A New Kind of Super-Resolution Reconstruction Algorithm Based on the ICM and the Constrained Cubic Spline Interpolation

Xiangguang Zhang¹², Zeyu Zheng²³, Ichio Asanuma², Yongsheng Xu¹,*
¹: Key Laboratory of Ocean Circulation and Wave, Institute of Oceanology, Chinese Academy of Sciences, 7 Nanhai Road, Qingdao, 266071, China
²: Department of Environmental Information, Tokyo University of Information Sciences, 4-1 Wakaba-ku, Chiba, 265-8501, Japan
³: Department of Physics and Centre for Computational Science and Engineering, National University of Singapore, 21 Lower Kent Ridge Road, Singapore 117542, Republic of Singapore
E-mail: zxg@qdio.ac.cn

Abstract

Super-resolution reconstruction of image is highly dependent on the data outliers. This work addresses the super-resolution reconstruction design of the Intersecting Cortical Model (ICM) algorithm applied to the constrained cubic spline interpolation. Based on a simplification of the Pulse-Coupled Neural Network (PCNN), we propose a design strategy to reduce the effects of outliers on the reconstructed image. Intersecting Cortical Model (ICM) has gained widely research as a new artificial neural network. It derives directly from the studies of the small mammal's visual cortex. Cubic spline interpolation is a useful technique to interpolate between known data points due to its stable and smooth characteristics. Unfortunately it does not prevent the high-frequency information, which is essential for many image processing applications. This article presents a new interpolation method that combines the smooth curve characteristics of spline interpolation, with the non-smoothing behavior of linear interpolation. The theory analysis and the simulation experiments of the image processing indicate that this kind of super-resolution reconstruction algorithm may reduce the effects of outliers effectively, while keeping the detail and quality of the image.

Key Words: Intersecting Cortical Model, Nonlinear filter, Median filter, Constrained Cubic Spline Interpolation, High-frequency Information

1. Introduction

An approach to improve digital image quality which has attracted large interest in the past decade is super resolution reconstruction (SRR). One of the major issues regarding SRR algorithms is their dependence on an accurate modeling of the SRR problem [1]. Outliers are defined as data points with different distributional characteristics than the assumed model, so this kind of reconstruction algorithm is highly dependent on the data outliers. We focus on the non-Gaussian outliers in this paper, and concentrate on reducing the effects of salt and pepper outliers.

In image processing, non-linear filter [2-6] has been studied widely because it can not only remove outlier effectively but also keep details of the image sufficiently. At present, there are
many typical nonlinear filter algorithms such as Median filter [7], Morphology filter [8], Stack filter [9], some improved filter algorithms based on the Median filter and so on.

Based on the studies of the cat’s visual cortex, the Pulse Coupled Neural Networks (PCNN) becomes a mathematical model based on the mammal's visual nerve nets. PCNN is different from the traditional multilayer neural networks in that it is a single layer model, which is suited to real-time image processing. This is a neural network that without any training needed, generates a sequence of binary images for the input digital image. To simplify the PCNN, Kinser introduced the Intersecting Cortical Model (ICM), earlier called the Unified Cortical Model (UCM) [10].

In order to solve this problem, which the salt and pepper outliers can affect the super-resolved images greatly, the paper suggests a kind of designing project of the Intersecting Cortical Model (ICM) algorithm applied to super-resolution reconstruction. The theory analysis and the simulation experiments of the image processing indicate that this algorithm has the ability to reduce the effects of outliers and protect the details of the image.

### 1.1. Intersecting Cortical Model

Eckhorn, Reitboeck, Amdt and Dicke had studied the cat visual cortex and discovered that the midbrain in an oscillating way creates binary images that extract different features from the visual impression [11]. Based on these so-called pulse images the actual image is created in the cat brain. They developed a neural network that simulated this behavior. The network thus creates binary output images from the input, which is the visual impression. Johnson realized that this could be used in the area of digital image processing and changed the model mentioned above for this purpose. The result was the Pulse-Coupled Neural Network (PCNN) [12]. This is a neural network that without any training needed, generates a sequence of binary images for the input digital image. To simplify the PCNN, Kinser introduced the Intersecting Cortical Model (ICM), earlier called the Unified Cortical Model (UCM). It works in same way as the PCNN and generates binary output images from a digital input image, but instead of five equations of PCNN the ICM now have three:

\[
F_y[n] = fF_y[n-1] + S_y + W(Y_y[n-1])P
\]  \hfill (1.1)

\[
Y_y[n] = \begin{cases} 1, & \text{if } F_y[n] > \Theta_y[n-1] \\ 0, & \text{otherwise} \end{cases}
\]  \hfill (1.2)

\[
\Theta_y[n] = g\Theta_y[n-1] + hY_y[n]
\]  \hfill (1.3)
$S$ is the input, $F$ is the state of the neuron, $Y$ is the output, $\Theta$ is the threshold and $f$, $g$ and $h$ are constants. $n = 1, 2, \ldots, N$ is the iteration number and $N$ is the number of iterations. The function $W(\cdot)$ calculates the communication from the neighboring neurons.

### 1.2. Constrained Cubic Spline Interpolation

Interpolation is used to estimate the value of a function between known data points without knowing the actual function. Interpolation methods can be divided into two main categories [13-14]:

Global interpolation. These methods rely on constructing a single equation that fits all the data points. This equation is usually a high degree polynomial equation. Although these methods result in smooth curves, they are usually not well suited for image processing applications, as they are prone to severe oscillation and overshoot at intermediate points.

Piecewise interpolation. These methods rely on constructing a polynomial of low degree between each pair of known data points. If a first degree polynomial is used, it is called linear interpolation. For second and third degree polynomials, it is called quadratic and cubic splines respectively. The higher the degree of the spline, the smoother the curve. Splines of degree m, will have continuous derivatives up to degree m-1 at the data points.

Linear interpolation result in straight line between each pair of points and all derivatives are discontinuous at the data points. As it never overshoots or oscillates, it is frequently used in image processing despite the fact that the curves are not smooth.

To obtain a smoother curve, cubic splines are frequently recommended. They are generally well behaved and continuous up to the second order derivative at the data points. Even though cubic splines are less prone to oscillation or overshoot than global polynomial equations, they do not prevent it. Thus, the use of cubic splines in image processing is limited to applications where oscillation and overshoot are acceptable or desirable.

#### 1.2.1. Traditional Cubic Splines

Consider a collection of known points $(x_0, y_0)$, $(x_1, y_1)$, ... $(x_{i-1}, y_{i-1})$, $(x_i, y_i)$, $(x_{i+1}, y_{i+1})$, ... $(x_n, y_n)$. To interpolate between these data points using traditional cubic splines, a third degree polynomial is constructed between each point. The equation to the left of point $(x_i, y_i)$ is indicated as $f_i$ with a $y$ value of $f_i(x_i)$ at point $x_i$. Similarly, the equation to the right of point $(x_i, y_i)$ is indicated as $f_{i+1}$ with a $y$ value of $f_{i+1}(x_i)$ at point $x_i$.

Traditionally the cubic spline function, $f_i$, is constructed based on the following criteria:

Curves are third order polynomials,
\[ f_i(x) = a_i + b_i x + c_i x^2 + d_i x^3 \]  
\hspace{1cm} (1.4)

Curves pass through all the known points,
\[ f_i(x_i) = f_{i+1}(x_i) = y_i \]  \hspace{1cm} (1.5)

The slope, or first order derivative, is the same for both functions on either side of a point,
\[ f'_i(x_i) = f'_{i+1}(x_i) \]  \hspace{1cm} (1.6)

The second order derivative is the same for both functions on either side of a point,
\[ f''_i(x_i) = f''_{i+1}(x_i) \]  \hspace{1cm} (1.7)

This results in a matrix of \( n-1 \) equations and \( n+1 \) unknowns. The two remaining equations are based on the border conditions for the starting point, \( f_1(x_0) \), and end point, \( f_n(x_n) \). Historically one of the following border conditions have been used \[8-9\]:

**Natural splines.** The second order derivative of the splines at the end points are zero.
\[ f''_1(x_0) = f''_n(x_n) = 0 \]  \hspace{1cm} (1.8a)

**Parabolic runout splines.** The second order derivative of the splines at the end points are the same as at the adjacent points. The result is that the curve becomes a parabolic curve at the end points.
\[ f''_1(x_0) = f''_n(x_n) = f''_{n-1}(x_{n-1}) \]  \hspace{1cm} (1.8b)

**Cubic runout splines.** The curve degrades to a single cubic curve over the last two intervals by setting the second order derivative of the splines at the end points to:
\[ f''_1(x_0) = 2f''_1(x_1) - f''_2(x_2) \]
\[ f''_n(x_n) = 2f''_n(x_{n-1}) - f''_{n-1}(x_{n-2}) \]  \hspace{1cm} (1.8c)

**Clamped spline.** The first order derivative of the splines at the end points are set to known values.
\[ f'_1(x_0) = f'(x_0) \]
\[ f'_n(x_n) = f'(x_n) \]  \hspace{1cm} (1.8d)

In traditional cubic splines equations 1.4 to 1.8 are combined and the \( n+1 \) by \( n+1 \) tridiagonal matrix is solved to yield the cubic spline equations for each segment [8]. As both the first and second order derivative for connecting functions are the same at every point, the result is a very smooth curve.

Even though traditional cubic splines are well behaved for many applications, it does not prevent the high-frequency information, which is essential for many image processing applications. Clearly this behavior is unacceptable for image processing applications, and we have little choice but to revert back to linear interpolation.

**1.2.2. Proposed Constrained Cubic Splines**
The principle behind the proposed constrained cubic spline is to prevent high frequency information by sacrificing smoothness. This is achieved by eliminating the requirement for equal second order derivatives at every point (equation 1.7) and replacing it with specified first order derivatives.

Thus, similar to traditional cubic splines, the proposed constrained cubic splines are constructed according to equations (1.5), (1.6) and (1.8a). Equation (1.7) is replaced by,

A specified first order derivative, or slope, at every point,

$$f_i'(x_i) = f_{i+1}'(x_i) = f_i''(x_i)$$  \hspace{1cm} (1.9)

The key step becomes the calculation of the slope at each point. Intuitively we know the slope will be between the slopes of the adjacent straight lines, and should approach zero if the slope of either line approaches zero.

A relatively simple equation that works well and satisfies these requirements is:

$$f_i''(x_i) = \frac{2}{x_{i+1} - x_i} + \frac{x_i - x_{i-1}}{y_{i+1} - y_i} \cdot \frac{y_i - y_{i-1}}{y_{i+1} - y_i} = 0 \hspace{1cm} (1.10a)$$

if slope changes sign at point.

Equation (1.10a) is only valid for intermediate points. The slope at the end points is based on rewriting equation (1.8a) to yield,

$$f'_i(x_0) = \frac{3(y_i - y_0)}{2(x_i - x_0)} - \frac{f''(x_i)}{2} \hspace{1cm} (1.10b)$$

$$f'_n(x_n) = \frac{3(y_n - y_{n+1})}{2(x_n - x_{n+1})} - \frac{f''(x_{n-1})}{2} \hspace{1cm} (1.10c)$$

As the slope at each point is known, it is no longer necessary to solve a system of equations. Each spline function, as given by equation (1.4), can be calculated based on the two adjacent points on each side. This is summarized in equations (1.11) to (1.16) below.

$$f'_i(x_{i-1}) = -\frac{2[f'_i(x_i) + 2f'_i(x_{i+1})]}{(x_i - x_{i+1})} + \frac{6(y_i - y_{i+1})}{(x_i - x_{i+1})^2} \hspace{1cm} (1.11)$$

$$f'_i(x_i) = -\frac{2[f'_i(x_i) + f'_i(x_{i+1})]}{(x_i - x_{i+1})} - \frac{6(y_i - y_{i+1})}{(x_i - x_{i+1})^2} \hspace{1cm} (1.12)$$

$$d_i = \frac{f'_i(x_i) - f'_i(x_{i+1})}{6(x_i - x_{i+1})} \hspace{1cm} (1.13)$$

$$c_i = \frac{x_i f'_i(x_{i+1}) - x_{i+1} f'_i(x_i)}{2(x_i - x_{i+1})} \hspace{1cm} (1.14)$$
The behavior of the proposed constrained cubic spline fits image processing needs well in cases where oscillation or overshoot cannot be tolerated.

2. New Methodology

Analyzing the proposed constrained cubic spline interpolation algorithm, we can find that the constrained cubic spline interpolation can keep the details of image sufficiently if the non-Gaussian outliers do not exist. But the image that is contaminated with the salt and pepper outliers, the effect of the constrained cubic spline interpolation algorithm is inadequate. To overcome this shortage, this paper suggests a kind of designing project of the Intersecting Cortical Model (ICM) algorithm applied to super-resolution reconstruction. Our algorithm can be described as following: first, according to the intensity of an outlier pixel having different characteristics with its surroundings, the proper set of the parameters of ICM will make the neuron corresponding to the outlier pixel output a pulse leading its neighborhood at iterations. Second, we use the median filter to remove the outliers. In the first step, we can find the position of salt and pepper outlier through the ICM, the median filter is used to reduce the outlier. In the end, we reconstruct the result of median filter by the constrained cubic spline interpolation algorithm.

3. Tests

The image “lena” and “testpad1” (256×256 8 bit) are used to test the performance of our super-resolution reconstruction algorithm. The results of the simulation experiments are listed in the Fig. 1. The outlier is 20% Salt-and-pepper noise. The PSNR (Peak Signal-to-Noise Ratio), Mean and Square deviation of those filters are listed in the Table 1 and Table 2. Analyzing the data of the table, we can find the following results: the PSNR value of the new algorithm is much bigger than that of the constrained cubic spline interpolation algorithm without removing the outlier, and the square deviation of the new algorithm is much smaller than that of the constrained cubic spline interpolation algorithm. Moreover, from the results of the image processing, we can find that the ability of this algorithm to keep the high-frequency details of the image is much better than that of the constrained cubic spline interpolation algorithm.
Fig. 1. Comparison of the simulation experiments (a, Original HR image of lena and testpad1; b, LR image and LR image with Salt and Pepper Outlier of lena and testpad1; c, The results of proposed super-resolution reconstruction algorithm of lena and testpad1, and d, The results of the constrained cubic spline interpolation algorithm of lena and testpad1).
Table 1. Comparison of PSNR, square deviation, mean of “lena” (noise density: 20%)

<table>
<thead>
<tr>
<th>Image type</th>
<th>Constrained cubic spline interpolation algorithm</th>
<th>Our proposed super-resolution reconstruction algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Square deviation</td>
<td>2056</td>
<td>186.53</td>
</tr>
<tr>
<td>Mean</td>
<td>-6.7854</td>
<td>0.3542</td>
</tr>
<tr>
<td>PSNR</td>
<td>16.874</td>
<td>28.635</td>
</tr>
</tbody>
</table>

Table 2. Comparison of PSNR, square deviation, mean of “testpad1” (noise density: 20%)

<table>
<thead>
<tr>
<th>Image type</th>
<th>Constrained cubic spline interpolation algorithm</th>
<th>Our proposed super-resolution reconstruction algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Square deviation</td>
<td>3046.8</td>
<td>306.42</td>
</tr>
<tr>
<td>Mean</td>
<td>3.524</td>
<td>0.4687</td>
</tr>
<tr>
<td>PSNR</td>
<td>14.375</td>
<td>25.364</td>
</tr>
</tbody>
</table>

The definition of the PSNR is following:

\[ PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) \text{ dB} \]  \hspace{1cm} (1.17)

And

\[ MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (Y_{i,j} - S_{i,j})^2 \]  \hspace{1cm} (1.18)

Where \( M \) and \( N \) respectively are the total number of pixels in the horizontal and vertical dimensions of the image; \( S_{i,j} \) is the value of original pixels and \( Y_{i,j} \) is the value of filtered image pixels.

4. Conclusions

The performance of our super-resolution reconstruction algorithm has been compared with that of the constrained cubic spline interpolation algorithm. Simulation experiments reveal that the proposed algorithm significantly outperforms constrained cubic spline interpolation algorithm by having much higher PSNR with real-time, consistent and stable performance.

5. Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments that helped to improve the paper. This work was partly supported by CAS (Chinese Academy of Sciences) Innovation Program Y22114101Q and CAS “100 Talent” Program Y32109101L. This research is also partly supported by the research project for a sustainable development of economic and social structure dependent on the environment of the eastern coast of Asia, by MEXT-Supported Program for the Strategic Research Foundation at Private Universities from 2008-2012.
References


*Corresponding author: Yongsheng Xu, Ph.D.

Key Laboratory of Ocean Circulation and Wave,
Institute of Oceanology,
Chinese Academy of Sciences,
7 Nanhai Road, Qingdao, 266071, China
Tel/Fax: +86-532-8289-8352
E-mail: yongsheng.xu@qdio.ac.cn