Variational Image Reconstruction from Arbitrarily Spaced Samples: A Fast Multiresolution Spline Solution

Muthuvel Arigovindan, Michael Sühling, Patrick Hunziker, and Michael Unser*,

EDICS: 2-INTR

Muthuvel Arigovindan, Michael Sühling, and Michael Unser are with Biomedical Imaging group, Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland; e-mail: muthuvel.arigovindan@epfl.ch, michael.suehling@epfl.ch, michael.unser@epfl.ch

Patrick Hunziker is with the Kantonsspital Basel, Petersgraben 5, CH-4031 Basel, Switzerland; e-mail: PHunziker@uhbs.ch

This work was supported in part by the Swiss Heart Foundation and the Swiss National Science Foundation under grant 200020-101821.

Abstract

We propose a novel method for image reconstruction from non-uniform samples with no constraints on their locations. We adopt a variational approach where the reconstruction is formulated as the minimizer of a cost that is a weighted sum of two terms: (i) the sum of squared errors at the specified points; (ii) a quadratic functional that penalizes the lack of smoothness. We search for a solution that is a uniform spline and show how it can be determined by solving a large, sparse system of linear equations. We interpret the solution of our approach as an approximation of the analytical solution that involves radial basis functions (RBFs) and demonstrate the computational advantages of our approach. Using the two-scale relation for B-splines, we derive an algebraic relation that links together the linear systems of equations specifying reconstructions at different levels of resolution. We use this relation to develop a fast multigrid algorithm. We demonstrate the effectiveness of our approach on some image reconstruction examples.

Index Terms

non-uniform interpolation, variational reconstruction, radial basis functions, thin-plate splines, B-splines, multiresolution reconstruction, multigrid algorithm

I. Introduction

The problem of non-uniform sampling and reconstruction has received considerable attention in the recent past. Examples include sampling and reconstruction schemes for the space of band-limited signals [1], [2], [3], [4], and more general shift invariant wavelet-like spaces [5]. Most of the works in this area address the problem with a restriction on the distribution of sample points, usually the maximum gap between the sample points. However, in many applications such as shape reconstruction, landmark interpolation, image recovery from the contours etc., such restrictions are not practical.

When there is no restriction on the distribution of sample points, the reconstruction problem is not uniquely defined and hence ill-posed. One way to handle such a problem is to adopt a variational approach where the reconstruction is formulated as an optimization problem. The cost function is typically a weighted sum of two terms: (i) the sum of squared errors at the specified points; (ii) a regularization term, usually a quadratic functional that penalizes the lack of smoothness. The regularization term takes care of the ill-posedness of the problem and gives a meaningful reconstruction. The weighting is set up such as to find a compromise between fitting the data well and penalizing reconstructions that are not smooth enough. The presence of the regularization is crucial for dealing with large sampling gaps; it allows these to be filled-in in a smooth way using the information from the surrounding samples.

The optimal solution for the variational formulation is given by the method of radial basis functions (RBFs) which are also known as thin-plate splines when a particular type of regularization is used. Thin-plate splines are optimal when the regularization belongs to a class rotationally invariant semi-norms [6] that are appropriate for most of the practical applications. The solution is expressed as a linear combination of shifted radial basis functions positioned at the location of the data points [6], [7], [8]. The optimal weights are determined as the solution of a linear system of equations.

While thin plate splines are undoubtedly among the preferred techniques for dealing with non-uniform interpolation in multiple dimensions [9], [10], they tend to break down numerically when there are too many data points; say, $M > 100$. The main difficulty is that the linear system to be solved is ill-conditioned. Moreover, the matrix is not sparse. Hence, solving the system soon becomes overly expensive; the complexity is in $\mathcal{O}(M^3)$, where M is the number of sample points. Various solutions have been proposed for reducing this complexity and improving the numerical behaviour [11], but there is still a long way to go for making them practical. Another fundamental limitation is that computing the weights is only a part of the effort. Indeed, the solution has to be resampled numerically if it is to be displayed on a regular grid; this will cost an additional $\mathcal{O}(MN)$ operations, where N is the number of grid points.

In 1D, the situation is not as dramatic because the optimal solution can be expressed in terms of non-uniform B-splines which are compactly supported as opposed to the RBFs which are not; this is the key for obtaining an efficient solution [12]. Unfortunately, this is not possible in higher dimensions; i.e., there are no compactly supported functions that reproduce spline-related RBFs.

In this paper, we circumvent the above mentioned difficulties by searching for the solution of the regularized reconstruction problem in the space of uniform splines where the basis functions are now attached to the reconstruction grid, as opposed to the data points. We consider Duchon's class of semi-norms for regularization. In other words, we are proposing to discretize thin plate splines using uniform B-splines with a degree that is matched to the underlying semi-norm. In this way, we have at least the guarantee that the solution in 1D coincides with the theoretical one, provided that the sample points are on the reconstruction grid. This helps to say qualitatively that the solution in general will not be very different from the exact analytical one.

The proposed approach has many advantages over using RBFs. First, the linear system for getting the B-spline coefficients is well-conditioned. Second, the system matrix is sparse resulting in much faster computation. Third, the formulation lends itself quite naturally to an efficient multiresolution and multigrid implementation, thanks to an interscale relation that is derived in the paper. This reduces the complexity of solving the linear system to $\mathcal{O}(N)$, where N is the number of grid points. Fourth, after solving the linear system, there is no expensive resampling step as in RBF reconstruction. The samples at the grid locations are obtained by a simple digital filtering [13]. Finally, we should note that unlike other purely discrete variational formulations of the image reconstruction problem [14], [15], [16], we get a solution that is defined in the continuous domain with all the corresponding advantages.

The present algorithm offers a lot of a user-flexibility. The number and location of the data points can be arbitrary without any incidence on the computational speed. One can reduce measurement noise by imposing more or less smoothness on the solution (regularization). One can also reconstruct the image at any desired resolution (step a); the solution will converge to the analytical one (RBF) as a gets sufficiently small—the rate is given by the order of the spline.

The paper is organized as follows. In Section II, we set up the variational reconstruction problem in the continuous domain and provide its analytic solution which involves RBFs. In Section III, we present our B-spline formulation and derive the corresponding linear system of equations. We also derive the interscale relation that relates the system matrices at various levels of resolution. We present the algorithm in the fourth section and give experimental results in the fifth section.

II. Variational Reconstruction

A. The Problem

Given measurements $\{f_i\}$ at locations $\{\mathbf{x}_i : \mathbf{x}_i \in \mathbb{R}^d\}$, the basic interpolation problem is to construct a continuously-defined function, $S(\mathbf{x})$, such that $S(\mathbf{x}_i) = f_i$. Since this problem is obviously ill-posed, one needs to introduce some additional constraint that makes the solution unique. In the variational approach [6], one searches for the solution that minimizes the cost function

$$\mathcal{M}_p(S) = \int\int ||D^p S(\mathbf{x})||^2 d\mathbf{x}$$

where D^p is the vector of all possible partial derivative operators of order p. For example, for $p = 2$ and $d = 2$,

$$\begin{aligned}\mathcal{M}_2(S) &= \int\int \left(\frac{\partial^2 S(x,y)}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 S(x,y)}{\partial x \partial y}\right)^2 \\ &\quad + \left(\frac{\partial^2 S(x,y)}{\partial y^2}\right)^2 dxdy.\end{aligned}$$

In practice, the measurements are often noisy or lacking precision, and it is not necessarily desirable to reconstruct a function $S(\mathbf{x})$ that interpolates the data f_i exactly. One therefore relaxes the interpolation constraint, trading some closeness of fit against more smoothness on the solution [6], [17]. In that case, the reconstruction is formulated as the minimizer of

$$\sum_i |S(\mathbf{x}_i) - f_i|^2 + \lambda \mathcal{M}_p(S).$$

where λ is a so-called regularization parameter that works as a trade-off factor. Note that, in this second version, we have an approximation problem rather than an interpolation. Interestingly, we can get back to the first case by selecting λ to be arbitrarily small (but non-zero).

B. The classical solution

The solution of the above problem is well-known in approximation theory and statistics [17], [18]. Duchon [6] was one of the first to establish that the solution can be written as

$$S(\mathbf{x}) = \left[\sum_i w_i \phi(||\mathbf{x} - \mathbf{x}_i||) \right] + p(\mathbf{x}),$$

which is made up of two terms. The first is a linear combination of so-called radial basis functions (RBF) which are radially symmetric, and positioned at the sampling locations. The second term, $p(\mathbf{x})$, is a polynomial that lies in the kernel of the smoothness semi-norm—i.e. $D^p p(\mathbf{x}) = 0$—and has no contribution to the cost function. The RBFs, $\phi(r)$, which solely depends on $\mathcal{M}_p(S)$, are given by

$$\phi(r) = \begin{cases} r^{2p-d} \log r, & \text{if } 2p - d \text{ is even} \\ r^{2p-d} & \text{if } 2p - d \text{ is odd} \end{cases}$$

where r is the radial distance from the origin. They belong to a specific class of radial basis functions called thin plate splines. The above analytical formula specifies a vector space for the solution. To fully describe the latter, one still needs to determine the appropriate weights through the solution of a linear system of equations. To this end one defines the two weight vectors $\mathbf{w} = [\cdots \ w_i \ \cdots]^T$ and $\mathbf{a} = [a_0 \ \cdots \ a_{p-1}]^T$ where the a_i's are the coefficients of the polynomial $p(\mathbf{x}) = \sum_{i=0}^{p-1} a_i q_i(\mathbf{x})$ with $q_i(\mathbf{x})$ being the basis for the kernel of D^p (e.g. monomials in x and y). The numerical solution for the approximation problem is given by

$$\begin{bmatrix} \mathbf{B} + \lambda \mathbf{I} & \mathbf{Q} \\ \mathbf{Q}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}$$

where $\{\mathbf{B}\}_{i,j} = \phi(||\mathbf{x}_i - \mathbf{x}_j||)$, and $\{\mathbf{Q}\}_{i,j} = q_j(\mathbf{x}_i)$.

For the 2D case ($d = 2$), the RBFs are given by $\phi(r) = r^{2p-2} \log r$. For the first order semi-norm

$$\mathcal{M}_1(S) = \int\int \left(\frac{\partial S(x,y)}{\partial x} \right)^2 + \left(\frac{\partial S(x,y)}{\partial y} \right)^2 dxdy,$$

the RBF is $\phi(r) = \log r$. For the second order semi-norm

$$\begin{aligned}\mathcal{M}_2(S) &= \int\int \left(\frac{\partial^2 S(x,y)}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 S(x,y)}{\partial x \partial y}\right)^2 \\ &\quad + \left(\frac{\partial^2 S(x,y)}{\partial y^2}\right)^2 dxdy,\end{aligned}$$

the RBF is $\phi(r) = r^2 \log r$.

