# New boundary effect free algorithm for fast and accurate image arbitrary scaling and rotation

Leonid Bilevich[a] and Leonid Yaroslavsky[a]

[a]Department of Physical Electronics, Faculty of Engineering, Tel Aviv University, 69978, Tel Aviv, Israel

## ABSTRACT

A new fast DCT-based algorithm for accurate image arbitrary scaling and rotation is described. The algorithm is free from boundary effects characteristic for FFT-based algorithm and ensures perfect interpolation with no interpolation errors. The algorithm is compared with other available algorithms in terms of the interpolation accuracy, computational complexity and suitability for real time applications.

**Keywords:** scaling, rotation, DCT, convolution, fast algorithm

## 1. INTRODUCTION

Scaling and rotation are basic image processing tasks with wide range of applications. The rotation is usually implemented through either three-pass method[1] (with 1D translations) or non-separable DFT-domain method.[2] Presently, for image scaling& rotation available are spline methods[3,4] implemented in spatial domain and discrete sinc-interpolation methods implemented as a convolution in DFT domain.[5,6] The spline methods work in image domain and tend to introduce image blurring. The discrete sinc-interpolation methods are potentially perfectly accurate but heavily suffer from boundary effects. We propose a novel DCT[7]-based scaling& rotation algorithm that implements the ideal discrete sinc-interpolation and at the same time is very substantially less vulnerable to boundary effects.

In video processing, a crucial issue is its computational complexity. Generally, there is a trade-off between the accuracy and computational complexity. In this paper, we, along with introducing a new improved DCT-based method of image arbitrary scaling& rotation, compare its computational complexity with the complexity of spatial domain interpolation methods and DFT-domain methods.

The definitions of different 2D DCT-related transforms used in this paper are provided in Appendix A.

## 2. DCT-DOMAIN SCALING AND ROTATION ALGORITHM

### 2.1 Inverse Scaled Rotated Discrete Cosine Transform (IScRotDCT)

Suppose the input real image $a_{m,n}$ of size $N \times N$ has to be scaled by arbitrary scaling factor $\sigma$ and rotated by arbitrary angle $\theta$ (in radians, positive $\theta$ corresponds to counter-clockwise rotation). We need to choose a method to perform this scaling& rotation. The Inverse Scaled Rotated Discrete Fourier Transform (IScRotDFT)[5,6] suffers from heavy boundary effects due to periodicity of

Further author information: (Send correspondence to Leonid Bilevich)
bilevich@eng.tau.ac.il; http://www.eng.tau.ac.il/~bilevich/
yaro@eng.tau.ac.il; http://www.eng.tau.ac.il/~yaro/

DFT. In order to eliminate these effects, we mirror-reflect the input image $a_{m,n}$ to double length (in both dimensions) and apply IScRotDFT to the mirror-reflected image. Constraining the output image $\tilde{a}_{k,l}$ to be real-valued, we obtain the *Inverse Scaled Rotated Discrete Cosine Transform (*IScRotDCT*)* defined as follows:

$$\begin{aligned}\tilde{a}_{k,l} = \frac{2}{\lfloor \sigma N \rfloor} \sum_{r=0}^{N^{(0)}-1} \sum_{s=0}^{N^{(0)}-1} \alpha_{r,s}^{C(1/2)} \\ \times \cos\left[\frac{\pi}{\sigma N}\left(C\left(k+1/2\right)+S\left(l+1/2\right)-\Delta_1'\right)r\right] \\ \times \cos\left[\frac{\pi}{\sigma N}\left(C\left(l+1/2\right)-S\left(k+1/2\right)-\Delta_2'\right)s\right],\end{aligned} \tag{1}$$

where $\tilde{a}_{k,l}$ is the scaled and rotated signal, $N^{(0)}$ is the "unified length" given by:

$$N^{(0)} = \min\left(N, \lfloor \sigma N \rfloor\right), \tag{2}$$

$\lfloor \cdot \rfloor$ is the rounding operator:

| | |
|---|---|
| $\sigma \geq 1$ | $\lfloor \sigma N \rfloor = \mathrm{CEIL}(\sigma N)$ |
| $\sigma < 1$ | $\lfloor \sigma N \rfloor = \mathrm{FLOOR}(\sigma N)$ |

, (3)

$C$ and $S$ are the "rotation factors":

$$S = \sin\theta,\ C = \cos\theta, \tag{4}$$

$\Delta_1'$, $\Delta_2'$ are the "centering factors":

$$\Delta_1' = (C+S)\frac{\lfloor \sigma N \rfloor}{2} - \sigma\frac{N}{2},\ \Delta_2' = (C-S)\frac{\lfloor \sigma N \rfloor}{2} - \sigma\frac{N}{2}, \tag{5}$$

$\alpha_{r,s}^{C(1/2)}$ is the "half-spectrum" given by:

$$\alpha_{r,s}^{C(1/2)} = \alpha_{r,s}^{C} \cdot$$

| | 0 | 1 | … | $N^{(0)}-2$ | $N^{(0)}-1$ |
|---|---|---|---|---|---|
| 0 | 1/4 | 1/2 | … | 1/2 | 1/4 |
| 1 | 1/2 | 1 | … | 1 | 1/2 |
| ⋮ | ⋮ | ⋮ | | ⋮ | ⋮ |
| $N^{(0)}-2$ | 1/2 | 1 | … | 1 | 1/2 |
| $N^{(0)}-1$ | 1/4 | 1/2 | … | 1/2 | 1/4 |

, (6)

and $\alpha_{r,s}^{C}$ is the DCT spectrum of the input image $a_{m,n}$:

$$\alpha_{r,s}^{C} = \mathrm{DCT}\left(a_{m,n}\right) = \frac{2}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} a_{m,n}\cos\left[\pi\frac{\left(m+1/2\right)r}{N}\right]\cos\left[\pi\frac{\left(n+1/2\right)s}{N}\right]. \tag{7}$$

*Notes about the derivation of Eq. (1):*

- The "centering factors" $\Delta_1'$ and $\Delta_2'$ are used in order to map the center of the input image to the center of the output image.
- The last row and the last column of the spectral coefficients $\alpha_{r,s}^{C}$ were halved in order to improve the speed of convergence (to zero) of the scaling& rotation point-spread function (PSF).

Note also that for integer $\sigma$ and $\theta = 0$ the Eq. (1) represents the DCT-domain zero-padding scaling algorithm.[5,8]

### 2.2 Implementation through DCT-domain convolution

*Conversion of* IScRotDCT *to convolution.* Let's define the modified "centering factors":

$$\tilde{\Delta}_1 = (C+S)\,\frac{\lfloor \sigma N \rfloor - 1}{2} - \sigma\frac{N}{2},\ \tilde{\Delta}_2 = (C-S)\,\frac{\lfloor \sigma N \rfloor - 1}{2} - \sigma\frac{N}{2}. \tag{8}$$

