ENHANCING LOCAL - TRANSMITTING LESS - IMPROVING GLOBAL

Benjamín Béjar and Martin Vetterli

Audiovisual Communications Laboratory (LCAV) - EPFL

ABSTRACT

Super-resolving a natural image is an ill-posed problem. The classical approach is based on the registration and subsequent interpolation of a given set of low-resolution images. However, achieving satisfactory results typically requires the combination of a large number of them. Such an approach would be impractical over heterogeneous rate-constrained wireless networks due to the associated communication cost and limited data available. In this paper, we present an approach for local image enhancement following the finite rate of innovation sampling framework, and motivate its application to the super-resolution problem over heterogeneous networks. Local estimates can be exchanged among the nodes of the network in order to regularize the super-resolution problem while, at the same time, reduce data exchange.

Index Terms— FRI sampling, super-resolution, distributed processing

1. INTRODUCTION

The goal of super-resolution is to produce a better quality image from a set of lower-resolution ones. After proper registration, the reconstruction of the high-resolution image is an ill-posed inverse problem. This issue can be alleviated by employing regularization techniques where some prior information about the image can be used in order to make the problem well posed [1]. Typically, a centralized scenario is assumed in the super-resolution problem. However, due to the ubiquitous presence of mobile wireless devices in our everyday life, distributed scenarios should be considered, too. This is a particularly challenging task since one faces two main additional problems. First, different devices have different characteristics in their acquisition systems leading to different noise levels and pixel resolutions. Second, communications should be done over a wireless network. Thus, being energy-efficient and keeping communication cost at a minimum are a must. To the best of our knowledge, these constraints have not been considered before in the context of super-resolution problems.

In this work, we propose a new method for local edge extraction based on the theory of sampling signals with *finite rate of innovation* [2]. Our motivation is to take advantage of local processing power in order to precisely extract image features such as edges that can be later used to regularize the image restoration problem. The use of local information (such as edges) is also well motivated by biological evidence on how the early stages of the visual system work (see [3] and references therein). Once nodes have extracted edges from the images they can exchange that information in order to better regularize the image enhancement part. Note that following this strategy, the exchange of raw pixels can be significantly reduced while, at the same time, producing perceptually better reconstructions.

This work has been conducted as part of the project HANDiCAMS under FET-Open grant no. 323944.

A method for step-edge estimation using the polynomial reproducing property of basis splines (B-splines) was initially proposed in [4] and later extended in [5] to the case of E-splines. One of the main issues with these two approaches is that they recast the problem as a set of one-dimensional problems making them less robust to noise. In our case, we propose to use all the image samples (within some region of interest) in the estimation process and not just those corresponding to a particular row. In our formulation this is possible since we consider that the observed signals (edges) have a finite length. In the simulations part, we will show that our method performs well in practice even when we observe a segment whose span is outside the boundaries of our observation window. In such cases, we also observe that windowing the data results in an improved estimation performance. We also show a motivating example of the developed theory to the case of image super-resolution when two devices with significantly different spatial resolutions are cooperating. Our experiments indicate that a significant reduction in the data volume to be exchanged is possible while achieving sharper reconstructions.

2. SAMPLING LINE SEGMENTS

2.1. Exponential Splines

Let us first start by introducing some background material that will be used in later in the paper. For a detailed treatment of E-splines and their properties we refer the reader to [6].

Let α be a complex number and consider the (one-dimensional) function

$$\beta_\alpha(t) = e^{\alpha t}\beta_0(t), \tag{1}$$

where $\beta_0(t)$ is the "box" function given by

$$\beta_0(t) = \begin{cases} 1 & 0 \le t < 1 \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

The function $\beta_\alpha(t)$ has the property of reproducing the exponential function $e^{\alpha t}$ by combining appropriately scaled integer-shifted replicas of it:

$$\sum_{n\in\mathbb{Z}} e^{\alpha n}\beta_\alpha(t-n) = e^{\alpha t}. \tag{3}$$

Higher order E-splines can be easily constructed by convolution. In particular, let $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_K]$ then, an E-spline of order K is given by

$$\beta_{\boldsymbol{\alpha}}(t) = \beta_{\alpha_1}(t) * \cdots * \beta_{\alpha_K}(t), \tag{4}$$

where $*$ denotes convolution. It becomes clear from (4) that the Fourier transform of $\beta_{\boldsymbol{\alpha}}(t)$ is given by

$$B_{\boldsymbol{\alpha}}(\omega) = \prod_{k=1}^{K} B_{\alpha_k}(\omega) = \prod_{k=1}^{K} \frac{e^{\alpha_k - j\omega} - 1}{\alpha_k - j\omega}. \tag{5}$$

