An Iterative Super Resolution Algorithm Based On Adaptive FIR Wiener Filtering

Kyle Xiang ZHANG, Yuk-Hee CHAN and Wan-Chi SIU

Centre for Multimedia Signal Processing
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong SAR

Abstract

Aiming at image/video super resolution applications, this paper presents a spatially adaptive FIR wiener filter for super resolution reconstruction. Missing high resolution samples can be estimated as the weighted sum of nearby low resolution samples. In the proposed algorithm, the optimal weighting coefficients for each of nearby low resolution samples are determined with a distance-based correlation model of samples and iteratively refined according to the most updated estimates of the missing high resolution samples. Both objective and subjective measurements in our simulations show that the proposed algorithm can produce a better result as compared with some conventional algorithms across different noise conditions.

I. Introduction

Super Resolution (SR) is one of the major methods to realize spatial resolution up-sampling. It aims to reverse the image formation process and then re-sample the restored signal at a higher sampling rate. SR has been studied over two decades and many algorithms[1-17] have been proposed.

Iterative back projection (IBP) [12] is one of the successful conventional SR methods. It simulates the image formation process to form a low resolution (LR) image from an estimated high resolution (HR) image. The difference between the resultant LR image and the corresponding observed LR image is then used to improve the estimate of the HR image. Accordingly, the HR image can be iteratively restored to minimize the difference between the observed LR images and the simulated ones. Based on IBP, a robust super resolution algorithm (RSR) [13] is further developed. It introduces a median estimator to solve the problem caused by the presence of outliers during the reconstruction process.

Papoulis and Gerchberg's POCS algorithm [14,15] is another successful iterative solution for SR. In this algorithm, the original HR image is interpreted as the solution of a projection function. This function iteratively projects estimated solutions onto several non-empty and closed-convex sets which are understood to be the prior information of the final solution.

Recently, a SR algorithm based on normalized convolution was proposed by Tuan Q. Pham et. al. [16]. This algorithm performs a robust polynomial fit over a local neighborhood based on a structure-adaptive applicability function. Each pixel is assigned a robust certainty value based on its intensity difference from its neighbors to control its influence in the polynomial fitting process.

Hardie [17] proposed a fast SR algorithm by using an adaptive wiener filter. In particular, a HR sample is estimated to be the weighted sum of its nearby LR samples. The weights are derived based on a simple space-invariant intensity correlation model among pixels, which does not work properly when noisy and edge regions are encountered.

In this paper, an iterative SR algorithm based on adaptive FIR Wiener filtering is proposed. This algorithm iteratively compensates for the cross-correlation estimation error caused by the presence of outliers in a local region. It is generally assumed that samples are gradually less correlated when they are more distant away from each other. However, the presence of noise and edges may break this assumption. By considering this factor, an iterative process is proposed to optimize the correlation estimation and softly minimize the effect of local outliers during the reconstruction process.

The remainder of this paper is organized as follows. In Section II, an observation model is described as the basis of the development of the proposed algorithm. Section III presents a detailed description of the proposed algorithm. Section IV provides some simulation results for performance evaluation. The conclusion is drawn in Section V.

II. Observation Model

The observation model is used to model the image formation process. It can be mathematically described as

\[ y = D(b * G(x)) + n \]  

where \( G(\bullet) \) is a geometrical transformation operator, \( b \) is a point spread function (PSF), \( D(\bullet) \) is a down-sampling operator and \( n \) is an additive random noise.

III. Proposed Algorithm

To start the process, all LR pixels in successive LR images are registered onto a high resolution grid. Through a spatial sliding window covering a 3d×3d region of the high resolution grid, where \( d \) is the ratio of the target resolution of the reconstructed HR image to the original resolution of a LR image, all LR pixels registered in the spatial sliding window are collected to estimate the \( d \times d \) HR pixels in the center of the window.

Let the number of the registered LR pixels in the window be \( L \) and the total number of the HR pixels to be estimated with
these $L$ LR pixels be $N = d \times d$. The $N$ estimated HR pixels estimated at iteration $t$ are lexicographically ordered to form vector $\tilde{x}^{(t)} = [\tilde{x}_1^{(t)}, \tilde{x}_2^{(t)}, \ldots, \tilde{x}_N^{(t)}]^T$. Specifically, $\tilde{x}_m^{(t)}$ is the estimate of the $m^{th}$ HR pixel obtained by

$$\tilde{x}_m^{(t)} = \frac{\sum_{k=1}^{L} \omega_{mk} y_k}{\sum_{k=1}^{L} \omega_{mk}} \text{ for } m = 1, 2, \ldots, N $$

in iteration $t$, where $\tilde{y} = [y_1, y_2, \ldots, y_L]^T$ are the registered LR pixels in the window, and $\omega_{mk} y_k / \sum_{k=1}^{L} \omega_{mk}$ is the normalized weight of the $k^{th}$ LR pixel when estimating $\tilde{x}_m^{(t)}$.

