

Abstract

In this contribution we present transform and neural network approaches to the interpolation of images. From transform point of view, the principles from [1] are modified for 1st and 2nd order interpolation. We present several new interpolation discrete orthogonal transforms. From neural network point of view, we present interpolation possibilities of multilayer perceptrons. We use various configurations of neural networks for 1st and 2nd order interpolation. The results are compared by means of tables.

Keywords:
interpolation, discrete orthogonal transform, multilayer perceptron

1. Discrete Fourier Transform

1.1 One-dimensional interpolation with zero completing

The interpolation feature of 1D DFT is fairly known and described in basic textbooks about image processing [4]. The algorithm is clear from (1) [1].

\[ X(k) = \frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j \frac{2\pi m n}{N}}, \quad k = 0, 1, ..., N-1, \]

\[ y(l) = X(l), \]

for \( l = 0, 1, ..., \frac{N}{2} - 1 \),

\[ Y \left( BN - \frac{N}{2} + l \right) = X \left( \frac{N}{2} + l \right), \]

for \( l = 1, 2, ..., \frac{N}{2} - 1 \),

\[ y(l) = 0, \]

for \( l = \frac{N}{2} + 1, \frac{N}{2} + 2, ..., BN - \frac{N}{2} - 1 \),

and

\[ Y \left( \frac{N}{2} \right) = X \left( BN - \frac{N}{2} \right) = \frac{1}{2} X \left( \frac{N}{2} \right). \] (1)

Then

\[ y(m) = B \sum_{l=0}^{BN-1} y(l) e^{j \frac{2\pi m n}{BN}}, \]

for \( m = 0, 1, ..., BN - 1 \).

1.2 Zero interleaving and filtration interpolation

The spectrum mirroring principle of a zero interleaved signal and a following lowpass filtration is shown in [4]. In a higher order subsampling, this scheme can be considered as one step of a multistage algorithm. In 2D case, we have more ways of zero interleaving, but mainly the one from [5], where we speak about the n-th order interpolation. Interpolations (subsampling) of 1st and 2nd order are the most important for us.

2. Discrete cas-cas Transform (DCCT)

2.1 Interpolation 1D DHYT with zero completing

The process is very similar to DFT interpolation, only interpolating function is identical with DHYT basis. Consequently, it is necessary to alter only initializing and terminating step in (1).

2.2 Interpolation with zero interleaving DCCT

The spectrum can be described by the equation

\[ Y_N = \begin{pmatrix} A & B \\ B & A \end{pmatrix} \] (2)

where \( Y_N \) is a matrix with dimension \( N \times N \) and \( A, B \) are matrices of dimension \( N/2 \times N/2 \).

Let us have the following transfer function of filter, which expresses the zero completing for image of half area. For image 8x8, type 1 is
\[
\begin{pmatrix}
1 & 1 & 1 & 0.5 & 0.5 & 0.5 & 1 & 1 \\
1 & 1 & 1 & 0.5 & 0.5 & 0.5 & 1 & 1 \\
1 & 1 & 0.5 & 0 & 0 & 0 & 0.5 & 1 \\
0.5 & 0.5 & 0 & 0 & 0 & 0 & 0.5 & 1 \\
0.5 & 0.5 & 0 & 0 & 0 & 0 & 0.5 & 1 \\
1 & 1 & 0.5 & 0 & 0 & 0 & 0.5 & 1 \\
1 & 1 & 0.5 & 0.5 & 0.5 & 1 & 1 & 1
\end{pmatrix}
\]

As a reconstructed image, we will call an image produced by following operation:

\[
y_{N_1\times N_2} = 2 \cdot (U_{H_1})^T \cdot (H_{N_1\times N_2} \odot Y_{N_1\times N_2}) \cdot U_{N_2}
\]

where

\[
Y = H \odot Y
\]

is so called Hadamard product, for which

\[
Y_y = H_y \odot Y_y
\]

2.3 Interpolation with zero completing with DCCT 2

The spectrum can be described by the formula

\[
Y_N = \begin{pmatrix} A \\
B \end{pmatrix} \begin{pmatrix} J_{N/2} \cdot B \cdot J_{N/2} & J_{N/2} \cdot A \cdot J_{N/2} \end{pmatrix}
\]

The transfer function of filter type 2 is the matrix

\[
\begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 0.5 & 0.5 & 0.5 \\
1 & 1 & 1 & 1 & 1 & 0.5 & 0.5 & 0.5 \\
1 & 1 & 1 & 1 & 1 & 0.5 & 0.5 & 0.5 \\
1 & 1 & 1 & 0.5 & 0.5 & 0 & 0 & 0 \\
1 & 1 & 1 & 0.5 & 0.5 & 0 & 0 & 0 \\
0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

The transfer function of filter type 3 is the matrix, which for image 8x8 is

\[
H_8 = \begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

Or, after taking a non-separable spectrum symmetry (around reverted diagonal) in account, we can have also another type of filter transfer function, which is good with rectangular transforms. We will call it the type 4 transfer function.

\[
H_8 = \begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0.5 \\
1 & 1 & 1 & 1 & 1 & 1 & 0.5 & 0 \\
1 & 1 & 1 & 1 & 1 & 0.5 & 0 & 0 \\
1 & 1 & 1 & 0.5 & 0 & 0 & 0 & 0 \\
1 & 1 & 0.5 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

3. 2D Interpolation of 1st Order - Conclusion

From the preceding parts and other experiences with 2D DOTs, we can derive Table 1.

