Enlargement or Reduction of Digital Images with Minimum Loss of Information

Michael Unser, Akram Aldroubi, and Murray Eden

IEEE Transactions on Image Processing. Vol 4. No 3. March 1995.

Objetive

- "Derive optimal spline algorithms for the enlargement or reduction of digital images by arbitrary (non integer) scaling factors" Δ.
- Optimality is managed by the authors in a least square sense.

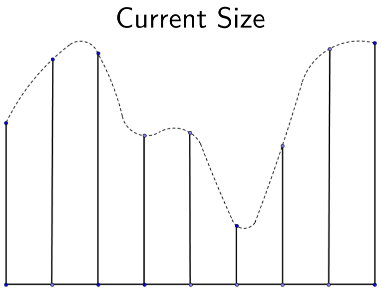
Classical Approach: Inteporlation Reconstruction + Resampling

- "Standard approaches fit the original data with a continuous model (image interpolation) and then resample two dimensional function in a new sample grid".

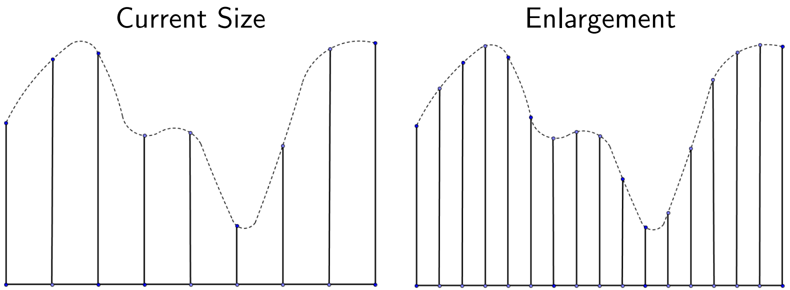
$$[f_1]_1 \xrightarrow{\text{Int. Rec.}} f_1 = [f_1]_1 * \varphi \xrightarrow{\text{Resamp. freq } \Delta} [f_1]_\Delta = [[f_1] * \varphi]_\Delta$$

- "Simple to implement but they tend to produce suboptimal results because they are not designed to minimize loss information".

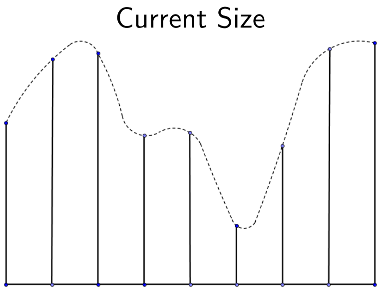
Inteporlation Reconstruction + Resampling

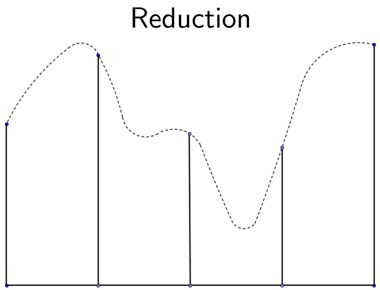


Inteporlation Reconstruction + Resampling



Inteporlation Reconstruction + Resampling





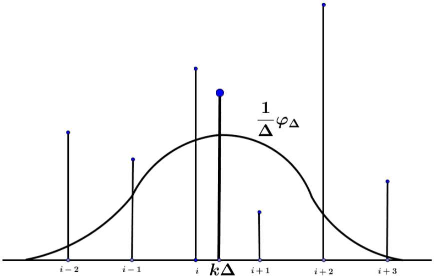
Perceptual Results of Inteporlation Reconstruction + Resampling

- "In the case of reduction, the situation is analogous to sampling a signal that has not previously bandlimited, a process that may induce aliasing errors".
- Results in the case of magnification have some distortions but they "tend to disappear when higher order of splines are applied".

Other common approach for Reduction (M2)

Results of reducing an image using **IR+R** are poor since they do not suppress high frequencies. A traditional approach for reduction that performs better is the following:

$$f_\Delta(k\Delta) = \frac{1}{\Delta} \sum_{i \in \mathbb{Z}} f_1(i) \varphi_\Delta(k\Delta - i)$$



where $\varphi_\Delta = \varphi(\bullet/\Delta)$.