A LR pixel’s weight is contributed by $\psi_{mk}^{(t)}$ and $\omega_{mk}$. Specifically, coefficients $\omega_{mk}$ are elements of matrix

$$W = \begin{bmatrix} \omega_{m1} & \omega_{m2} & \cdots & \omega_{mN} \\ \omega_{m2} & \omega_{m2} & \cdots & \omega_{mN} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{mL} & \omega_{m2L} & \cdots & \omega_{mNL} \end{bmatrix} = R^{-1} \cdot P $$

where $R$ is the auto-correlation matrix of $\{ y_k | k = 1, \ldots, L \}$, and $P$ is the cross-correlation matrix of $\{ y_k | k = 1, \ldots, L \}$ and $\{ x_m | m = 1, \ldots, N \}$.

While $\omega_{mk}$ is determined by (i) the correlation among LR pixels and (ii) the correlation between LR pixels and HR pixels, $\psi_{mk}^{(t)}$ is an iteration-variant term used to rectify $\omega_{mk}$. In general, the correlation of two pixels is modeled as a function of their distance[17]. This simple model is space-invariant and does not reflect the real situation in the regions containing noise and edges. Accordingly, $\omega_{mk}$ cannot weight $y_k$ properly in these regions. $\psi_{mk}^{(t)}$ is used to improve the weighting effect by rectifying $\omega_{mk}$. Its details will be addressed in the last part of this section.

By considering that the observed LR pixel $y_k$ is corrupted with an additive zero mean random noise $n$ and its noisy free version is $f_k = y_k - n_k$, where $n_k$ is the noise magnitude for sample $y_k$, we have

$$R = E\{\tilde{y} \cdot \tilde{y}^T\} = E\{\tilde{f} \cdot \tilde{f}^T\} + \sigma_n^2 \cdot I $$

and

$$P = E\{\tilde{y} \cdot \tilde{x}^T\} = E\{\tilde{f} \cdot \tilde{x}^T\} $$

where $E\{\bullet\}$ is the expectation operator and $\sigma_n^2$ is the local variance of the noise.

In theory, $f_k$ are samples of $\{f(i,j)\}$, a blurred version of the original HR image $\{x(i,j)\}$, and hence we have $f(i,j) = x(i,j) * b(i,j)$, where $b$ is the blurring function and $(i,j)$ denotes a position in the HR grid. Accordingly, the cross-covariance of $x(i,j)$ and $f(i,j)$ and the auto-covariance of $f(i,j)$ can be, respectively, determined by

$$r_x(i,j) = r_x(i,j) * b(i,j) $$

and

$$r_f(i,j) = r_f(i,j) * b(i,j) * b(-i,-j) $$

In general, the covariance between two HR pixels can be modeled as

$$r_x(\Delta i, \Delta j) = \sigma_x^2 \cdot \exp(-\gamma (\sqrt{\Delta i^2 + \Delta j^2})^2 $$

where $(\Delta i, \Delta j)$ is the distance of the two HR pixels, $\gamma$ is a tuning parameter and $\sigma_x^2$ is the local variance of the HR image. In particular, $\sigma_x^2$ is estimated as

$$\sigma_x^2 = \max\{\sigma_x^2 - \sigma_n^2 \cdot C \cdot \sigma_s^1 \} $$

where $\sigma_s^2$ is the variance of the $\{ x_m | m = 1, \ldots, L \}$ and

$$C = \int \exp(-\gamma (\sqrt{\Delta i^2 + \Delta j^2})^2) \cdot (b(i,j) * b(-i,-j)) \cdot dijd $$

With $r_{xx}(\Delta i, \Delta j)$ on hand, elements of $E\{f(i,j)^T\}$ and $E\{f(i,j) \tilde{x}^T\}$ in eqns. (4) and (5) and hence $W$ can be determined with $r_{xx}(\Delta i, \Delta j)$ and $r_{xx}(\Delta i, \Delta j)$, where $(\Delta i, \Delta j)$ is the distance of the two involved pixels in the HR grid.

$\psi_{mk}^{(t)}$, the parameter used to rectify $\omega_{mk}$, is initialized to be $\psi_{mk}^{(t)} = 1$, and then iteratively updated based on $\tilde{x}_m^{(t)}$, the most updated estimate of the $m^{th}$ HR pixel, as follows.

$$\psi_{mk}^{(t)} = \begin{cases} 1 & \text{if } t = 1 \\ \exp(-y_k - \Delta x_m^{(t-1)} / \sigma_s^1) & \text{if } t > 1 \end{cases} $$

Starting from iteration $t=1$, HR pixels are iteratively estimated until they converge to the final estimates. When $t=1$, we have $\psi_{mk}^{(t)} = 1$ and hence the 1st iteration of the proposed algorithm is equivalent to an adaptive Wiener filtering algorithm (different from the one proposed in [17]). When $t>1$, the algorithm evaluates the intensity difference between a particular LR pixel and the average of the HR pixel estimates in the local region and, in the current estimation, deemphasizes the contribution of the LR pixels the intensity values of which are very different from the local intensity mean of the estimated HR pixels. This helps to reject those outliers and improve the estimation performance in noisy and edge areas.