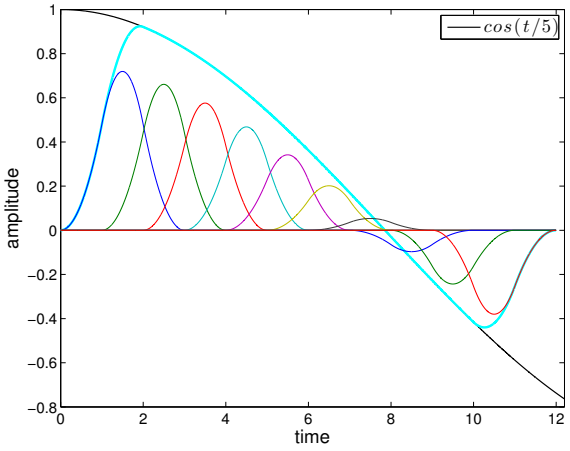

Fig. 1. Reproduction property of an E-spline function. The function $cos(t/5)$ is locally approximated by adding scaled shifted versions of an E-spline.

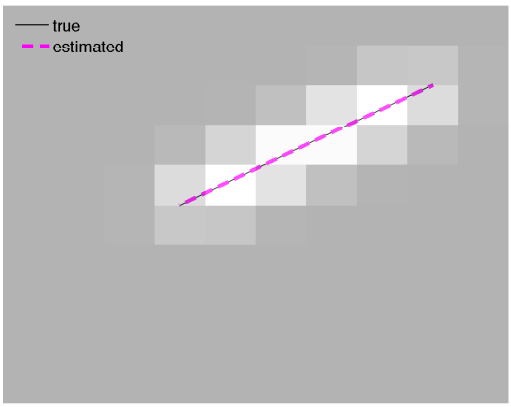

Fig. 2. Example of an image obtained from sampling a line segment. The true and the estimated segment are also displayed.

In this paper, we will be considering the particular case where the α_k's are purely imaginary (*i.e.*, $\alpha_k = j\omega_k$). Furthermore, we will consider pairs of frequencies $(-j\omega_k, j\omega_k)$, which will give rise to real-valued E-spline functions that reproduce complex exponentials, that is

$$\sum_{n\in\mathbb{Z}} c_n(\omega_k)\beta_{\boldsymbol{\alpha}}(t-n) = e^{\alpha_k t}, \tag{6}$$

where (for this particular choice of E-splines) the coefficients $c_n(\omega_k)$ are given by

$$c_n(\omega_k) = e^{j\omega_k n}/B_{\boldsymbol{\alpha}}(\omega_k). \tag{7}$$

It is important to remark that the reproduction property will allow us to compute continuous-time quantities from the discrete samples. An example of the reproduction property of E-splines is illustrated in Figure 1.

2.2. Sampling Line Segments

Let $f(\mathbf{x})$ denote the two-dimensional signal that we want to capture. During acquisition the observed signal is convolved with a low-pass filter (due to the lens) and sampled on a regular grid. Mathematically, this operation is described as

$$y_{nm} = \langle f(\mathbf{x}),\, \phi_{nm}(\mathbf{x})\rangle = \int_{\mathbb{R}^2} f(\mathbf{x})\,\phi_{nm}(\mathbf{x})\,d\mathbf{x}, \tag{8}$$

where y_{nm} is the sample (pixel) at the (n,m) position in the image, $\mathbf{x} = [x,y]^T \in \mathbb{R}^2$ is a two-dimensional point, and $\phi_{nm}(\mathbf{x}) = \phi(x-n, y-m)$ is the sampling kernel. Consider a line segment between two points $\mathbf{x}_1$ and $\mathbf{x}_2$, and let C be a parametric representation of the segment:

$$C = \{\mathbf{x}\in\mathbb{R}^2, \text{ s.t. } \mathbf{x} = t\,\mathbf{x}_1 + (1-t)\,\mathbf{x}_2,\ 0\le t\le 1\}. \tag{9}$$

