0}^{N/2-1} (f_m^p(d) - f_m^p(d + N/2)) \\
  &\times \exp(-j\pi vd(2v' + 1)/2) \exp(-j\pi vd(2v' + 1)/2).
\end{align*}
\]
Consequently,
\[
F(2u' + 1, \langle -2s(2u' + 1) \rangle) = \sum_{d=0}^{N/2-1} f^p_s(d) e^{-j2\pi d(u'+1/2)/N}/2,
\]
\[
F(\langle -m(2u' + 1) \rangle, 2u' + 1) = \sum_{d=0}^{N/2-1} f^p_m(d) e^{-j2\pi d(u'+1/2)/N}/2.
\]

This proves the following discrete Fourier slice theorem for ODPRT:

**Theorem 3** (Discrete Fourier slice theorem for ODPRT). For the ODPRT \{f^p_s, f^p_m\} of \(f(x, y) \in l^2(Z_N^2)\), where \(N = 2^n\),
\[
F(\langle -m(2u + 1) \rangle, 2u + 1) = \sum_{d=0}^{N/2-1} f^p_m(d) e^{-j2\pi d(v+1/2)/N}/2 = F^p_m(v),
\]
\[
F(2u + 1, \langle -2s(2u + 1) \rangle) = \sum_{d=0}^{N/2-1} f^p_s(d) e^{-j2\pi d(u+1/2)/N}/2 = F^p_s(u),
\]
where \(F(u, v)\) is the DFT of the 2-D function \(f(x, y)\), \(m \in Z_N\), \(s, u, v \in Z_{N/2}\).

Eqs. (12) and (13) show that the spectrum of the ODPRT projections corresponds to the odd frequency spectrum of the original function. Note that \(F^p_m\) and \(F^p_s\) are the so-called reduced DFT [13] of \(f^p_m\) and \(f^p_s\). It can be easily shown from Eqs. (12) and (13) that \(F^p_m\) and \(F^p_s\) are related to \(F^p_{m'}\) and \(F^p_{s'}\) as follows:
\[
F^p_m(v) = F^p_m(2v + 1),
\]
\[
F^p_s(u) = F^p_s(2u + 1),
\]
where \(u, v \in Z_{N/2}\). For example, the 1-D discrete Fourier transform of \(f^p_1(d)\) w.r.t. \(d\) is the slice \(F^p_1(v)\) in the 2-D discrete Fourier transform of \(f\), as shown in Fig. 1.

---

**2.2. Circular convolution property**

ODPR also possess the circular convolution property as DPRT. We are given a 2-D function \(g(x, y)\) which is the result of a 2-D circular convolution between two 2-D functions \(f(x, y)\) and \(h(x, y)\), of which all functions are of size \(N \times N\), where \(N = 2^n\), that is,
\[
g(x, y) = f(x, y) \ast_2 h(x, y),
\]
where \(\ast_2\) stands for 2-D circular convolution. We are also given that the ODPRT of \(g(x, y)\), \(h(x, y)\), \(f(x, y)\) are \(g^p_m(d)\) and \(g^p_s(d)\) and \(h^p_m(d)\) and \(h^p_s(d)\) and \(f^p_1(d)\) and \(f^p_1(d)\), respectively, where \(m \in Z_N\); \(s, d \in Z_{N/2}\).

From the definition of ODPRT,
\[
g^p_m(l) = g^c_m(l) - g^c_m(l + N/2)
\]
\[
= \sum_{d=0}^{N-1} f^c_m(l) h^c_m((l - d)N)
\]
\[
= - \sum_{d=0}^{N-1} f^c_m(l) h^c_m((l - d - N/2)N)
\]
\[
= \sum_{d=0}^{N/2-1} f^c_m(l) h^c_m((l - d)N)
\]
\[\begin{align*}
&\sum_{d=0}^{N/2-1} f_m^p(l + N/2)h_m^p((l - d + N/2)_N) \\
&- \sum_{d=0}^{N/2-1} f_m^p(l)h_m^p((l - d + N/2)_N) \\
&- \sum_{d=0}^{N/2-1} f_m^p(l + N/2)h_m^p((l - d)_N) \\
&- \sum_{d=0}^{N/2-1} f_m^p(l)_N \\
&\quad (f_m^p(l) - f_m^p(l + N/2))(h_m^p((l - d)_N) \\
&- h_m^p((l - d + N/2)_N)) \\
&= \sum_{d=0}^{N/2-1} f_m^p(l)h_m^p((l - d)_N),
\end{align*}\]

where \(l = 0, 1, \ldots, N/2 - 1\).