Other common approach for Reduction (M2)

From the previous expression we get:

$$
\begin{aligned}
f_\Delta(k\Delta) &= \frac{1}{\Delta}\sum_{i\in\mathbb{Z}} f_1(i)\varphi_\Delta(k\Delta - i) \\
&= \frac{1}{\Delta}[[f_1] * \varphi_\Delta]_\Delta \\
&= [[f_1] * (\frac{1}{\Delta}\varphi_\Delta)]_\Delta
\end{aligned}
$$

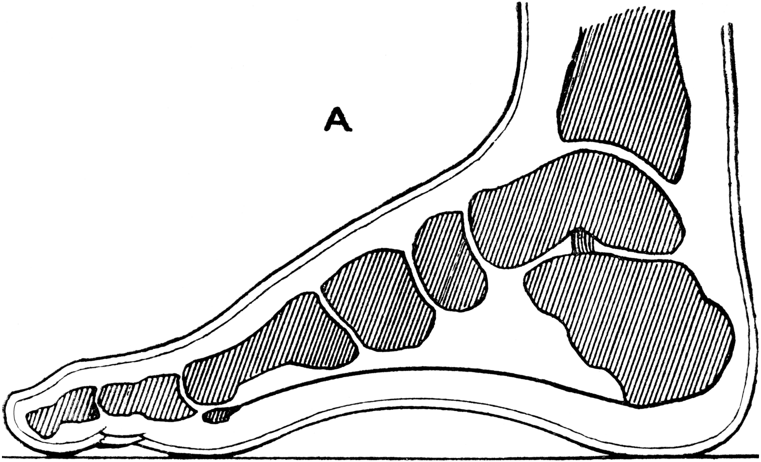
This is quite similar to **IR+R**:

$$[[f_1] * \varphi]_\Delta$$

Instead of using φ for reconstruction, here we use $\frac{1}{\Delta}\varphi_\Delta$, a "low pass version" of it.

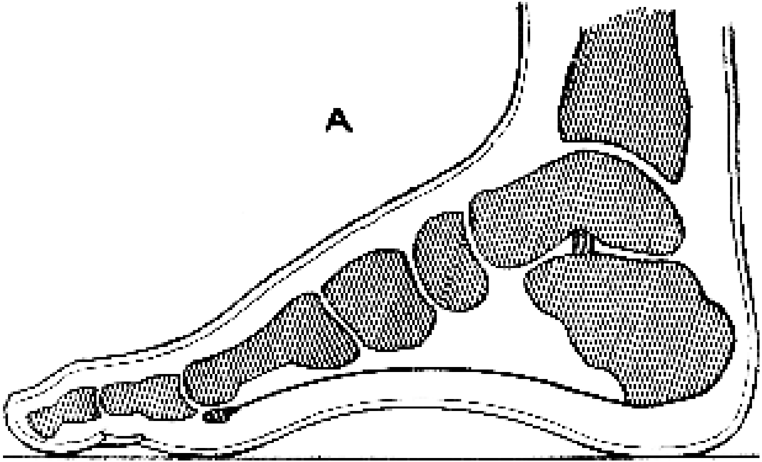
Comparision: IR+R ($[[f_1] * \varphi]_\Delta$) vs M2 ($[[f_1] * (\frac{1}{\Delta}\varphi_\Delta)]_\Delta$)

φ=Box, $\Delta = 10$.



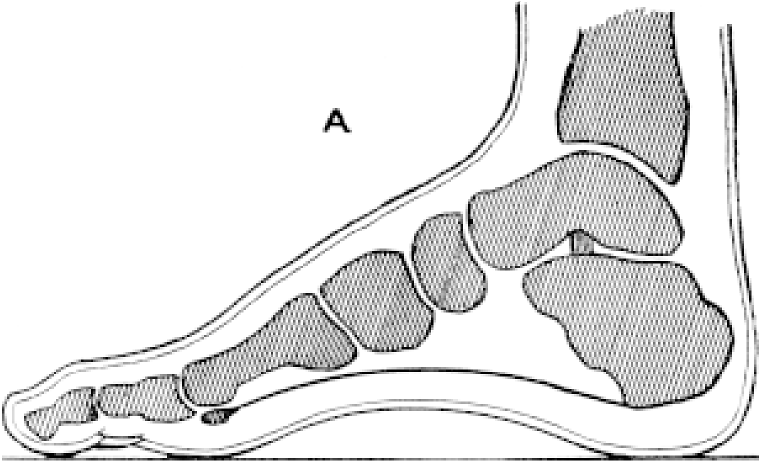
Comparision: IR+R ($[[f_1] * \varphi]_\Delta$) vs M2 ($[[f_1] * (\frac{1}{\Delta}\varphi_\Delta)]_\Delta$)

φ=Box, $\Delta = 10$.



Comparision: IR+R ($[[f_1] * \varphi]_\Delta$) vs M2 ($[[f_1] * (\frac{1}{\Delta}\varphi_\Delta)]_\Delta$)

φ=Box, $\Delta = 10$.



Comparision: IR+R ($[[f_1] * \varphi]_\Delta$) vs M2 ($[[f_1] * (\frac{1}{\Delta}\varphi_\Delta)]_\Delta$)

φ=Box, $\Delta = 10$.