4. 2D interpolation of 2nd order (eventually even B_1, B_2 order)

4.1 2D interpolation of 2nd order with zero completing

In this case, an image is separable subsampled, consequently the interpolation can be done separately: zero completing as in 1D case for the row spectrum and for the spectrum of transformed and completed rows in spectrum after column transform.

<table>
<thead>
<tr>
<th>2D DOT MIRROWING</th>
<th>DCT II</th>
<th>DCCT</th>
<th>DFT</th>
<th>DCCT2</th>
<th>DST</th>
<th>MHT</th>
<th>DCT III</th>
<th>DLT</th>
<th>HT</th>
</tr>
</thead>
<tbody>
<tr>
<td>TYPE</td>
<td>no</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>FILTER TYPE</td>
<td>-</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3.4</td>
<td>3.4</td>
<td>3.4</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Tab. 1 Interpolation qualities of DOTs in transform of zero interleaved signal (1st order subsampling)
4.2 Zero interleaving interpolation

Spectrum of transforms from Tab. 1, which have feature of mirroring the spectra of image zero interleaved, is for image $\hat{x}_{H\times H}$.

A. 2D DFT and DCCT

$$Y_{H\times H} = \begin{pmatrix}
\mathbf{A}_{H/2} & 0 \\
0 & \mathbf{A}_{H/2}
\end{pmatrix}
\begin{pmatrix}
\mathbf{A}_{H/2} & \mathbf{A}_{H/2} \\
\mathbf{A}_{H/2} & \mathbf{A}_{H/2}
\end{pmatrix}
\begin{pmatrix}
\mathbf{H}_{H/2} & 0 \\
0 & \mathbf{H}_{H/2}
\end{pmatrix}
$$

(11)

B. 2D DST, MHT, WST, IDCT II and IDLT

$$Y_{H\times H} = \begin{pmatrix}
\mathbf{J}_{H/2} & \mathbf{A}_{H/2} \\
\mathbf{A}_{H/2} & \mathbf{J}_{H/2}
\end{pmatrix}
\begin{pmatrix}
\mathbf{J}_{H/2} & 0 \\
0 & \mathbf{J}_{H/2}
\end{pmatrix}
\begin{pmatrix}
\mathbf{H}_{H/2} & 0 \\
0 & \mathbf{H}_{H/2}
\end{pmatrix}
\begin{pmatrix}
\mathbf{J}_{H/2} & 0 \\
0 & \mathbf{J}_{H/2}
\end{pmatrix}
$$

(12)

Then

$$y_{H\times H} = 4 \cdot \{U_{H}\}^T \cdot (H_{H\times H} \circ Y_{H\times H}) \cdot U_{H}$$

(13)

Results for image NELA of size 256x256 pixels are presented in Tab. 2 and 3.

<table>
<thead>
<tr>
<th>2D DOT</th>
<th>overlap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8x8/8x8</td>
</tr>
<tr>
<td>DFT, DCCT</td>
<td>42,941</td>
</tr>
<tr>
<td>DCT III</td>
<td>47,208</td>
</tr>
<tr>
<td>DST</td>
<td>55,433</td>
</tr>
<tr>
<td>MHT</td>
<td>68,362</td>
</tr>
<tr>
<td>DLT</td>
<td>82,017</td>
</tr>
<tr>
<td>WST</td>
<td>42,628</td>
</tr>
</tbody>
</table>

Tab. 2 Dependence of MSE between 1st order interpolated image and original from different DOTS, block size and overlap size, where nominator is size of transform block and denominator is size of block retained as a result of interpolation.

<table>
<thead>
<tr>
<th>2D DOT</th>
<th>overlap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8x8/8x8</td>
</tr>
<tr>
<td>DFT, DCCT</td>
<td>166,953</td>
</tr>
<tr>
<td>DCT III</td>
<td>1074,254</td>
</tr>
<tr>
<td>DST</td>
<td>1444,731</td>
</tr>
<tr>
<td>MHT</td>
<td>1960,868</td>
</tr>
<tr>
<td>DLT</td>
<td>1502,382</td>
</tr>
<tr>
<td>WST</td>
<td>168,396</td>
</tr>
</tbody>
</table>

Tab. 3 Dependence of MSE between 2nd order interpolated image and original from different DOTS, block size and overlap size, where nominator is size of transform block and denominator is size of block retained as a result of interpolation.

5. Multilayer Perceptron (MLP) as a Non-linear Interpolator

To perform neural network interpolation of the 1st and the 2nd order, we concentrate on the multilayer perceptron (MLP). The basic MLP building unit is the model of an artificial neuron. This unit computes the weighted sum of the inputs plus the bias weight and passes this sum through the activation function (usually sigmoid).