Let $f(\mathbf{x}) = \delta(\mathbf{x}\in C)$, where $\delta(\cdot)$ is the Dirac's delta function. Then we have that

$$\begin{aligned} y_{nm} &= \int_{\mathbb{R}^2} \delta(\mathbf{x}\in C)\,\phi_{nm}(\mathbf{x})\,d\mathbf{x} = \int_C \phi_{nm}(\mathbf{x})\,d\mathbf{x} \\ &= \int_0^1 \phi_{nm}(C(t))\,\|C'(t)\|\,dt \\ &= \|\mathbf{x}_1-\mathbf{x}_2\| \int_0^1 \phi_{nm}(t\mathbf{x}_1 + (1-t)\mathbf{x}_2)\,dt, \end{aligned} \tag{10}$$

where we have made explicit the parametrization of the curve C with respect to t, and where $C'(t)$ is the gradient of $C(t)$. An example of an image obtained from sampling a line segment is displayed in Figure 2.

Assume that the kernel $\phi_{nm}(\mathbf{x})$ is able to reproduce exponentials (e.g., E-spline kernel) such that

$$\sum_{n\in\mathbb{Z}}\sum_{m\in\mathbb{Z}} c_{nm}(\boldsymbol{\omega})\phi_{nm}(\mathbf{x}) = e^{j\boldsymbol{\omega}^T\mathbf{x}}, \tag{11}$$

for some frequency vector $\boldsymbol{\omega}\in\mathbb{R}^2$. If we now consider the weighted sum of samples we have

$$\begin{aligned} F(\boldsymbol{\omega}) &= \sum_{n,m\in\mathbb{Z}} c_{nm}(\boldsymbol{\omega})y_{nm} \\ &= \|\mathbf{x}_1-\mathbf{x}_2\| \int_0^1 e^{j\boldsymbol{\omega}^T(t\mathbf{x}_1+(1-t)\mathbf{x}_2)}\,dt \\ &= \|\mathbf{x}_1-\mathbf{x}_2\|\, e^{j\boldsymbol{\omega}(\mathbf{x}_1+\mathbf{x}_2)/2}\ \mathrm{sinc}(\boldsymbol{\omega}^T(\mathbf{x}_1-\mathbf{x}_2)/2\pi), \end{aligned} \tag{12}$$

where the last equality follows after manipulating the integration result and where $\mathrm{sinc}(x) = \sin(\pi x)/\pi x$.

2.3. Parameter Estimation

Based on Equation (12) we can now establish a procedure for determining the points that define the segment. If we restrict ourselves to low frequencies, it is clear from (12) that we can get an estimate of

$$\mathbf{s} = \mathbf{x}_1 + \mathbf{x}_2 \tag{13}$$

by just looking at the phase of $F(\boldsymbol{\omega})$. For that purpose, we need at least a pair of distinct frequencies in order to solve a system of equations of the form

$$\begin{bmatrix} \boldsymbol{\omega}_1^T \\ \vdots \\ \boldsymbol{\omega}_L^T \end{bmatrix} \mathbf{s} = \begin{bmatrix} \sphericalangle F(\boldsymbol{\omega}_1) \\ \vdots \\ \sphericalangle F(\boldsymbol{\omega}_L) \end{bmatrix}, \tag{14}$$

for a set of L distinct frequencies $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_L$. In practice, due to noise and other corruptions, the above system of equations needs to be solved in the least-squares sense. In order to resolve the endpoints of the segment we will use the magnitude of the frequency response given by

$$|F(\boldsymbol{\omega})| = \left|\mathrm{sinc}(\boldsymbol{\omega}^T(2\mathbf{x}_1 - \mathbf{s})/2\pi)\right|, \tag{15}$$

where we have made use of relation (13). In particular, consider the ratios

$$r_{ij} = \left|\frac{F(\boldsymbol{\omega}_i)}{F(\boldsymbol{\omega}_j)}\right| = \left|\frac{\mathrm{sinc}(\boldsymbol{\omega}_i^T(2\mathbf{x}_1 - \mathbf{s})/2\pi)}{\mathrm{sinc}(\boldsymbol{\omega}_j^T(2\mathbf{x}_1 - \mathbf{s})/2\pi)}\right|, \tag{16}$$

where we have made use of relation (13).