IV. Simulations

Simulations were carried out to evaluate the performance of the proposed algorithm. A set of 8-bit gray level aerial images shown in Fig. 1 were used as the original HR images in the simulation. Their LR versions were obtained with the observation model shown in Section II. In particular, each original HR image was blurred with a Gaussian PSF, corrupted with additive zero-mean Gaussian random noise of variance $\sigma_n^2$, and then down-sampled by 2 along each dimension. The resultant LR images were then enlarged to reconstruct the original with various SR algorithms including the proposed one. In other words, each HR image was estimated with 1 LR image and the enlarging ratio $d$ was 2. A spatial sliding...
window of size $6 \times 6$ was used during the reconstruction when the proposed algorithm was used. Tuning parameter $\gamma$ was selected to be 0.2. Fig. 2 shows the simulation results of several SR algorithms [12-16] for visual comparison.

Table 1 shows the performance achieved with different SR algorithms in terms of PSNR under different noise conditions. In particular, PSNR is defined as

$$
PSNR = 10 \cdot \log_{10} \left( \frac{255^2}{\frac{1}{N_0} \sum_{(i,j)} (x(i,j) - \hat{x}(i,j))^2} \right) \text{ in dB}
$$

(12)

where $x$ and $\hat{x}$ are, respectively, the original high resolution image and its estimate, and $N_0$ is the total number of image pixels involved in the comparison.

V. Conclusions

This paper presents a SR adaptive FIR wiener filter with cross-correlation refinement. Super resolution reconstruction normally suffers from the presence of outliers. Outliers corrupt the reconstruction of the high resolution image since they could be over considered. The proposed method helps to minimize the degradation caused by the outliers in a local region during the reconstruction process. Simulations demonstrated the effectiveness of the proposed algorithm. It suppresses random additive noise and produces outputs of better subjective and objective quality as compared with some conventonal iterative SR algorithms.

VI. Acknowledgement

This work was supported by Centre for Multimedia Signal Processing, The Hong Kong Polytechnic University (PolyU Grant 1-BB9T).

References

<table>
<thead>
<tr>
<th>Methods</th>
<th>( \sigma_n^2 = 50 )</th>
<th>( \sigma_n^2 = 100 )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Image.1</strong></td>
<td>23.12</td>
<td>28.32</td>
</tr>
<tr>
<td><strong>Image.2</strong></td>
<td>24.54</td>
<td>29.88</td>
</tr>
<tr>
<td><strong>Image.3</strong></td>
<td>24.78</td>
<td>29.78</td>
</tr>
<tr>
<td><strong>Image.4</strong></td>
<td>23.42</td>
<td>29.56</td>
</tr>
<tr>
<td><strong>Image.5</strong></td>
<td>24.90</td>
<td>30.44</td>
</tr>
<tr>
<td><strong>Image.6</strong></td>
<td>25.12</td>
<td>30.75</td>
</tr>
<tr>
<td><strong>Image.7</strong></td>
<td>25.97</td>
<td>31.25</td>
</tr>
<tr>
<td><strong>Image.8</strong></td>
<td>23.51</td>
<td>29.54</td>
</tr>
<tr>
<td><strong>Image.9</strong></td>
<td>23.79</td>
<td>29.60</td>
</tr>
<tr>
<td><strong>Image.10</strong></td>
<td>24.10</td>
<td>29.81</td>
</tr>
<tr>
<td><strong>Image.11</strong></td>
<td>24.25</td>
<td>30.01</td>
</tr>
<tr>
<td><strong>Image.12</strong></td>
<td>24.46</td>
<td>30.27</td>
</tr>
<tr>
<td><strong>Image.13</strong></td>
<td>24.94</td>
<td>30.82</td>
</tr>
<tr>
<td><strong>Image.14</strong></td>
<td>25.31</td>
<td>31.30</td>
</tr>
<tr>
<td><strong>Image.15</strong></td>
<td>26.56</td>
<td>32.56</td>
</tr>
<tr>
<td><strong>Image.16</strong></td>
<td>27.28</td>
<td>33.28</td>
</tr>
<tr>
<td><strong>Image.17</strong></td>
<td>27.84</td>
<td>33.84</td>
</tr>
<tr>
<td><strong>Image.18</strong></td>
<td>28.40</td>
<td>34.40</td>
</tr>
<tr>
<td><strong>Image.19</strong></td>
<td>29.02</td>
<td>35.02</td>
</tr>
<tr>
<td><strong>Image.20</strong></td>
<td>29.61</td>
<td>35.61</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td>24.44</td>
<td>29.52</td>
</tr>
</tbody>
</table>

Table 1 PSNR performance of various SR algorithms including IBP[12], RSR[13], POCS[14,15], SANC[16], AWF[17] and the proposed under different noise conditions.