Comparision: IR+R ($[[f_1] * \varphi]_\Delta$) vs M2 ($[[f_1] * (\frac{1}{\Delta}\varphi_\Delta)]_\Delta$)

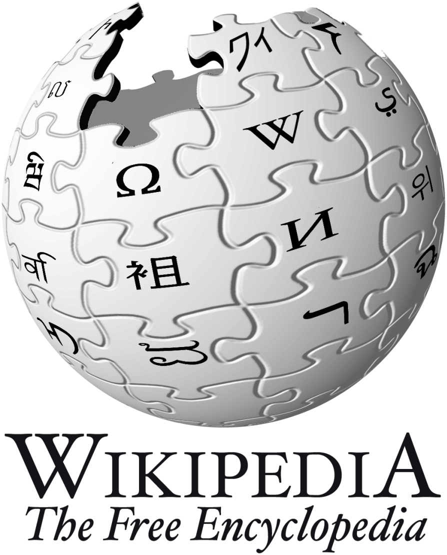
φ=Box, $\Delta = 10$.

Comparision: IR+R ($[[f_1] * \varphi]_\Delta$) vs M2 ($[[f_1] * (\frac{1}{\Delta}\varphi_\Delta)]_\Delta$)

φ=Box, $\Delta = 10$.

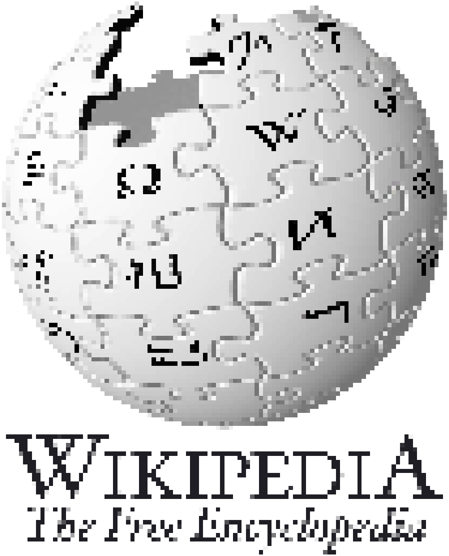
Comparision: IR+R ($[[f_1] * \varphi]_\Delta$) vs M2 ($[[f_1] * (\frac{1}{\Delta}\varphi_\Delta)]_\Delta$)

φ=Box, $\Delta = 10$.



Comparision: IR+R ($[[f_1] * \varphi]_\Delta$) vs M2 ($[[f_1] * (\frac{1}{\Delta}\varphi_\Delta)]_\Delta$)

φ=Box, $\Delta = 10$.



Comparision: IR+R ($[[f_1] * \varphi]_\Delta$) vs M2 ($[[f_1] * (\frac{1}{\Delta}\varphi_\Delta)]_\Delta$)

φ=Box, $\Delta = 10$.

Comparision: IR+R ($[[f_1] * \varphi]_\Delta$) vs M2 ($[[f_1] * (\frac{1}{\Delta}\varphi_\Delta)]_\Delta$)

φ=Box, $\Delta = 10$.

Comparision: IR+R ($[[f_1] * \varphi]_\Delta$) vs M2 ($[[f_1] * (\frac{1}{\Delta}\varphi_\Delta)]_\Delta$)

φ=Box, $\Delta = 10$.

Comparision: IR+R ($[[f_1] * \varphi]_\Delta$) vs M2 ($[[f_1] * (\frac{1}{\Delta}\varphi_\Delta)]_\Delta$)

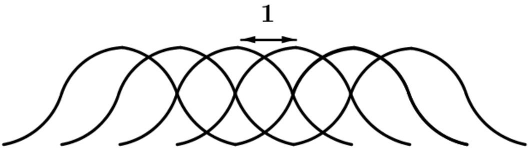
φ=Box, $\Delta = 10$.

Least Square Approach. Notation

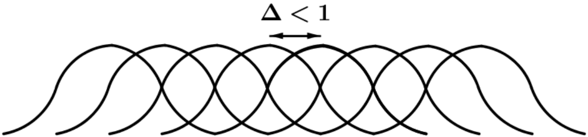
- β^n: nth order β-spline.
- $V_{\beta^n} = \{c * \beta^n : c \in l_2\}$: Subspace of representable signals for the reconstruction kernel β^n.
- $\beta^n_\Delta = \beta^n(\bullet/\Delta)$: Kernel scaled to step size Δ.
- $V_{\beta^n_\Delta} = \{c *_\Delta \beta^n_\Delta : c \in l_2\}$: Subspace of representable signals for the reconstruction kernel β^n_Δ.

