A Sub-pixel Image Magnification Using Adaptive Linear Interpolation

DSU Dongseo University Hoon Yoo

Presentation Outline

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

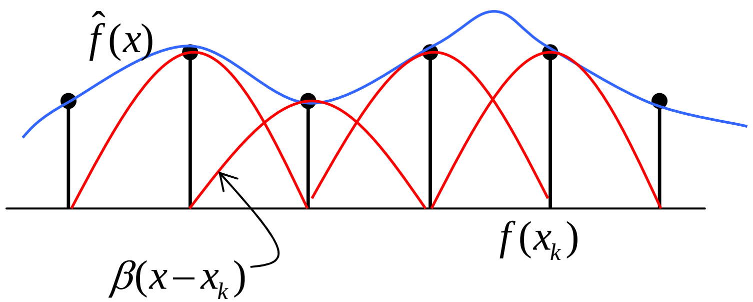
Introduction

- 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

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

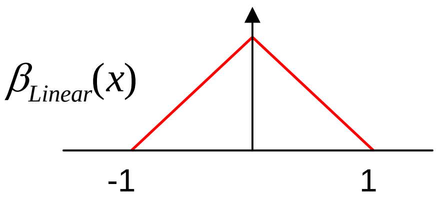


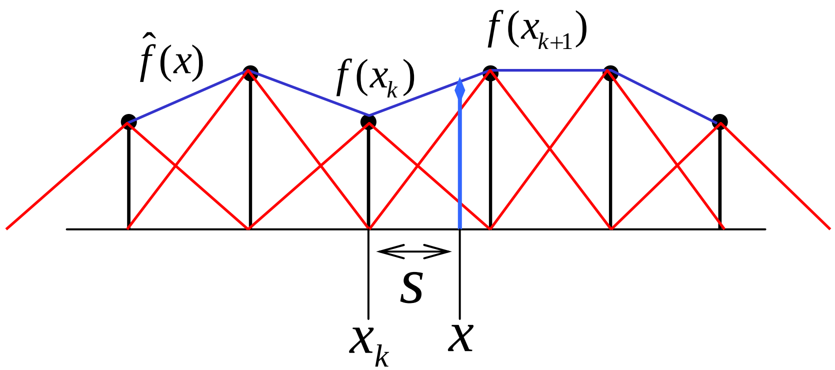
Interpolation Kernel

Conventional Interpolation

- Linear

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





$$s = x - x_k,\quad 1-s = x_{k+1} - x$$

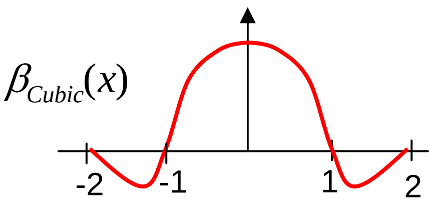
$$\text{where}\quad x_k \le x \le x_{k+1},\ 0 < s \le 1$$

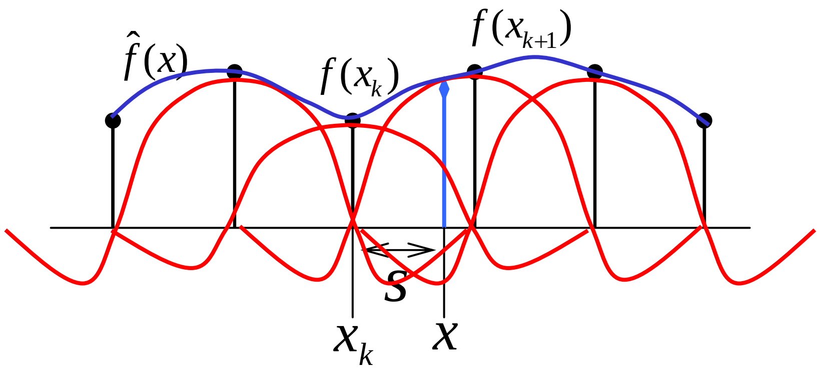
$$\hat{f}(x) = \sum_k f(x_k)\beta(x-x_k) = (1-s)f(x_k) + sf(x_{k+1})$$

Conventional Interpolation

- Keys' Cubic Convolution Interpolation

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





$$\begin{aligned} \hat{f}(x) = {} & f(x_{k-1})(\alpha s^3 - 2\alpha s^2 + \alpha s) \\ & + f(x_{k+0})((\alpha+2)s^3 - (\alpha+3)s^2 + 1) \\ & + f(x_{k+1})(-(\alpha+2)s^3 + (2\alpha+3)s^2 - \alpha s) \\ & + f(x_{k+2})(-\alpha s^3 + \alpha s^2) \end{aligned}$$