Also, let's define a spectrum $\alpha_{r,s}^{C(1/2),ZP}$ that is obtained by zero-padding or truncation of DCT spectrum $\alpha_{r,s}^{C}$:

$$\alpha_{r,s}^{C(1/2),ZP} = \begin{array}{c|cccccccc} & 0 & 1 & \dots & N-2 & N-1 & N & \dots & \lfloor \sigma N \rfloor - 1 \\ \hline 0 & \frac{1}{4}\alpha_{r,s}^{C} & \frac{1}{2}\alpha_{r,s}^{C} & \dots & \frac{1}{2}\alpha_{r,s}^{C} & \frac{1}{4}\alpha_{r,s}^{C} & & & \\ 1 & \frac{1}{2}\alpha_{r,s}^{C} & & & & \frac{1}{2}\alpha_{r,s} & & & \\ \vdots & \vdots & & \alpha_{r,s}^{C} & & \vdots & & 0 & \\ N-2 & \frac{1}{2}\alpha_{r,s}^{C} & & & & \frac{1}{2}\alpha_{r,s}^{C} & & & \\ N-1 & \frac{1}{4}\alpha_{r,s}^{C} & \frac{1}{2}\alpha_{r,s}^{C} & \dots & \frac{1}{2}\alpha_{r,s}^{C} & \frac{1}{4}\alpha_{r,s}^{C} & & & \\ \hline N & & & & & & & & \\ \vdots & & & 0 & & & & 0 & \\ \lfloor \sigma N \rfloor - 1 & & & & & & & & \end{array} \tag{9}$$

for $\sigma \geq 1$ and

$$\alpha_{r,s}^{C(1/2),ZP} = \begin{array}{c|ccccc} & 0 & 1 & \dots & \lfloor \sigma N \rfloor - 2 & \lfloor \sigma N \rfloor - 1 \\ \hline 0 & \frac{1}{4}\alpha_{r,s}^{C} & \frac{1}{2}\alpha_{r,s}^{C} & \dots & \frac{1}{2}\alpha_{r,s}^{C} & \frac{1}{4}\alpha_{r,s}^{C} \\ 1 & \frac{1}{2}\alpha_{r,s}^{C} & & & & \frac{1}{2}\alpha_{r,s}^{C} \\ \vdots & \vdots & & \alpha_{r,s}^{C} & & \vdots \\ \lfloor \sigma N \rfloor - 2 & \frac{1}{2}\alpha_{r,s}^{C} & & & & \frac{1}{2}\alpha_{r,s}^{C} \\ \lfloor \sigma N \rfloor - 1 & \frac{1}{4}\alpha_{r,s}^{C} & \frac{1}{2}\alpha_{r,s}^{C} & \dots & \frac{1}{2}\alpha_{r,s}^{C} & \frac{1}{4}\alpha_{r,s}^{C} \end{array} \tag{10}$$

for $\sigma < 1$.
Then the Eq. (1) can be converted to the form of convolution:

$$\begin{aligned} \tilde{a}_{k,l} = {} & \frac{1}{\lfloor \sigma N \rfloor} \cos\left[\frac{\pi}{2\sigma N}\left[C\left(l^2-k^2\right) - 2Skl\right]\right] \\ & \times \left\{ \sum_{r=0}^{\lfloor \sigma N \rfloor - 1} \sum_{s=0}^{\lfloor \sigma N \rfloor - 1} \left\{ \alpha_{r,s}^{C(1/2),ZP} \cos\left[\frac{\pi}{2\sigma N}\left[C\left(s^2-r^2\right) - 2Srs + 2\left(\tilde{\Delta}_1 r - \tilde{\Delta}_2 s\right)\right]\right]\right\} \right. \\ & \qquad \left. \times \cos\left[\frac{\pi}{2\sigma N}\left[C\left((k-r)^2-(l-s)^2\right) + 2S\left(k-r\right)\left(l-s\right)\right]\right] \right. \\ & - \sum_{r=0}^{\lfloor \sigma N \rfloor - 1} \sum_{s=0}^{\lfloor \sigma N \rfloor - 1} \left\{ \alpha_{r,s}^{C(1/2),ZP} \sin\left[\frac{\pi}{2\sigma N}\left[C\left(s^2-r^2\right) - 2Srs + 2\left(\tilde{\Delta}_1 r - \tilde{\Delta}_2 s\right)\right]\right]\right\} \\ & \qquad \left. \times \sin\left[\frac{\pi}{2\sigma N}\left[C\left((k-r)^2-(l-s)^2\right) + 2S\left(k-r\right)\left(l-s\right)\right]\right] \right\} \\ & - \frac{1}{\lfloor \sigma N \rfloor} \sin\left[\frac{\pi}{2\sigma N}\left[C\left(l^2-k^2\right) - 2Skl\right]\right] \end{aligned} \tag{11}$$

$$
\begin{aligned}
&\times \left\{ \sum_{r=0}^{\lfloor \sigma N \rfloor - 1} \sum_{s=0}^{\lfloor \sigma N \rfloor - 1} \left\{ \alpha_{r,s}^{C(1/2),ZP} \sin\left[ \frac{\pi}{2\sigma N} \left[ C\left(s^2 - r^2\right) - 2Srs + 2\left(\tilde{\Delta}_1 r - \tilde{\Delta}_2 s\right) \right] \right] \right\} \right. \\
&\qquad \times \cos\left[ \frac{\pi}{2\sigma N} \left[ C\left((k-r)^2 - (l-s)^2\right) + 2S(k-r)(l-s) \right] \right] \\
&\quad + \sum_{r=0}^{\lfloor \sigma N \rfloor - 1} \sum_{s=0}^{\lfloor \sigma N \rfloor - 1} \left\{ \alpha_{r,s}^{C(1/2),ZP} \cos\left[ \frac{\pi}{2\sigma N} \left[ C\left(s^2 - r^2\right) - 2Srs + 2\left(\tilde{\Delta}_1 r - \tilde{\Delta}_2 s\right) \right] \right] \right\} \\
&\qquad \left. \times \sin\left[ \frac{\pi}{2\sigma N} \left[ C\left((k-r)^2 - (l-s)^2\right) + 2S(k-r)(l-s) \right] \right] \right\} \\
&+ \frac{1}{\lfloor \sigma N \rfloor} \cos\left[ \frac{\pi}{2\sigma N} \left[ S\left(k^2 - l^2\right) - 2Ckl \right] \right] \\
&\times \left\{ \sum_{r=0}^{\lfloor \sigma N \rfloor - 1} \sum_{s=0}^{\lfloor \sigma N \rfloor - 1} \left\{ \alpha_{r,s}^{C(1/2),ZP} \cos\left[ \frac{\pi}{2\sigma N} \left[ S\left(s^2 - r^2\right) - 2Crs + 2\left(\tilde{\Delta}_1 r + \tilde{\Delta}_2 s\right) \right] \right] \right\} \right. \\
&\qquad \times \cos\left[ \frac{\pi}{2\sigma N} \left[ S\left((l-r)^2 - (k-s)^2\right) + 2C(l-r)(k-s) \right] \right] \\
&\quad - \sum_{r=0}^{\lfloor \sigma N \rfloor - 1} \sum_{s=0}^{\lfloor \sigma N \rfloor - 1} \left\{ \alpha_{r,s}^{C(1/2),ZP} \sin\left[ \frac{\pi}{2\sigma N} \left[ S\left(s^2 - r^2\right) - 2Crs + 2\left(\tilde{\Delta}_1 r + \tilde{\Delta}_2 s\right) \right] \right] \right\} \\
&\qquad \left. \times \sin\left[ \frac{\pi}{2\sigma N} \left[ S\left((l-r)^2 - (k-s)^2\right) + 2C(l-r)(k-s) \right] \right] \right\} \\
&- \frac{1}{\lfloor \sigma N \rfloor} \sin\left[ \frac{\pi}{2\sigma N} \left[ S\left(k^2 - l^2\right) - 2Ckl \right] \right] \\
&\times \left\{ \sum_{r=0}^{\lfloor \sigma N \rfloor - 1} \sum_{s=0}^{\lfloor \sigma N \rfloor - 1} \left\{ \alpha_{r,s}^{C(1/2),ZP} \sin\left[ \frac{\pi}{2\sigma N} \left[ S\left(s^2 - r^2\right) - 2Crs + 2\left(\tilde{\Delta}_1 r + \tilde{\Delta}_2 s\right) \right] \right] \right\} \right. \\
&\qquad \times \cos\left[ \frac{\pi}{2\sigma N} \left[ S\left((l-r)^2 - (k-s)^2\right) + 2C(l-r)(k-s) \right] \right] \\
&\quad + \sum_{r=0}^{\lfloor \sigma N \rfloor - 1} \sum_{s=0}^{\lfloor \sigma N \rfloor - 1} \left\{ \alpha_{r,s}^{C(1/2),ZP} \cos\left[ \frac{\pi}{2\sigma N} \left[ S\left(s^2 - r^2\right) - 2Crs + 2\left(\tilde{\Delta}_1 r + \tilde{\Delta}_2 s\right) \right] \right] \right\} \\
&\qquad \left. \times \sin\left[ \frac{\pi}{2\sigma N} \left[ S\left((l-r)^2 - (k-s)^2\right) + 2C(l-r)(k-s) \right] \right] \right\}.
\end{aligned}
$$