The proof for \(g^p_s(d)\) is similar. To summarize, we have

\[g_m^p(d) = h_m^p(d) \otimes f_m^p(d),\]

\[g_s^p(d) = h_s^p(d) \otimes f_s^p(d),\]

where \(m \in \mathbb{Z}_N; s, d \in \mathbb{Z}_{N/2}\). The symbol \(\otimes\) stands for 1-D circular convolution. Consequently, we have the following theorem:

**Theorem 4** (Convolution property of ODPRT). For functions \(g, f, h \in \ell^1(\mathbb{Z}_N^2)\), where \(g\) is the 2-D cyclic convolution of \(f\) and \(h\), i.e.,

\[g(x, y) = f(x, y) \otimes_2 h(x, y),\]

where \(\otimes_2\) stands for 2-D circular convolution; \(x, y \in \mathbb{Z}_N^2\). Then, the ODPRT of \(g\) can be obtained from the ODPRT of \(f\) and \(h\) as follows:

\[g_m^p(d) = h_m^p(d) \otimes f_m^p(d),\]  \hspace{1cm} (16)

\[g_s^p(d) = h_s^p(d) \otimes f_s^p(d),\]  \hspace{1cm} (17)

where \(m \in \mathbb{Z}_N; s, d \in \mathbb{Z}_{N/2}\). The symbol \(\otimes\) stands for 1-D circular convolution. The functions \(g_m^p(d)\) and \(g_s^p(d)\), \(h_m^p(d)\) and \(h_s^p(d)\), \(f_m^p(d)\) and \(f_s^p(d)\) are the ODPRTs of \(g(x, y), h(x, y), f(x, y)\), respectively.

### 3. 2-D Circular convolution using ODPRT

We have demonstrated in Eqs. (16) and (17) that ODPRT also possesses the circular convolution property. As shown in the equations, every set of orthogonal projections, \(g_m^p(d)\) and \(g_s^p(d)\), is the result of the 1-D circular convolution of the orthogonal projections of \(f\) and \(h\). It shows that the computation of a 2-D circular convolution can be decomposed to a number of 1-D ones. The length of the 1-D convolution depends on the length of the orthogonal projections. Due to the recursive structure of ODPRT, projections of different length have resulted. An example can be found in Fig. 2 for a 2-D circular convolution of two \(8 \times 8\) 2-D functions using the fast ODPRT algorithm [8]. In the figure, the function \(h\) is assumed to be known a priori. Hence its ODPRT is assumed to be pre-computed. Fig. 2 also shows how the final result is reconstructed using the fast inverse ODPRT algorithm [8]. As it can be seen in the figures, the fast inverse ODPRT algorithm has a structure similar to the forward one. This feature will be particularly useful in actual implementation.

#### 3.1. The algorithm

Let us summarize the procedure for computing the 2-D circular convolution using ODPRT as follows:

Given two 2-D functions \(f(x, y)\) and \(h(x, y)\) with size \(N \times N\) such that \(x, y \in \mathbb{Z}_N^2\):

1. Generate the ODPRT projections, \(h_m^p(d)\) and \(h_s^p(d)\), \(f_m^p(d)\) and \(f_s^p(d)\), recursively using the fast ODPRT algorithm as shown in [8].
2. Perform 1-D circular convolutions on \(h_m^p(d)\) and \(h_s^p(d)\), \(f_m^p(d)\) and \(f_s^p(d)\).
3. Repeat steps 1 and 2 with the spatially aliased functions, \(f^1\) and \(h^1\), as input. Stop if trivial solution is obtained (for instance, when \(f^1\) and \(h^1\) are functions with size \(2 \times 2\)).

For the implementation of 1-D circular convolutions, we can apply the split-radix FFT algorithm [2] if the input data are complex. In the case that the input data are real, we can use the real-valued FFT to realize the 1-D convolution. 1-D short convolution [13]
algorithms can also be used to either complex or real cases if the sequence length is short.

3.2. Computational complexity

Let us have an analysis on the computational complexity of the proposed approach. Assume that the input data are real. Let $M_n^r$ and $A_n^r$ be the numbers of real multiplications and additions, respectively, needed to perform an $N \times N$ real circular convolution with the proposed ODPRT algorithm, where $N = 2^n$. We know that an $N \times N$ 2-D circular convolution can be decomposed into an $N/2 \times N/2$ 2-D circular convolution and $3N/2$ length-$N/2$ 1-D circular convolutions. In this case, we can determine that the number of multiplications required by the algorithm is as follows:

$$M_n^r = M_{n-1}^r + \frac{3N}{2} \hat{M}_{n-1}^r,$$

where $\hat{M}_n^r$ represents the number of multiplications required for the computation of a length-$2^n$ circular convolution. On the other hand, we know that the number of additions required by the proposed algorithm is contributed from three parts:

1. for the realization of the $(N/2 \times N/2)$ 2-D convolution;
2. for the realization of the $3N/2$ length-$N/2$ 1-D convolution;
3. for the realization of the forward and inverse ODPRT.
Table 1
Number of non-trivial real operations for real circular convolution of size \(N \times N\)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(M'_n)</td>
<td>(A'_n)</td>
<td>(M'_n)</td>
<td>(A'_n)</td>
</tr>
<tr>
<td>8</td>
<td>76</td>
<td>756</td>
<td>160</td>
<td>928</td>
</tr>
<tr>
<td>16</td>
<td>436</td>
<td>4108</td>
<td>1024</td>
<td>5120</td>
</tr>
<tr>
<td>32</td>
<td>2500</td>
<td>21,116</td>
<td>5888</td>
<td>28,672</td>
</tr>
<tr>
<td>64</td>
<td>13,540</td>
<td>104,028</td>
<td>31,232</td>
<td>147,456</td>
</tr>
<tr>
<td>128</td>
<td>69,412</td>
<td>496,156</td>
<td>156,672</td>
<td>720,896</td>
</tr>
<tr>
<td>256</td>
<td>340,900</td>
<td>2,308,508</td>
<td>755,712</td>
<td>3,407,872</td>
</tr>
<tr>
<td>512</td>
<td>1,621,156</td>
<td>10,537,116</td>
<td>3,543,040</td>
<td>15,728,640</td>
</tr>
<tr>
<td>1024</td>
<td>7,524,004</td>
<td>47,376,028</td>
<td>16,261,120</td>
<td>65,308,160</td>
</tr>
</tbody>
</table>

The number of additions required for the proposed algorithm becomes

\[
A'_n = A'_{n-1} + \frac{3N}{2} A'_n + \frac{N^2}{2} (3n + 5),
\]

(19)

where \(A'_n\) represents the number of additions required for the computation of a length-\(2^n\) circular convolution.

The number of real arithmetic operations required for computing an \(N\)-point circular convolution is as follows [2].

\[
M'_n = 2^{n-1}(2n-3) + 3,
\]

(20)

\[
A'_n = 2^{n-1}(6n-7) + 5.
\]

(21)

We are given that \(M'_1 = 4\) and \(A'_1 = 16\). Table 1 gives a comparison of the proposed approach to other approaches. In this table, we have assumed that the function to be convolved with the input data is fixed. This is the case in many filtering operations where the impulse response of the filter is known. In this case, the ODPRT of the function is pre-computed and stored. The result shows that the proposed approach achieves at least an 18% reduction in the number of multiplications. In fact, the total number of operations (multiplications and additions) is also fewer than the traditional approaches. We believe that a good performance can be achieved when implementing the proposed algorithm in general computing systems.

4. Blind image restoration

4.1. Blind image restoration based on ODPRT

In many practical situations, image formation process can be adequately formulated by the following linear model [1]:

\[
g(x, y) = \sum_{x' = 0}^{N-1} \sum_{y' = 0}^{N-1} h(x - x', y - y') f(x', y') + n(x, y),
\]

(22)

where \(N\) is the image size, \(f(x, y)\) is the original image, \(g(x, y)\) is the observed image, \(h(x, y)\) is the PSF and \(n(x, y)\) is the additive noise due to the imaging system. The problem of blind image restoration is to recover an unknown original image \(f(x, y)\) from a given blurred image \(g(x, y)\) without prior knowledge of the PSF \(h(x, y)\). It is here understood that \(f(x, y)\) is positive (i.e. real and non-negative) and has compact support.

The blind image restoration problem can be speeded up by using ODPRT. Assume that the image size \(N\) is sufficiently greater than the support of \(f(x, y)\) and \(h(x, y)\) such that the linear convolution in Eq. (22) can be rewritten into a circular convolution form as follows:

\[
g(x, y) = f(x, y) \otimes_2 h(x, y) + n(x, y).
\]

(23)

Due to the circular convolution property, the ODPRT of the observed image in the absence of noise is given
as follows:

\[ g_m^P(d) = h_m^P(d) \otimes f_m^P(d), \]

\[ g_s^P(d) = h_s^P(d) \otimes f_s^P(d), \]

\[ g'(x, y) = g(x, y) + g(x + N/2, y) + g(x, y + N/2) \]

\[ + g(x + N/2, y + N/2), \]

where \( m \in \mathbb{Z}_N; s, d, x, y \in \mathbb{Z}_{N/2}. \) For the ease of presentation, we shall only consider the operations performed on \( g_m^P(d) \). Similar operations are performed on \( g_s^P(d) \) hence will not be further discussed in this paper.