Least Square Approach. Step-Size

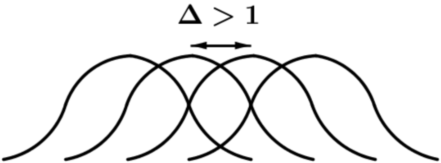
Current Size



Enlargement

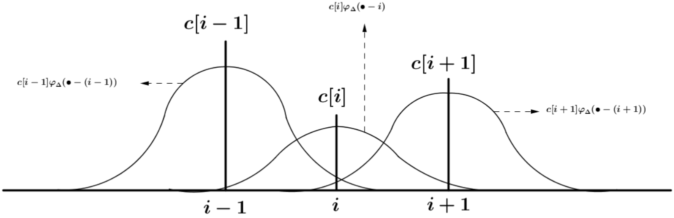


Reduction



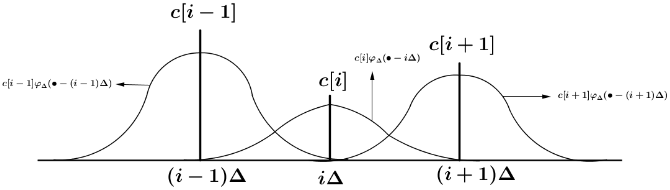
Clarification in notation!!

$$c * \varphi_\Delta \notin V_{\varphi_\Delta}$$



Wrong:

$$c *_\Delta \varphi_\Delta \in V_{\varphi_\Delta}$$



Right:

Least Square Approach. Problem

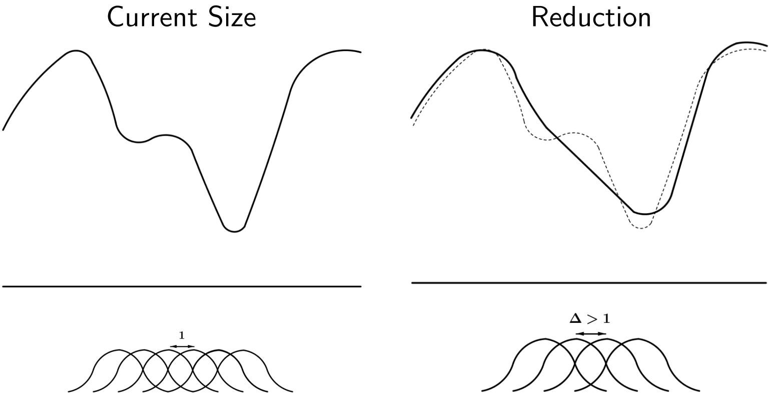
Definitions:

- $f_1^n = c_1 * \beta^n$: Interp. reconstruction for step size 1.
- $f_\Delta^n = c_\Delta *_\Delta \beta_\Delta^n$: Reconstruction for step size Δ.

Statement:

- $[f_1^n]_1 \to$ Given.
- $[f_\Delta^n]_\Delta \to$ **To Find!**.
- $f_\Delta^n \in V_{\beta_\Delta^n}$ is the minimum error approximation of $f_1^n \in V_{\beta^n}$.
- $\Rightarrow f_\Delta^n = P_{V_{\beta_\Delta^n}}(f_1^n)$.

Least Square Approach. Problem



Least Square Approach. Bi-orthogonal Basis

Let $\{x_i\}_{i \in I}$ and $\{y_i\}_{i \in I}$ be a Bi-orthogonal bases for the space V, this is:

$$< x_i, y_j >= \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

Then any $v \in V$ can be expressed as:

$$v = \sum_{i \in I} < v, y_i > x_i$$

Observe one basis is used for projection an the other for reconstruction.

Least Square Approach. Constructing Bi-orthogonal pairs

Given the basis $B_\varphi : \{\varphi(\bullet - k)\}_{k\in\mathbb{Z}}$, let's find it's bi-orthogonal pair in V_φ. Let $B_{p*\varphi} : \{(p * \varphi)(\bullet - k)\}_{k\in\mathbb{Z}}$.
We must satisfy,

$$< \varphi(\bullet - i), (p * \varphi)(\bullet - j) >= \delta(i, j)$$

Equivalently,

$$[\varphi * (p * \varphi)^\vee] = \delta$$

This implies,

$$p = [\varphi * \varphi^\vee]^{-1}$$

The function $\mathring{\varphi} := [\varphi * \varphi^\vee]^{-1} * \varphi$ is named the **dual**.
Then φ and $\mathring{\varphi}$ induces Bi-orthogonal basis in V_φ.

Least Square Approach. Multiples representation of the same space: Bases

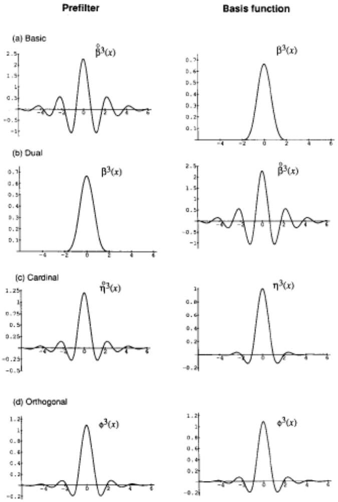


Fig. 2. Optimal prefilters and basis functions for four equivalent representation of cubic spline polynomial approximations.