Since radial basis functions do not have a finite support, the matrices in the above system of equation are dense. Also, since the magnitude of the functions grow with distance from the centre, the matrices are poorly conditioned. This makes RBFs impractical and numerically unstable when the number of samples is large.

C. Relation to B-splines

For the unidimensional problem, the thin-plate RBF is given by $\phi(x) = |x|^{2p-1}$, which is a polynomial of degree $2p-1$, with a discontinuity of order $2p-2$ at the origin. This implies that the solution is a polynomial spline of degree $2p-1$ with knots at the x_i's. It turns out that these splines have basis functions, the so-called non-uniform B-splines, which are compactly supported and therefore much better conditioned than the radial basis functions [12], [19]. The corresponding numerical technique is called "smoothing splines" and is widely used in practical applications [18]. In the special case where the samples are uniformly spaced, i.e. $x_i = i \cdot a$, there is still another simplification since all the B-splines become shifted replicates of each other. The corresponding solution can be represented as $S(x) = \sum_k c_k \beta^{2p-1}(x/a - k)$, where $\beta^{2p-1}(x)$ is the central B-spline of degree $2p-1$. This means that the approximation problem can be discretized exactly if we work with B-splines of size a. Moreover, the expansion coefficients c_k of the smoothing spline can be evaluated very efficiently by recursive digital filtering [13]. Now, if there are gaps in the samples but the available ones are positioned on the grid (integer multiples of a), then the solution can still be expressed as a linear combination of B-splines with step size a, but the algorithm does not have a simple filtering interpretation anymore. The two cases of interest to us are $p = 1$ and $p = 2$, which lead to piecewise linear and cubic spline solutions, respectively.

Unfortunately, for dimensions higher than one, there are no compactly supported functions that span the same space as the radial basis functions. Thus, a uniform B-spline discretization of the problem is not rigorously exact anymore. However, a B-spline basis with a degree that is matched to p remains the best possible choice among all tensor product shift-invariant bases, because it has a high enough order of differentiation and it is compatible with the optimal 1D solution. The slight discrepancy with the optimal analytical solution that this may generate is largely compensated by the computational advantages (sparse matrices, multi-resolution) provided by this type of representation. Additionally, the error can be made arbitrarily small by decreasing the sampling step a. In this last respect, B-splines offer another advantage: for a given support size, they are the refinable functions that result in the smallest discretization error [20]. Specifically, for the B-spline of degree n which is of support $L = n + 1$, the approximation error decays like $\mathcal{O}(a^L)$, which is the maximum rate possible [21]. In other words, they provide the best quality for a given computational cost.

III. B-spline Discretization and the Solution

In this section, we present the proposed B-spline discretization of the problem and specify its solution. We discretize the variational reconstruction problem by searching for the optimal solution within the space of uniform splines.

A. B-spline Formulation

Given a set of noisy measurements $\{f_i\}$ of the image at sampling locations $\{x_i, y_i\}$, the approximation problem is now to find a uniform spline $S(x, y)$ of the form

$$S(x,y) = \sum_{l=0}^{(N-1)} \sum_{k=0}^{(N-1)} c_{k,l} \beta^n(x/a - k)\beta^n(y/a - l) \tag{1}$$

such that

$$\mathcal{C}(S) = \sum_i |S(x_i, y_i) - f_i|^2 + \lambda \int\int ||D^p S||^2 dxdy \tag{2}$$

is minimized. Obviously, the degree of the B-spline should be chosen such that the regularization term does not blow up; i.e., $n \geq p$.

The analytical part of this discretization process is to express the second part of the cost function in term of the expansion coefficients $c_{k,l}$. The regularization term is

$$\int\int ||D^p S||^2 dxdy = \sum_{q_1+q_2=p} \binom{p}{q_1} D_{q_1,q_2}$$

where

$$D_{q_1,q_2} = \int\int \left[\frac{\partial^p S(x,y)}{\partial x^{q_1} \partial y^{q_2}}\right]^2 dxdy$$

Define $\alpha_{q,a}(x) = \frac{d^q}{dx^q}\beta^n(x/a)$. Then

$$\begin{aligned} D_{q_1,q_2} &= \sum_{l,n}\sum_{k,m} c_{k,l}c_{m,n} \\ &\quad \left[\int \alpha_{q_1,a}(x-k)\alpha_{q_1,a}(x-m)dx\right] \\ &\quad \left[\int \alpha_{q_2,a}(y-l)\alpha_{q_2,a}(y-n)dy\right], \end{aligned}$$

which yields

$$D_{q_1,q_2} = \sum_{l,n}\sum_{k,m} c_{k,l}c_{m,n}\eta_{q_1}((m-k)a)\eta_{q_2}((n-l)a)$$

where

$$\eta_q(x) = \frac{d^q}{dx^q}\beta^n(x/a) * \frac{d^q}{dx^q}\beta^n(-x/a) \tag{3}$$

Define,

$$\gamma_q(k) = \eta_q(ka)$$

$$r_{q_1,q_2}(k,l) = \gamma_{q_1}(k)\gamma_{q_2}(l) \tag{4}$$

Then

$$\begin{aligned} D_{q_1,q_2} &= \sum_{l,n}\sum_{k,m} c_{k,l}c_{m,n}r_{q_1,q_2}(m-k,n-l) \\ &= \sum_{k,l} c_{k,l}\sum_{m,n} c_{m,n}r_{q_1,q_2}(m-k,n-l) \\ &= \langle r_{q_1,q_2}(k,l) * c_{k,l},\ \ c_{k,l}\rangle \end{aligned}$$

Hence,

$$\int\int ||D^pS||^2 dxdy = \sum_{q_1+q_2=p}\binom{p}{q_1}\langle r_{q_1,q_2}(k,l) * c_{k,l},\ \ c_{k,l}\rangle$$

Finally,

$$\int\int ||D^pS||^2 dxdy = \langle r(k,l) * c_{k,l},\ \ c_{k,l}\rangle$$

with $r(k,l) = \sum_{q_1+q_2=p}\binom{p}{q_1}r_{q_1,q_2}(k,l)$; in other words, the regularization term is a quadratic form of the $c_{k,l}$'s with a special convolutional structure.

In the filter component (4), $\gamma_q(k)$ is the discrete B-spline kernel of order $2n-2q+1$, convolved with the finite difference operator of order $2p$. This can be verified by using the following properties of splines [13]: (i) the derivative of a B-spline is a B-spline of reduced degree convolved with a finite difference operator; (ii) the convolution of two B-splines results in a B-spline of increased order. Table I gives the regularization filters for the first order and the second order semi-norms.

p	$R(z_1, z_2)$
1	$\triangle_*{}^2(z_1)B^{2n-1}(z_1)B^{2n+1}(z_2)+$ $\triangle_*{}^2(z_2)B^{2n-1}(z_2)B^{2n+1}(z_1)$
2	$\frac{1}{(a)^2}\big[\triangle_*{}^4(z_1)B^{2n-3}(z_1)B^{2n+1}(z_2)+$ $2\triangle_*{}^2(z_1)B^{2n-1}(z_1)\triangle_*{}^2(z_2)B^{2n-1}(z_2)+$ $\triangle_*{}^4(z_2)B^{2n-3}(z_2)B^{2n+1}(z_1)\big]$

TABLE I

REGULARIZATION FILTERS. $\triangle_*{}^{2n}(z) = (z - 2 + z^{-1})^n$ (CENTRAL FINITE DIFFERENCE OPERATOR OF ORDER $2n$);
$B^n(z) = \sum_{k=-n/2}^{n/2} \beta^n(k) z^{-k}$ (DISCRETE B-SPLINE KERNEL OF DEGREE n).

To introduce the corresponding matrix formulation, we define the coefficient and data vectors

$$\mathbf{c} = [c_{0,0} \cdots c_{N-1,0} \cdots c_{N-1,N-1}]^T.$$

$$\mathbf{f} = [\cdots \ f_i \ \cdots].$$

Then the cost is given by

$$\mathcal{C}(S) \quad = \quad ||\mathbf{f} - \mathbf{S}\mathbf{c}||^2 + \lambda \mathbf{c}^T \mathbf{R} \mathbf{c}$$

where

$$\{\mathbf{S}\}_{i,Nl+k} = \beta^n(x_i/a - k)\beta^n(y_i/a - l),$$

and $\mathbf{R}$ is a block-circulant matrix that correspond to the discrete filter $r(k,l)$. Through vector differentiation, we get the minimizer of the above cost as a solution of the following equation:

$$[\mathbf{D} + \lambda \mathbf{R}]\, \mathbf{c} = \mathbf{b} \tag{5}$$

where $\mathbf{D} = \mathbf{S}^T\mathbf{S}$ and $\mathbf{b} = \mathbf{S}^T\mathbf{f}$

B. Interscale Relation

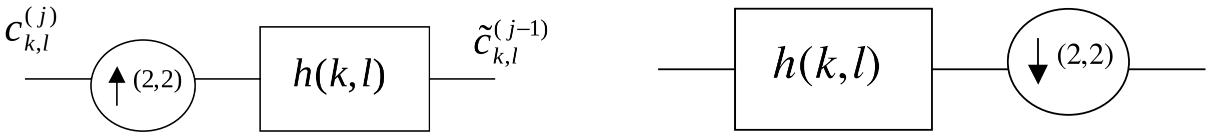
Let us now consider signal reconstructions at different scales. Specifically, let 2^j be the reconstruction grid size (scale j) and

$$S_j(x,y) = \sum_{l=0}^{(N-1)/2^j} \sum_{k=0}^{(N-1)/2^j} c_{k,l}^{(j)} \beta^n(x/2^j - k)\beta^n(y/2^j - l) \tag{6}$$

be the reconstructing function. A key property of the central B-spline of odd degree is the two-scale relation:

$$\beta^n(x/2^j) = \sum h(k)\beta^n(x/2^{j-1} - k) \tag{7}$$

where $h(k) = 2^{-n} \begin{pmatrix} n+1 \\ k \end{pmatrix}$ is the binomial filter. Consider a set of coefficients $\{c_{k,l}^{(j)}\}$ representing a 2D signal at scale j. It is well known from wavelet theory [22], [23] that the same signal can be represented at scale $j-1$ by the coefficients $\{\tilde{c}_{k,l}^{(j-1)}\}$ that are obtained by upsampling $\{c_{k,l}^{(j)}\}$ and filtering with h. The schematic is given in the figure 1(a) where $h(k,l) = h(k)h(l)$. This operation can be represented by a matrix, $\mathbf{U}_j$, that is obtained from the circulant matrix corresponding to the filter $h(k,l)$ after suppression of its odd index columns. The adjoint operation is the downsamping operation represented in the figure 1(b). The equivalent matrix is $\mathbf{U}_j^T$.



(a) Upsampling operation

(b) Downsampling operation

Fig. 1.