To restore the original function \( f(x, y) \), we can make use of the Fourier phase of \( g_m^P(d) \). By the discrete Fourier slice theorem as shown in Eqs. (12) and (13), we know that the Fourier spectrum of \( g_m^P(d) \) is related to the 2-D Fourier spectrum of \( g(x, y) \), i.e., \( G(u, v) \), by the following formulation:

\[
G((-m(2v+1))_N, 2v+1) = \sum_{d=0}^{N/2-1} g_m^P(d) \exp(-j2\pi(v+1/2)d/(N/2))
\]

\[
= G_m^P(v), \quad (24)
\]

where \( m \in \mathbb{Z}_N; v \in \mathbb{Z}_{N/2}. \) We can obtain similar relationships between \( F_m^P(d) \) and \( F(u, v) \) as well as between \( H_m^P(d) \) and \( H(u, v) \). Furthermore, it is known that

\[ G(u, v) = F(u, v)H(u, v). \]

Hence,

\[
G((-m(2v+1))_N, 2v+1) = F((-m(2v+1))_N, 2v+1) \times H((-m(2v+1))_N, 2v+1). \quad (25)
\]

It implies

\[
G_m^P(v) = F_m^P(v)H_m^P(v), \quad (26)
\]

\[
\angle G_m^P(v) = \angle H_m^P(v) + \angle F_m^P(v). \quad (27)
\]

Eq. (27) suggests that, from the phase of the ODPRT of the observed image, i.e., \( \angle G_m^P(v) \), we can obtain \( \angle F_m^P(v) \) if we have prior knowledge of \( \angle H_m^P(v) \) or if \( \angle H_m^P(v) \) simply equals to zero. It is possible because, in many situations, the image blurs, including the out-of-focus blurs, can be modeled by a conjugate symmetric function such that \( \angle H(u, v) \) is either 0 or \( \pi \). It can be shown that, in this case, \( \angle H_m^P(v) \) will also be either 0 or \( \pi \). Consequently, \( \angle F_m^P(v) \) can be determined by using Eq. (27). Then by using Eq. (24), we can obtain the phase of the original image, i.e., \( \angle F(u, v) \), from \( \angle F_m^P(v) \). We are interested in the phase of the original image because it is well known that the Fourier phase of an image is sufficient for its representation and reconstruction under certain mild conditions. In the case of finite-length 1-D sequence, it was demonstrated in [3,4] that a finite number of Fourier-phase coefficients are sufficient for exact reconstruction. Hayes [3] also introduced the uniqueness conditions under which a multidimensional (MD) sequence is exactly defined by its Fourier phase; these include (i) finite support \( R(N) \) (i.e., the sequence is non-zero only in the finite MD interval \([1, N]\) of an MD sequence), (ii) lack of symmetric factors in its Fourier transform, and (iii) knowledge of the phase of its \( M \)-point FFT, provided that \( M > 2N \). Hence, only if we know \( \angle F(u, v) \), we can uniquely reconstruct \( f(x, y) \) using the algorithm given in [3].

4.2. Applying constraints

The approach mentioned above sounds ideal; however, there are many difficulties encountered in practice. Although \( \angle H_m^P(v) \) in many situations is equal to either 0 or \( \pi \), we do not have the exact information of which coefficients of \( \angle H_m^P(v) \) is 0 or which of them is \( \pi \). Different blurring functions, even if they are conjugate symmetric, have different patterns of 0 and \( \pi \) in their Fourier phase. To solve this problem, we make use of an approach given in [7], which suggests that the magnitude of a function can be retrieved if its Fourier phase modulo \( \pi \) is known. Furthermore, we impose another set of constraints based on the relationship between different ODPRT projections. More precisely, we first transform the blurred image \( g(x, y) \) using the fast ODPRT to become \( g_m^P(d) \) and \( g_s^P(d) \) as shown in [8]. We perform \( 3N/2 \) 1-D reduced FFTs on \( g_m^P(d) \) and \( g_s^P(d) \) to obtain \( G_m^P(v) \) and \( G_s^P(u) \). The following two constraints are then imposed on \( G_m^P(v) \) and \( G_s^P(u) \). The first one is the same as that in [7]. Let

\[
\psi(G_m^P(v)) = \angle G_m^P(v) \mod \pi. \quad (28)
\]
The phase of the estimated image is updated as follows:

\[
\angle F_m^p(v)_{t+1} = \begin{cases}
\psi\{G_m^p(v)\} \\
\psi\{G_m^p(v)\} + \pi
\end{cases}
\]

for \(|\psi\{G_m^p(v)\} - \angle F_m^p(v)_{t}| < (\pi/2),

for \(|\psi\{G_m^p(v)\} - \angle F_m^p(v)_{t}| \geq (\pi/2),

\text{where } F_m^p(v), \text{ is the estimation of } F_m^p(v) \text{ after } "t" \text{ iterations. (Note that the operations on } G_N^p(u) \text{ are similar, hence will not be repeatedly described.)}