*"Regular" and "flipped" convolutions.* From Eq. (11) we see that computation of the scaled and rotated image $\tilde{a}_{k,l}$ boils down to computation of four "regular" convolutions of the form

$$
b_{k,l}^{(1)} = \sum_{r=0}^{\lfloor \sigma N \rfloor - 1} \sum_{s=0}^{\lfloor \sigma N \rfloor - 1} \alpha_{r,s}^{(1)} h_{k-r,l-s}^{(1)}, \tag{12}
$$

where $\alpha_{r,s}^{(1)}$ is given by either

$$
\alpha_{r,s}^{C(1/2),ZP} \sin\left[ \frac{\pi}{2\sigma N} \left[ C\left(s^2 - r^2\right) - 2Srs + 2\left(\tilde{\Delta}_1 r - \tilde{\Delta}_2 s\right) \right] \right]
$$

or

$$\alpha_{r,s}^{C(1/2),ZP}\cos\left[\frac{\pi}{2\sigma N}\left[C\left(s^2-r^2\right)-2Srs+2\left(\tilde{\Delta}_1 r-\tilde{\Delta}_2 s\right)\right]\right],$$

$h_{r,s}^{(1)}$ is given by either

$$\sin\left[\frac{\pi}{2\sigma N}\left[C\left(r^2-s^2\right)+2Srs\right]\right]\ \text{or}\ \cos\left[\frac{\pi}{2\sigma N}\left[C\left(r^2-s^2\right)+2Srs\right]\right],$$

and to computation of four "flipped" convolutions of the form

$$b_{l,k}^{(2)}=\sum_{r=0}^{\lfloor\sigma N\rfloor-1}\sum_{s=0}^{\lfloor\sigma N\rfloor-1}\alpha_{r,s}^{(2)}h_{l-r,k-s}^{(2)}, \tag{13}$$

that are converted to the transpose of the "regular" convolution:

$$b_{l,k}^{(2)}=\left(b_{k,l}^{(2)}\right)^T=\left(\sum_{r=0}^{\lfloor\sigma N\rfloor-1}\sum_{s=0}^{\lfloor\sigma N\rfloor-1}\alpha_{r,s}^{(2)}h_{k-r,l-s}^{(2)}\right)^T, \tag{14}$$

where $\alpha_{r,s}^{(2)}$ is given by either

$$\alpha_{r,s}^{C(1/2),ZP}\sin\left[\frac{\pi}{2\sigma N}\left[S\left(s^2-r^2\right)-2Crs+2\left(\tilde{\Delta}_1 r+\tilde{\Delta}_2 s\right)\right]\right]$$

or

$$\alpha_{r,s}^{C(1/2),ZP}\cos\left[\frac{\pi}{2\sigma N}\left[S\left(s^2-r^2\right)-2Crs+2\left(\tilde{\Delta}_1 r+\tilde{\Delta}_2 s\right)\right]\right],$$

$h_{r,s}^{(2)}$ is given by either

$$\sin\left[\frac{\pi}{2\sigma N}\left[S\left(r^2-s^2\right)+2Crs\right]\right]\ \text{or}\ \cos\left[\frac{\pi}{2\sigma N}\left[S\left(r^2-s^2\right)+2Crs\right]\right].$$

For simplicity of notation for the following derivation, let's denote $b_{k,l}^{(1)}$ and $b_{k,l}^{(2)}$ by $b_{k,l}$, $\alpha_{r,s}^{(1)}$ and $\alpha_{r,s}^{(2)}$ by $\alpha_{r,s}$, $h_{r,s}^{(1)}$ and $h_{r,s}^{(2)}$ by $h_{r,s}$.

*"Even" and "odd" "regular" convolutions.* Let's convert all four versions of $h_{r,s}$ to the sum of two summands:

$$\begin{aligned}h_{r,s}&=\sin\left[\frac{\pi}{2\sigma N}\left[C\left(r^2-s^2\right)+2Srs\right]\right]\\&=\sin\left[\frac{\pi}{2\sigma N}C\left(r^2-s^2\right)\right]\cos\left[\frac{\pi}{\sigma N}Srs\right]+\cos\left[\frac{\pi}{2\sigma N}C\left(r^2-s^2\right)\right]\sin\left[\frac{\pi}{\sigma N}Srs\right],\end{aligned} \tag{15}$$

$$\begin{aligned}h_{r,s}&=\cos\left[\frac{\pi}{2\sigma N}\left[C\left(r^2-s^2\right)+2Srs\right]\right]\\&=\cos\left[\frac{\pi}{2\sigma N}C\left(r^2-s^2\right)\right]\cos\left[\frac{\pi}{\sigma N}Srs\right]-\sin\left[\frac{\pi}{2\sigma N}C\left(r^2-s^2\right)\right]\sin\left[\frac{\pi}{\sigma N}Srs\right],\end{aligned} \tag{16}$$

$$h_{r,s}=\sin\left[\frac{\pi}{2\sigma N}\left[S\left(r^2-s^2\right)+2Crs\right]\right] \tag{17}$$

$$= \sin\left[\frac{\pi}{2\sigma N} S\left(r^2 - s^2\right)\right] \cos\left[\frac{\pi}{\sigma N} Crs\right] + \cos\left[\frac{\pi}{2\sigma N} S\left(r^2 - s^2\right)\right] \sin\left[\frac{\pi}{\sigma N} Crs\right],$$

$$\begin{aligned} h_{r,s} &= \cos\left[\frac{\pi}{2\sigma N}\left[S\left(r^2 - s^2\right) + 2Crs\right]\right] \\ &= \cos\left[\frac{\pi}{2\sigma N} S\left(r^2 - s^2\right)\right] \cos\left[\frac{\pi}{\sigma N} Crs\right] - \sin\left[\frac{\pi}{2\sigma N} S\left(r^2 - s^2\right)\right] \sin\left[\frac{\pi}{\sigma N} Crs\right]. \end{aligned} \tag{18}$$

We note from Eqs. (15)-(18) that in all four cases the first summand is an even function of $r$ and $s$ and the second summand is an odd function of $r$ and $s$, so we can represent the function $h_{r,s}$ in the following form:

$$h_{r,s} = h^e_{r,s} + h^o_{r,s}, \tag{19}$$

where $h^e_{r,s}$ is the even function of $r$, $s$ (i.e., $h^e_{r,s} = h^e_{-r,s}$, $h^e_{r,s} = h^e_{r,-s}$, $h^e_{r,s} = h^e_{-r,-s}$) and $h^o_{r,s}$ is the odd function of $r$, $s$ (i.e., $h^o_{r,s} = -h^o_{-r,s}$, $h^o_{r,s} = -h^o_{r,-s}$, $h^o_{r,s} = h^o_{-r,-s}$). (Note also that the multiplier generating the product $h^o_{r,s}$ is either $\sin\left[\pi Srs/\left(\sigma N\right)\right]$ or $\sin\left[\pi Crs/\left(\sigma N\right)\right]$, so $h^o_{r,0} = 0$ for all $r$ and $h^o_{0,s} = 0$ for all $s$.)
So the "regular" convolution

$$b_{k,l} = \sum_{r=0}^{\lfloor \sigma N \rfloor - 1} \sum_{s=0}^{\lfloor \sigma N \rfloor - 1} \alpha_{r,s} h_{k-r,l-s} \tag{20}$$

can be represented as a sum of "even" and "odd" "regular" convolutions:

$$b_{k,l} = b^e_{k,l} + b^o_{k,l}, \tag{21}$$

where $b^e_{k,l}$ is the "even" "regular" convolution given by

$$b^e_{k,l} = \sum_{r=0}^{\lfloor \sigma N \rfloor - 1} \sum_{s=0}^{\lfloor \sigma N \rfloor - 1} \alpha_{r,s} h^e_{k-r,l-s} \tag{22}$$

and $b^o_{k,l}$ is the "odd" "regular" convolution given by

$$b^o_{k,l} = \sum_{r=0}^{\lfloor \sigma N \rfloor - 1} \sum_{s=0}^{\lfloor \sigma N \rfloor - 1} \alpha_{r,s} h^o_{k-r,l-s}. \tag{23}$$

DCT-*based convolution.* The "even" and "odd" "regular" convolutions can be computed using DCT-based algorithms:

$$\begin{aligned} b^e_{k,l} = \frac{\lfloor \sigma N \rfloor}{2} \big[&\mathrm{IDCT}\left(\xi^{CC}_{p,q} \eta^{CI}_{p,q}\right) + \mathrm{IDcS/CT}\left(\xi^{SC}_{p,q} \eta^{CI}_{p,q}\right) \\ &+ \mathrm{IDC/cST}\left(\xi^{CS}_{p,q} \eta^{CI}_{p,q}\right) + \mathrm{IDcST}\left(\xi^{SS}_{p,q} \eta^{CI}_{p,q}\right)\big] \end{aligned} \tag{24}$$

and

$$\begin{aligned} b^o_{k,l} = \frac{\lfloor \sigma N \rfloor}{2} \big[&\mathrm{IDcST}\left(\xi^{CC}_{p,q} \eta^{SI}_{p,q}\right) - \mathrm{IDC/cST}\left(\xi^{SC}_{p,q} \eta^{SI}_{p,q}\right) \\ &- \mathrm{IDcS/CT}\left(\xi^{CS}_{p,q} \eta^{SI}_{p,q}\right) + \mathrm{IDCT}\left(\xi^{SS}_{p,q} \eta^{SI}_{p,q}\right)\big], \end{aligned} \tag{25}$$

where

$$\xi_{p,q}^{CC} = \mathrm{DCT}\left(\alpha_{r,s}\right),\ \xi_{p,q}^{SS} = \mathrm{DcST}\left(\alpha_{r,s}\right), \tag{26}$$

$$\xi_{p,q}^{SC} = \mathrm{DcS/CT}\left(\alpha_{r,s}\right),\ \xi_{p,q}^{CS} = \mathrm{DC/cST}\left(\alpha_{r,s}\right), \tag{27}$$

$$\eta_{p,q}^{CI} = \mathrm{DCT}^{I}\left(h_{r,s}^{e}\right),\ \eta_{p,q}^{SI} = \mathrm{DcST}^{I}\left(h_{r,s}^{o}\right), \tag{28}$$

and different DCT-related transforms involved in these formulae are defined in Appendix A.

## 3. COMPUTATIONAL COMPLEXITY

We assume that both the input image $a_{m,n}$ and the output image $\tilde{a}_{k,l}$ are 2D real matrices. In order to compute 2D scaled image, rotated image and scaled& rotated image by DCT or DFT based methods, one needs to generate the cos/sin or exp factors, perform real or complex additions/ multiplications and compute DCT or DFT transforms.

- The computational complexity of multiplication of $N$ complex numbers is $4N$ real multiplications and $2N$ real additions, in total $6N$ real floating-point operations (flops).
- The 1D DFT is implemented by the Fast Fourier Transform (FFT) algorithm that needs $3\frac{7}{9}N\log_2 N - 4\frac{16}{27}N$ flops.[9] The 2D separable DFT algorithm needs $7\frac{5}{9}N^2\log_2 N - 9\frac{5}{27}N^2$ flops.
- The "numerically stable" 1D DCT algorithm needs $2\frac{1}{3}N\log_2 N - 2\frac{2}{9}N$ floating point operations (flops) and the "numerically stable" 1D DCT type-I algorithm needs $2\frac{1}{3}N\log_2 N - 2\frac{8}{9}N$ flops.[10] The 2D non-separable DCT algorithm needs $3\frac{3}{4}N^2\log_2 N - 2N^2$ flops.[11]

Also, the following implementation details were taken into account:

- The 2D DFT-based scaling algorithm was implemented separably by 1D scaling of rows followed by 1D scaling of columns.
- For the three-pass rotation with translation using splines we noticed that the spline coefficients have to be computed only at the first pass and only for the top half of rows. At the second and third passes these coefficients are reused.

Computational complexity of discussed image scaling/rotation methods is listed in Table 1. From Table 1 it is clear that the computational complexity of the DCT-based scaling& rotation method is higher than that of the low-order splines. From the other hand, the DCT-based scaling method is equivalent to very high-order spline and provides very accurate scaling results that outperform those of low-order splines. In order to provide very high quality scaling using spline interpolation, one has to use a high-order spline, e.g., B-spline of order $p$ with computational complexity $(4p+1)\left[1 + N/\left(\lfloor \sigma N \rfloor\right)\right]$ flops per one output sample.[12] In this case the complexity increases with the order of spline and becomes comparable and even higher that the complexity of the DCT-based scaling. So the DCT-based scaling& rotation method is a natural choice for very accurate image scaling& rotation with reasonable computational complexity.

Table 1: Comparison of computational complexities of 2D scaling, rotation and scaling& rotation algorithms.

| Type of the method | Number of flops per one output sample |
|---|---|
| 2D scaling implemented with central B-spline of degree $p$ ($p$ odd)[12] | $(4p+1)\left(1+\frac{N}{\lfloor \sigma N \rfloor}\right)$ |
| 2D DFT-based scaling with DFT-based cyclic convolution[5,6] | $22\frac{2}{3}\log_2\lfloor \sigma N \rfloor - 15\frac{5}{9} + \frac{1}{\lfloor \sigma N \rfloor^2}\left[6\left(N^{(0)}\right)^2 + 7\frac{5}{9}N^2\log_2 N - 9\frac{5}{27}N^2\right]$ |
| 2D DFT-based scaling with DFT-based "linear" convolution[13] | $90\frac{2}{3}\log_2\lfloor \sigma N \rfloor + 10\frac{4}{9} + \frac{1}{\lfloor \sigma N \rfloor^2}\left[6\left(N^{(0)}\right)^2 + 7\frac{5}{9}N^2\log_2 N - 9\frac{5}{27}N^2\right]$ |
| Rotation with three-pass algorithm[1] with translation implemented with central B-spline of degree $p$ ($p$ odd)[1] | $7p+2$ |
| Rotation with three-pass algorithm with translation implemented in DFT domain[1,14] | $22\frac{2}{3}\log_2 N - 18\frac{5}{9}$ |
| Rotation with three-pass algorithm with translation implemented in DCT domain[5,15] | $21\log_2 N - 11$ |
| DFT-based rotation (with DFT-based cyclic convolution)[2] | $30\frac{2}{9}\log_2 N - 12\frac{20}{27}$ |
| DFT-based scaling& rotation (with DFT-based cyclic convolution)[5,6] | $22\frac{2}{3}\log_2\lfloor \sigma N \rfloor - 15\frac{5}{9} + \frac{1}{\lfloor \sigma N \rfloor^2}\left[6\left(N^{(0)}\right)^2 + 7\frac{5}{9}N^2\log_2 N - 9\frac{5}{27}N^2\right]$ |
| DCT-based scaling and rotation | $330\log_2\lfloor \sigma N \rfloor - 55 + \frac{1}{\lfloor \sigma N \rfloor^2}\left[4\left(N^{(0)}\right)^2 + 3\frac{3}{4}N^2\log_2 N - 2N^2\right]$ |