Least Square Approach. Multiples representation of the same space: Change of Coordinates

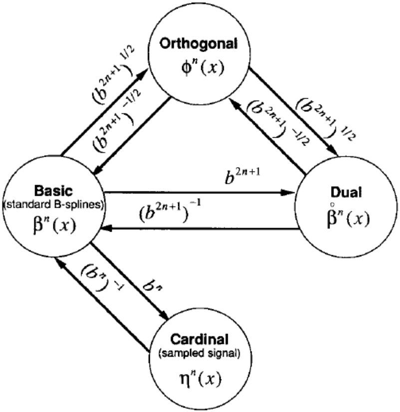


Fig. 1. Digital filters for the conversion between several equivalent polynomial spline representations of signals.

Least Square Approach. Orthogonal Projection

- **General Sampling Theorem**: The orthogonal projection of a function $f \in L_2$ on V_φ is given by:

$$f_{V_\varphi} = [f * (\mathring{\varphi})^\vee] * \varphi$$

In the article this is implemented as follows:

$$f_\Delta^n = P_{V_{\beta_\Delta^n}}(f_1^n) = [f_1^n * (\beta_\Delta^n)^\vee]_\Delta *_\Delta \mathring{\beta}_\Delta^n$$

Remark: Here the projection is done using the dual as reconstruction basis. In Diego's article is the opposite.

Least Square Approach. Step 1: Calculating Interpolation Coefficients c_1

"Determine the β-spline coefficients of f_1^n that interpolates the digital signal $[f_1^n]_1$".

$$f_1^n = c_1 * \beta^n \Rightarrow c_1 = [f_1^n]_1 * [\beta^n]_1^{-1}$$

Remark: f_1^n can be expressed in terms of the cardinal spline β_{int}^n of order n as follows:

$$f_1^n = [[f_1^n]_1 * [\beta^n]_1^{-1}]_1 * \beta^n = [f_1^n]_1 * ([\beta^n]_1^{-1} * \beta^n) = [f_1^n]_1 * \beta_{int}^n$$

Least Square Approach. Step 2: Math Derivation of the Sampling Function ξ_Δ^n

$$
\begin{aligned}
f_\Delta^n &= [f_1^n * (\beta_\Delta^n)^\vee]_\Delta *_\Delta \mathring{\beta}_\Delta^n \\
&= [(c_1 * \beta^n) * (\beta_\Delta^n)^\vee]_\Delta *_\Delta \mathring{\beta}_\Delta^n \\
&= \Delta[c_1 * \left(\frac{1}{\Delta}\beta^n * (\beta_\Delta^n)^\vee\right)]_\Delta *_\Delta \mathring{\beta}_\Delta^n \\
&= \Delta[c_1 * \xi_\Delta^n]_\Delta *_\Delta \mathring{\beta}_\Delta^n
\end{aligned}
$$

The function $\xi_\Delta^n = \frac{1}{\Delta}\beta^n * (\beta_\Delta^n)^\vee$ is called the Sampling Function.
It's important to notice that this function has compact support.
Remark: ξ_Δ^n corresponds to the cross correlation $\frac{1}{\Delta}a_{\beta^n,\beta_\Delta^n}$.

Least Square Approach. Take care using the notation!

One would be tempted to write

$$[c_1 * \xi_\Delta^n]_\Delta = c_1 * [\xi_\Delta^n]_\Delta,$$

but that's a mistake!!. Observe that

$$\begin{aligned}
[c_1 * \xi_\Delta^n]_\Delta &= \left\{(c_1 * \xi_\Delta^n)(k\Delta)\right\}_{k\in\mathbb{Z}} \\
&= \left\{\sum_{i=-\infty}^{\infty} c_1(i)\xi_\Delta^n(k\Delta - i)\right\}_{k\in\mathbb{Z}}
\end{aligned}$$

On the other hand,

$$\begin{aligned}
c_1 * [\xi_\Delta^n]_\Delta &= \left\{(c_1 * [\xi_\Delta^n]_\Delta)(k)\right\}_{k\in\mathbb{Z}} \\
&= \left\{\sum_{i=-\infty}^{\infty} c_1(i)\xi_\Delta^n(\Delta(k - i))\right\}_{k\in\mathbb{Z}}
\end{aligned}$$

Least Square Approach. Step 3: Post Filter q

Resample the signal at a step-size Δ corresponds to find the sequence $[f_\Delta^n]_\Delta$.
From the conditions:

$$f_\Delta^n = \Delta[c_1 * \xi_\Delta^n]_\Delta *_\Delta \mathring{\beta}_\Delta^n$$
$$f_\Delta^n = [f_\Delta^n]_\Delta *_\Delta (\beta_\Delta^n)_{int}$$

We get,

$$\begin{aligned}[f_\Delta^n]_\Delta &= \Delta[c_1 * \xi_\Delta^n]_\Delta * [\beta_\Delta^n * (\beta_\Delta^n)^\vee]_\Delta^{-1} * [\beta_\Delta^n]_\Delta \\ &= \Delta[c_1 * \xi_\Delta^n]_\Delta * (\Delta[\beta^{2n+1}])^{-1} * [\beta^n] \\ &= [c_1 * \xi_\Delta^n]_\Delta * q\end{aligned}$$

Remark: The postfilter $q = [\beta^{2n+1}]^{-1} * [\beta^n]$ converts from dual to cardinal spline representation.