The second constraint arises from the fact that some of the ODPRT projections of a symmetric function are the same. First let us consider the DPRT projections and we shall make use of the relationship between DPRT and ODPRT to illustrate the above point. For an \(N \times N\) symmetric function \(h(x, y)\) such that \(h(x, y)\) is equal to \(h(N - x, y), h(x, N - y), h(N - x, N - y)\) and \(h(y, x), \text{ we have}\)

\[
h^b_0(d) = \sum_{y=0}^{N-1} h(d, y) = \sum_{y=0}^{N-1} h(y, d)
\]

\[
= \sum_{x=0}^{N-1} h(x, d) = h^b_0(d),
\]

\[
h^b_{N/4}(d) = \sum_{y=0}^{N-1} h(\langle d + 2yN/4 \rangle_N, y)
\]

\[
= \sum_{y=0}^{N-1} h(y, \langle d + yN/2 \rangle_N)
\]

\[
= \sum_{x=0}^{N-1} h(x, \langle d + xN/2 \rangle_N) = h^c_{N/2}(d),
\]

\[
h^c_{N-m}(d) = \sum_{x=0}^{N-1} h(x, \langle d + (N - m)x \rangle_N)
\]

\[
= \sum_{x=0}^{N-1} h(\langle N - x \rangle_N, \langle d + (N - m) \rangle_N \times (N - x)_N)
\]

Similarly, it can be shown that

\[
h^c_{N/2-s}(d) = h^c_s(d).
\]

Since \(h^c_m(d)\) is equal to \(h^c_{N-m}(d)\), \(H^c_m(v)\) is also equal to \(H^c_{N-m}(v)\). For a pair of DPRT projections of the observed image, for example, \(g_m^e(d)\) and \(g_{N-m}^e(d)\), we have

\[
G_m^c(v) = H_m^c(v)F_m^c(v)
\]

and

\[
G_{N-m}^c(v) = H_{N-m}^c(v)F_{N-m}^c(v) = H_m^c(v)F_{N-m}^c(v).
\]

Then,

\[
G_m^c(v)F_{N-m}^c(v) = H_m^c(v)F_m^c(v)F_{N-m}^c(v)
\]

\[
= G_{N-m}^c(v)F_m^c(v).
\]

We have

\[
G_m^c(v)/G_{N-m}^c(v) = F_m^c(v)/F_{N-m}^c(v).
\]

Similarly, we have

\[
G^b_m(v)/G^b_{N/2-s}(v) = F^b_m(v)/F^b_{N/2-s}(v),
\]

\[
G^b_0(v)/G^b_0(v) = F^b_0(v)/F^b_0(v),
\]

\[
G^b_{N/2}(v)/G^b_{N/4}(v) = F^b_{N/2}(v)/F^b_{N/4}(v).
\]

From Eqs. (14) and (15) we know that the spectrum of the DPRT projections is related to the DPRT projections as follows:

\[
F_m^p(v) = F_m^2(2v + 1),
\]

\[
F^b_s(u) = F^b_s(2u + 1).
\]

Hence we have a similar set of formulations as Eqs. (34)–(37) for ODPRT:

\[
G_m^p(v)/G^p_{N-m}(v) = F_m^p(v)/F^p_{N-m}(v),
\]

\[
G^b_s(v)/G^b_{N/2-s}(v) = F^b_s(v)/F^b_{N/2-s}(v),
\]
\[ G_0^P(v)/G_0^q(v) = F_0^P(v)/F_0^q(v), \]
\[ G_{N/2}^P(v)/G_{N/4}^q(v) = F_{N/2}^P(v)/F_{N/4}^q(v). \]

(40)

(41)

The above equations show that each ODPRT projection pair of the observed image is linearly related by the corresponding ODPRT projections of the original image. Hence, we can write out a new constraint to impose on the iteration process. We use Eq. (38) as an example. By Eq. (38), we use the ratio of \( G_m^P(v)/G_{N-m}^q(v) \) to obtain \( F_m^P(v) \) and \( F_{N-m}^P(v) \):

\[ F_m^P(v)_{i+1} = F_m^P(v)_i G_m^P(v)/G_{N-m}^q(v), \]

(42)

\[ F_{N-m}^P(v)_{i+1} = F_{N-m}^P(v)_i G_{N-m}^q(v)/G_m^P(v). \]

(43)

Motivated by various relaxation techniques for iterative algorithms [14], Eqs. (42) and (43) are modified as follows:

\[ F_m^P(v)_{i+1} = (1 - \alpha)F_m^P(v)_i + \alpha F_{N-m}^P(v)_i G_m^P(v)/G_{N-m}^q(v), \]

(44)

\[ F_{N-m}^P(v)_{i+1} = (1 - \alpha)F_{N-m}^P(v)_i + \alpha F_m^P(v)_i G_{N-m}^q(v)/G_m^P(v), \]

(45)

where the relaxation parameter \( \alpha \) is a scalar between 0 and 1. Eqs. (44) and (45) can be equivalently written as

\[ F_m^P(v)_{i+1} = F_m^P(v)_i + \alpha[F_{N-m}^P(v)_i G_m^P(v)/G_{N-m}^q(v) - F_m^P(v)_i], \]

(46)

\[ F_{N-m}^P(v)_{i+1} = F_{N-m}^P(v)_i + \alpha[F_m^P(v)_i G_{N-m}^q(v)/G_m^P(v) - F_{N-m}^P(v)_i]. \]

(47)

Several special cases of Eqs. (46) and (47) are immediately apparent. If \( \alpha \) is a fixed constant and is equal to 1, Eqs. (46) and (47) will correspond to the original iteration formulations shown in Eqs. (42) and (43). When \( \alpha = 0 \), it produces the trivial result \( F_m^P(v)_{i+1} = F_m^P(v)_i \). Intermediate values of \( \alpha \), i.e., \( 0 < \alpha < 1 \), correspond to what is commonly referred to as the under-relaxed version of Eqs. (42) and (43).

After imposing the above two constraints on the determination of \( F_m^P(v) \) and \( F_m^q(u) \), we perform \( 3N/2 \) 1-D reduced IFFTs on \( F_m^P(v) \) and \( F_m^q(u) \) to obtain \( f_m^P(d) \) and \( f_m^q(s) \). The operation is then recursively applied to the spatially aliased function \( f' \), until a trivial solution is obtained. Finally, an inverse ODPRT is applied to obtain \( f(x, y) \). This completes one iteration.

Let us summarize the procedure of the new iterative algorithm for restoring an image \( f(x, y) \) of dimension \( N \times N \) using ODPRT, it is illustrated in Fig. 3 and described as follows:

1. Perform ODPRT, which involves solely additions, to the observed image \( g(x, y) \) with size \( M \times M \) (\( M > 2N \)). We have \( 3M/2 \) length-M/2 projections of the image and a spatially aliased function \( g \).

For each projection, the \( M/2 \) projections of \( g_m(d) \) is denoted as \( G_m^P(v) = \{G_m^P(v)\exp(j \angle G_m^P(v))\} \) (where \( v = 0, 1, \ldots , M/2 - 1 \). Then, we derive \( \psi(G_m^P(v)) \) for different \( m \) by Eq. (28). (The computation for \( g_m^q(s) \) is similar and will not be described further.)

2. Form another sequence \( t(x, y) \), which is given by

\[
\begin{align*}
0, & \quad 0 \leq x, y < \text{Int}(K/2), \\
\text{Int}(K/2) \leq x, & \quad y < N - \text{Int}(K/2), \\
0, & \quad N - \text{Int}(K/2) \leq x, \\
y < M - 1, & \quad y < M - 1,
\end{align*}
\]

(48)

where \( f(x, y) \) represents the estimation of \( f(x, y) \) after “n” iterations. Note that \( f(x, y)_0 = g(x, y) \). The parameter \( K \) is the length of the blurring filter, and \( \text{Int}(K) \) indicates the closest integer smaller than \( K \).

3. Find the ODPRT of the function \( t(x, y)_i \) to obtain \( t_m^P(d)_i \), \( t_m^q(s)_i \), and \( t'(x, y)_i \).

4. Perform \( 3M/2 \) 1-D \( M/2 \)-point reduced FFTs to the transformed results. We obtain \( T_m^P(v)_i \), \( T_m^q(u)_i \).

5. Impose the following constraints to increase the rate of convergence of the algorithm:

(i) Impose the constraints in Eqs. (46) and (47) to obtain \( P_m^P(v)_i \).
(ii) Impose the constraint in Eq. (29):
\[
\tilde{F}_p^m(v)_{i+1} = \begin{cases} 
  |P_m^c(v)|\psi\{G_m^p(v)\} \\
  |P_m^c(v)|\psi\{G_m^p(v)\} + \pi
\end{cases} \begin{array}{l}
\text{if } |\psi\{G_m^p(v)\} - L P_m^p(v)| < (\pi/2), \\
\text{if } |\psi\{G_m^p(v)\} - L P_m^p(v)| \geq (\pi/2).
\end{array}
\]

6. Perform 3M/2 M/2-point inverse reduced FFTs to the newly estimated signal \( \tilde{F}_p^m(v)_{i+1} \) to obtain \( \tilde{f}_p^m(v)_{i+1} \).
7. Repeat steps 3–6 with \( t'(x',y) \) as the input until trivial solution is obtained.
8. Perform inverse ODPRT to obtain a new estimate \( f(x,y)_{i+1} \).

Steps 2–8 are repeated until the algorithm converges.
Table 2
Number of non-trivial real operations for one iteration of restoration process

<table>
<thead>
<tr>
<th>$M$ (2$^m$)</th>
<th>Proposed approach</th>
<th>Row–column Polynomial transforms [13]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$O_0$</td>
<td>$O_r$</td>
</tr>
<tr>
<td>128</td>
<td>1,078,176</td>
<td>1,835,008</td>
</tr>
<tr>
<td>256</td>
<td>5,100,448</td>
<td>8,388,608</td>
</tr>
<tr>
<td>512</td>
<td>23,548,832</td>
<td>37,748,736</td>
</tr>
<tr>
<td>1024</td>
<td>106,779,552</td>
<td>167,772,160</td>
</tr>
</tbody>
</table>

4.3. Computational complexity

We have analyzed the computational complexity of the original and the new algorithms. For the original algorithm, since the iterative algorithm is characterized by repeated transformations between time and frequency domains, 2-D FFT and inverse 2-D FFT, each with size $M \times M$, are computed in each iteration. By using the conventional row–column 2-D FFT algorithm, it requires $16M^2 \times \log_2 M$ operations for each iteration.

For the ODPRT algorithm, the $M \times M$ forward and inverse 2-D FFT are converted into $3M/2$ length-$M/2$ 1-D forward and inverse reduced FFTs. Besides, a spatially aliased function is also generated which will be used as the input for the next stage of operations. The procedure is then repeated recursively. Assume that the length-$M$ ($M = 2^m$) reduced FFT [13] has the real multiplicative and additive complexity of $\tilde{M}_m^r$ and $\tilde{A}_m^a$, respectively. The total arithmetic complexity, $O_m$, of the proposed approach is as follows:

$$O_m = O_{m-1} + 2 \frac{3M}{2} (\tilde{M}_m^r + \tilde{A}_m^a) + \frac{M^2}{2} (3m + 5).$$

Table 2 shows the actual arithmetic operations required for different $M$ by different approaches. In the table the reduced FFT is computed using the Rader–Brenner algorithm [13]. The arithmetic complexity of using the polynomial transform for the computation of 2D-FFT is also shown. We can see that the proposed algorithm requires fewer operations than the traditional approaches [13].

4.4. Experimental results

Experimental results are presented in this section. We examine the behavior of the new algorithm when an image is degraded by a Gaussian blurring filter. A 128 $\times$ 128 image with no noise is shown in Fig. 4(a), and 4(b) is a blurred version of this image with a $15 \times 15$ Gaussian filter, with $\sigma = 6.5$ pixels. Figs. 4(c), (d) and (e) are the restored images using the approach in [7] with a 256 $\times$ 256 point 2-D FFT and a 256 $\times$ 256 point 2-D IFFT after 50, 100, 500 iterations, respectively. Figs. 4(f), (g) and (h) are the restored image using the ODPRT approach with $\alpha = 0.97$ after 50, 100, 500 iterations, respectively. From these two groups of images, we see that the restored images are properly recovered by both algorithms, but the ODPRT algorithm gives results with a better quality.

In order to compare the rate of convergence of two algorithms, a plot of SNR versus the number of iterations for the example in Fig. 4 is shown in Fig. 6. We observe that the proposed algorithm improves the rate of convergence. Less than 100 iterations are required to give an output image with 19 dB for the ODPRT approach, but 500 iterations are required for the original approach to give the same quality image. Since the number of operations of the proposed approach in each iteration is less than the original one, it shows that the proposed approach is more efficient. In addition, the estimate using the original approach converges at 19 dB, while the estimates using the ODPRT approach converges at 23.26 dB. It shows that the proposed approach also improves the quality of the estimated image by more than 3 dB. Table 3 summarizes the results for different images. It shows that the proposed approach gives an improved performance up to 3.74 dB.

It is known that all restoration procedures demonstrate little tolerance for estimated image support size. In case that the image support size is overestimated, Fig. 5 shows the restorations results using the original and the proposed approaches. In this simulation, the true image support is 98 $\times$ 98 pixels. The overestimated support is 100 $\times$ 100 pixels. It is seen that the proposed algorithm is highly robust to overestimation of the support size. Around 3 dB improvement in SNR is obtained as compared to the original approach (Fig. 6).
Fig. 4. (a) Original image; (b) blurred image by a $15 \times 15$ Gaussian filter, $\sigma = 6.5$ pixels; (c)–(e) image reconstructed by the original approach [7] after 50, 100, 500 iterations, respectively; and (f)–(h) image reconstructed by the proposed algorithm after 50, 100, 500 iterations, respectively.

Table 3
Results of the original approach and the proposed algorithm for different blurred images after 50 iterations

<table>
<thead>
<tr>
<th>Image</th>
<th>Blur filter</th>
<th>Blurred image (SNR/dB)</th>
<th>Result (SNR/dB)</th>
<th>Improvement in SNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Original [7]</td>
<td>Proposed</td>
</tr>
<tr>
<td>Lenna</td>
<td>Gaussian $\sigma = 6.5$ pixels $15 \times 15$</td>
<td>9.59</td>
<td>14.74</td>
<td>16.56</td>
</tr>
<tr>
<td>Bird</td>
<td>Gaussian $\sigma = 6.5$ pixels $15 \times 15$</td>
<td>11.53</td>
<td>20.98</td>
<td>24.72</td>
</tr>
<tr>
<td>Boat</td>
<td>Gaussian $\sigma = 4$ pixels $13 \times 13$</td>
<td>11.77</td>
<td>19.40</td>
<td>22.17</td>
</tr>
<tr>
<td>Airplane</td>
<td>Gaussian $\sigma = 2.56$ pixels $11 \times 11$</td>
<td>12.55</td>
<td>19.94</td>
<td>22.44</td>
</tr>
<tr>
<td>Peppers</td>
<td>Smooth (boxcar average) $9 \times 9$</td>
<td>12.51</td>
<td>18.10</td>
<td>19.33</td>
</tr>
</tbody>
</table>
Fig. 5. Simulation results of the original approach and the proposed algorithm for overestimated supports: (a) original approach (SNR = 17.68 dB) after 388 iterations and (b) proposed algorithm (SNR = 20.34 dB) after 244 iterations.

Fig. 6. Plot of SNR versus number of iterations using the original and ODPRT approach.

Fig. 7. The restored image which is blurred with the same filter in Fig. 4 and Gaussian noise with: (a) BSNR = 40 dB, (b) BSNR = 50 dB, and (c) BSNR = 60 dB.

The results of the algorithms in the presence of additive white Gaussian noise are presented in Fig. 7. The blurred signal-to-noise ratio (BSNR) is used as a measure of the extent of degradation due to noise.

It is defined as the ratio of the energy of the blurred signal to the energy of noise:

$$\text{BSNR} = 10 \log_{10} \left( \frac{\text{variance blurred image}}{\text{variance noise}} \right).$$

In Fig. 7, the image “Lenna” is blurred with the same filter as in Fig. 4. The restored images using the ODPRT approach with BSNR 40, 50 and 60 dB are shown in Fig. 7(a), (b) and (c), respectively. Although we observe some distortion, the estimated image converges to an acceptable quality at termination.

5. Conclusion

In this paper, the useful properties of ODPRT are studied. The unique feature of ODPRT is that we can reduce a 2-D problem to become some 1-D ones. As compared with the DPRT, ODPRT has the advantage that it eliminates most of the redundant operations due to its orthogonal structure. Besides, ODPRT retains most of the useful properties of DPRT such as the discrete Fourier slice theorem and circular convolution property. They allow ODPRT to be applicable to many image processing problems. We have demonstrated the applications of ODPRT in the computations of 2-D circular convolution and blind image restoration. Significant improvement is obtained in each case as compared with the traditional approaches.
These applications indeed can be further improved by noting that not all projections are significant in the reconstruction process. This feature of DPRT has been adopted in [10] for image coding applications. Further reduction in computational complexity can be achieved by ignoring the insignificant projections.

Acknowledgements

This work is supported by the Hong Kong Polytechnic University under Research Grant A418.

References