## 4. EXPERIMENTAL VERIFICATION AND PERFORMANCE COMPARISON

In this Section we compare the DCT-based scaling&rotation algorithm with conventional scaling& rotation algorithms.

*Example of rotated image.* The image rotated by angle $\theta = 45°$ by the DCT-based scaling& rotation algorithm is demonstrated in Figure 1. We observe that the rotated image is free of boundary effects and the corners of the rotated image contain information that was mirror-reflected from the central part of the image. Also, we can verify that the rotated image is centered correctly.

*Comparison of different scaling&rotation methods.* As test images for checking 2D scaling, an image "text" and a pseudo-random image with uniform spectrum of size $256 \times 256$ were used. In order to evaluate the accuracy of image resampling by the suggested DCT-domain scaling & rotation algorithm in comparison with conventional bilinear/ bicubic[16]-based scaling & rotation algorithms, we employed iterative Scaling/ Rotation & DeRotation/ DeScaling sequence of operations (zooming-in of the original image by factor $\sigma$ ($\sigma \geq 1$) and rotation by angle $\theta$ followed by rotation by the negative angle $-\theta$ and zooming-out by the reciprocal factor $1/\sigma$). For the ideal resampling procedure the resulting "zoom&rotate-back" image has to be identical to the original image.

The results of 20-step iterative "zoom&rotate-back" of "text" image ($\sigma = 1.1$, $\theta = 11°$) by bilinear/ bicubic/ DCT-based scaling & rotation algorithms are shown in Figure 2.

Figure 1: Demonstration of image rotated by the DCT-based scaling&rotation algorithm. Rotated image ($\sigma = 1,\ \theta = 45°$).

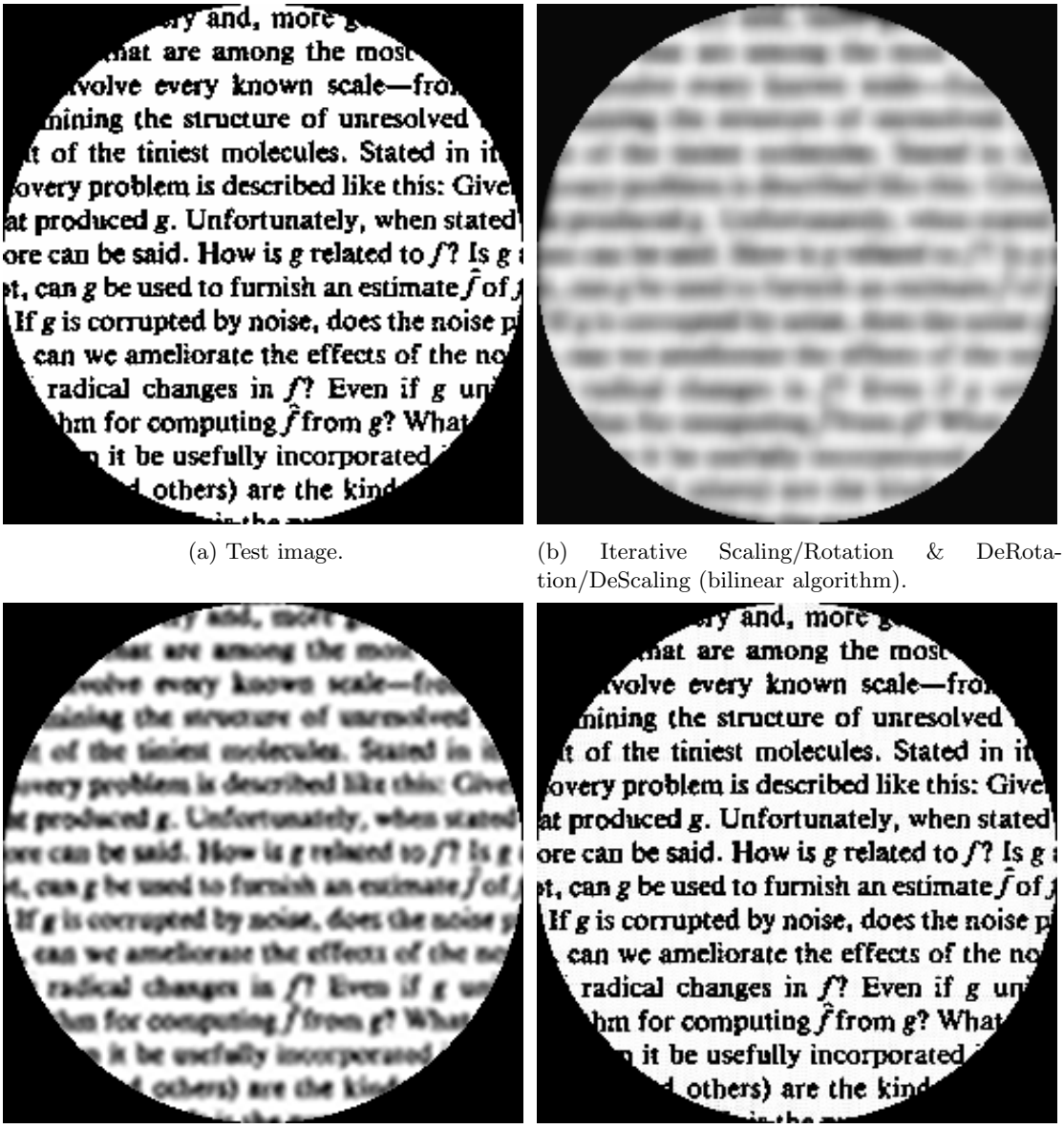
The results of 20-step iterative "zoom&rotate-back" of "random" image ($\sigma = 1.1,\ \theta = 11°$) by bilinear/ bicubic/ DCT-based scaling & rotation algorithms are shown in Figure 3.

Evolution of image spectra in such an iterative Scaling/ Rotation & DeRotation/ DeScaling test one can evaluate from comparison of spectra of "zoom&rotate-back"-iterated images. The magnitudes of DFT-spectra after 20-step iterative "zoom&rotate-back" of "random" image ($\sigma = 1.1,\ \theta = 11°$) by bilinear/ bicubic/ DCT-based scaling & rotation algorithms are shown in Figure 4.

Comparing the presented results one can see that the bilinear/ bicubic scaling & rotation methods tend to blur scaled & rotated images, also, DFT-based scaling & rotation methods suffer from boundary artifacts. The suggested DCT-based scaling & rotation method produces perfectly sharp images and is virtually free of boundary effects, demonstrating virtually perfect accuracy of resampling.

## 5. CONCLUSION

A novel DCT-domain algorithm for image scaling& rotation by arbitrary scaling factors and angles is presented. It was demonstrated that the proposed algorithm is virtually free of boundary effects and ensures virtually perfect reconstruction accuracy. Thanks to the availability of fast FFT-type algorithms for computing DCT and DCT-related transforms, the algorithm has a reasonable computational complexity and represents a valuable alternative to known scaling& rotation algorithms in image and video processing applications that need a very high resampling accuracy.



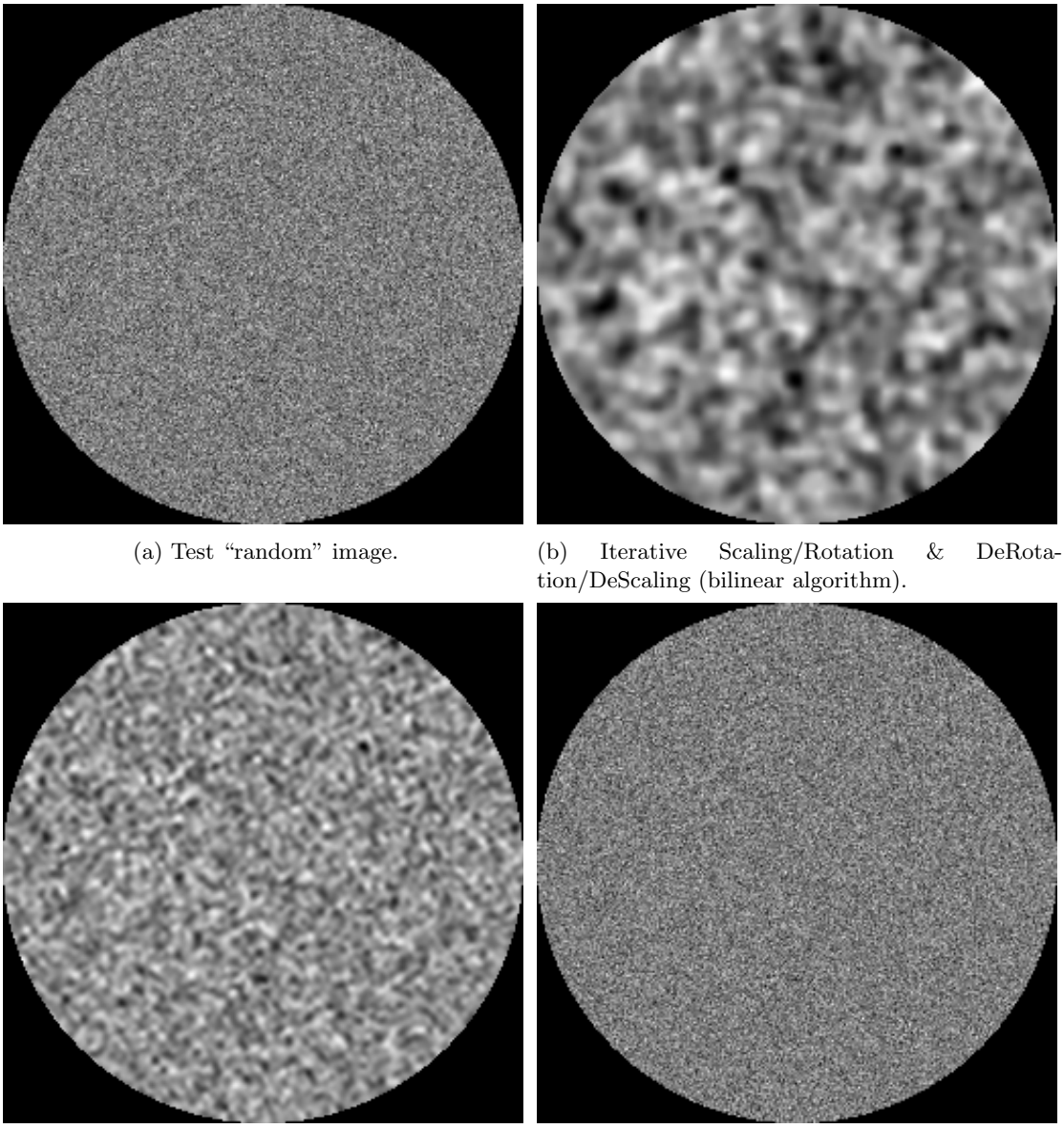
(a) Test image.

(b) Iterative Scaling/Rotation & DeRotation/DeScaling (bilinear algorithm).

(c) Iterative Scaling/Rotation & DeRotation/DeScaling (bicubic algorithm).

(d) Iterative Scaling/Rotation & DeRotation/DeScaling (DCT algorithm).

Figure 2: Iterative Scaling/Rotation & DeRotation/DeScaling of the test "text" image (a) by the bilinear algorithm (b), by the bicubic algorithm (c), and by the DCT-domain scaling&rotation algorithm (d) (after 20 iterations).



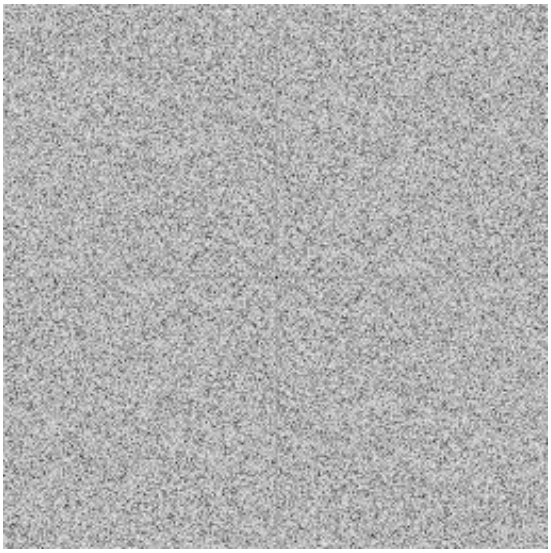
(a) Test "random" image.

(b) Iterative Scaling/Rotation & DeRotation/DeScaling (bilinear algorithm).

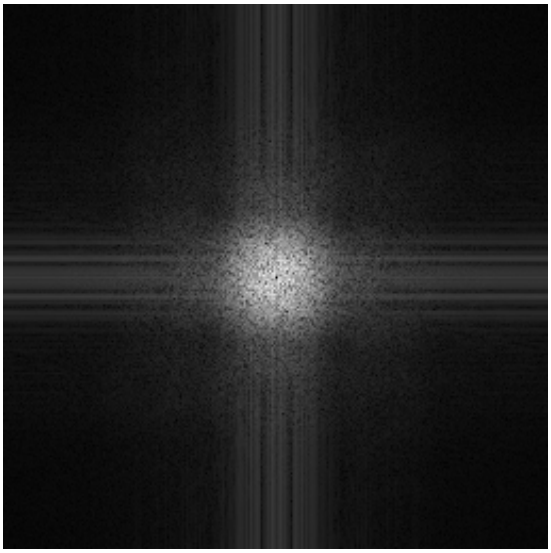
(c) Iterative Scaling/Rotation & DeRotation/DeScaling (bicubic algorithm).

(d) Iterative Scaling/Rotation & DeRotation/DeScaling (DCT algorithm).