Least Square Approach. Diagram Summary

Least Square Approach. Diego's article generalization

From the condition $f^n_\Delta = \Delta[c_1 * \xi^n_\Delta]_\Delta *_\Delta \mathring{\beta}^n_\Delta$ we get,

$$
\begin{aligned}
f^n_\Delta &= \Delta[c_1 * \xi^n_\Delta]_\Delta * [a_{\beta^n_\Delta}]^{-1}_\Delta *_\Delta \beta^n_\Delta \\
&= [c_1 * \xi^n_\Delta]_\Delta * [a_{\beta^n}]^{-1} *_\Delta \beta^n_\Delta
\end{aligned}
$$

then,

$$
\begin{aligned}
c_\Delta &= [c_1 * \xi^n_\Delta]_\Delta * [a_{\beta^n}]^{-1} \\
&= \frac{1}{\Delta}[c_1 * a_{\beta^n,\beta^n_\Delta}]_\Delta * [a_{\beta^n}]^{-1}
\end{aligned}
$$

This last expression corresponds to (55) of Diego's article:

$$
c_s = \frac{1}{s}[a_\varphi]^{-1} * [c * a_{\varphi,\varphi_s}]_s
$$

for the particular case $\varphi = \beta^n$

Evaluation of the Sampling Kernels. Degree 0

$$\xi_\Delta^0(x) = \begin{cases} b, & 0 \le |x| < a_1 \\ b - \frac{b(|x|-a_1)}{a_2-a_1}, & a_1 \le |x| < a_2 \\ 0, & a_2 \le |x| \end{cases}$$

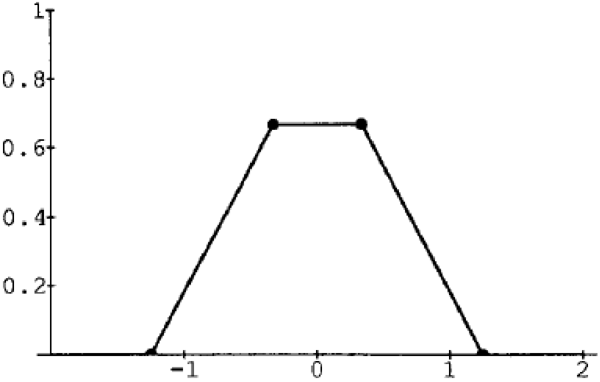


Fig. 4. Example of trapezoidal sampling function for a zero order spline model ($\Delta = 3/2$).

Evaluation of the Sampling Kernels. Degree 1

$$\xi^1_\Delta(x) = \begin{cases} b_{i0} + b_{i1}|x| + b_{i2}x^2 + b_{i3}|x|^3, & |x| \in [a_{i-1}, a_i) \\ 0, \text{otherwise} \end{cases}$$

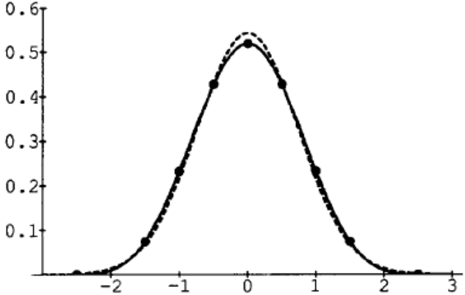


Fig. 5. Example of a modified sampling function (solid line) for a first order spline model ($\Delta = 3/2$). This function is a cubic spline with knot points at the positions marked by the small circles. The Gaussian approximation given by (34) is superimposed with a dashed line (relative mean square error = 0.145%).

Evaluation of the Sampling Kernels. General Case

- "As *n* increases this function converges to a Gaussian as a consequence of the Central Limit Theorem".
- "We use the fact that the global variance of a convolution is equal to the sum of the variance of its individual components".

$$\xi_{\Delta}^{n}(x) \cong \begin{cases} \frac{1}{\sqrt{2\pi}\sigma_n} \exp\left\{\frac{-x^2}{2\sigma_n^2}\right\}, & |x| < \frac{n+1}{2}(1+\Delta) \\ 0, & \text{otherwise} \end{cases}$$

with standard deviation

$$\sigma_n = \sqrt{\frac{n+1}{12}(1+\Delta^2)}.$$

Article's Results. Box: Resampling vs Least Squares

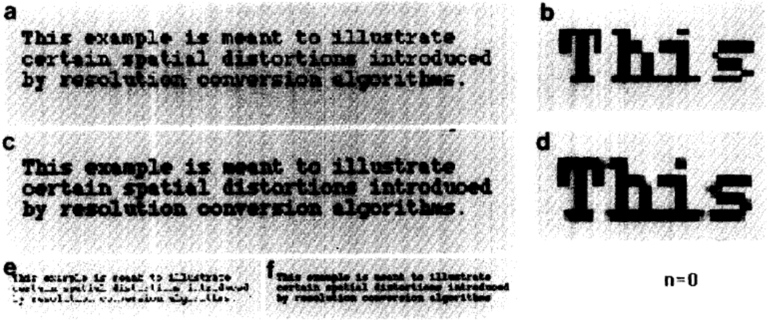


Fig. 6. Examples of image magnification and reduction using a zeroth-order model: (a) ISC0 with $\Delta = 1/\sqrt{2}$; (b) enlarged detail of (a); (c) LSSC0 with $\Delta = 1/\sqrt{2}$; (d) enlarged detail of (c); (e) ISC0 with $\Delta = \sqrt{2}$; (f) LSSC0 with $\Delta = \sqrt{2}$.

Article's Results. Hat: Resampling vs Least Squares

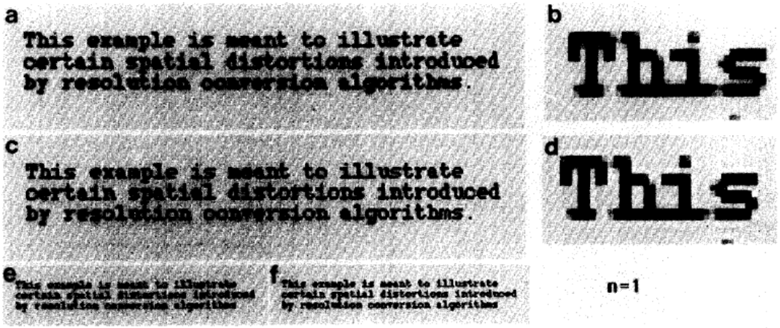


Fig. 7. Examples of image magnification and reduction using a first-order model: (a) ISC1 with $\Delta = 1/\sqrt{2}$; (b) enlarged detail of (a); (c) LSSC1 with $\Delta = 1/\sqrt{2}$; (d) enlarged detail of (c); (e) ISC1 with $\Delta = \sqrt{2}$; (f) LSSC1 with $\Delta = \sqrt{2}$.

Article's Results. Cubic Spline: Resampling vs Least Squares

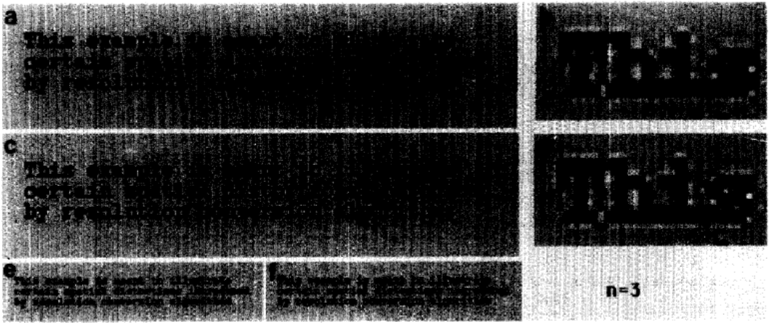


Fig. 8. Examples of image magnification and reduction using a cubic spline model: (a) ISC3 with $\Delta = 1/\sqrt{2}$; (b) LSSC3 with $\Delta = 1/\sqrt{2}$; (c) ISC3 with $\Delta = \sqrt{2}$; (d) LSSC3 with $\Delta = \sqrt{2}$.

Comments from the authors

- "Our experimental results demonstrate the superiority of least square scale conversion (LSSC) over interpolative scale conversion in a consistent fashion. This observation is specially true for image reduction".
- "LSSC1 appears to yield images with better visual quality, probably because the oscillation near the borders of the objects are less pronounced than they are for cubic splines".

Summary of methods discussed

- **IR+R** $\rightarrow [[f_1] * \varphi]_\Delta$
- **M2** $\rightarrow [[f_1] * (\frac{1}{\Delta}\varphi_\Delta)]_\Delta$
- **LS** $\rightarrow [[([f_1] * \varphi) * \mathring{\varphi}_\Delta]_\Delta *_\Delta \varphi_\Delta]_\Delta$

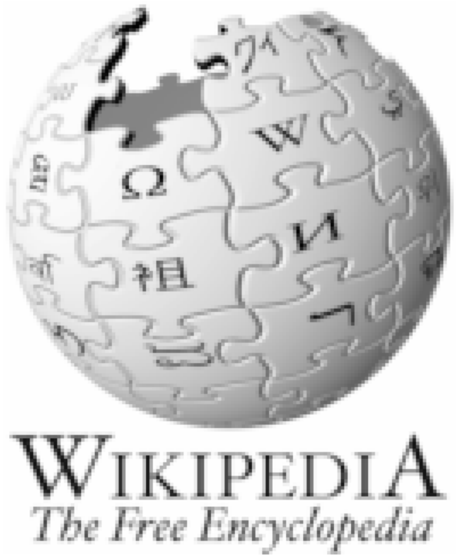
Comparision: M2 vs LS

Hat, $\Delta = 8$

WIKIPEDIA
The Free Encyclopedia

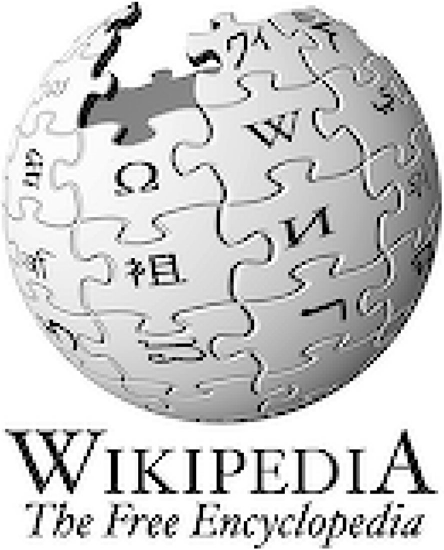
Comparision: M2 vs LS

Hat, $\Delta = 8$



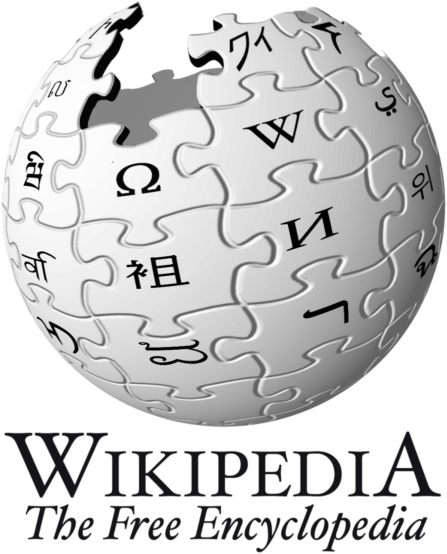
Comparision: M2 vs LS

Hat, $\Delta = 8$



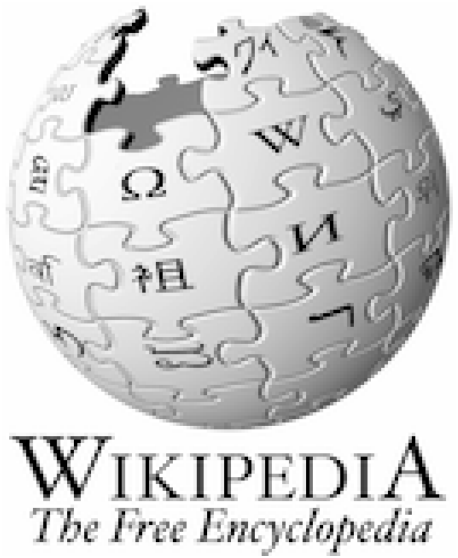
Comparision: M2 vs LS

Bspline3i, $\Delta = 8$



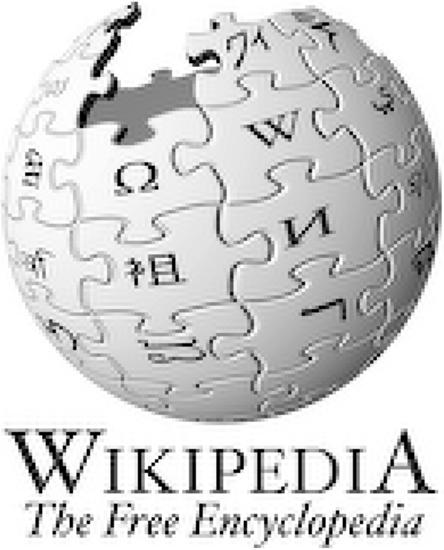
Comparision: M2 vs LS

Bspline3i, $\Delta = 8$



Comparision: M2 vs LS

Bspline3i, $\Delta = 8$



Comparision: M2 vs LS

Hat, $\Delta = 8$

Comparision: M2 vs LS

Hat, $\Delta = 8$

Comparision: M2 vs LS

Hat, $\Delta = 8$

Comparision: M2 vs LS

Bspline3i, $\Delta = 8$

Comparision: M2 vs LS

Bspline3i, $\Delta = 8$

Comparision: M2 vs LS

Bspline3i, $\Delta = 8$

Obrigado!