Least Squares Image Resizing

Atanas Gotchev[1], Karen Egizarian[2], and Tapio Saramaki[3]

Abstract – **- This invited paper considers the problem of reducing the size of a digital image as a problem of constructing a proper polynomial spline approximation to a function $s(t) \in L_2(R)$. From this point of view, the crucial problem is to design an appropriate reconstruction (basis) function. Then, the analysis (projecting) function can be formed as biorthogonal to the reconstruction one. The basis functions are chosen among the class of the symmetrical and compactly supported modified B-splines having both good approximation properties and efficient realization structures. We review the theory of orthogonal projections both in continuous (minimizing the L2 norm) and discrete domain (minimizing the l_2 norm) and propose a constructive compromise yielding an efficient decimation structure possessing good anti-aliasing properties. It is shown, by means of examples, that with a considerably lower computational complexity the proposed structure provides practically the same quality for the restored images as the best existing structures.**

Keywords – **decimation, orthogonal projection, B-splines**

I. Introduction

Many image processing applications demand an effective resizing algorithm. These applications include, among others, digital zooming effects, equalizing the resolution for different imaging and printing devises, and building multiresolution pyramids. The problem of image interpolation (reconstruction) can be accomplished by first fitting an appropriate continuous model to the discrete data and then resampling the resulting image on a *finer* grid [1,2]. The problem of image size reduction is rather different, since the decimation is vulnerable to aliasing effects if one samples the same continuous model at a *sparser* grid. The classical digital signal processing theory dictates that a preliminary low-pass (*anti-aliasing*) filtering is needed. The ideal low-pass filtering is not a practically realizable option because it corresponds to the use of the sinc impulse response that is infinitely supported. Hence, the investigation efforts have been concentrated in searching for solutions based on decimation (pre-filtering) functions that are compactly supported and have good anti-aliasing proper ties. Some designs using polynomial or B-spline combinations have seemed to be perspective candidates [1,2]. These solutions have been originally designed to be good interpolators (with good anti-imaging properties). The research question is how to adapt them to the decimation problem. More formally, by considering those kernels as basis functions generating polynomial spline subspaces, the problem of down-sampling can be transferred to the problem of designing a new signal approximation with the minimum loss of information, thereby naturally leading to a certain form of *least squares* (LS).

For B-spline bases, algorithms for rational [3] and for arbitrary [4] scale conversion have been proposed, aimed at minimizing the continuous L_2 error norm. In order to have an input *continuous* function at hands, those algorithms advise a preliminary spline model fitting, as in the interpolation case, and a subsequent continuous-time processing.

Another standpoint insists on minimizing the discrete l_2 norm, based on the fact that the images are discrete and the quality variations are assessed by means of the signal-to noise ratio (SNR) that is a discrete l_2 measure. The drawback of using the classical discrete LS is that there is a demand to compute a *pseudo-inverse* matrix [9]. This computation is not preferable even for the banded matrices that express compactly supported bases.

In this paper, a new way is proposed for treating the L_2 and l_2 solutions for the image decimation problem and it is shown how they can be merged into a high-quality and cost-efficient hybrid scheme.

II. Signal Decimation Problem

By assuming a separable basis functions, we downgrade the image size reduction problem to the 1-D signal decimation problem.

Consider an initial discrete signal $x(k)$ for $k = 0, 1, ..., L_{in} - 1$ as taken from a continuous function $s(t)$ sampled over a uniform grid in the interval $[a, b]$: $\boldsymbol{\tau} = [\tau_0, ..., \tau_{L_{in}-1}]$, $\tau_{k+1} - \tau_k = h_{in}$, $\tau_0 = a$, $\tau_{L_{in}-1} = b$ that is, $x(k) = s(\tau_k)$. This signal is desired to be decimated into a signal $y(l)$ for $l = 0, 1, ..., L_{out} - 1$ over a new uniform grid, determined by a larger step h_{out} with $h_{out} > h_{in}$. Without loss of generality it is assumed that $h_{out} = 1$, $a = 0$, and $b = L_{out} - 1$. Hence, the output grid will be placed on the integer coordinates. The quantity $h \equiv h_{in} = (L_{out} - 1)/(L_{in} - 1) < 1$ indicates the decimation ratio. Fig. 1 illustrates the process.

The decimated signal should preserve the original signal features as well as possible, i.e., having a proper reconstruction function one should be able to get a good approximation to the original signal from the decimated one. Hence, we seek for a solution in a form of a projection onto some linear space generated by this well chosen reconstruction basis. In Section III, we briefly review the problem of construct-

[1]Atanas Gotchev, Member of IEEE is with the Institute of Signal Processing, Tampere University of Technology, P.O.Box 553, FIN-33101 Tampere, Finland E-mail: agotchev@cs.tut.fi

[2]Karen Egiazarian, Senior Member of IEEE is with the Institute of Signal Processing, Tampere University of Technology, P.O.Box 553, FIN-33101 Tampere, Finland E-mail: karen@cs.tut.fi

[3]Tapio Saramaki, Fellow of IEEE is with the Institute of Signal Processing, Tampere University of Technology, P.O.Box 553, FIN-33101 Tampere, Finland E-mail: ts@cs.tut.fi

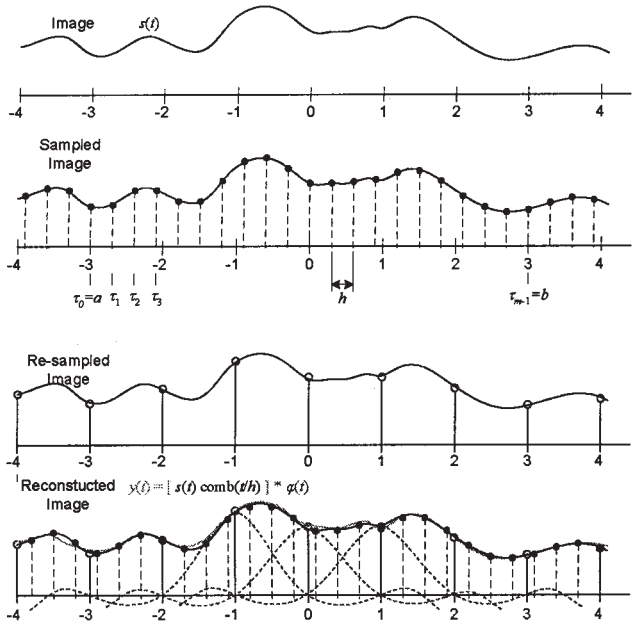

Fig. 1. Image decimation problem in 1-D

ing different polynomial spline approximations to a function $s(t) \in L_2(R)$ being parameterized by its scale. Then we return back to the problem of signal decimation and comment how it can be solved using the proposed formalism.

A. An interpolative solution

Before reviewing the LS approaches, we present a solution that has been considered as good candidate for continuous (at an arbitrary sparser grid) signal decimation [5,11]. Consider a continuous model fitting for the discrete data given at the coordinate grid $\boldsymbol{\tau}$. To preserve the values of the given discrete sequence we write the model as:

$$x_h(t) = \sum d_h(i)\varphi(t/h - i) \tag{1}$$

where the modeling coefficients are obtained by the recursive pre-filtering with the all-pole filter formed by the basis function sampled at the integers as follows:

$$d_h(k) = \sum_i x(i)(q)^{-1}(k-i) \quad \text{where} \quad q(k) = \varphi(k). \tag{2}$$

Equivalently,

$$h_h(t) = \sum_{i=1}^{m} x_h(i)\varphi_{\text{int}}(t/h - 1), \tag{3}$$

where

$$\varphi_{\text{int}}(t) = \sum_{i=1}^{m} (q)^{-1}(i)\varphi(t-i). \tag{4}$$

Eq. (3) can be interpreted as continuous domain filtering of the impulse train $\sum x(i)\delta(t/h - i)$ by the function $\varphi_{\text{int}}(t)$, rescaled to have its nodes over the grid $\boldsymbol{\tau}$. Its Fourier transform $\Phi_{\text{int}}(\omega) = \Phi(\omega)/\Phi(e^{j\omega})$ has zeros clustered around multiples of $F_h = 1/h$. If we now resample $x_h(t)$ at the integers we will encounter aliasing effects due to the fact that $F_h > F_1 = 1$. In the contrary, we can take the continuous filter as function specified at the integers (non-rescaled), with zeros around $F_1 = 1$, thereby generating good anti-aliasing properties. This is equivalent to changing the order of operation: first the continuous filtering and resampling is performed followed by the recursive digital filtering. Writing the continuous filtering and resampling as

$$x_h(k) = \sum_{i=0}^{L_{in}-1} x(i)\varphi(k - hi) \tag{5}$$

the subsequent digital filtering leads to

$$y(l) = \sum_k q^{-1}(k) \sum_{i=0}^{L_{in}-1} x(i)\varphi(l - k - hi). \tag{6}$$

While this solution has shown satisfactory performance for certain class of signals [5], it is not optimal in LS sense. This fact has some effect when resizing images.

III. Orthogonal Projection Paradigm

A. Function spaces and generating bases

Consider the following *shift-invariant* function space $V(\varphi)$ that is a closed subspace of L_2 and generated by a function φ as:

$$V(\varphi) = \left\{ q(t) = \sum_{l=-\infty}^{\infty} c(i)\varphi(t-i) \quad c \in l^2 \right\}. \tag{7}$$

Any function $s(t) \in L_2$ can be orthogonally projected into $V(\varphi)$ by finding the corresponding discrete-time sequence $c(k)$. It gives the LS approximation with respect to the L_2 norm defined by

$$||s||_{L_2}^2 = \langle s, s \rangle = \int s(\xi)s(\xi)d\xi. \tag{8}$$

The resulting approximation is given by

$$\tilde{s}(t) = \sum_{i=-\infty}^{\infty} c(i)\varphi(t-i). \tag{9}$$

If the basis $\phi(t-i)$ is not orthogonal, but only linearly independent, then the orthogonal projection of $s(t) \in L_2$ onto V is obtained by

$$c(i) = \langle s(\xi)\hat{\varphi}(\xi - i) \rangle = \int s(\xi)\hat{\varphi}(\xi - i)d\xi \tag{10}$$

Here, $\hat{\varphi}(t)$ is the dual (bi-orthogonal) basis of $\varphi(t)$ satisfying

$$\hat{\varphi}(t) \in V(\varphi) \tag{11}$$

and

$$\hat{\varphi}(t) = \sum_{i=-\infty}^{\infty} (p)^{-1}(i)\varphi(t-i) \tag{12}$$

where

$$p(i) = \int \varphi(\xi)\varphi(\xi - i)d\xi. \tag{13}$$

The sequence $p^{-1}(i)$ is the convolution inverse of the autocorrelation sequence $p(i)$ [6]. If φ is a symmetrical and compactly supported function over N, then $p(i)$ is a symmetrical sequence of length $2N+1$.

B. Spline-like basis functions

We choose our basis function φ to be a compactly supported piece-wise function constructed by B-splines:

$$\varphi(t) = \beta^{\text{mod}}(t) = \sum_{n=0}^{N} \sum_{m \in 0} \gamma_{nm} \beta^n (t-m) \qquad (14)$$

where $\beta^n(t)$ is the B-spline of degree n being symmetrical around the origin defined as

$$\beta^n(t) = \Delta^{n+1} * \frac{x_+^n}{n!} * \delta\Big(t + \frac{n+1}{2}\Big) \qquad (15)$$

where

$$t_+^n = \begin{cases} t^n & \text{if } t \geq 0 \\ 0 & \text{otherwise,} \end{cases} \qquad (16)$$

and where $\Delta = \delta(t) - \delta(t-1)$ denotes the backward *finite difference*, and $\delta(t)$ is the Dirac's mass distribution [4].

Examples of such functions, apart the B-splines themselves, are the *modified* B-splines and *moms* studied in [1,2]. They have shown their superiority as bases for image reconstruction (interpolation) both because of their good performance and low computational complexity. B-splines are compactly supported over the interval $[N/2, N/2)$ and are the most regular functions having the maximum order with the given support. While the combinations (14) are not so regular, because of the added low-degree terms, they can be optimized to have good asymptotic approximation properties [2] or good anti-imaging properties in frequency domain needed for the signal reconstruction [1]. Furthermore, the function given by Eq. (14) can be formed in such a way as to have the support of the highest degree B-spline attending the combination, thereby resulting in the same computational complexity as with classical B-splines of the same degree. Those, so-called *splines of minimal support* [7] can be presented also as consisting of polynomial pieces of degree N at every interval between integers for the interval of its support, that is $N+1$:

$$\varphi(t) = \sum_{i=0}^{N} \sum_{m=0}^{N} c_m(i) \Big(t + \frac{N+1}{2} - i\Big)^m. \qquad (17)$$

The latter form is more suitable for practical realizations. The polynomial coefficients $c_m(i)$ in the i-th interval are formed as linear combinations of the corresponding polynomial coefficients $c_m^n(i)$ of the B-spline of degree n, attending the i-th interval of the modified kernel

$$c_m(i) = \sum_{n=0}^{N} \gamma_{mn} c_m^n(i). \qquad (18)$$

IV. Least Squares Solutions

There are two main problems in making a down-scale projection (decimation). First, the continuous function $s(t)$ is not known. Hence, we cannot perform a true L_2 orthogonal projection in the form of Eqs. (9), (10). One can avoid this inconvenience by modeling this function with a continuous model. Thus, the subsequent projection would minimize the squared error between the approximation and the model. Second, even if an appropriate continuous function is provided, solving the integral given by Eq. (10) is rather problematic. This is due to the fact that, while the synthesis (reconstruction) function is compactly supported, the analysis one is not. Hence, in the continuous convolution described by Eq. (10), there are two functions being infinitely supported.

Equally well, we can try to stay entirely in the discrete domain seeking for a solution minimizing a certain discrete norm. We shall present this alternative in the next subsection.

A. Minimizing l_2 norm

This solution is very well known and described in most of the textbooks considering function approximations, bases, and signal expansions [8,9]. We present it here briefly, in order to use it later as a reference when comparing different solutions.

The discrete norm to be minimized is induced by the following discrete inner product over the grid $\boldsymbol{\tau}$:

$$\langle u, v \rangle \equiv h \sum_{i=0}^{L_{in}-1} u(\tau_i) v(\tau_i) \qquad (19)$$

being an approximation to the continuous inner product. It induces a semi-norm given by

$$||s||_{l^2}^2 = \langle s, s \rangle = h \sum_{i=0}^{L_{in}-1} s(\tau_i) s(\tau_i). \qquad (20)$$

This semi-norm serves in quantifying the distance between the approximation and the initial samples as follows:

$$||s - \tilde{s}||_{l^2}^2 = \min_{y \in V} ||s - y||_{l^2}^2. \qquad (21)$$

Here, s is assumed to be known only on the grid $\boldsymbol{\tau}$, that is, $s(hk) = x(k)$. Assume $\tilde{s}(t)$ is reconstructed by the basis $\varphi(t-i)$, as in (9), i.e. $\tilde{s}(\tau_k) = \sum_{j=0}^{L_{out}-1} c_j \varphi(\tau_k - j) = x(k)$. The solution for the unknown coefficients c_j, $j = 0, 1, ..., L_{out}-1$ is given by the following system of normal equations [8]:

$$\begin{aligned} \sum_{j=0}^{L_{out}-1} \Big(\sum_{k=0}^{L_{in}-1} \varphi(\tau_k - i) \varphi(\tau_k - j) \Big) c_j = \\ = \sum_{k=0}^{L_{in}-1} \varphi(\tau_k - i) x(k) \\ \text{for} \quad i = 0, ..., L_{out} - 1 \end{aligned} \qquad (22)$$

or in the matrix form as follows:

$$(\boldsymbol{\Phi}^T \boldsymbol{\Phi}) \boldsymbol{c} = \boldsymbol{\Phi}^T \boldsymbol{x}, \qquad (23)$$

$$\boldsymbol{c} = (\boldsymbol{\Phi}^T \boldsymbol{\Phi})^{-1} \boldsymbol{\Phi}^T \boldsymbol{x}. \qquad (24)$$

The matrix $[\boldsymbol{\Phi}]_{L_{in} \times L_{out}}$ where $L_{in} \geq L_{out}$, takes the coefficients' L_{out}-dimensional vector $\boldsymbol{c}$ to the reconstructed signal version $\tilde{s}(\tau_k)$ (L_{in}-dimensional). The matrix $[\boldsymbol{P}]_{L_{out} \times L_{out}} = \boldsymbol{\Phi}^T \boldsymbol{\Phi}$ Grammian of the reconstruction matrix $[\boldsymbol{\Phi}]$. Usually, the system given by Eq. (22) is solved by efficient methods for the matrix (pseudo) inversion taking advantage of the band structure of the matrix $\boldsymbol{P}$. These methods include, among others, the Cholesky factorization, and the

Givens or Householder QR decompositions [9]. For example, when performing the Cholesky factorization, the total number of floating point operations is (cf. [9]) $0.5(L_{in}N(N+3)+L_{out}(N-1)(N+1))+N(2N-1)+O(L_{in}+L_{out})$.

B. Minimizing L_2 norm

Remember that the initial signal is a discrete sequence $x(k)$ generated by uniformly sampling a certain *unknown* continuous function $s(t)$ according to $x(k)=s(\tau_k)=s(h_k)$. This makes it impossible to minimize directly the error $||s-\tilde{s}||^2_{L_2}$. Instead, a continuous model can be used:

$$x_s(t)=\sum x(k)K_\varepsilon\Big(\frac{t}{h}-k\Big) \tag{25}$$

for minimizing the error $||x_s-\tilde{s}||^2_{L_2}$. The function K_ε, possibly depending on a certain scale parameter ε, takes the samples $x(k)$ into a continuous function $x_\varepsilon(t)\in L_2$. Having the model given by Eq. (25), one can confidently apply the theory of Subsection III.A, namely, the coefficients $c(i)$ can be determined as in Eq. (10) as follows

$$\begin{aligned}
c(i)&=\langle x_s(\xi)\hat{\varphi}(\xi-i)\rangle=\int x_s(\xi)\hat{\varphi}(\xi-i)d\xi\\
c(i)&=\sum_n p^{-1}(n)\int x_s(\xi)\varphi(i-n-\xi)d\xi \qquad (26)\\
c(i)&=\sum_n p^{-1}(n)\int\Big\{\sum_k x(k)K_\varepsilon\Big(\frac{\xi}{h}-k\Big)\Big\}\varphi(i-n-\xi)d\xi\\
c(i)&=\sum_n p^{-1}(n)\Big\{\sum_k x(k)\int K_\varepsilon\Big(\frac{\xi}{h}-k\Big)\varphi(i-n-\xi)d\xi\Big\}
\end{aligned}$$

Consider the convolution integral

$$\phi(t)=\int k_\varepsilon\Big(\frac{\xi}{h}\Big)\varphi(t-\xi)d\xi. \tag{27}$$

It involves two continuous kernels with different sizes. By properly selecting the kernel K_ε this integral can be solved yielding the following linear model:

$$d(t)=h\sum_k x(k)\phi(t-hk). \tag{28}$$

Then, the procedure continues by sampling the latter at the integers followed by digital filtering with the convolution inverse p^{-1}. For a symmetrical sequence $p(k)$ with length $2N+1$ and $P(z)=\sum_{k=-N}^{N}p(k)z^{-k}$, digital filtering with $p^{-1}(k)$ means an efficient realization of the IIR filter $1/P(z)$ [10]. Fig. 2 shows the entire algorithm.

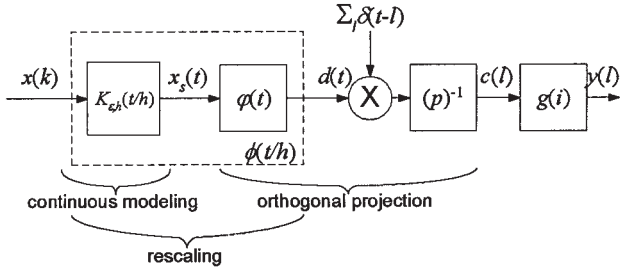

Fig. 2. Continuous LS decimation

1) *Solution of Munoz et al.* A very elegant solution entirely consistent with the continuous domain framework established in Section II has been developed in [4]. Their work concerns primarily B-splines of degree N as basis functions.

The first step in the algorithm is to match an interpolative, *continuous-domain, spline model* to the discrete data $x(k)$ by constraining the new function to have the same values at the coordinate grid τ. Thus, the modeling spline coefficients can be obtained applying Eqs. (3) and (4). This is equivalent to substituting the kernel K_ε by the interpolating function

$$K_\varepsilon(t)=\sum_{i=1}^{m}(b^N)^{-1}(i)\beta^N(t-i). \tag{29}$$

The benefit of the method proposed in [4] is that it involves in the integral given by Eq. (27) two splines. A convolution of two splines (of degrees N and M, correspondingly) is again a spline of degree $N+M+1$ [8,10]. When splines are represented by their B-spline expansion this integration can be separated into low-complexity discrete-time operations such as *finite differences* and *running sums* and the continuous interpolation with the B-spline kernel of degree $N+M+1$ [4]. After some rearrangements in Fig. 2, the final algorithm can be expressed as in Fig. 3 with the following steps [4]:

- Inverse filtering with $(b^N)^{-1}$ for obtaining the initial spline coefficients.
- $(N+1)$ running sums $\Delta^{-(N+1)}$.
- Re-sampling by a factor of $1/h$ using the spline model of degree $2N+1$.
- $(N+1)$ finite differences $\Delta^{(N+1)}$.
- Inverse filtering with $(b^{2N+1})^{-1}$ for generating the new spline coefficients.
- FIR filtering with the kernel b^N for obtaining the resized signal.

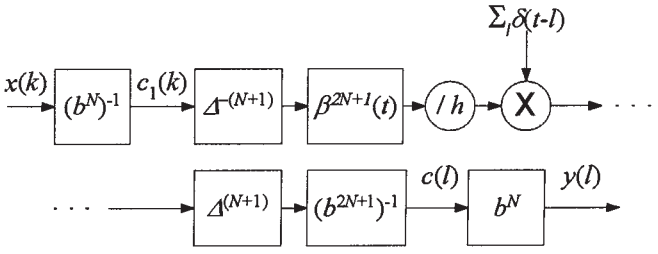

Fig. 3. Continuous LS signal decimation using finite differences

V. A Hybrid Method

To overcome the difficulties in solving the integral given by Eq. (27) we model the kernel K_ε by the Gaussian function

$$K_\varepsilon(t)=e^{-x^2/(4\varepsilon)}/(2\sqrt{\pi\varepsilon}). \tag{30}$$

While this is not an interpolating kernel, it allows us to approximate the integral given by Eq. (27) when $\varepsilon\to 0$, as

$$\lim_{\varepsilon\to 0}K_\varepsilon(t)=\lim_{\varepsilon\to 0}e^{-x^2/(4\varepsilon)}/(2\sqrt{\pi\varepsilon})=\delta(x). \tag{31}$$

It can be though as emphasizing the fact that we are interested in minimizing the error in L_2 sense especially around the given discrete points. This approximation has sense, especially for a relatively small sampling step h, which is the practical case in most of the images of interest. We will illustrate this assertion later by our experiments.

By using the above simplification, we get a particular form of Eq. (28), where $\phi(t)$ is formally replaced by $\varphi(t)$ as:

$$d(t) = h \sum_k x(k)\varphi(t - hk) \tag{32}$$

and

$$c(i) = \sum_n p^{-1}(n)\Big\{h \sum_k x(k)\varphi(i - n - hk)\Big\}. \tag{33}$$

Eq. (33) can be regarded as a form of Eq. (24) where the matrix inversion $(\mathbf{\Phi}^T\mathbf{\Phi})^{-1}$ is substituted by the digital recursive filtering $1/P(z)$.

When the compactly supported piece-wise polynomial functions in the form of Eq. (14) are used, the inner sum in Eq. (33) can be realized very efficiently by using the so-called transposed modified Farrow structure [11,5,12]. We will discuss this realization in the next subsection.

A. *Transposed Farrow structure*

Recall that our basis functions are composed from B-splines up to degree N in a combination preserving the support of the highest degree B-spline. Hence, they are formed of polynomial pieces as in Eq. (17).

By substituting Eq. (17) into Eq. (32) and sampling at the integers we get

$$d(l) = h \sum_k x(k) \sum_{i=0}^{N} \sum_{m=0}^{N} c_m(i)\Big(l + \frac{N+1}{2} - i - hk\Big)^m. \tag{34}$$

Changing the order of the summations results in the following practically realizable form

$$d(l) = h \sum_{m=0}^{N} \sum_{i=0}^{N} c_m(i) \sum_k x(k)\Big(l + \frac{N+1}{2} - i - hk\Big)^m. \tag{35}$$

The innermost sum contains input signal samples inside an interval of the *unity length* weighted by fraction values raised to various powers of m for $m = 0, 1, , N$. Denoting those fractions by

$$\mu_k = |j - \tau_k| \quad \text{for} \quad l \le \tau_k < l + 1 \tag{36}$$

results in the practical implementation scheme shown in Fig. 4, known as the transposed modified Farrow structure [11,5,12]. It contains $N+1$ filters with coefficients $c_m(i)$ determined by the B-splines involved. The sampling rate conversion [innermost summation in Eq. (35)] is made in the accumulator blocks. The outputs of these blocks are used as inputs to the fixed filters when $\mu_k < \mu_{k-1}$. Then, the accumulators are reset for the new sample summation.

This structure realizes exactly the matrix multiplication between the transposed basis sampled at the grid $\boldsymbol{\tau}$ and the signal vector: $\mathbf{\Phi}^T\boldsymbol{x}$. In $\mathbf{\Phi}$, the norm of each basis vector $\varphi(\boldsymbol{\tau} - \boldsymbol{l})$ differs from $1/h$. This would not be a problem if we would realize the l_2 error norm minimization given by Eq. (24). But, formally replacing it by Eq. (35), we have no preservation of the constant in the output. For some re-sampling ratio range this can cause visible periodic texture artifacts in the smooth areas of the reconstructed image. Therefore, we have to take care of the constant preservation in the *intermediate* stage given by Eq. (34) before the IIR filtering. It has turned out the above-mentioned problem can be solved by using the local scaling factors h_l as follows:

$$d(l) = h_l \sum_k x(k)\varphi(l - hk) \tag{37}$$

where

$$(1/h_l) = \sum_k \varphi(l - hk). \tag{38}$$

The local scaling factors h_l depend on $\boldsymbol{\tau}$. When performing uniform re-sampling on images, the same basis and the same grid are used for all column-wise (row-wise) transformations, hence the scaling factors are the same. To obtain them, what is needed is one more re-sampling operation on a vector with the column (row) length composed of unities.

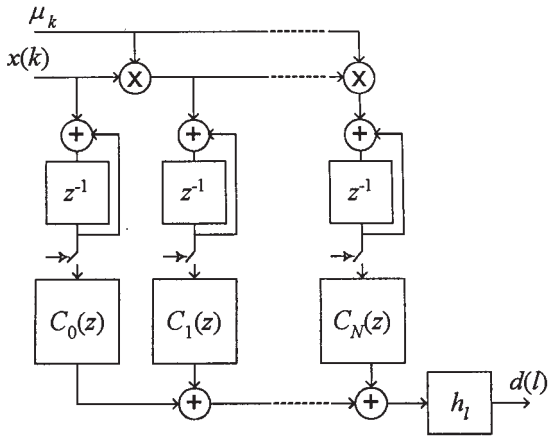

Fig. 4. Transposed modified Farrow structure for piece-wise polynomial signal decimation

B. *Relation with the l_2 solution*

The method can be represented in the following matrix form

$$\boldsymbol{c} = (\boldsymbol{H}\boldsymbol{P})^{-1}\mathbf{\Phi}^T\boldsymbol{x}. \tag{39}$$

Here, the matrix $[\boldsymbol{P}]_{L_{out}\times L_{out}}$ is a band matrix containing $p(k)$ along its rows and $[\boldsymbol{H}]_{L_{out}\times L_{out}}$ is a diagonal matrix containing the scaling factors as given by Eq. (38). To illustrate how close this solution is to the corresponding l_2 solution, the norm or the error matrix

$$\boldsymbol{E} = \mathbf{\Phi}(\mathbf{\Phi}^T\mathbf{\Phi})^{-1} - (\boldsymbol{H}\boldsymbol{P})^{-1}\mathbf{\Phi}^T \tag{40}$$

has been computed for different values of h. Fig. 5 shows the result for the case of cubic B-spline basis function.

It can be seen that the error between the true discrete LS and the hybrid method is relatively low for decimation ratios up to some value close to 0.9. After this value, the error increases considerably because the proposed model is no longer adequate.

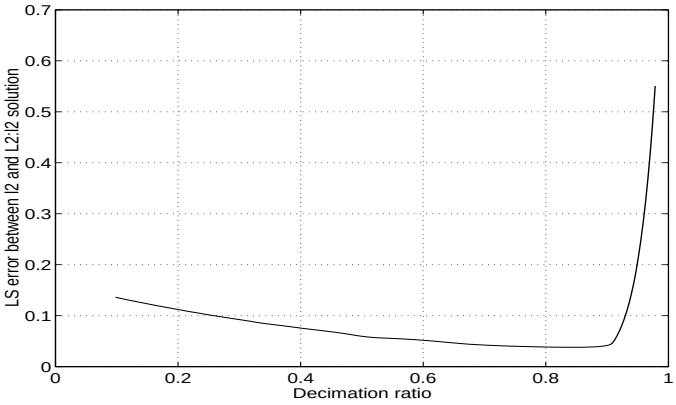

Fig. 5. Error norm $||\boldsymbol{E}||_{l_2}$ for cubic B-spline basis. $L_{in} = 512$

VI. Comparative Analysis of Different Decimation Schemes

A. *Frequency-domain analysis*

The schemes presented above can be considered as forms of continuous-domain filtering of the impulse train $x_(t) = \sum_k x(k)\delta(t/h - k)$. This allows us to compare their (s continuous frequency responses.

The interpolative scheme of Subsection II.A for the grid on the integers has the frequency response given by

$$\Phi_{\text{int}}(\omega) = \Phi(\omega)/\Phi(e^{j\omega}). \tag{41}$$

The continuous LS scheme of Subsection IV.A can be presented in the frequency domain as

$$\Phi_{L_2}(\omega) = \frac{\Phi(\omega)\Phi(h\omega)}{P(e^{j\omega})\Phi(e^{jh\omega})}, \tag{42}$$

where $P(e^{j\omega})$ is the frequency response of the autocorrelation sequence given by Eq. (13).

The new scheme takes in the frequency domain the following rather simple form:

$$\Phi_{l_2}(\omega) = \Phi(\omega)/P(e^{j\omega}). \tag{43}$$

The frequency responses are shown in Fig. 6 for the case of cubic B-spline basis, that is, $\varphi(t) = \beta^3(t)$. As can be seen,

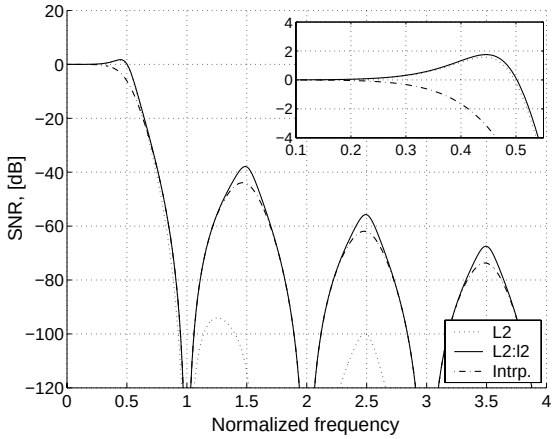

Fig. 6. Continuous frequency responses of different decimation systems

all schemes possess anti-aliasing properties around multiples of $F_{\text{out}} = 1$ and are rather similar in the transition band. The main difference appears in the pass-band where the LS schemes provide better performances especially for frequencies close to 0.5. The bump appearing in this region reflects the duality in decimation and interpolation processes [13].

Next, we compare three different kernels realizing the hybrid LS method. The first kernel is the following linear combination of the cubic and linear B-splines: $\varphi_{\text{mod}}(t) = \beta^3(t) + \sum_{i=-1}^{1} \gamma_{3i}\beta^1(t-i)$ [1]. The second kernel is the following linear combination of the cubic B-spline and its second derivative $\varphi_{moms}(t) = \beta^3(t) + \gamma\frac{d^2\beta^1(t)}{dt^2}$ [2]. The third kernel is the cubic B-spline itself. Their frequency characteristics for some well-optimized γ's are shown in Fig. 7. What is seen is that the modified kernel, initially optimized to possess good anti-imaging properties [1], has a flatter frequency response for the dual function. This, combined with the flatter region of the interpolator near to the cut-off frequency, would give better preservation of the high-frequency details in the processed image.

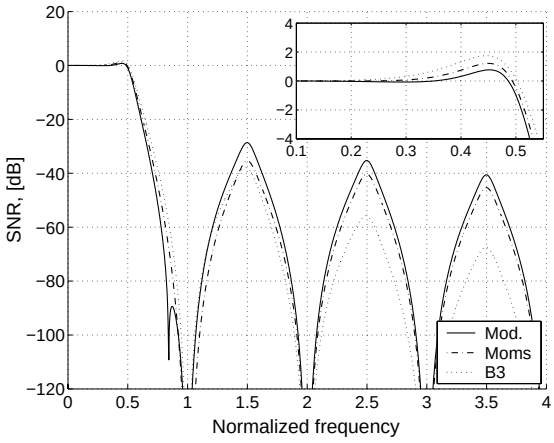

Fig. 7. Continuous frequency responses of systems realizing l_2 LS signal decimation with different kernels. For moms case $\gamma = 1/42$; for modified case $\gamma_{30} = 2\gamma_{31} = -0.0714$

B. *Quality assessment*

We compare the quality by measuring the SNR between the original image and the image resulting from a two-step processing: *decimation* followed by *reconstruction* to the initial size. In these two steps, the respective operators have been chosen to be dual each other, thus complying with the LS paradigm. We have realized also the interpolative scheme described in Subsection II.A.

The well-known *Barbara* and *Lena* test images of size 512×512 have been processed. We changed the target size from 51×51 to 511×511 and built the SNR curves for different methods, as shown in Figs. 8 and 9. We related also the SNRs ?60 of the L_2 solution and our solution (denoted as $L_2 : l_2$) to the pure discrete (l_2) solution. The proposed method performs equally well compared to the original l_2 solution up to a certain target size close to the original image size. After that, it fails. The same is true also for the

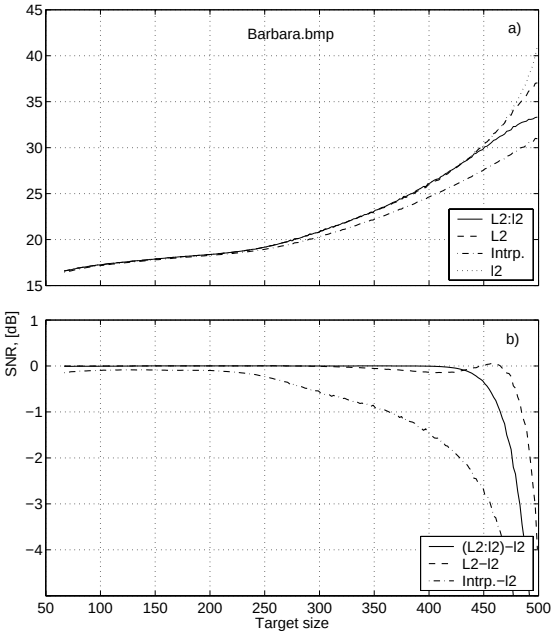

Fig. 8. Fig. 8. a) SNR for different decimation schemes; b) relative SNR to Target size l2 scheme. *Barbara* image processed

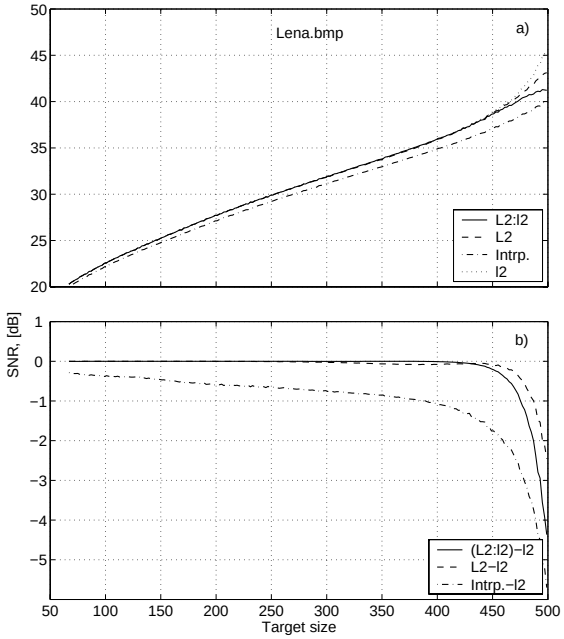

Fig. 9. Same as in Fig. 8 for *Lena* image

L_2 solution when compared to the l_2 solution, although for a higher size, because its corresponding continuous model is more adequate than ours for large and close to unity sampling steps. Although the relative differences to the l_2 solution can be quite large (up to 5 dB), the absolute values of SNR are higher than 32 dB, indicating subtle variances from the original image.

As far as the interpolative method is concerned, it performs always worse than the LS methods. This is less visible for small target sizes, where more aliasing is introduced for the frequency content close to 0.5. However, for moderate

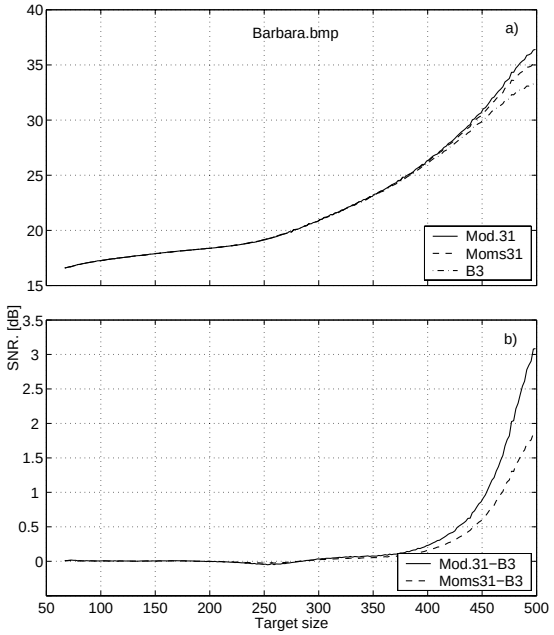

Fig. 10. (a) SNR for different decimation kernels. (b) Relative SNR to the cubic spline kernel. *Barbara* image processed

decimation ratios it is evident that providing the good anti-aliasing properties is not enough. Some better performance in the pass-band is needed and it is provided by the higher order IIR filtering assuring the LS optimality.

Three different kernels are compared in terms of their LS decimation performance in Fig. 10. The benefit from using modified B-spline basis vs. other bases is outlined by the higher SNR, especially for low decimation ratios.

C. Computational complexity assessment

The computational complexity has been measured in terms of the number of multiplications and additions required to process one image row. We refer to Figs. 2 and 3 that help to evaluate the number of operations for each step (see also [5,4] for details).

For a kernel with the highest degree being N and the input

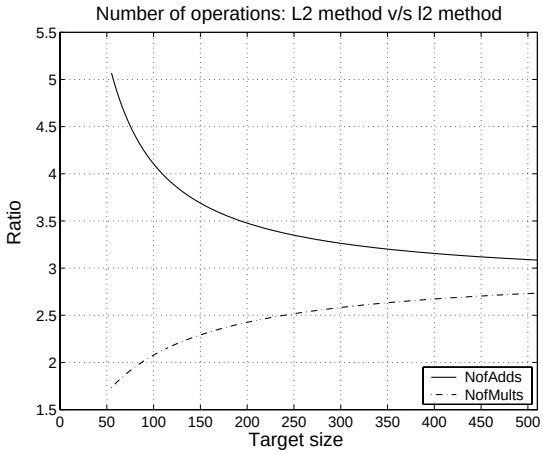

Fig. 11. 12 NOR: L_2 method versus $L_2 : l_2$ method

and output signals of lengths L_{in} and L_{out}, respectively, the numbers are summarized in Tables 1 and 2.

Table 1. Number of operations for L_2 method [4]

Algorithm step		Multiplications	Additions
1.	Pre-filtering	(N 1)/2)(2m-1)	(N 1)/2)(2m 1)
2.	Running sums	—	(N+1)m
3.	Interpolation		
	- Output grid	(2N+1)(2N+2)n	(2N+1)(2N+2)n
	- Filtering	(2N+1)n	(2N+1)n
4.	Running diff.	—	(N+1)n
5.	IIR filtering	N(2n 1)	N(2n 1)
6.	FIR filtering	(N+1)/2)n	(N+1)/2)n
	Total:	(N 1)/2)(2m 1)+ + (4N^2+10.5N+4)n-N	3mN (N 1)/2)+ + (4N^2+11.5N+5)n-N

Table 2. Number of operations for $L_2 : l_2$ method

Algorithm step		Multiplications	Additions
1.	Pre-filtering	(N 1)/2)(2m-1)	(N 1)/2)(2m 1)
2.	Running sums	—	(N+1)m
3.	Interpolation		
	- Output grid	(2N+1)(2N+2)n	(2N+1)(2N+2)n
	- Filtering	(2N+1)n	(2N+1)n
4.	Running diff.	—	(N+1)n
5.	IIR filtering	N(2n 1)	N(2n 1)
6.	FIR filtering	(N+1)/2)n	(N+1)/2)n
	Total:	(N 1)/2)(2m 1)+ + (4N^2+10.5N+4)n-N	3mN (N 1)/2)+ + (4N^2+11.5N+5)n-N

Fig. 11 shows the number-of-operations ratio (NOR) between two methods. As can be seen, the proposed method reduces the computational complexity at least by factor of two in terms of the number of multiplications and at least by a factor of three in terms of the number of additions. This reduction is mainly due to the efficient computational structure discussed in Subsection V.A and being a result of the compromise made when considering the continuous signal model given by Eq. (32).

In order to be entirely consistent with the continuous LS, the method described in [4] demands a pre-filtering to get the continuous spline model and an interpolation with higher degree spline. The proposed $L_2 : l_2$ method, in turn, helps to lower the computations by compromising between continuous norm minimization and its discrete counterpart. The L_2 method also involves running sums in its second step. While this procedure can be realized rather easily, it should be maintained carefully because it risks introducing overflow or round-off errors, especially for large size images. Munoz *et al*. [4] have successfully avoided those potential problems by introducing some small amount of additional computations (not included in the comparison above). As far as our method is concerned, it is not risky in this sense. We have already commented the need of local normalization that, in the case of images, is not so costly additional operation. The computational complexity of our method can be reduced even further by exploiting the symmetries of the sub-filters $C_m(z)$ in the transposed Farrow structure of Fig. 4 (see [5] for a detailed analysis).

VII. Conclusions

In this paper, we reviewed the recent advances in performing image resizing and presented a novel method based on a hybrid form of the LS. By taking the limit case in our continuous model, we managed to simplify the L_2 norm minimization problem and to make it resembling to its discrete counterpart. However, the new solution is not the classical discrete LS since, in fact, we replaced the matrix inversion – the most costly operation in the discrete case – by a more efficient IIR filtering. We commented some important details in realizing the transposed Farrow structure, namely the need for a local scaling (normalization). We showed that our method dramatically lowers the computational complexity when the quality remains practically the same as in the classical discrete case for a wide range of decimation ratios. Our method competes successfully in quality with the state-of-the-art method developed in [4] while being at least twice faster. We compared also the performance of a number of spline-based kernels, all of them having the same computational complexity, but different approximation or anti-aliasing properties. It was demonstrated that by properly optimizing the basis functions, the proposed solution can be made very close to the classical l_2 solution.

Acknowledgement

This work was supported by the Academy of Finland, project No. 44876 (Finnish Centre of Excellence Program).

References

[1] A. Gotchev, K. Egiazarian, and T. Saramäki, "Optimization Techniques in Designing Piece-wise Polynomial Interpolators of Minimal Support", *Int. Journal Wavelets, Multiresolutiom and Information Processing*, in press.

[2] T. Blu, P. Thévenaz, and M. Unser, "MOMS: Maximal-Order Interpolation of Minimal Support", *IEEE Trans. Image Proc*., vol. 10, no. 7, pp. 1069-1080, July 2001.

[3] Yu-Ping Wang, "Scale-space Derived from B-spline", *IEEE Trans. PAMI*, 1998.

[4] A. Munos, T. Blu, and M. Unser, "Least-Squares Image Resizing Using Finite Differences", *IEEE Trans. Image Proc.*, vol. 10, No.9, pp. 1365-1378, September, 2001.

[5] D. Babic, J. Vesma, T. Saramäki, and M. Renfors, "Implementation of the transposed Farrow structure", in *Proc. Int. Symposium Circuits and Systems, ISCAS'2002*, Phoenix, Arizona, USA, May 2002, vol. IV, pp. 5-8.

[6] A. Aldroubi and M. Unser, "Sampling procedures in function spaces and asymptotic equivalence with Shannon's sampling theory", *Numer. Funct. Anal. Optim*. vol. 15, No.1/2, pp. 1-21, 1994.

[7] A. Ron, "Factorization theorems for univariate spline on regular grids", *Israel Journal of Mathematics*, vol. 70, No.1, pp. 48-68, 1990.

[8] C. de Boor, *A Practical Guide to Splines*, Springer-Verlag, Second Edition, 2001.

[9] A. Björk, *Numerical methods for least squares problems*, SIAM, 1996.

[10] M. Unser, A. Aldroubi, M.Eden, “B-spline signal processing: Part I – Theory and Part II – Efficient design and applications”, *IEEE Trans. Signal Processing*, vol. 41, pp. 821-848, 1993.

[11] A. Gotchev, J. Vesma, T. Saramaki, and K. Egiazarian “Multiscale image representations based on modified B-splines”, in *Proc. Int. Workshop on Spectral Techniques and Logic Design*, SPEGLOG’2000, Tampere, Finland, June 2000, pp. 431-452.

[12] T. Hentschel, and G. Fettweis, “Continuous-time digital filters for sam-ple-rate conversion in reconfigurable radio terminals”, *Proc. European Wireless*, Dresden, Germany, Sept. 2000, pp. 55-59.

[13] M. Unser, A. Aldroubi and M. Eden, “Polynomial spline signal approximation: Filter design and asymptotic equivalence with Shannon’s sampling theorem”, *IEEE Trans. Inform. Theory*, vol. 38, pp. 95-103, January 1992.