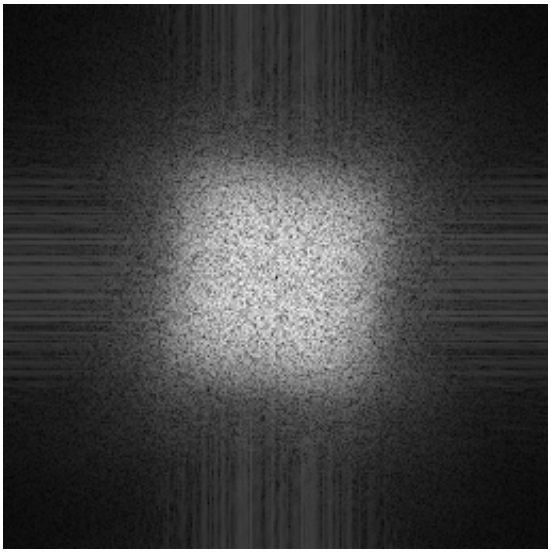
Figure 3: Iterative Scaling/Rotation & DeRotation/DeScaling of the test "random" image (a) by the bilinear algorithm (b), by the bicubic algorithm (c), and by the DCT-domain scaling& rotation algorithm (d) (after 20 iterations).



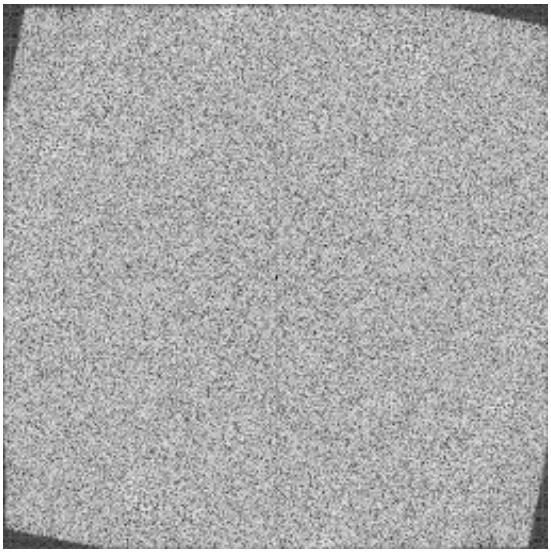
(a) Magnitude of DFT spectrum of the test image.



(b) Magnitude of DFT spectrum after iterative scaling & rotation (bilinear algorithm).



(c) Magnitude of DFT spectrum after iterative scaling & rotation (bicubic algorithm).



(d) Magnitude of DFT spectrum after iterative scaling & rotation (DCT algorithm).

Figure 4: Magnitude of DFT spectrum of iterative Scaling/Rotation & DeRotation/DeScaling of the test "random" image (a) by the bilinear algorithm (b), by the bicubic algorithm (c), and by the DCT-domain scaling&rotation algorithm (d) (after 20 iterations).

## APPENDIX A. DEFINITIONS OF DCT-RELATED TRANSFORMS

- DCT $\quad (k = 0, \ldots, N_1 - 1, \quad l = 0, \ldots, N_2 - 1, \quad r = 0, \ldots, N_1 - 1, \quad s = 0, \ldots, N_2 - 1)$

$$\begin{aligned} \alpha_{r,s} &= \mathrm{DCT}\left(a_{k,l}\right) \\ &= \frac{2}{\sqrt{N_1 N_2}} \sum_{k=0}^{N_1-1} \sum_{l=0}^{N_2-1} a_{k,l} \cos\left[\pi \frac{(k+1/2)\, r}{N_1}\right] \cos\left[\pi \frac{(l+1/2)\, s}{N_2}\right]. \end{aligned} \tag{29}$$

- DC/cST $\quad (k = 0, \ldots, N_1 - 1, \quad l = 0, \ldots, N_2 - 1, \quad r = 0, \ldots, N_1 - 1, \quad s = 1, \ldots, N_2)$

$$\begin{aligned} \alpha_{r,s} &= \mathrm{DC/cST}\left(a_{k,l}\right) \\ &= \frac{2}{\sqrt{N_1 N_2}} \sum_{k=0}^{N_1-1} \sum_{l=0}^{N_2-1} a_{k,l} \cos\left[\pi \frac{(k+1/2)\, r}{N_1}\right] \sin\left[\pi \frac{(l+1/2)\, s}{N_2}\right]. \end{aligned} \tag{30}$$

- DcS/CT $\quad (k = 0, \ldots, N_1 - 1, \quad l = 0, \ldots, N_2 - 1, \quad r = 1, \ldots, N_1, \quad s = 0, \ldots, N_2 - 1)$

$$\begin{aligned} \alpha_{r,s} &= \mathrm{DcS/CT}\left(a_{k,l}\right) \\ &= \frac{2}{\sqrt{N_1 N_2}} \sum_{k=0}^{N_1-1} \sum_{l=0}^{N_2-1} a_{k,l} \sin\left[\pi \frac{(k+1/2)\, r}{N_1}\right] \cos\left[\pi \frac{(l+1/2)\, s}{N_2}\right]. \end{aligned} \tag{31}$$

- DcST $\quad (k = 0, \ldots, N_1 - 1, \quad l = 0, \ldots, N_2 - 1, \quad r = 1, \ldots, N_1, \quad s = 1, \ldots, N_2)$

$$\begin{aligned} \alpha_{r,s} &= \mathrm{DcST}\left(a_{k,l}\right) \\ &= \frac{2}{\sqrt{N_1 N_2}} \sum_{k=0}^{N_1-1} \sum_{l=0}^{N_2-1} a_{k,l} \sin\left[\pi \frac{(k+1/2)\, r}{N_1}\right] \sin\left[\pi \frac{(l+1/2)\, s}{N_2}\right]. \end{aligned} \tag{32}$$

- $\mathrm{DCT}^I \quad (k = 0, \ldots, N_1, \quad l = 0, \ldots, N_2, \quad r = 0, \ldots, N_1, \quad s = 0, \ldots, N_2)$

$$\begin{aligned} \alpha_{r,s} &= \mathrm{DCT}^I\left(a_{k,l}\right) \\ &= \frac{1}{2\sqrt{N_1 N_2}} \Bigg[ a_{0,0} + (-1)^r a_{N_1,0} + (-1)^s a_{0,N_2} + (-1)^{r+s} a_{N_1,N_2} \\ &\quad + 2 \sum_{k=1}^{N_1-1} a_{k,0} \cos\left(\pi \frac{kr}{N_1}\right) + 2 \sum_{l=1}^{N_2-1} a_{0,l} \cos\left(\pi \frac{ls}{N_2}\right) \\ &\quad + 2\,(-1)^s \sum_{k=1}^{N_1-1} a_{k,N_2} \cos\left(\pi \frac{kr}{N_1}\right) + 2\,(-1)^r \sum_{l=1}^{N_2-1} a_{N_1,l} \cos\left(\pi \frac{ls}{N_2}\right) \\ &\quad + 4 \sum_{k=1}^{N_1-1} \sum_{l=1}^{N_2-1} a_{k,l} \cos\left(\pi \frac{kr}{N_1}\right) \cos\left(\pi \frac{ls}{N_2}\right) \Bigg]. \end{aligned} \tag{33}$$

- $\mathrm{DcST}^I \quad (k = 1, \ldots, N_1 - 1, \quad l = 1, \ldots, N_2 - 1, \quad r = 1, \ldots, N_1 - 1, \quad s = 1, \ldots, N_2 - 1)$

$$\begin{aligned} \alpha_{r,s} &= \mathrm{DcST}^I\left(a_{k,l}\right) \\ &= \frac{2}{\sqrt{N_1 N_2}} \sum_{k=1}^{N_1-1} \sum_{l=1}^{N_2-1} a_{k,l} \sin\left(\pi \frac{kr}{N_1}\right) \sin\left(\pi \frac{ls}{N_2}\right). \end{aligned} \tag{34}$$