The coefficient vector $\mathbf{c}_j = [\cdots c_{k,l}^{(j)} \cdots]$ of the image reconstruction at scale j must satisfy the equation

$$\mathbf{A}_j \mathbf{c}_j = \mathbf{b}_j, \tag{8}$$

with $\mathbf{A}_j = \mathbf{D}_j + \lambda \mathbf{R}_j$ where the matrices $\mathbf{D}_j$ and $\mathbf{R}_j$ are defined as in the previous section with a being replaced by 2^j. The block circulant matrix $\mathbf{R}_j$ is generated from the filter

$$r_j(k,l) = \sum_{q_1+q_2=p} r_{q_1,q_2,j}(k,l)$$

where

$$r_{q_1,q_2,j}(k,l) = \gamma_{q_1,j}(k)\gamma_{q_2,j}(l), \tag{9}$$

$$\gamma_{q,j}(k) = \eta_{q,j}(k2^j), \tag{10}$$

$$\eta_{q,j}(x) = \frac{d^q}{dx^q}\beta^n(-x/2^j) * \frac{d^q}{dx^q}\beta^n(x/2^j) \tag{11}$$

The following theorem gives an important property of the filters $\{r_{q_1,q_2,j}(k,l)\}$.

Theorem 1: The filters $\{r_{q_1,q_2,j}(k,l)\}$ satisfy

$$r_{q_1,q_2,j+1}(k,l) = [h_a(k,l) * r_{q_1,q_2,j}(k,l)]_{\downarrow(2,2)}$$

where $h_a(k,l) = h(k,l) * h(k,l)$ with $h(k,l)$ being the 2D version of the binomial filter in the two-scale relation (7), and where $\downarrow (2,2)$ denotes down sampling by a factor of 2 in each dimension.

Using the above theorem, we prove the following:

Corollary 1: The system matrices at scales j and $j+1$ are related by

$$\mathbf{A}_{j+1} = \mathbf{U}_j^T \mathbf{A}_j \mathbf{U}_j \quad (12)$$

$$\mathbf{b}_{j+1} = \mathbf{U}_j^T \mathbf{b}_j \quad (13)$$

where $\mathbf{U}_j$ is the subsampling matrix described in this section.

Proof: Applying the two-scale relation, we get $\mathbf{S}_{j+1} = \mathbf{S}_j \mathbf{U}_j$. Hence

$$\mathbf{b}_{j+1} = \mathbf{S}_{j+1}^T \mathbf{f} = \mathbf{U}_j^T \mathbf{S}_j^T \mathbf{f} = \mathbf{U}_j^T \mathbf{b}_j$$

and

$$\mathbf{D}_{j+1} = \mathbf{S}_{j+1}{}^T \mathbf{S}_{j+1} = \mathbf{U}_j^T \mathbf{S}_j{}^T \mathbf{S}_j \mathbf{U}_j = \mathbf{U}_j^T \mathbf{D}_j \mathbf{U}_j \quad (14)$$

Now, consider $\mathbf{U}_j^T \mathbf{R}_j \mathbf{U}_j$. The equivalent filter is $[h_a(k,l) * r_j(k,l)]_{\downarrow 2}$. By Theorem 1, it is equal to $r_{j+1}(k,l)$, which is equivalent to writing

$$\mathbf{R}_{j+1} = \mathbf{U}_j^T \mathbf{R}_j \mathbf{U}_j \quad (15)$$

Combining (14) and (15), we get

$$\mathbf{A}_{j+1} = \mathbf{U}_j^T \mathbf{A}_j \mathbf{U}_j.$$

■

IV. Reconstruction Algorithm

We have shown that the reconstruction problem is equivalent to solving a system of linear equations (cf. (5)). A key property is that the present system is sparse and well-conditioned in contrast with the RBF method where the matrix is dense and ill-conditioned. We also have the flexibility to choose the step size a with the guarentee that the solution converges to the analytical one (RBF) when a is sufficiently small. The main difficulty here is that the number of unknowns $\{c_{k,l}\}$ for a given step size a is typically very large. Therefore the choice of the method for solving the linear system becomes crucial in order to do the reconstruction in affordable time. Enabled by the right choice of the basis functions and our theorem on interscale relation, we develop an efficient multiresolution algorithm to achieve fast reconstruction.

A. Multiresolution Strategy

Since the dimension of the linear system is very large and depends upon the resolution of reconstruction, one naturally thinks of multiresolution.

Let a_0 be the required resolution and $N = 2^J + 1$. Then the coarsest resolution will be $2^{J-1}a_0$. The idea is to solve $\mathbf{c}_{J-1}$ exactly and to interpolate the solution to the next finer scale using the two-scale relation for B-spline [13]. This serves as initialization for the computation of $\mathbf{c}_{J-2}$ and so on. At the end of the process, $\mathbf{c}_0$ gives the reconstruction at the required resolution.

Note that the regularization matrix $\mathbf{R}_j$ has a generic form independent of the resolution, while the matrix $\mathbf{D}_j$ is clearly scale-dependent since it is defined by the sample points. Because of the compact support of B-splines, the cost for the direct evaluation of $\mathbf{D}_j$ is the same at each scale and is proportional to the number of data points, which is typically quite large. To cut down on this cost, we evaluate the matrix at the fine scale and use our inter-scale relation to derive the matrices at coarser resolutions. Here the computation is by sparse matrix multiplication which makes the complexity now depend on the resolution level. This enables us to compute all the matrices very efficiently with complexity $\mathcal{O}(N \times N)$; that is, a complexity that is proportional to the number of grid points (i.e., the number of unknown B-spline coefficients).

We handle the computation of the coefficient vectors at each scale from their initializations using the multigrid iteration described below.

B. Multigrid Iteration

A multigrid iteration is obtained by using classical iterators as the building blocks. We first describe the classical iterative scheme and then the multigrid iteration. A classical iterative scheme gives a way to get a refined estimate of the solution from a given estimate. Let $\mathbf{c}_j^{(k)}$ be the current estimate of the solution for level j. A refinement step is represented as follows:

$$\mathbf{c}_j^{(k+1)} = \mathbf{c}_j^{(k)} + \omega \hat{\mathbf{A}}_j^{-1}(\mathbf{b}_j - \mathbf{A}_j \mathbf{c}_j^{(k)}) \tag{16}$$

where $\hat{\mathbf{A}}_j^{-1}$ denotes an approximate inverse and ω is a damping factor. For the estimate $\mathbf{c}_j^{(k)}$, the error $\mathbf{e}_j$ and the residue $\mathbf{r}_j$ are given by

$$\mathbf{e}_j = \mathbf{c}_j - \mathbf{c}_j^{(k)}$$

$$\mathbf{r}_j = \mathbf{b}_j - \mathbf{A}_j \mathbf{c}_j^{(k)}$$

When $\hat{\mathbf{A}}_j$ in (16) is the diagonal of $\mathbf{A}_j$, then the iterator is called damped-Jacobi; if instead it is the lower triangular part of $\mathbf{A}_j$, the algorithm yields the so-called Gauss-Seidel iterator. Both implementations have the same computational complexity. See [24] for a comprehensive treatment of such numerical schemes.

It is important here to note that such iterators have a smoothing effect on the error. The larger k, the smoother the error, and for sufficiently large k, there will not be significant improvement in the error anymore. Since the error is smooth after a few relaxations (iterations), it can be well represented at lower resolution. In other words, one can try to compute the error at a coarser scale.

To do this we first consider the residual equation:

$$\mathbf{A}_j \mathbf{e}_j = \mathbf{r}_j$$

Then the computation of error is formulated as

$$\mathbf{A}_{j+1} \tilde{\mathbf{c}}_{j+1} = \tilde{\mathbf{b}}_{j+1}$$

where $\tilde{\mathbf{b}}_{j+1}$ is obtained by filtering and downsampling $\mathbf{r}_j$; that is,

$$\tilde{\mathbf{b}}_{j+1} = \mathbf{U}_j^T \mathbf{r}_j$$

Here $\tilde{\mathbf{c}}_{j+1}$ is the lower dimensional version of the error to be computed. $\tilde{\mathbf{c}}_{j+1}$ is computed by the same iterative scheme with zero initialization. Afterwards one gets back the error $\mathbf{e}_j = \mathbf{U}_j \tilde{\mathbf{c}}_{j+1}$, which is used to correct $\mathbf{c}_j^{(k)}$. This is called coarse-grid correction.

The advantage is twofold. First, it is computationally efficient to iterate in lower dimension. Second, a smooth error becomes somewhat bumpier at a coarser resolution. Hence, relaxation on the coarser grid further reduces the error. However, the coarse grid correction is not perfect and also introduce some amount of non-smooth error. Hence it is customary to do a few relaxations after coarse-grid correction. The overall scheme is computationally more effective than relaxation at the finest grid. However, when the dimension of the system at the coarser grid is large, relaxation will also stall there. Since the problem here is exactly the same as the original one, one can think of applying the same three-step procedure recursively. This is called a multigrid V-cycle [25], [24]. Figure 2 gives the schematic of multigrid V-cycle. Here $\hat{\mathbf{c}}_j$ is the initialization for V-cycle that is obtained by interpolating $\mathbf{c}_{j+1}$ (solution for the level $j + 1$) using the B-spline two-scale filter.

Whenever applicable, multigrid algorithms are extremely efficient; in fact, they are among the best numerical methods available. There are also some general convergence proofs [24] that are directly applicable to our case because of the existence of the interscale relation (12) that relates the systems of equations at successive scales.

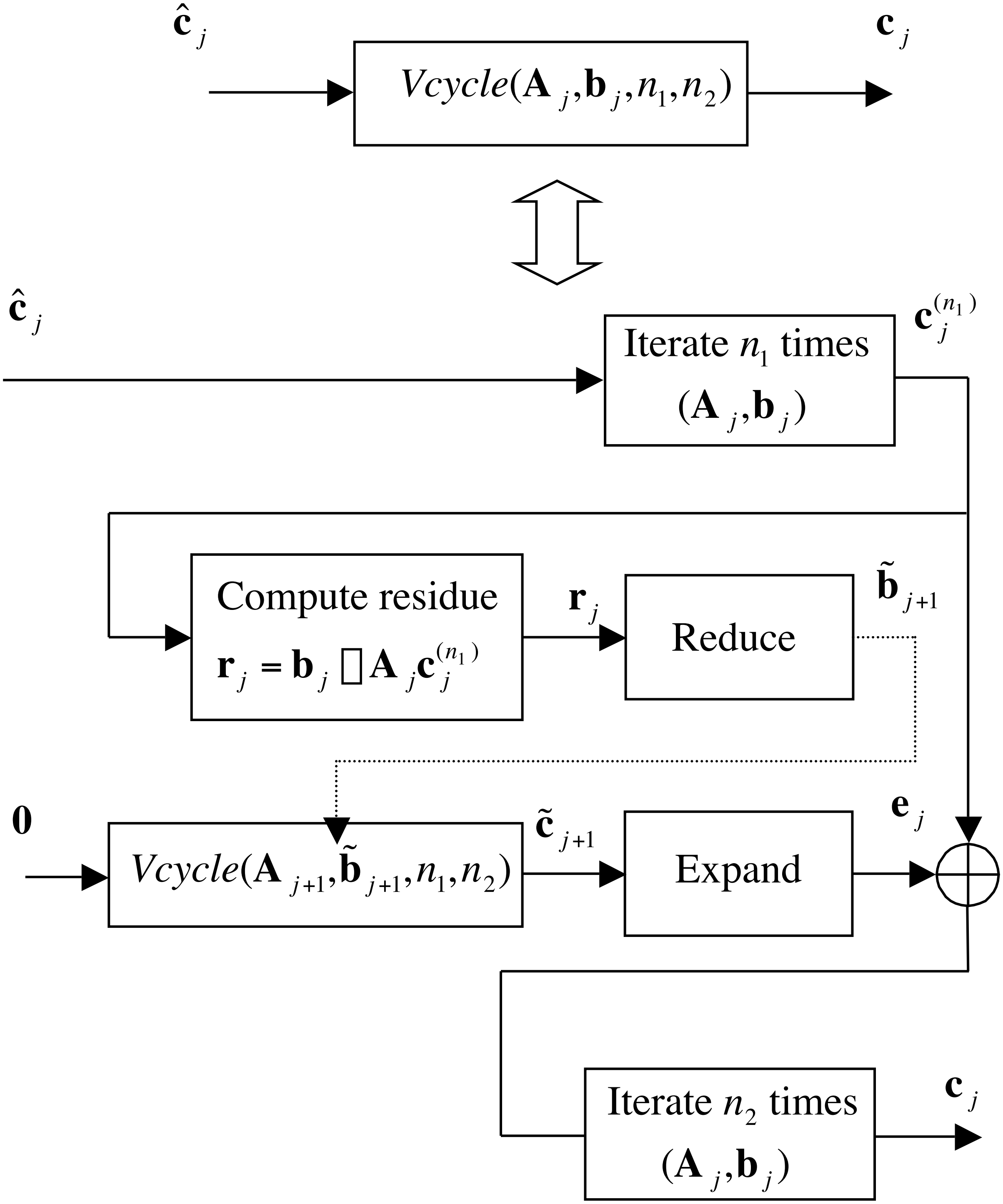

$\hat{\mathbf{c}}_j$
$Vcycle(\mathbf{A}_j, \mathbf{b}_j, n_1, n_2)$
$\mathbf{c}_j$
$\hat{\mathbf{c}}_j$
Iterate n_1 times
$(\mathbf{A}_j, \mathbf{b}_j)$
$\mathbf{c}_j^{(n_1)}$
Compute residue
$\mathbf{r}_j = \mathbf{b}_j \square \mathbf{A}_j \mathbf{c}_j^{(n_1)}$
$\mathbf{r}_j$
Reduce
$\tilde{\mathbf{b}}_{j+1}$
$\mathbf{0}$
$Vcycle(\mathbf{A}_{j+1}, \tilde{\mathbf{b}}_{j+1}, n_1, n_2)$
$\tilde{\mathbf{c}}_{j+1}$
Expand
$\mathbf{e}_j$
Iterate n_2 times
$(\mathbf{A}_j, \mathbf{b}_j)$
$\mathbf{c}_j$

Fig. 2. Schematic of multigrid V-cycle

The parameters that affect the computational complexity of a multigrid V-cycle are the number of iterations before and after the coarse-grid correction, n_1 and n_2 respectively.

C. Implementation Issues

The forgoing discussions cover the main philosophy of our method. However, to get the full benefit of the multiresolution/multigrid algorithm described above and to make the algorithm almost real-time, efficient implementation of the building blocks in the algorithm is crucial. The implementation should meet stringent constraints on computational complexity and storage requirements.

To get a flavor of what is involved, let us consider the structure of the matrix $\mathbf{A}_0$ or $\mathbf{D}_0$ for cubic reconstruction. The matrix is block-band diagonal of width seven, and each block is band-diagonal of width seven. The whole matrix is symmetric and each block in the matrix is symmetric as well. In our implementation, we exploit this structure to reduce storage requirements. The non-redundant elements are stored in a sparse format without explicit indexing. The basic matrix iterators and modules (especially Gauss-Seidel iteration) are coded specifically for this sparse format.

The sample matrix $\mathbf{D}_0$ is computed from the sample points directly into the above mentioned sparse format. The matrices $\mathbf{A}_1, \ldots, \mathbf{A}_{J-1}$ are also precomputed using relation (12) directly into the sparse format exploiting the special structure of $\mathbf{U}_j$. In this way, we are able to compute the matrices $\mathbf{A}_1, \ldots, \mathbf{A}_{J-1}$ with around $20(N \times N)$ multiplications in the case of reconstruction with linear splines.

V. Experimental Results

For our experiments, we consider three kinds of scenarios: (i) reconstruction from a subset of samples of a given digital image; (ii) reconstruction after a geometric transformation of a given image; and (iii) reconstruction from sparse samples of a synthetic phantom. In the first case, the inputs sample points are obtained by randomly choosing the pixels or choosing the pixels along some selected contours. In the second case, the input sample points are obtained by a geometric transformation of a uniform grid points. In the last case, we choose samples from a synthetic phantom along some lines. We provide the samples to our algorithm in a list format $\{x_i, y_i, f_i\}$.

In all our experiments we consider two settings: (i) linear spline reconstruction with Duchon's first semi-norm as the regularization; (ii) cubic spline reconstruction with Duchon's second semi-norm as the regularization. Unless stated otherwise, we adjusted λ empirically for the best visual results. We observed that the most favorable value of λ is image specific and typically proportional to the noise variance when the data is corrupted by noise.

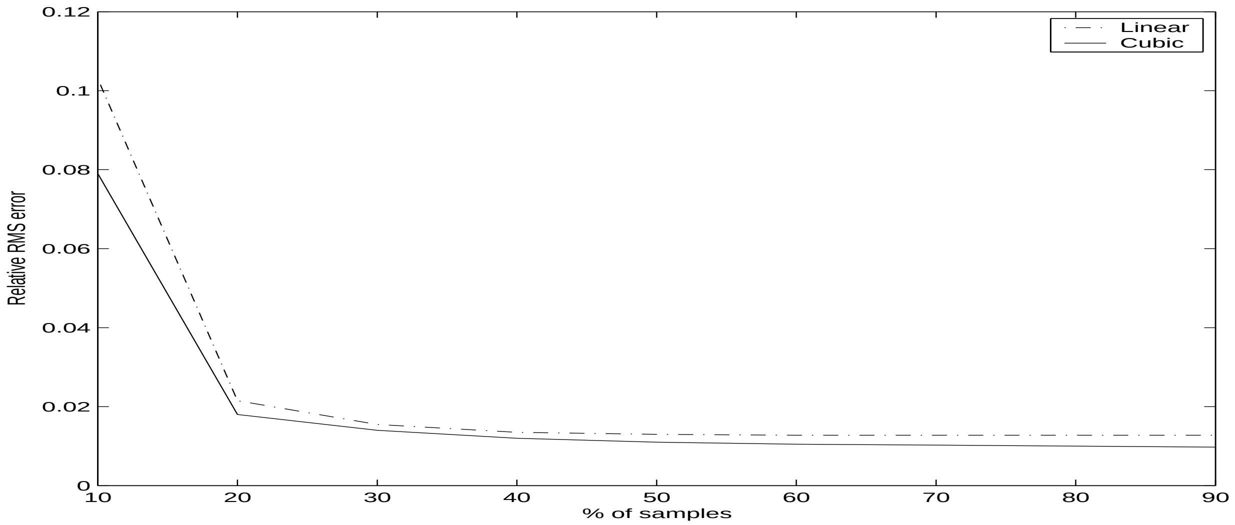


Fig. 3. Reconstruction error for various degrees of subsampling

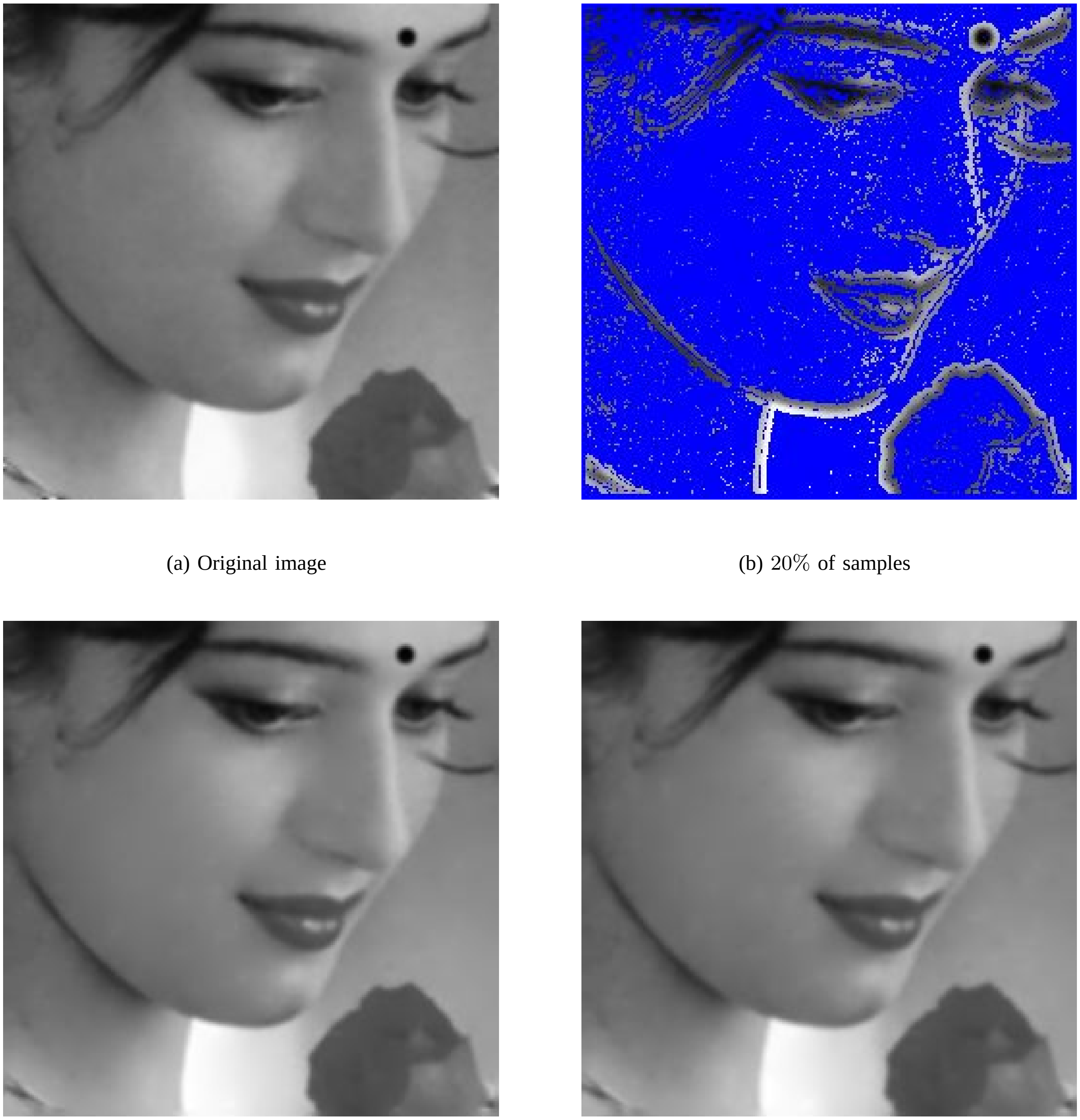
A. Reconstruction from Incomplete Data

For the experiments in this category, we define the reconstruction error as $e_r = \frac{||\mathbf{I}_o - \mathbf{I}_r||}{||\mathbf{I}_o||}$, where $\mathbf{I}_o$ is the original image and $\mathbf{I}_r$ is the reconstructed image. Note that this will be an underestimate of the performance of the algorithm—especially when the initial data is sampled arbitrarily—since it does not correspond to the objective function that we minimize in our formulation of the problem.

We first applied the algorithm on a face image. We sub-sampled the image by applying a binary mask obtained by thresholding the Laplacian of the image. Figure 3 gives the reconstruction error for both linear and cubic reconstructions for various numbers of input samples. Cubic spline reconstruction gives lower reconstruction error as one would expect. Figure 4 compares the reconstructed images from 20% of the samples. This example demonstrates that our algorithm is able to handle both large and small gaps simultaneously in an efficient way.

In figure 5, we give lower resolution reconstructions from the same set of samples. One can clearly see the artifacts; they are somewhat reduced in the case of cubic spline reconstruction.

Figure 6 displays reconstruction results for an MRI brain image. The reconstruction was from 30% of samples that were retained in the same way as in the previous experiment. In this case, the improvement of the cubic spline reconstruction over the linear one is more visible even though both images have same reconstruction error. The image from linear spline reconstruction shows more bright spots than that of cubic spline reconstruction. This is due to the fact that the contours are less prominent than those of



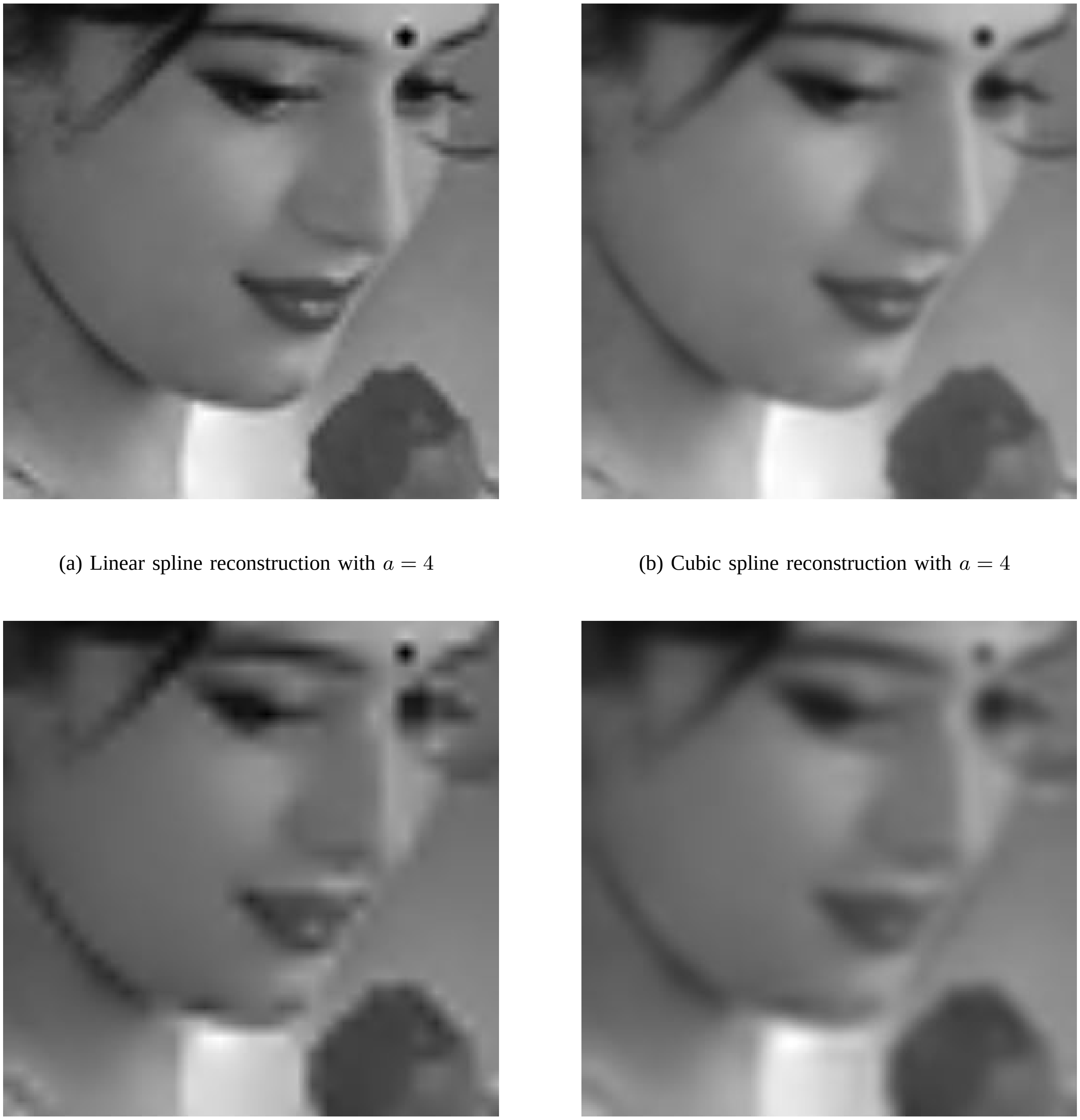
(a) Original image

(b) 20% of samples

(c) Reconstructed image using linear splines

(d) Reconstructed image using cubic splines

Fig. 4. Reconstruction from high Laplacian points. Sampling density: 20%



(a) Linear spline reconstruction with $a = 4$

(b) Cubic spline reconstruction with $a = 4$

(c) Linear spline reconstruction with $a = 8$

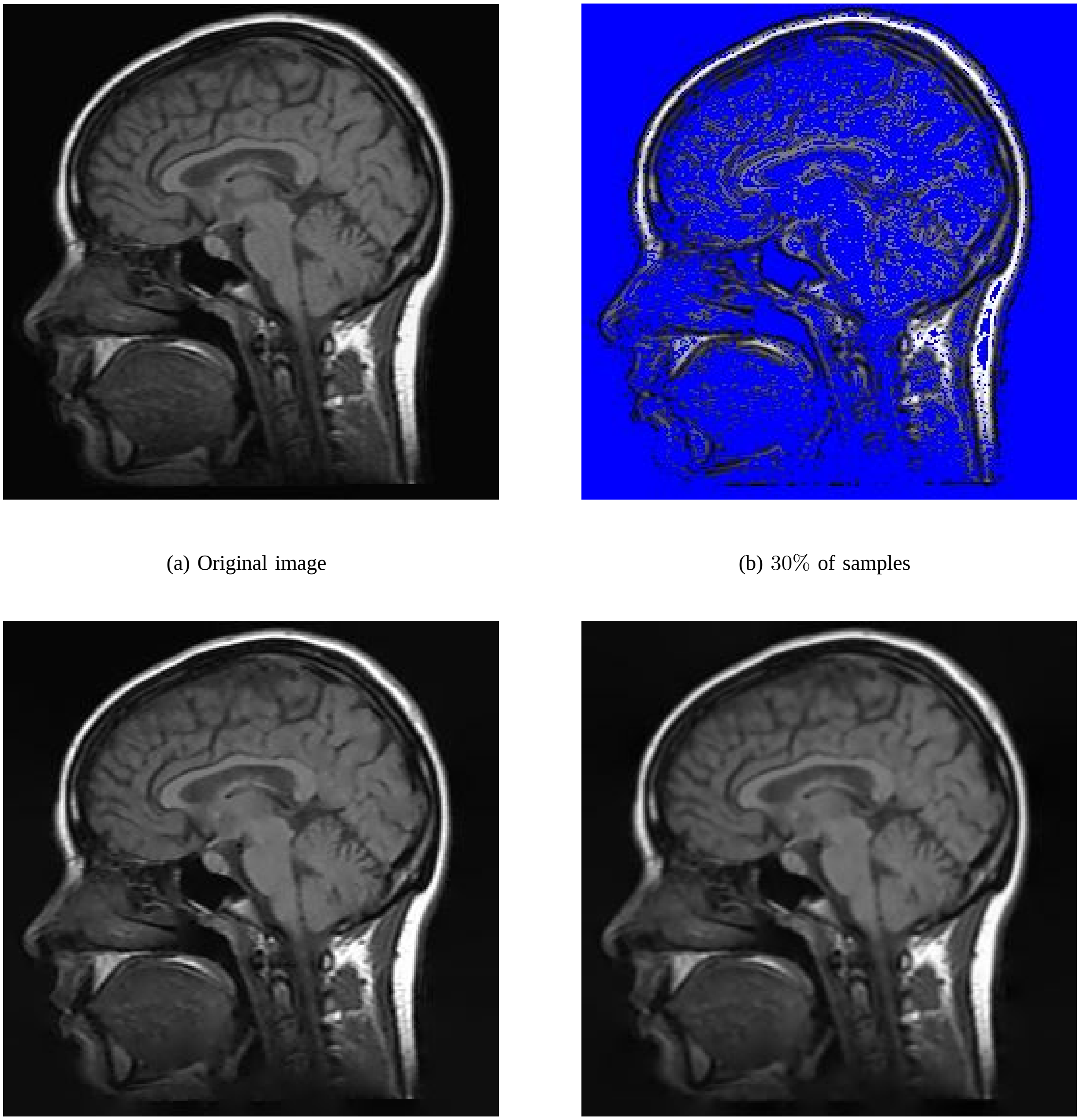
(d) Cubic spline reconstruction with $a = 8$

Fig. 5. Lower resolution reconstruction

the previous image and that regularization with second order semi-norm is more suitable when there are discontinuities in the contours. Figure 7 shows the images for a larger smoothing factor. In this case as well, both cubic spline and linear spline reconstructions have the same reconstruction error. However, cubic spline reconstruction is smoother than the linear one with less artifacts (bright spots). Figure 8 gives lower resolution reconstructions of the MRI image from the same set of samples as in the figure 6. One can clearly see the artifacts in this case too.

The above examples demonstrate the ability of the algorithm to reconstruct data when there are large sampling gaps, something that is typically not possible with the reconstruction algorithms for bandlimitted functions mentioned in the introduction [2], [3]. However, we must admit that our algorithm will fill the sampling gaps smoothly by extrapolating the available information (samples). For this reason, it cannot correctly recover image parts for which the contour or texture information has been lost. Note that this effect may also be used to our advantage for suppressing unwanted objects or features in images, as demonstrated in the next example. Here, we started with the contour map of 4(b) and applied a coarse binary mask to suppress all samples in the regions of the round spot on the face and the rose in the lower left corner. The corresponding reconstruction is given in Figure 9. It is still looking realistic, even though the selected objects have entirely disappeared. This is due to the regularization term that smoothly extrapolates the missing pixel values from the surroundings.

The proposed algorithm is obviously also applicable to the case of random sub-sampling, which is the context in which reconstruction algorithm for band-limited functions are usually tested [2], [3]. An example of such a reconstruction is given in Figure 10. As one would expect, the quality is inferior to that obtained with the reconstruction from high-Laplacian points using the same percentage of samples ($e_r = 0.0204$ vs. 0.0145). We have verified experimentally that the reconstruction errors obtained under these conditions are essentially equivalent to those of alternative techniques for bandlimited functions, provided that the reconstruction parameters are matched (same reconstruction density and λ small). This behavior is consistent with the property that a cubic spline is a very good approximation of a bandlimited function [26], and that our algorithms provides the least squares solution as λ tends to zero (data term only). A very significant advantage of our method is its computational speed: this is due to two important ingredients that are specific to our formulation: (1) the use of compactly supported basis functions, and (2) a most efficient multi-grid algorithm that can solve the linear system of equations with a complexity that is essentially proportional to the number of reconstructed samples.



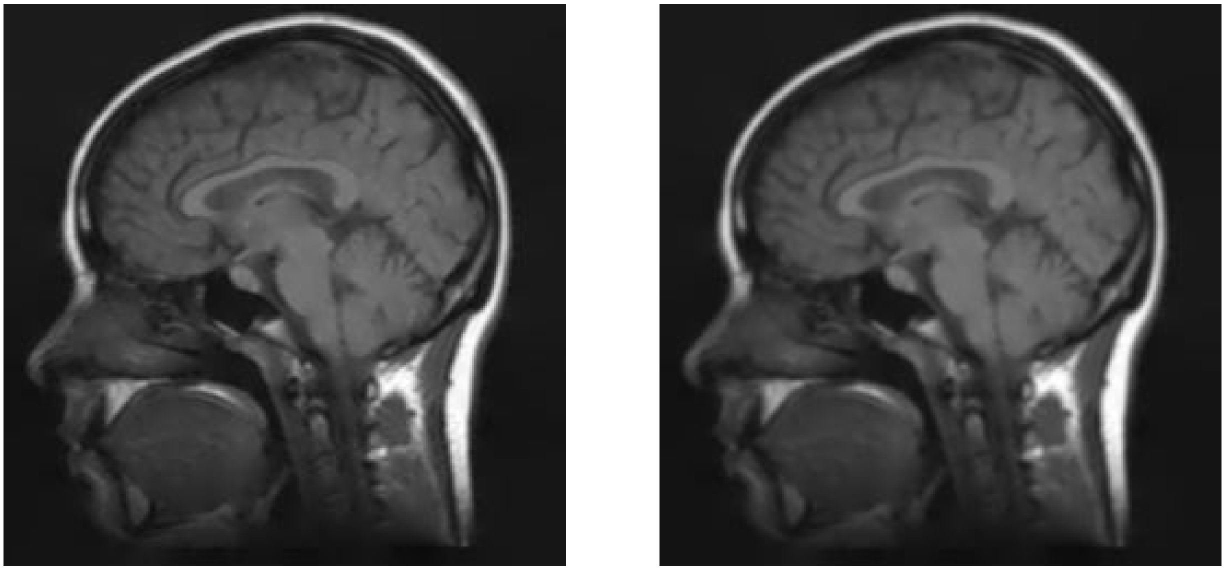
(a) Original image

(b) 30% of samples

(c) Reconstructed image using linear spline. $\lambda = 1.5848 \times 10^{-3}$

(d) Reconstructed image using cubic spline. $\lambda = 1.5848 \times 10^{-3}$

Fig. 6. Reconstruction from high Laplacian points. Sampling density: 30%



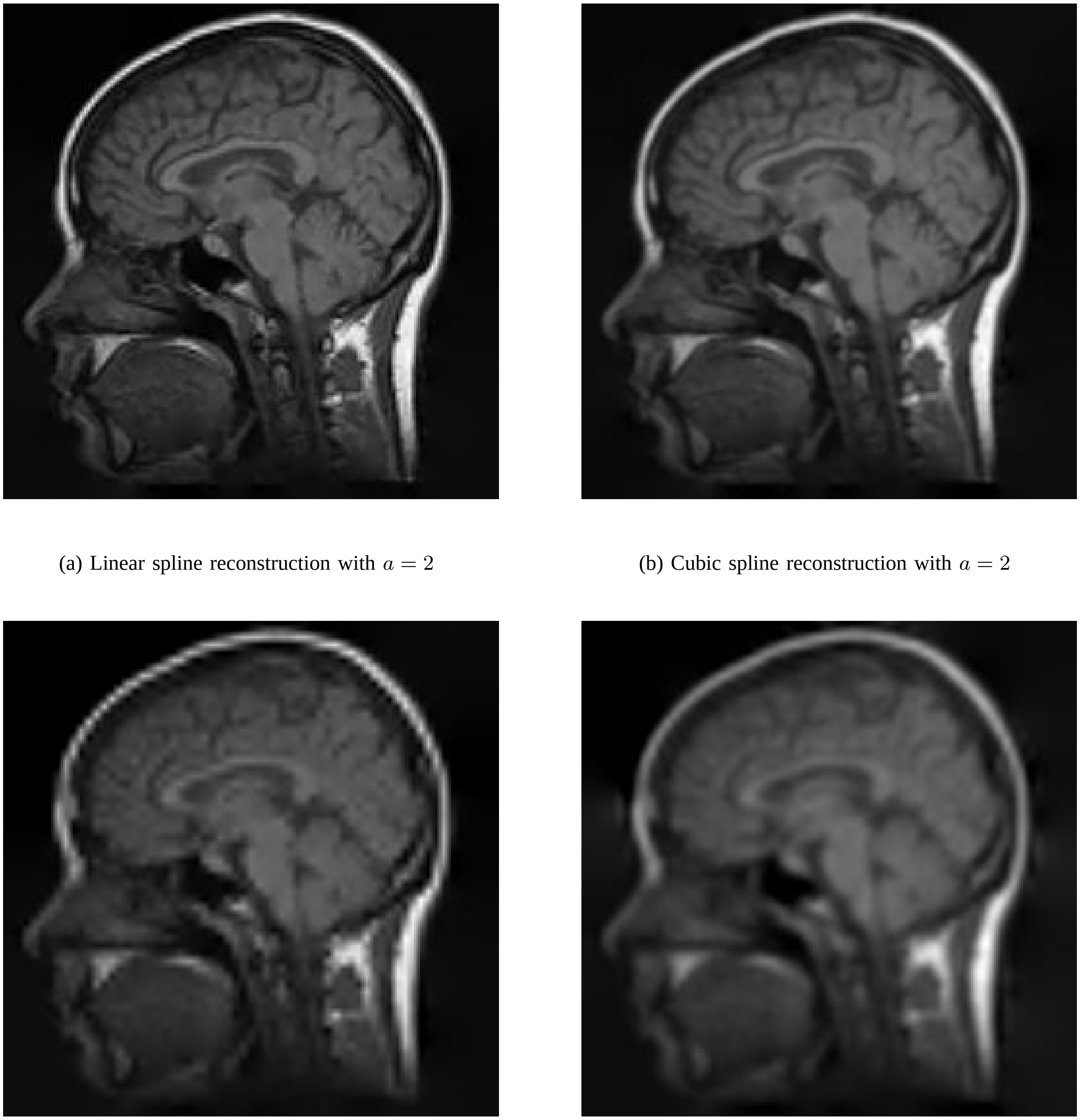
(a) Reconstructed image using linear spline. $\lambda = 7.9430 \times 10^{-1}$

(b) Reconstructed image using cubic spline. $\lambda = 7.9430 \times 10^{-1}$

Fig. 7. Effect of λ

B. Reconstruction with geometric transformation (texture mapping)

Texture mapping typically refers to the process of geometrically transforming a given source image or pattern in order to simulate its mapping onto a 3D surface. There are potentially two ways to do this: (i) applying the inverse transformation for each pixel position in the target image to get the interpolated value from the source image; (ii) applying the transformation of each source pixel and using our non-uniform reconstruction method to get the target image. The second method has the clear advantage that it uses the information present in the source image completely, whereas, there might be some loss of information with the first approach (unused pixels in the source image). Our method will give the least squares fit in the regions where the input samples (transformed source pixels) outnumber the reconstruction grid points (target pixels). This reduces reconstruction artifacts. Figure 11 gives an example of texture mapping generated using our algorithm, where the lena image is mapped onto a cylinder. The key feature of our technique is that there are no aliasing artifacts and that the sharpness of the pictorial information is essentially preserved when λ is small.



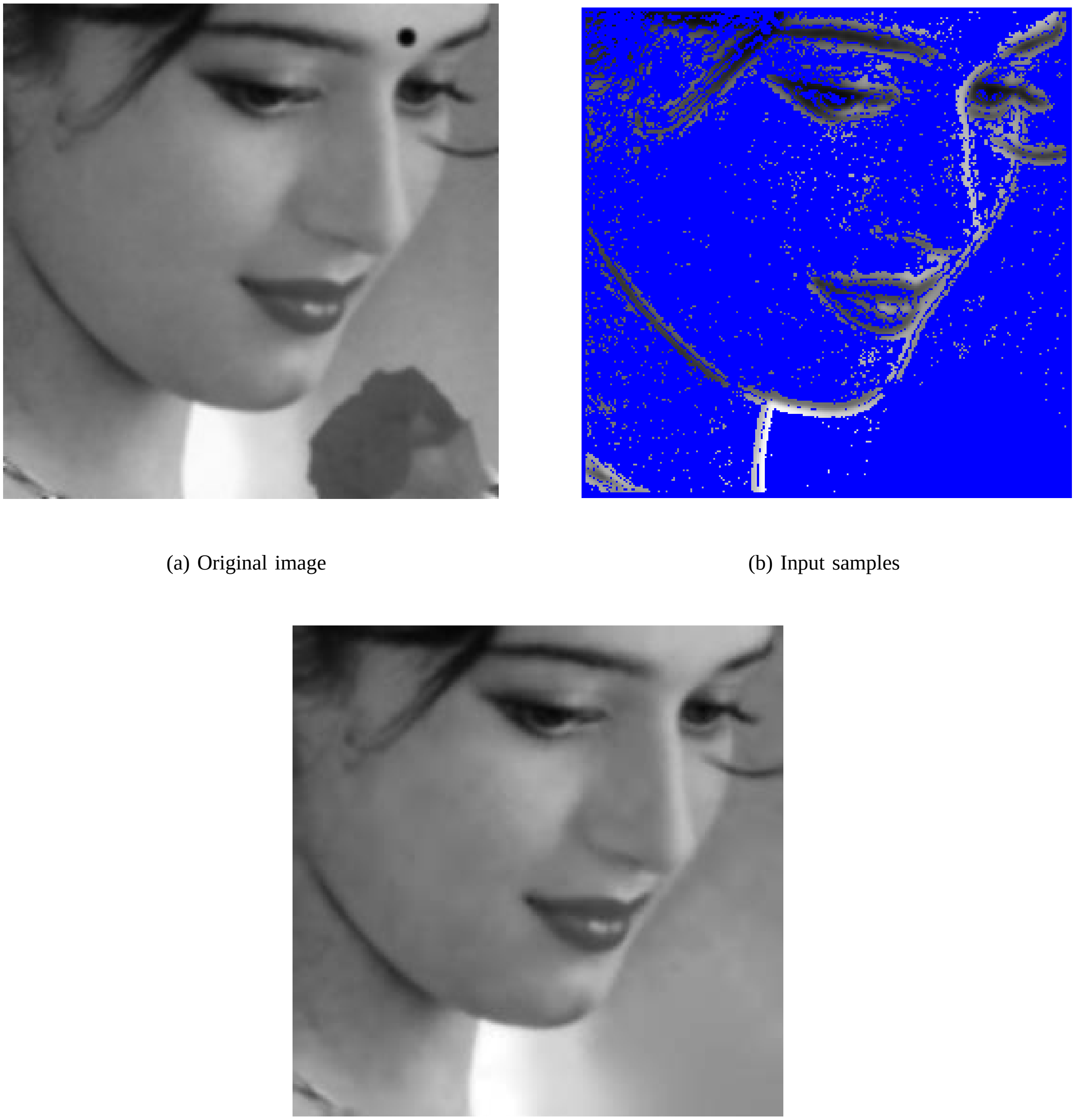
(a) Linear spline reconstruction with $a = 2$

(b) Cubic spline reconstruction with $a = 2$

(c) Linear spline reconstruction with $a = 4$

(d) Cubic spline reconstruction with $a = 4$

Fig. 8. Lower resolution reconstruction

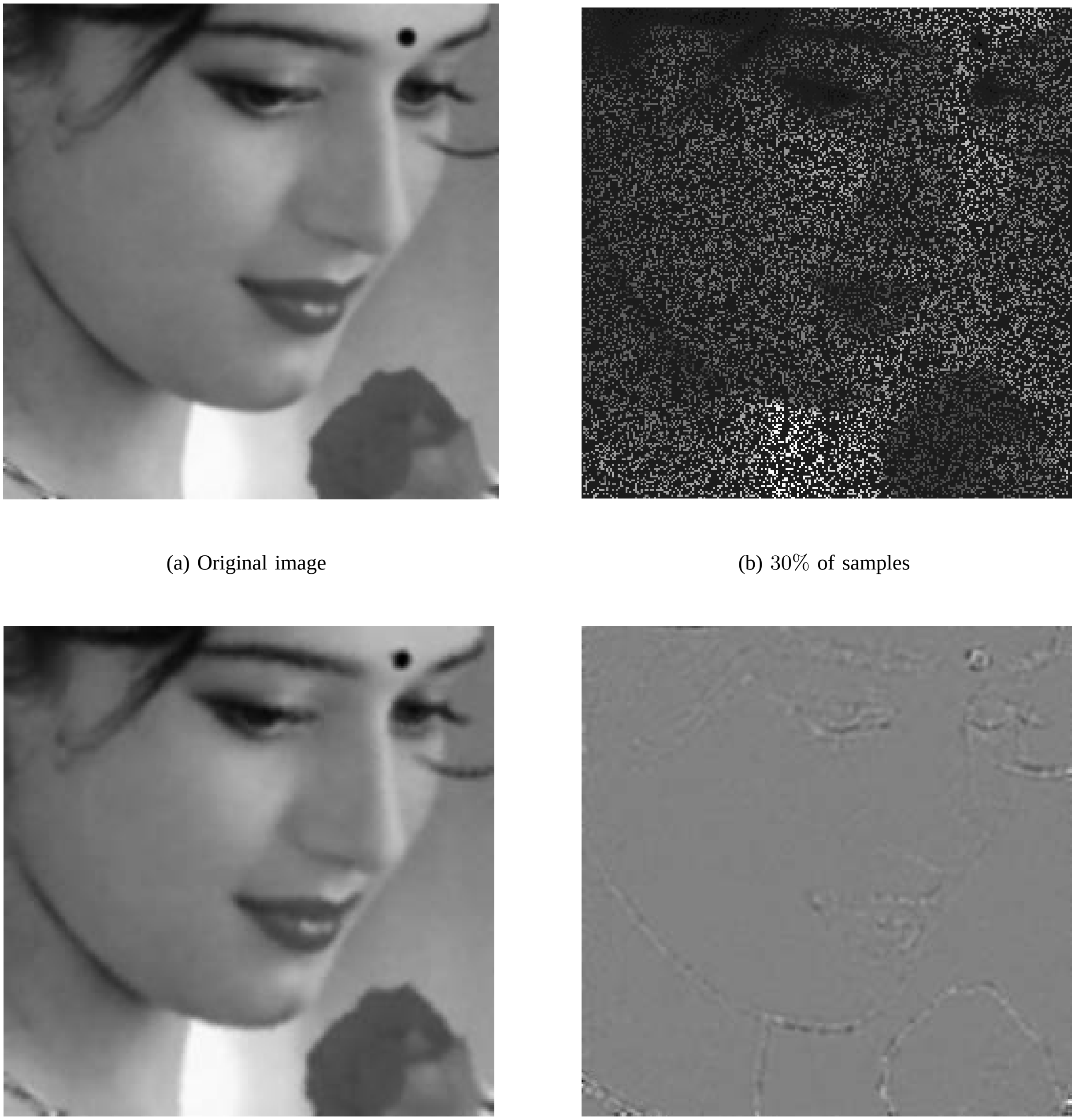


(a) Original image

(b) Input samples

(c) Reconstructed image

Fig. 9. Reconstruction from partial contours



(a) Original image

(b) 30% of samples

(c) Reconstructed image. $\lambda = 3.086 \times 10^{-1}$. $e_r = 0.0204$

(d) Rescaled error image

Fig. 10. Reconstruction from random samples

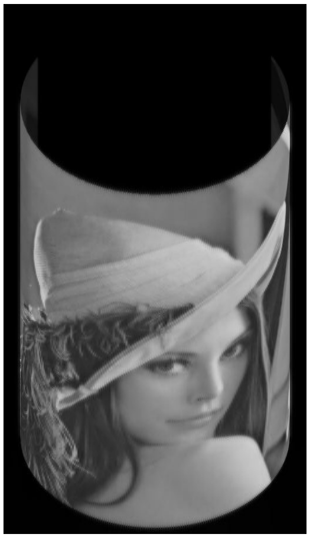
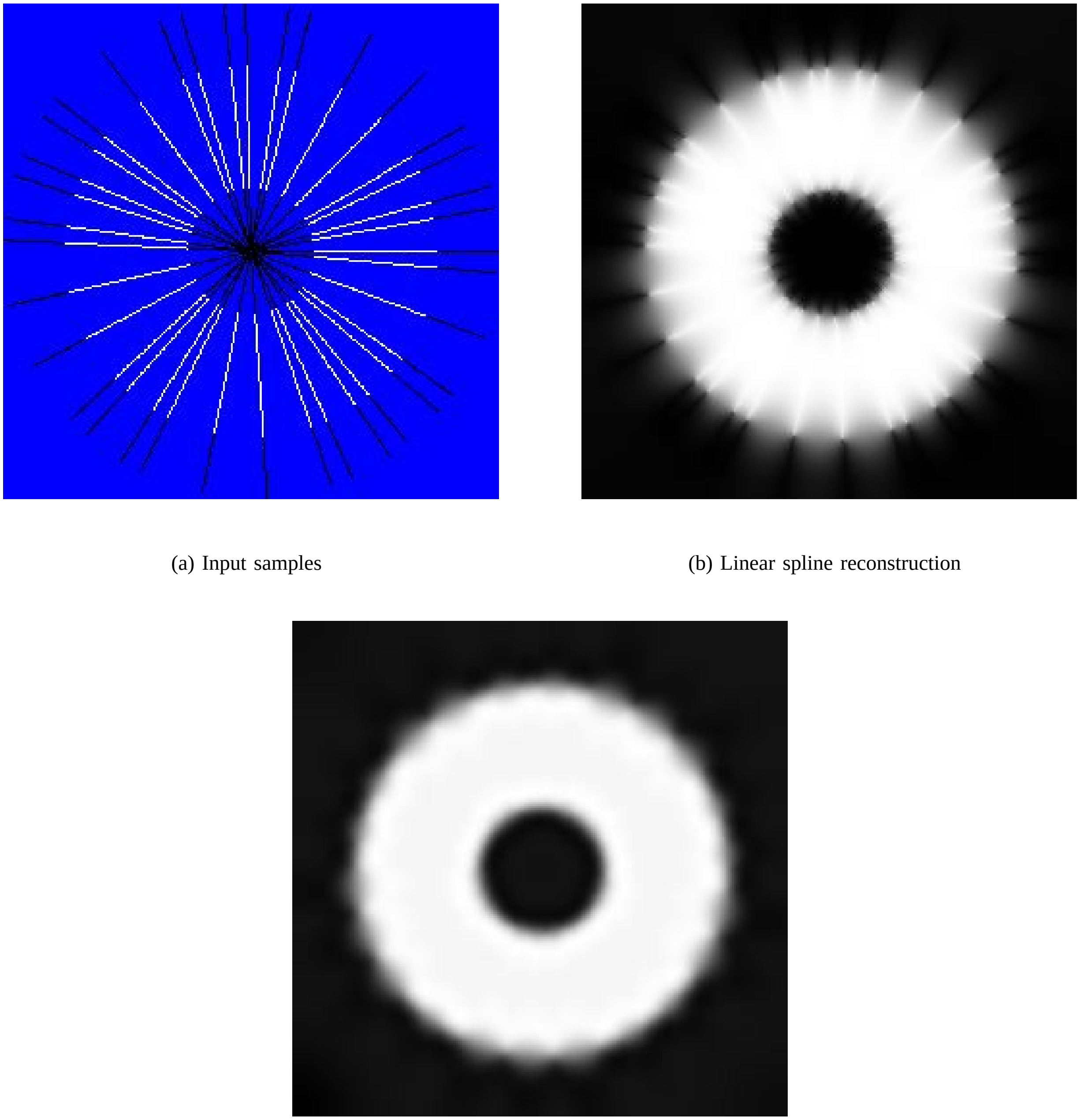


Fig. 11. Example of texture mapping: reconstruction with geometric transformation

C. Phantom reconstruction

Our next test image takes the value 255 inside a circular ring and zero outside. We sample this phantom along some radial lines. The data is sampled non-uniformly along the angular dimension and uniformly in the radial direction. The reconstruction is challenging because the samples are sparse and the boundaries are lost. The results are given in the figure 12. One can clearly see that the cubic spline reconstruction gives better reconstruction of boundaries. This is due to the fact that the underlying radial basis function for the first order smoothness semi-norm is less suitable for recovering lost boundaries. In fact, the analytical RBF method with $p = 1$ is not numerically stable in 2D since the radial basis function $\log r$ is unbounded at the sample locations.

In all the cases above, in order to make the comparison meaningful, we set up the iterations such that the residue of the linear system goes to the machine precision. To achieve this, it was required to have $n_1 = n_2 = 32$ for a reconstruction in a 256×256 grid; it took 2 seconds and 0.8 seconds on a 1.8 GHz Macintosh G5 system for cubic and linear splines, respectively. However, we were able get a visually acceptable reconstruction with $n_1 = n_2 = 2$ that took 0.8 seconds and 0.2 seconds using cubic splines and linear splines, respectively. Also, we should mention that the number and the distribution of input samples have negligible effect on the speed of the algorithm. The speed is primarily determined by the size of the reconstruction grid. This factor makes our method quite attractive in the noisy situations where one needs to have more input samples than the number of reconstruction grid points.



(a) Input samples

(b) Linear spline reconstruction

(c) Cubic spline reconstruction

Fig. 12. Reconstruction of boundary.

The present algorithm is very general; it includes previously published spline algorithms as particular cases. For instance, consider the case when the input data is a standard digital image. When $a = 1$, the algorithm gives the smoothing spline approximation of the image and is functionally equivalent to filtering algorithm described in [27]. In fact, because of the multigrid implementation, the computational complexity of our algorithm is in the same order as that of the FFT algorithm given in [27] with the advantage that the present scheme works for arbitrary samples. When $\lambda \to 0$, it gives the standard B-spline interpolation [13]. One can also generate least squares pyramid, by choosing $\lambda \to 0$ and $a = 2^i$. By choosing an appropriate reconstruction grid size and a diagonal affine transformation, one can achieve least squares rescaling [28], as well as more general types of geometric transformations as illustrated in the figure 11.

VI. Conclusion

We developed a new method for regularized image reconstruction from arbitrarily spaced samples. We chose to reconstruct a continuously-defined function that is a uniform spline and selected a regularization term within Duchon's class of smoothness semi-norms. We interpreted our scheme as a way to discretize the radial basis function (RBF) method which gives the optimal analytical solution of the approximation problem in the continuous domain. The key point of our scheme is that it uses basis functions (B-splines) that are well conditioned; this makes our approach much more stable numerically and computationally advantageous than the classical RBF method. We provided a multiresolution formulation that allowed us to accelerate the reconstruction by way of a fast multigrid algorithm; the key ingredient here is an algebraic relation that links the reconstruction equations at different resolutions. Our algorithm has a number of advantageous features that should make it attractive for practical applications:

- The user can select the resolution a of the reconstruction grid. This parameter controles the trade-off between computational complexity and reconstruction accuracy. For a sufficiently small, the reconstruction converges to the solution of the RBF problem.
- The algorithm allows for a trade-off between smoothness and closeness of fit through the adjustment of the regularization parameter λ. When $\lambda \to 0$, it provides an interpolation of the data.
- The algorithm has the ability to handle arbitrary sample locations.
- It has a complexity that depends primarily on the number of reconstruction grid points. There is essentially no dependence on the location and the number of samples, which is unusual in this type of problem.

- It provides a stable reconstruction in all cases. In particular, this means that the method has the ability to handle large data gaps.

APPENDIX

Proof of theorem

Define

$$\beta_j^n(x) = \beta^n(x/2^j)$$

$$\alpha_{q,j}(x) = \frac{d^q}{dx^q}\beta_j^n(x)$$

Let $h(k)$ be the two-scale filter. Since

$$\beta_{j+1}^n(x) = \sum_k h(k)\beta_j^n(x - k2^j)$$

We get

$$\alpha_{q,j+1}(x) = \sum_k h(k)\alpha_{q,j}(x - k2^j) \tag{17}$$

From (11), $\eta_{q,j}(x) = \alpha_{q,j}(x) * \alpha_{q,j}(-x)$. Hence from (17) we get

$$\eta_{q,j+1}(x) = \sum_k h_a(k)\eta_{q,j}(x - k2^j) \tag{18}$$

where $h_a(k) = h(k) * h(k)$. Substituting $x = m2^{j+1}$ yields

$$\eta_{q,j+1}(m2^{j+1}) = \sum_k h_a(k)\eta_{q,j}(m2^{j+1} - k2^j) \tag{19}$$

From (10), the above equation gives

$$\gamma_{q,j+1}(m) = \sum_k h_a(k)\gamma_{q,j}(2m - k) \tag{20}$$

Hence $\gamma_{q,j+1}(k) = [\gamma_{q,j}(k) * h_a(k)]_{\downarrow 2}$. Now, from (9), we get

$$r_{q_1,q_2,j}(k,l) = \gamma_{q_1,j}(k)\gamma_{q_2,j}(l)$$

Due to separability

$$r_{q_1,q_2,j+1}(k,l) = [h_a(k,l) * r_{q_1,q_2,j}(k,l)]_{\downarrow(2,2)} \tag{21}$$

where $h_a(k,l) = h(k,l) * h(k,l)$. Note that the proof also carries over to any refinable function other than splines.

REFERENCES

[1] C. Cenker, H. Feichtinger, and H. Steier, “Fast iterative and non-iterative reconstruction of band-limited functions from irregular sampling values,” in *ICASSP*, 1991, pp. 1773–1776.

[2] T. Strohmer, “Computationally attractive reconstruction of bandlimited images from irregular samples,” *IEEE Transactions on Image Processing*, vol. 6, pp. 540–548, Apr. 1997.

[3] ——, “Fast scattered data approximation with neumann and other boundary conditions,” *Linear Algebra and Application*, To appear.

[4] Y. Eldar and A. Oppenheim, “Filterbank reconstruction of bandlimited signals from nonuniform and generalized samples,” *IEEE Transactions on Signal Processing*, vol. 48, No. 10, pp. 2864–2875, Oct. 2000.

[5] A. Aldroubi and K. Grochenig, “Nonuniform sampling and reconstruction in shift invariant spaces,” *SIAM Review*, vol. 43, pp. 585–620, 2001.

[6] J. Duchon, “Splines minimizing rotation-invariant semi-norms in sobolev spaces,” *Constructive Theory of Functions of Several Variables*, pp. 85–100, 1977.

[7] C. A. Micchelli, “Interpolation of scattered data: distance matrices and conditionally positive definite functions,” *Constructive Approximation*, vol. 2, pp. 11–22, 1986.

[8] J. Meinguet, “Multivariate interpolation at arbitrary points made simple,” *Zeitschrift fur Angewandte Mathematik und Physik*, vol. 30, pp. 292–304, 1979.

[9] F. L. Bookstein, *Morphometric Tools for Landmark Data*. Cambridge University Press, 1991.

[10] W. R. Fright, J. C. Carr, and R. K. Beatson, “Surface interpolation with radial basis function for medical imaging,” *IEEE Transactions on Medical Imaging*, vol. 16, No. 1, pp. 96–107, 1997.

[11] R. Beatson, W. A. Light, and S. Billings, “Fast solutions of radial basis function interpolation equations: Domain decomposistion methods,” *SIAM Journal of Scientific Computing*, vol. 22, No. 5, pp. 1717–1740, 2000.

[12] C. de Boor, *A Practical Guide to Splines*. Springer-Verlag, 1978.

[13] M. Unser, “Splines: A perfect fit for signal and image processing,” *IEEE Signal Processing Magazine*, vol. 16, No. 2, pp. 22–38, Nov. 1999.

[14] ——, “Multigrid adaptive image processing,” *IEEE International Conference on Image Processing*, vol. 1, pp. 49–52, Oct. 1995.

[15] M.-H. Yaou and W.-T. Chang, “Wavelet transform in scattering data interpolation,” *Electronic Letters*, vol. 29, No. 21, pp. 1835–1837, Oct. 1993.

[16] R. Szeliski, “Fast surface interpolation using hierarchical basis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, No. 6, pp. 513–528, June 1990.

[17] G. Wabha, *Spline Models for Observational Data*. SIAM, 1990.

[18] R. L. Eubank, *Nonparametric regression and spline smoothing*. Marcel Dekker, Inc, 1999.

[19] I. J. Schoenberg, “Spline functions and the problem of graduation,” *Proc. Nat. Acad. Sci.*, vol. 52, pp. 652–654, 1964.

[20] M. Unser, “Approximation power of biorthogonal wavelet expansions,” *IEEE Transactions on Signal Processing*, vol. 44, pp. 519–527, 1996.

[21] T. Blu, P. Thevenaz, and M. Unser, “Moms: Maximal-order interpolation of minimal support,” *IEEE Transactions on Image Processing*, vol. 10, pp. 1069–1080, 2001.

[22] G. Strang and T. Nguyen, *Wavelets and filter banks*. Wellesley-Cambridge Press, 1997.

[23] M. Vetterli and J. Kovacevic, *Wavelets and subband coding*. Prentice Hall, 1995.

[24] W. Hakbusch, *Iterative Solutions of Large Sparse Systems of Equations*. Springer-Verlag, 1994.

[25] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*. SIAM, 2000.

[26] A. Aldroubi, M. Unser, and M. Eden, “Cardinal spline filters: Stability and convergence to the ideal sinc interpolator,” *Signal Processing*, vol. 28, pp. 127–138, 1992.

[27] M. Berman, “Automated smoothing of image and other regularly spaced data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 460–468, 1994.

[28] A. Munoz, T. Blu, and M. Unser, “Least-squares image resizing using finite differences,” *IEEE Transactions on Image Processing*, vol. 10, pp. 1365–1378, 2001.