A Sub-pixel Image Magnification Using Adaptive Linear Interpolation

Hoon Yoo
Presentation Outline

• Introduction
• Basic Concept of Interpolation
• Conventional Interpolation
• Previous Adaptive Linear Interpolation
• Proposed Method
• Example of Proposed Method
• Simulation Results
• Conclusions
Introduction

• Image interpolation plays a key role in the image processing literature
  – Image magnification
  – Image compression
  – Mosaicking color filter array
  – De-interlacing
  – Lifting-based wavelet transform

• Sub-pixel image interpolation has many applications in the multimedia industry
  – Mosaicking, de-interlacing, YUV format conversion, lifting-based wavelet transform, etc
Basic Concept of Interpolation

- With given discrete samples $f(x_k)$, generating continuous function as follows

$$\hat{f}(x) = \sum_k f(x_k) \beta(x - x_k)$$
Conventional Interpolation

• Linear

\[ \beta(x) = \begin{cases} 
1 - |x|, & 0 < |x| \leq 1 \\
0, & \text{elsewhere} 
\end{cases} \]

\[ \hat{f}(x) = \sum_k f(x_k) \beta(x - x_k) = (1 - s)f(x_k) + sf(x_{k+1}) \]
Conventional Interpolation

- **Keys’ Cubic Convolution Interpolation**

\[
\beta(x) = \begin{cases} 
(\alpha + 2)|x|^3 - (\alpha + 3)|x|^2 + 1, & 0 < |x| \leq 1 \\
\alpha|x|^3 - 5\alpha|x|^2 + 8\alpha|x| - 4\alpha, & 1 < |x| \leq 2 \\
0, & \text{elsewhere}
\end{cases}
\]

\[
\hat{f}(x) = f(x_{k-1})(\alpha s^3 - 2\alpha s^2 + \alpha s) + f(x_{k+0})((\alpha + 2)s^3 - (\alpha + 3)s^2 + 1) + f(x_{k+1})(-(\alpha + 2)s^3 + (2\alpha + 3)s^2 - \alpha s) + f(x_{k+2})(-\alpha s^3 + \alpha s^2)
\]
Conventional Interpolation

- Keys’ Cubic Convolution Interpolation

\[
\hat{f}(x) = f(x_{k-1})(\alpha s^3 - 2\alpha s^2 + \alpha s) \\
+ f(x_{k+0})(\alpha + 2)s^3 - (\alpha + 3)s^2 + 1) \\
+ f(x_{k+1})(-\alpha + 2)s^3 + (2\alpha + 3)s^2 - \alpha s) \\
+ f(x_{k+2})(\alpha s^3 + \alpha s^2)
\]

\[
\alpha = -\frac{1}{2}
\]

\[
\hat{f}(x) = f(x_{k-1})(-s^3 + 2s^2 - s) / 2 \\
+ f(x_{k+0})(3s^3 - 5s^2 + 2) / 2 \\
+ f(x_{k+1})(-3s^3 + 4s^2 + s) / 2 \\
+ f(x_{k+2})(s^3 - s^2) / 2
\]

\[
\hat{f}(x) = \frac{-1}{16} f(x_{k-1}) + \frac{9}{16} f(x_{k+0}) + \frac{9}{16} f(x_{k+1}) + \frac{1}{16} f(x_{k+2})
\]
Previous Adaptive Linear Interpolation

- Warped Distance Linear Interpolation
  \[
  \hat{f}(x) = (1-s)f(x_k) + sf(x_{k+1})
  \]

- Definition of warped distance as follows
  \[
  s' = s - kAs(s - 1)
  \]
  \[
  A = \frac{|f(x_{k+1}) - f(x_{k-1})| - |f(x_{k+2}) - f(x_k)|}{L - 1}
  \]

- Note that
  - The variable \( A \) is a pixel-based parameter
  - The variable \( k \) is an image-based parameter
    \((k = 8 \text{ fixed for the Lena image})\)
Previous Adaptive Linear Interpolation

- Shifted Linear Interpolation

\[ \hat{f}(x) \]

- The optimal \( t \) is around 0.21
- Efficient implementation consists of IIR filtering + constant warping \((s \leftarrow s - 0.21)\)
Proposed Method

- Warping distance $s$, based on MMSE (minimization mean square error)

$$\hat{f}(x) = (1-s)f(x_k) + sf(x_{k+1})$$

- Use distance $s$ as a pixel-based parameter
- Introduce a system to calculate $s$
  - Including low pass filter and MMSE
Proposed Method