- Keys' Cubic Convolution Interpolation

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

$\downarrow$ $\alpha = -\frac{1}{2}$

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

$\downarrow$ $s = \frac{1}{2}$

$$\hat{f}(x) = \tfrac{-1}{16} f(x_{k-1}) + \tfrac{9}{16} f(x_{k+0}) + \tfrac{9}{16} f(x_{k+1}) + \tfrac{-1}{16} f(x_{k+2})$$

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

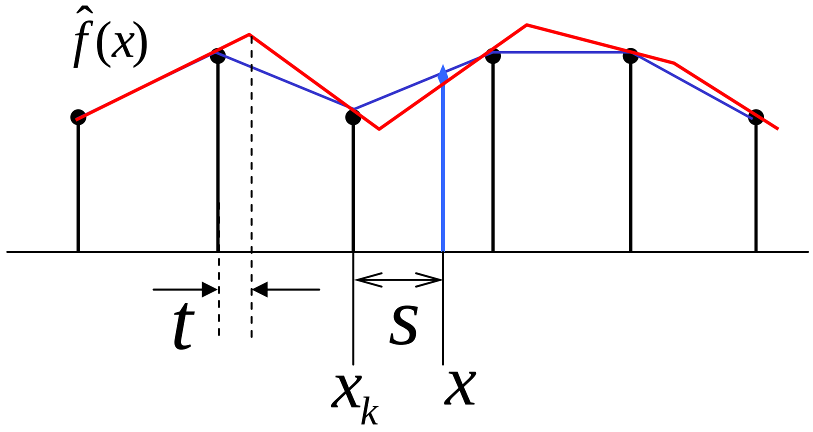
$$s' = s - kAs(s-1)$$

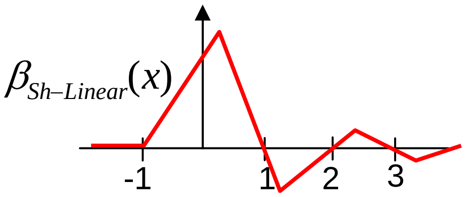
$$A = \frac{|f(x_{k+1}) - f(x_{k-1})| - |f(x_{k+2}) - f(x_k)|}{L-1}$$

- Note that
  - The variable A is a pixel-based parameter
  - The variable k is an image-based parameter (k = 8 fixed for the Lena image)

Previous Adaptive Linear Interpolation

- Shifted Linear Interpolation





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

Proposed Method

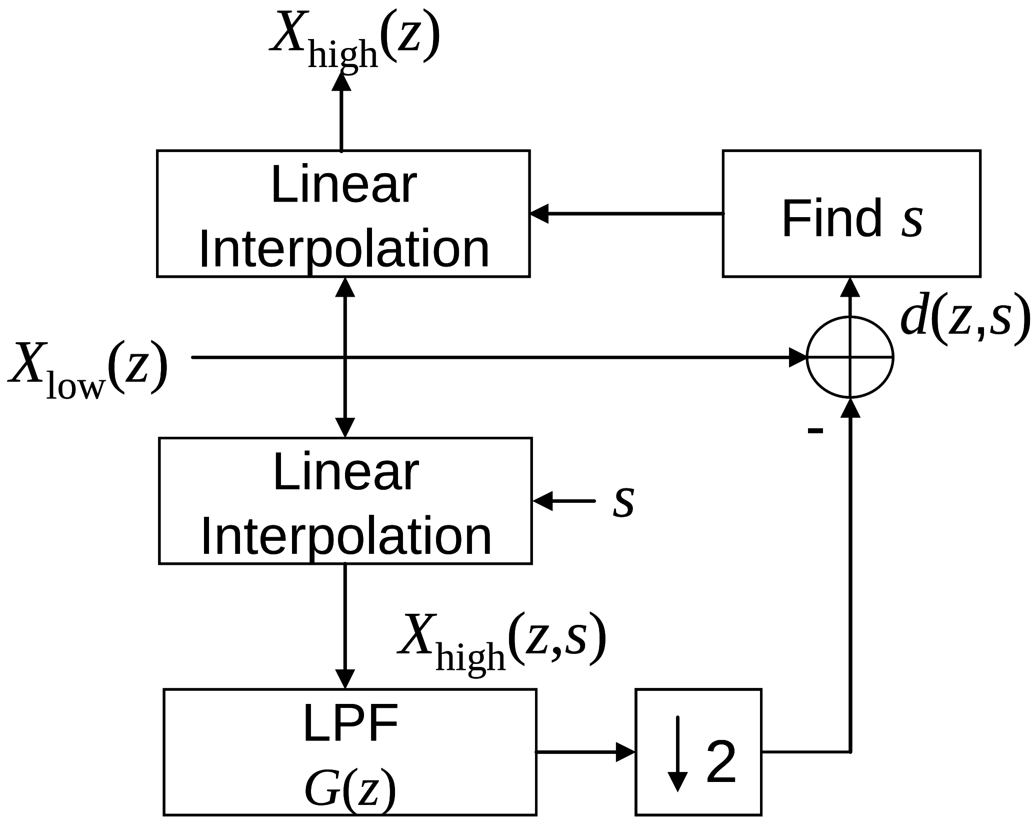
- Warping distance s, based on MMSE (minimization mean square error)