Since we are considering low frequencies, we can safely skip the absolute value in the last part of (16). Keeping this in mind we can compute several ratios in order to estimate $\mathbf{x}_1$ as the solution to the following nonlinear least-squares problem

$$\hat{\mathbf{x}}_1 = \arg\min_{\mathbf{x}} g(\mathbf{x}), \tag{17}$$

where $g(\mathbf{x})$ is given by

$$\begin{aligned} g(\mathbf{x}) &= \frac{1}{2}\sum_{i,j}\left(\frac{\mathrm{sinc}(\boldsymbol{\omega}_i^T(2\mathbf{x} - \mathbf{s})/2\pi)}{\mathrm{sinc}(\boldsymbol{\omega}_j^T(2\mathbf{x} - \mathbf{s})/2\pi)} - r_{ij}\right)^2 \\ &= \frac{1}{2}\sum_{i,j}\left(\frac{\mathrm{sinc}_i(\mathbf{x})}{\mathrm{sinc}_j(\mathbf{x})} - r_{ij}\right)^2 \end{aligned} \tag{18}$$

Having computed $\mathbf{x}_1$, we can find the other extreme point by using (13). A simple strategy to (locally) solve for (17) is to use a gradient-descent strategy or the Gauss-Newton method.

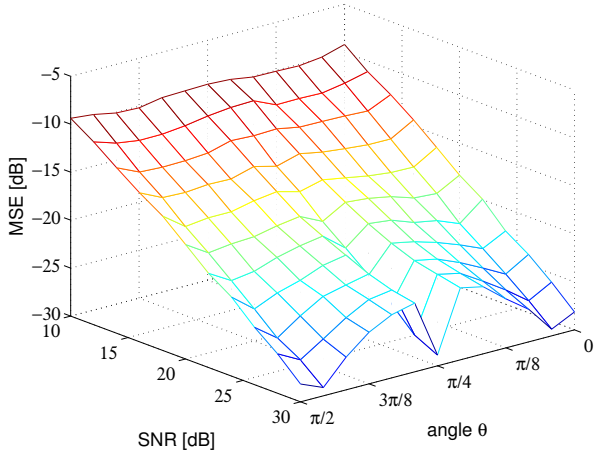

Fig. 3. Mean Squared Error as a function of the signal to noise ratio, and for different line slopes. The MSE is measured as the average relative error between the original image and the reconstructed one using the estimated parameters.

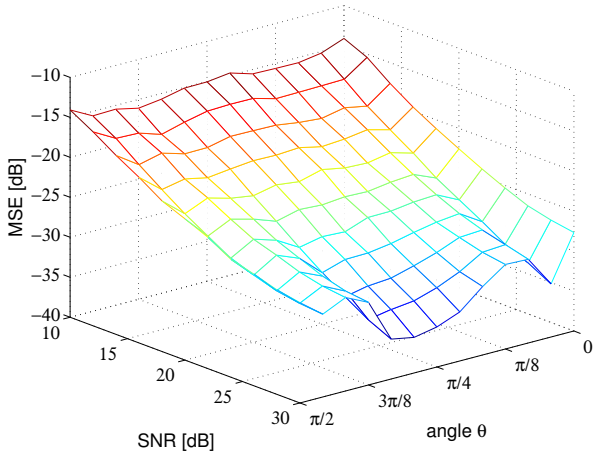

Fig. 4. Mean Squared Error as a function of the signal to noise ratio, and for different line slopes. The MSE is measured as the average relative error between the original image and the reconstructed one using the estimated parameters. The data is windowed using an isotropic 2D Gaussian window prior to parameter estimation.

3. NUMERICAL SIMULATIONS

For all the experiments we consider centered and symmetric E-spline sampling kernel obtained as the tensor product of 1D kernels with parameters $\boldsymbol{\alpha} = j2\pi(-2, -1, 1, 2)/50$.

3.1. Line Estimation

We proceed now to evaluate the estimation performance of the proposed scheme in the presence of noise. For that purpose, we randomly generate lines with different orientations. In particular, we generate lines centered in the image, and with different slopes in the range $[0,\ \pi/2]$. We then sample them using an E-spline kernel and produce an image of 10 by 10 pixels. The sampled image is corrupted (pixel wise) with independent additive Gaussian noise samples. If we denote $\mathbf{I}$ as the noiseless sampled image and σ^2 as the (per pixel) noise variance then, the Signal to Noise Ratio (SNR) is defined as

$$SNR = \frac{\sum_{nm}\mathbf{I}^2(n,m)}{r_x r_y \sigma^2}, \tag{19}$$

where r_x and r_y are the horizontal and vertical pixel resolutions of the image, respectively. It is important to mention that the parameters of the line (slope and intercept with the y-axis) can be easily retrieved from the estimated endpoints $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$.

The results of the simulation are depicted in Figure 3 where it is displayed the average relative error between the original (noiseless) image and the reconstructed one using the estimated parameters. As it can be seen, the performance degrades slightly as we move away from $\theta = \pi/4$, being less severe as we approach the extremes $\theta = 0$ and $\theta = \pi/2$. The reason is that in those cases we are having boundary effects that cause a small bias in the solution. In order to reduce such effect, we could window the observed data using an isotropic window such as a Gaussian 2D window of 10×10 samples. The estimation results with windowing are depicted in Figure 4.

As it can be seen in Figure 4 a better reconstruction in terms of MSE can be obtained (gain around 5 dB). However, in such case the

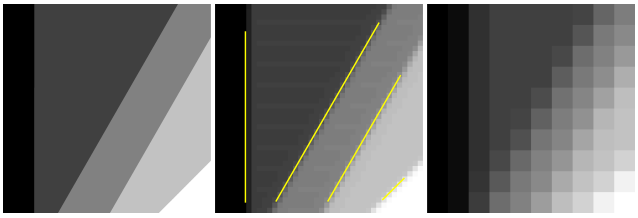

Fig. 5. Original high-resolution image (left), captured from device 1 (middle) and from device 2 (right). Yellow lines show the estimated edges of device 1.

most challenging lines to estimate are horizontal and vertical lines. This is to be expected since in those cases we are dealing with a 1D estimation problem, hence reducing the amount of samples (by windowing) has a greater effect on the variance of the estimator.

3.2. Image Enhancement

In order to motivate the applicability of the proposed edge-detection framework, consider the acquisition of a scene by two devices with very different spatial resolutions as illustrated in Figure 5. For simplicity, let's assume that the images are registered (i.e., they have a common coordinate system). A 1000×1000 high-resolution synthetic image has been generated and sampled with an E-spline kernel. The same sampling kernel is used for the two devices but with different scaling factor $1:4$ (i.e., kernel of device 2 is 4 times wider). As a result, the images acquired from device 1 has dimensions of 40×40 pixels while the one from device 2 is only 10×10 pixels. At that resolution, the edges in the image cannot be properly extracted. However, device 1 has enough resolution to perform reliable estimation (see Figure 5) and communicate those estimates to device 2. Note that in the acquired images we do not directly observe sampled lines but instead, edges are present in the acquired data. Edges in an image can be well approximated by the use of a step-function [4] and the support of the discontinuity can be retrieved by using finite differences of the samples [7]. More concretely, the image samples around an edge are given by

$$y_{nm} = A\langle h(\langle \mathbf{n},\, \mathbf{x}\rangle),\, \phi_{nm}(\mathbf{x})\rangle, \tag{20}$$

where $h(t)$ is the step function, $\mathbf{n}$ is the normal vector to the curve defining the edge and A denotes the amplitude of the step-edge. By taking sample differences along the x-direction (see [7]) we get

$$\begin{aligned} d_{nm} = y_{nm+1} - y_{nm} &= A\langle \frac{\partial h(\langle \mathbf{n},\, \mathbf{x}\rangle)}{\partial x},\, \tilde{\phi}_{nm}(\mathbf{x})\rangle \\ &\propto \langle \delta(C(t)),\, \tilde{\phi}_{nm}(\mathbf{x})\rangle, \end{aligned} \tag{21}$$

where $\tilde{\phi}_{nm}(\mathbf{x}) = \phi_{nm}(x,y) * \beta_0(x)$. It is clear from (21) that by working with sample differences we are back to our estimation framework (cf. Section 2). Since the above model holds only locally (around edges), an edge detector is first used to determine regions of interest following a similar procedure as in [4]. A Canny edge detector [8] is used in our simulations. For each pixel marked as an edge, we center a 10×10 window and extract a patch of the image. We then take the finite differences along the x-dimension and perform the estimation of the segment parameters on the differentiated patch. Estimates from neighboring pixels are finally clustered and averaged. The results of these operations on the acquired image of device 1 are displayed in Figure 5. Once device 1 has identified

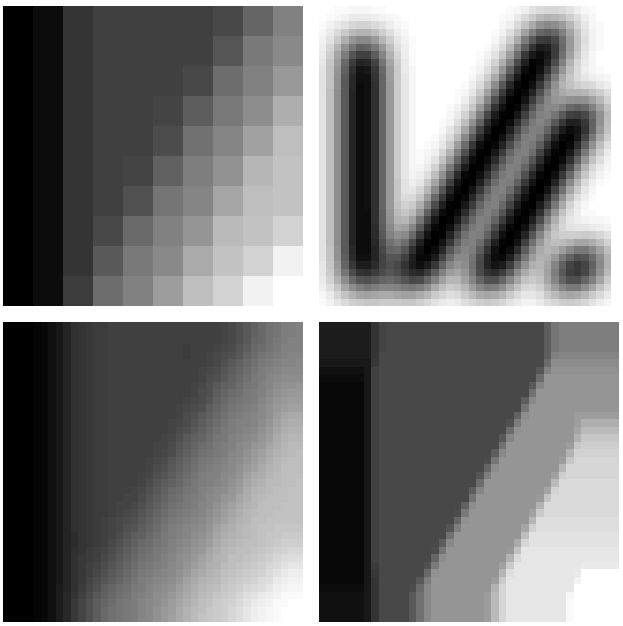

Fig. 6. Acquired image (top-left), generated mask (top-right), initial guess (bottom-left), reconstruction (bottom-right).

edges in the image, it communicates those to device 2. In our particular example this would mean the transmission of 4 segments (8 points). This information can be used to enhance the resolution of its own acquisition. In particular, we consider the enhancement of the image captured by device 2 by solving the following optimization problem:

$$\begin{aligned} \hat{\mathbf{Y}} = \arg\min_{\mathbf{Y}} \|\mathbf{Y}_0 - \mathbf{Y}\|_F^2 + \lambda_x \|\mathrm{vec}(\mathbf{M} \odot (\mathbf{YD}))\|_1 \\ + \lambda_y \|\mathrm{vec}(\mathbf{D}^T\mathbf{Y})\|_1, \end{aligned} \tag{22}$$

where $\mathbf{Y}_0$ is an initial estimate of our target image (e.g., obtained by interpolation), $\mathbf{M}$ is a mask image that reproduces the sampling of the segments (low value where segments are located and close to 1 outside the support), and $\mathbf{D}$ implements the finite difference operator. The operator $\mathrm{vec}(\cdot)$ vectorizes a matrix, $\odot$ denotes element-wise product, $\|\cdot\|_F$ is the matrix Frobenius norm while $\|\cdot\|_1$ is the ℓ_1 vector norm. The scalars λ_x and λ_y help to balance the reconstruction between data fidelity and smoothness on its respective dimension. In Figure 6 we have displayed the reconstructed image at device 2 using an up-sampling factor of 4. Problem (22) is solved using $(\lambda_x, \lambda_y) = (0.5, 0.1)$ and bicubic interpolation to generate the initial guess $\mathbf{Y}_0$. As it can be appreciated, a perceptually good reconstruction can be obtained by using the side information provided by device 1 (just a few bytes). The reconstruction gets poorer towards the boundaries where no constraints have been imposed.

4. CONCLUSIONS

In this paper we have presented a new method for the estimation of edges in an image using the properties of E-splines. We have illustrated its performance and motivated its application to the problem of image enhancement over heterogeneous wireless networks. This work constitutes a first step towards a more general theory for distributed and cooperative image super-resolution.

5. REFERENCES

[1] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *Signal Processing Magazine, IEEE*, vol. 20, no. 3, pp. 21–36, May 2003.

[2] M. Vetterli, P. Marziliano, and T. Blu, "Sampling signals with finite rate of innovation," *Signal Processing, IEEE Transactions on*, vol. 50, no. 6, pp. 1417–1428, Jun 2002.

[3] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," *Computational Models of Visual Processing*, pp. 3–20, 1991.

[4] L. Baboulaz and P. L. Dragotti, "Exact Feature Extraction Using Finite Rate of Innovation Principles With an Application to Image Super-Resolution," *Image Processing, IEEE Transactions on*, vol. 18, no. 2, pp. 281–298, Feb 2009.

[5] A. Hirabayashi and P. L. Dragotti, "E-spline sampling for precise and robust line-edge extraction," in *Image Processing (ICIP), 2010 17th IEEE International Conference on*, Sept 2010, pp. 909–912.

[6] M. Unser and T. Blu, "Cardinal exponential splines: part I - theory and filtering algorithms," *Signal Processing, IEEE Transactions on*, vol. 53, no. 4, pp. 1425–1438, April 2005.

[7] P.L. Dragotti, M. Vetterli, and T. Blu, "Sampling moments and reconstructing signals of finite rate of innovation: Shannon meets Strang-Fix," *Signal Processing, IEEE Transactions on*, vol. 55, no. 5, pp. 1741–1757, May 2007.

[8] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986.