- $\mathrm{IDCT} \quad (k = 0, \ldots, N_1 - 1, \quad l = 0, \ldots, N_2 - 1, \quad r = 0, \ldots, N_1 - 1, \quad s = 0, \ldots, N_2 - 1)$

$$\begin{aligned} a_{k,l} &= \mathrm{IDCT}\left(\alpha_{r,s}\right) \\ &= \frac{1}{2\sqrt{N_1 N_2}} \left\{ \alpha_{0,0} + 2 \sum_{r=1}^{N_1-1} \alpha_{r,0} \cos\left[\pi \frac{(k+1/2)\, r}{N_1}\right] \right. \\ &\quad + 2 \sum_{s=1}^{N_2-1} \alpha_{0,s} \cos\left[\pi \frac{(l+1/2)\, s}{N_2}\right] \\ &\quad \left. + 4 \sum_{r=1}^{N_1-1} \sum_{s=1}^{N_2-1} \alpha_{r,s} \cos\left[\pi \frac{(k+1/2)\, r}{N_1}\right] \cos\left[\pi \frac{(l+1/2)\, s}{N_2}\right] \right\}. \end{aligned} \tag{35}$$

- $\mathrm{IDC/cST} \quad (k = 0, \ldots, N_1 - 1, \quad l = 0, \ldots, N_2 - 1, \quad r = 0, \ldots, N_1 - 1, \quad s = 1, \ldots, N_2)$

$$\begin{aligned} a_{k,l} &= \mathrm{IDC/cST}\left(\alpha_{r,s}\right) \\ &= \frac{1}{2\sqrt{N_1 N_2}} \left\{ (-1)^l \alpha_{0,N_2} + 2\,(-1)^l \sum_{r=1}^{N_1-1} \alpha_{r,N_2} \cos\left[\pi \frac{(k+1/2)\, r}{N_1}\right] \right. \\ &\quad + 2 \sum_{s=1}^{N_2-1} \alpha_{0,s} \sin\left[\pi \frac{(l+1/2)\, s}{N_2}\right] \\ &\quad \left. + 4 \sum_{r=1}^{N_1-1} \sum_{s=1}^{N_2-1} \alpha_{r,s} \cos\left[\pi \frac{(k+1/2)\, r}{N_1}\right] \sin\left[\pi \frac{(l+1/2)\, s}{N_2}\right] \right\}. \end{aligned} \tag{36}$$

- $\mathrm{IDcS/CT} \quad (k = 0, \ldots, N_1 - 1, \quad l = 0, \ldots, N_2 - 1, \quad r = 1, \ldots, N_1, \quad s = 0, \ldots, N_2 - 1)$

$$\begin{aligned} a_{k,l} &= \mathrm{IDcS/CT}\left(\alpha_{r,s}\right) \\ &= \frac{1}{2\sqrt{N_1 N_2}} \left\{ (-1)^k \alpha_{N_1,0} + 2 \sum_{r=1}^{N_1-1} \alpha_{r,0} \sin\left[\pi \frac{(k+1/2)\, r}{N_1}\right] \right. \\ &\quad + 2\,(-1)^k \sum_{s=1}^{N_2-1} \alpha_{N_1,s} \cos\left[\pi \frac{(l+1/2)\, s}{N_2}\right] \\ &\quad \left. + 4 \sum_{r=1}^{N_1-1} \sum_{s=1}^{N_2-1} \alpha_{r,s} \sin\left[\pi \frac{(k+1/2)\, r}{N_1}\right] \cos\left[\pi \frac{(l+1/2)\, s}{N_2}\right] \right\}. \end{aligned} \tag{37}$$

- $\mathrm{IDcST} \quad (k = 0, \ldots, N_1 - 1, \quad l = 0, \ldots, N_2 - 1, \quad r = 1, \ldots, N_1, \quad s = 1, \ldots, N_2)$

$$
\begin{aligned}
a_{k,l} &= \mathrm{IDcST}\left(\alpha_{r,s}\right) \qquad (38)\\
&= \frac{1}{2\sqrt{N_1 N_2}} \left\{ (-1)^{k+l} \alpha_{N_1,N_2} + 2\,(-1)^{l} \sum_{r=1}^{N_1-1} \alpha_{r,N_2} \sin\left[\pi \frac{(k+1/2)\,r}{N_1}\right] \right.\\
&\quad + 2\,(-1)^{k} \sum_{s=1}^{N_2-1} \alpha_{N_1,s} \sin\left[\pi \frac{(l+1/2)\,s}{N_2}\right]\\
&\quad \left. +4 \sum_{r=1}^{N_1-1} \sum_{s=1}^{N_2-1} \alpha_{r,s} \sin\left[\pi \frac{(k+1/2)\,r}{N_1}\right] \sin\left[\pi \frac{(l+1/2)\,s}{N_2}\right] \right\}.
\end{aligned}
$$

## REFERENCES


1. M. Unser, P. Thévenaz, and L. Yaroslavsky, "Convolution-based interpolation for fast, high-quality rotation of images," *IEEE Trans. Image Process.* **4**, pp. 1371–1381, Oct. 1995.
2. R. W. Cox and R. Tong, "Two- and three-dimensional image rotation using the FFT," *IEEE Trans. Image Process.* **8**, pp. 1297–1299, Sept. 1999.
3. M. Unser, "Splines: a perfect fit for signal and image processing," *IEEE Signal Proc. Mag.* **16**, pp. 22–38, Nov. 1999.
4. C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, 1978.
5. L. Yaroslavsky, "Fast discrete sinc-interpolation: a gold standard for image resampling," in *Advances in Signal Transforms: Theory and Applications*, J. Astola and L. Yaroslavsky, eds., *EURASIP Book Series on Signal Processing and Communications* **7**, ch. 8, pp. 337–405, Hindawi, 2007.
6. L. Yaroslavsky, "Discrete transforms, fast algorithms, and point spread functions of numerical reconstruction of digitally recorded holograms," in *Advances in Signal Transforms: Theory and Applications*, J. Astola and L. Yaroslavsky, eds., *EURASIP Book Series on Signal Processing and Communications* **7**, ch. 3, pp. 93–141, Hindawi, 2007.
7. N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE T. Comput.* **23**, pp. 90–93, Jan. 1974.
8. J. I. Agbinya, "Interpolation using the discrete cosine transform," *Electron. Lett.* **28**, pp. 1927–1928, Sept. 24, 1992.
9. S. G. Johnson and M. Frigo, "A modified split-radix FFT with fewer arithmetic operations," *IEEE T. Signal Proces.* **55**, pp. 111–119, Jan. 2007.
10. G. Plonka and M. Tasche, "Fast and numerically stable algorithms for discrete cosine transforms," *Linear Algebra Appl.* **394**, pp. 309–345, Jan. 1, 2005.
11. S. C. Chan and K. L. Ho, "A new two-dimensional fast cosine transform algorithm," *IEEE T. Signal Proces.* **39**, pp. 481–485, Feb. 1991.
12. M. Unser, A. Aldroubi, and M. Eden, "Fast B-spline transforms for continuous image representation and interpolation," *IEEE Trans. Pattern Anal. Mach. Intell.* **13**, pp. 277–285, Mar. 1991.
13. L. R. Rabiner, R. W. Schafer, and C. M. Rader, "The chirp $z$-transform algorithm and its application," *Bell Syst. Tech. J.* **48**, pp. 1249–1292, May-June 1969.
14. L. Yaroslavsky, "Efficient algorithm for discrete sinc interpolation," *Appl. Optics* **36**, pp. 460–463, Jan. 10, 1997.
15. P. Yip and K. R. Rao, "On the shift property of DCT's and DST's," *IEEE T. Acoust. Speech* **35**, pp. 404–406, Mar. 1987.
16. R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE T. Acoust. Speech* **29**, pp. 1153–1160, Dec. 1981.