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

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

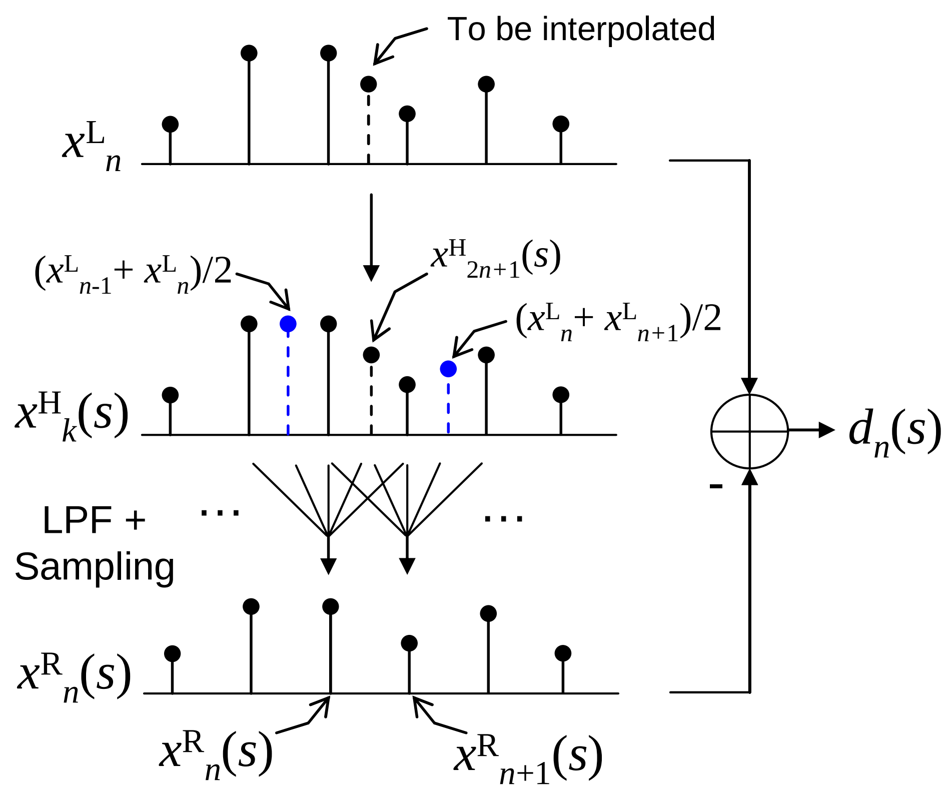
Proposed Method

- Generic diagram



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

$$x^{\mathrm{R}}{}_{n}(s) = \sum_{m=-(M-1)/2}^{(M-1)/2} g_{m+(M-1)/2} x^{\mathrm{H}}{}_{2n+m}(s)$$

- Apply 3-tap low pass filter {1,2,1}/4
- Define cost function as follows

$$C(s) = \frac{(x^{\mathrm{L}}{}_{n} - x^{\mathrm{R}}{}_{n}(s))^2 + (x^{\mathrm{L}}{}_{n+1} - x^{\mathrm{R}}{}_{n+1}(s))^2}{2}$$

- We have

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

$$c_0 = \frac{x^{\mathrm{L}}{}_{n+1} - x^{\mathrm{L}}{}_{n}}{4},\ c_1 = \frac{x^{\mathrm{L}}{}_{n-1} - x^{\mathrm{L}}{}_{n}}{8},\ c_2 = \frac{2x^{\mathrm{L}}{}_{n} - 3x^{\mathrm{L}}{}_{n+1} + x^{\mathrm{L}}{}_{n+2}}{8} = -c_0 + \frac{x^{\mathrm{L}}{}_{n+2} - x^{\mathrm{L}}{}_{n+1}}{8}$$

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

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

$$c_0 = \frac{x^{\mathrm{L}}{}_{n+1} - x^{\mathrm{L}}{}_{n}}{4},\ c_1 = \frac{x^{\mathrm{L}}{}_{n-1} - x^{\mathrm{L}}{}_{n}}{8},\ c_2 = -c_0 + \frac{x^{\mathrm{L}}{}_{n+2} - x^{\mathrm{L}}{}_{n+1}}{8}$$

Simulation Results

- Objective quality test (PSNR)

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

Simulation Results

- Subjective quality test

Conventional Linear

Proposed Linear

Simulation Results

- Subjective quality test

WaDi Linear

Proposed Linear

Simulation Results

- Subjective quality test

Shifted Linear

Proposed Linear

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