- Generic diagram

\[ X_{\text{high}}(z) \]

\[ X_{\text{low}}(z) \]

\[ \text{Linear Interpolation} \]

\[ \text{Find } s \]

\[ d(z,s) \]

\[ \text{s} \]

\[ X_{\text{high}}(z,s) \]

\[ \text{LPF} \]

\[ G(z) \]

\[ \downarrow 2 \]
Proposed Method

- Systematic approach to use MMSE method

\[
x^R_n(s) = \sum_{m=-(M-1)/2}^{(M-1)/2} g_{m+(M-1)/2} x^H_{2n+m}(s)
\]
Example of Proposed Method

- Low complexity version

\[ x^R_n(s) = \sum_{m=-(M-1)/2}^{(M-1)/2} g_{m+(M-1)/2} x^H_{2n+m}(s) \]

- Apply 3-tap low pass filter \( \{1,2,1\}/4 \)

- Define cost function as follows

\[ C(s) = \frac{(x^L_n - x^R_n(s))^2 + (x^L_{n+1} - x^R_{n+1}(s))^2}{2} \]

- We have

\[ C(s) = \frac{(sc_0 + c_1)^2 + (sc_0 + c_2)^2}{2} \]

\[ c_0 = \frac{x^L_{n+1} - x^L_n}{4}, \quad c_1 = \frac{x^L_{n-1} - x^L_n}{8}, \quad c_2 = \frac{2x^L_n - 3x^L_{n+1} + x^L_{n+2}}{8} = -c_0 + \frac{x^L_{n+2} - x^L_{n+1}}{8} \]
Example of Proposed Method

• Get $s$ to minimize the cost as follows

$$C(s) = \frac{(sc_0 + c_1)^2 + (sc_0 + c_2)^2}{2}$$

• We have

$$s = -\frac{c_1 + c_2}{c_0} \begin{cases} \text{if } s < 0, \text{ then } s = 0 \\ \text{if } s > 1, \text{ then } s = 1 \end{cases}$$

$$c_0 = \frac{x_{n+1}^L - x_n^L}{4}, \quad c_1 = \frac{x_{n-1}^L - x_n^L}{8}, \quad c_2 = -c_0 + \frac{x_{n+2}^L - x_{n+1}^L}{8}$$
Simulation Results

- Objective quality test (PSNR)

<table>
<thead>
<tr>
<th>Images</th>
<th>Linear</th>
<th>Cubic Convolution</th>
<th>Wadi-Linear</th>
<th>Shifted Linear</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena</td>
<td>33.28</td>
<td>34.25</td>
<td>34.09</td>
<td>34.15</td>
<td>34.62</td>
</tr>
<tr>
<td>Peppers</td>
<td>31.57</td>
<td>31.96</td>
<td>31.61</td>
<td>31.80</td>
<td>32.00</td>
</tr>
<tr>
<td>Baboon</td>
<td>23.28</td>
<td>23.59</td>
<td>23.42</td>
<td>23.54</td>
<td>23.66</td>
</tr>
<tr>
<td>Airplane</td>
<td>30.33</td>
<td>31.08</td>
<td>30.48</td>
<td>31.07</td>
<td>31.15</td>
</tr>
<tr>
<td>Goldhill</td>
<td>31.01</td>
<td>31.49</td>
<td>31.45</td>
<td>31.40</td>
<td>31.64</td>
</tr>
<tr>
<td>Barbara</td>
<td>25.25</td>
<td>25.40</td>
<td>25.34</td>
<td>25.28</td>
<td>25.38</td>
</tr>
</tbody>
</table>
Simulation Results

• Subjective quality test

Conventional Linear                  Proposed Linear
Simulation Results

- Subjective quality test

![WaDi Linear](image1)
![Proposed Linear](image2)
Simulation Results

- Subjective quality test

![Shifted Linear](image1.png)  ![Proposed Linear](image2.png)
Simulation Results

- Subjective quality test

Cubic Convolution  
Proposed Linear
Conclusions

• A Pixel-based adaptive linear interpolation has been presented
• A generic system and its low complexity version have been proposed
• Simulation results show that the proposed method
  – Give better visual quality
  – Give better objective quality in terms of PSNR
  – than previous methods such as conventional linear, cubic convolution, warped distance, and shifted linear interpolation