In a multilayer configuration, the outputs of the units in one layer form the inputs to the next layer. The weights of the network are usually computed by training the network using the backpropagation algorithm.

A multilayer perceptron represents nested sigmoidal scheme [6], [3] - its form for the single output neuron is

$$F(x, w) = \varphi \left( \sum_i w_{ij} \varphi \left( \sum_k w_{jk} \varphi \left( \ldots \varphi \left( \sum_l w_{li} x_i \right) \right) \right) \right)$$

(14)

where $\varphi()$ is a sigmoidal activation function, $w_{ij}$ is the synaptic weight from neuron $j$ in the last hidden layer to the single output neuron $o$, and so on for the other synaptic weights, $x_i$ is the $i$-th element of the input vector $x$. The weight vector $w$ denotes the entire set of synaptic weights ordered by layer, then neurons in a layer, and then number in a neuron.

Here we use the standard backpropagation algorithm without any modifications. Our notation for the MLP architecture is following: for a 3-layer neural network, the configuration n-m-p means that the MLP contains n input neurons, m hidden neurons, and p output neurons.

We are interested in 2 approaches in using MLP as the 1st order interpolator:
- non-block approach - 1 pixel is interpolated from 4 surrounding pixels
- block approach - 4x4 block is taken, 8 missing values are reconstructed from 8 original pixels

For 2nd order interpolation, we use block approach only; 4x4 block is taken, 12 missing values are reconstructed from 4 original pixels.

For training the neural networks, a 512x512 training image constructed from fragments of images of various poses and sizes was used. Lena 256x256 image (called here NELA) was used for testing.

The results for 1st order interpolation for non-block and block approaches are shown in Tab. 4 and 5, respectively.

The results for 2nd order interpolation for block approach are shown in Tab. 6.

From the presented tables it can be seen, that within one chosen method there is no significant difference between the chosen MLP architecture, i.e. the simplest architecture can be used. For 1st order interpolation, very fast training can be noticed (by 1 cycle we mean presenting training image to the network).
Tab 4. 1st order interpolation results for NELA 256x256 by various MLP configurations, trained by standard backpropagation, learning rate parameter=0.1, shuffled patterns, non-block approach

<table>
<thead>
<tr>
<th>Network</th>
<th>MSE</th>
<th>PSNR [dB]</th>
<th>Number of cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-1</td>
<td>16.82</td>
<td>35.87</td>
<td>5</td>
</tr>
<tr>
<td>4-2-1</td>
<td>16.5</td>
<td>35.96</td>
<td>3</td>
</tr>
<tr>
<td>4-8-1</td>
<td>16.97</td>
<td>35.83</td>
<td>1</td>
</tr>
<tr>
<td>4-16-1</td>
<td>17.65</td>
<td>35.66</td>
<td>3</td>
</tr>
<tr>
<td>4-8-16-1</td>
<td>17.24</td>
<td>35.76</td>
<td>6</td>
</tr>
</tbody>
</table>

Tab 5. 1st order interpolation results for NELA 256x256 by various MLP configurations, trained by standard backpropagation, learning rate parameter=0.1, shuffled patterns, block approach (4x4 blocks)

<table>
<thead>
<tr>
<th>Network</th>
<th>MSE</th>
<th>PSNR [dB]</th>
<th>Number of cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-8</td>
<td>25.8</td>
<td>34.02</td>
<td>3</td>
</tr>
<tr>
<td>8-16-8</td>
<td>29.17</td>
<td>33.48</td>
<td>6</td>
</tr>
<tr>
<td>8-16-32-8</td>
<td>27.17</td>
<td>33.79</td>
<td>30</td>
</tr>
</tbody>
</table>

Tab 6. 2nd order interpolation results for NELA 256x256 by various MLP configurations, trained by standard backpropagation, learning rate parameter=0.001, shuffled patterns, block approach (4x4 blocks)

<table>
<thead>
<tr>
<th>Network</th>
<th>MSE</th>
<th>PSNR [dB]</th>
<th>Number of cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-12</td>
<td>128.4</td>
<td>27.05</td>
<td>40</td>
</tr>
<tr>
<td>4-8-12</td>
<td>128.3</td>
<td>27.05</td>
<td>200</td>
</tr>
<tr>
<td>4-8-10-12</td>
<td>128</td>
<td>27.06</td>
<td>300</td>
</tr>
</tbody>
</table>

About authors...

Jaroslav POLEC received the Ing. (MSc.) degree in electrical engineering, PhD. degree in telecommunications and Doc. degree from FEI STU in Bratislava. At present he is the associate professor at the Department of Telecommunications of FEI STU in Bratislava. His research interests include signal processing and probability models.

Miloš ORAVEC received the Ing. (MSc.) degree in electrical engineering and Dr. degree in applied informatics from FEI STU in Bratislava. At present he is lecturer at the Department of Telecommunications of FEI STU in Bratislava. His research interests include signal processing and neural networks.

References:


