Finite Element Techniques for Removing the Mixture of Gaussian and Impulsive Noise

Bishnu P. Lamichhane

***Abstract*—The finite element method has become a very powerful and popular tool to solve boundary value problems coming from science and engineering. Here, we consider a scattered data fitting method based on the finite element method and apply the method to remove the mixture of Gaussian and impulsive noise from an image. Numerical results show the performance of the approach.**

***Index Terms*—Finite element interpolation and smoothing, Delaunay triangulation, Voronoi diagram, scattered data interpolation, impulsive and Gaussian noise**

I. Introduction

An interesting problem in image processing is to recover a clear image out of a noisy image. In real situations, a noise model is not known a priori. Nevertheless, the most common noise type encountered in real applications are impulsive noise, Gaussian noise or mixture of both. We refer to [1]–[3] for more details on digital image processing.

Here, we consider finite element techniques to denoise a digital image corrupted with the mixture of impulsive and Gaussian noise- so called ’mixed noise’. Recently, finite element methods have been applied in different areas of image processing [4]–[8]. As the finite element method is successfully used to remove high density impulsive noise, see, e.g., [6], [9], we show that finite element methods are also suitable for removing high density impulsive noise as well as the mixture of Gaussian and impulsive noise from an image. The removal of an impulsive noise is based on first identifying the noisy and pure pixels of the image and then fitting a suitable surface using only pure pixels of the image. The interpolation is based on Delaunay triangulation and Voronoi diagrams [10]–[12], whereas we use finite element smoothing based on minimization of a functional involving the gradient of a piecewise polynomial function to recover a clear image when the image is corrupted with the mixture of Gaussian and impulsive noise. Our finite element smoothing is closely related to the steady state solution of linear diffusion with appropriate boundary condition [13], [14]. This smoothing approach is simpler and more efficient compared to other related data smoothing techniques like thin plate splines, radial basis functions and finite element thin plate splines [15]–[17]. The functional minimization approach is often known as an energy method or variational approach [18], [19].

We recall that a digital image is an array of numbers, where the size of the array determines the size of the image. An intensity image, for example, can be stored as a single matrix of size $m \times n$, where each element of the matrix represents the intensity of the image pixel or gray level. On the other hand, an array of size $m \times n \times 3$ can be used for a color image where the image has size $m \times n$, and each pixel has three values representing the red, green, and blue intensities that make up the color. Although the idea can easily be extended to other types of image, we restrict ourselves to intensity images.

In this paper, we use finite element smoothing to remove the mixture of Gaussian and impulsive noise. The Gaussian noise is normally distributed, additive and affects almost all pixels of the image, whereas the impulsive noise corrupts some randomly selected pixels of the image. Those pixels corrupted by the impulsive noise are just random impulses and so do not carry any information of the image. In case of impulsive noise, we assume that the pixel location of the impulsive noise can be identified. In particular, we consider salt and pepper noise as a model of impulsive noise. Salt and pepper noise is fixed value impulsive noise where the noisy pixels are random and are set either to white or black in an intensity image, see [2]. Therefore, all the pixels having the largest or smallest values are regarded as corrupted pixels and are discarded for the interpolation and smoothing purpose. In case of mixture of Gaussian and impulsive noise, the impulsive noise is applied after the Gaussian noise, and therefore, the pixels with the largest or smallest values can still be regarded as the corrupted pixels. However, if the Gaussian noise comes after the impulsive noise or a complicated impulsive noise is applied, a reliable impulse detector should be applied to identify the impulsive pixels. Such impulse detectors are discussed in [6], [20]–[24]. Using chi-square goodness-of-fit test to accurately detect the corrupted pixels, an impulsive noise suppression scheme for images is proposed in [20]. This impulsive noise elimination filter shows a good performance even for images with 95% corrupted pixels. The impulse detector in [21] is based on a comparison of pixel samples within a narrow rank window by the rank and absolute value, whereas the one presented in [22] is based on an image statistic and neighboring pixels. An improvement of the approach in [22] is given in [24]. Using statistical tools to detect the impulsive pixels and non-linear filtering scheme based on ANFIS, another efficient method is presented in [23] for highly corrupted images. As our impulsive noise model is quite simple and the main goal of the paper is to introduce a finite element method to remove the mixture of Gaussian and impulsive noise, we do not consider these impulse detectors.

The paper is organized as follows. In the next section,

Copyright (c) 2008 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org,
The author is with Centre for Mathematics and its Applications, Mathematical Sciences Institute, Australian National University, Canberra, ACT 0200, Australia, (email: blamichha@yahoo.com)

we briefly review scattered data interpolation method based on Delaunay triangulation and Voronoi diagram applied to filter out salt and pepper noise. Section III is devoted to the introduction and analysis of our finite element smoothing technique based on minimization of a functional involving the gradient of a piecewise polynomial function. We show the existence and uniqueness of the the minimization problem under very mild assumptions. The finite element smoothing technique developed here is applied to denoise an image corrupted with salt and pepper noise as well as Gaussian noise in the last section. The results from finite element methods are compared with standard techniques like median filter and wavelets.

II. Finite Element Interpolation

Assume that $\mathcal{G} = \{(x_i, y_i)\}_{i=0}^N$ is a set of scattered points in $\mathbb{R}^2$, and a function f is given on $\mathcal{G}$ with $z_i = f(x_i, y_i)$ for $i = 0, \cdots, N$. Let Ω be the convex hull of the set of points $\mathcal{G}$. The problem of **scattered data interpolation** is to find a function $p : \bar{\Omega} \to \mathbb{R}$ so that $p(x_i, y_i) = z_i$ for $i = 0, \cdots, N$. There is a vast amount of literature devoted to the scattered data interpolation. We refer to [25], [26] for extensive surveys on this subject. Here, we restrict ourselves to the finite element interpolation on two dimensions.

In general, the finite element method consists of the following three steps. In the first step the domain Ω is decomposed into non-overlapping triangles or quadrilaterals. In the second step local basis functions are defined in each triangle or quadrilateral. Finally, these local basis functions are glued together to form a set of global basis functions which span the finite element space. We refer to [27]–[29] for a mathematical definition of the finite element method. In the two-dimensional case, if the interpolant is to be at least continuous, the decomposition should be geometrically conforming.

Definition 1. *Let $\Omega \subset \mathbb{R}^2$ be a polygoonal domain. The collection of disjoint polygonal subdomains $\mathcal{T}$ with $\bar{\Omega} = \cup_{T \in \mathcal{T}} \bar{T}$ forms a* **geometrically conforming decomposition** *of Ω if the intersection between the boundaries of any two different subdomains $\partial T_l \cap \partial T_k$, $k \neq l$, $T_k, T_l \in \mathcal{T}$ is either empty, a vertex or a common edge.*

In the following, we restrict ourselves to the situation where the subdomains are triangles or quadrilaterals.

Let Ω, the convex hull of $\mathcal{G}$, be a polygonal domain. We note that a polygonal domain in two dimensions is an open bounded region whose boundary consists of pieces of lines. The most efficient and popular way of decomposing the domain Ω into triangles with vertices in $\mathcal{G}$ when Ω is the convex hull of the scattered points is the Delaunay triangulation.

Definition 2. *Given a set $\mathcal{G}$ of points in $\mathbb{R}^2$, a* **Delaunay triangulation** *for $\mathcal{G}$ is a conforming decomposition $\mathcal{T}$ of convex hull of $\mathcal{G}$ into triangles with vertices in $\mathcal{G}$ such that no point in $\mathcal{G}$ is inside the circumcircle of any triangle in $\mathcal{T}$.*

A Delaunay triangulation of a finite set of points in the plane is a triangulation that minimizes the standard deviations of the angles of the triangles. Therefore, they are in some sense optimal for finite element interpolation [27], [29], [30]. Using locally optimal Delaunay triangulation, adaptive image approximation is applied in [8] based on piecewise linear finite elements, and a Delaunay triangulation based impulsive noise removal method is presented in [6]. Here, we briefly recall the method for the subsequent application.

The Voronoi diagram is used for the nearest-neighbor interpolation, and is the dual graph of the Delaunay triangulation for the same set of points.

Definition 3. *For a set of points $\mathcal{G} \subset \mathbb{R}^2$, the* **Voronoi diagram** *is the decomposition of the plane into convex polygons such that each polygon contains exactly one generating point from $\mathcal{G}$ and every point in a given polygon is closer to its generating point than to any other point in $\mathcal{G}$. A convex polygon $V_{\mathbf{x}}$ associated with the generating point $\mathbf{x} \in \mathcal{G}$ is called the* **Voronoi cell** *for the point $\mathbf{x} \in \mathcal{G}$.*

In other words, the Voronoi cell $V_{\mathbf{x}}$ for the point $\mathbf{x} \in \mathcal{G}$ has the property that the distance of every $\mathbf{y} \in V_{\mathbf{x}}$ from $\mathbf{x}$ is less than or equal to the distance of $\mathbf{y}$ from any other point in $\mathcal{G}$. The circle circumscribed about a Delaunay triangle has its center at the vertex of a Voronoi cell, see the right graph of Figure 1. The idea of Delaunay triangulation and Voronoi diagram is also extended to higher dimension. An efficient algorithm for computing Delaunay triangulation and Voronoi diagrams are presented in [11], see [10], [12] for more mathematical detail.

As an example of Delaunay triangulation and Voronoi diagram, we define a set $\mathcal{G}_1 = \{(0.1, 0.4), (0.5, 0.1), (0.45, 0.5), (0.3, 0.6), (0.3, 0.3), (0.1, 0.4), (0.9, 0.8), (0.3, 0.9), (0.2, 0.1), (0.8, 0.9)\}$, and generate the Delaunay triangulation and the Voronoi diagram of $\mathcal{G}_1$. We have shown the Delaunay triangulation and the Voronoi diagram of $\mathcal{G}_1$ in the left and middle graphs of Figure 1, respectively. The right graph of Figure 1 shows the circumcircle of a triangle with its center at a vertex of the Voronoi diagram as shown with a filled circle. Once we have a decomposition, the interpolation can be done by defining a suitable basis for the piecewise polynomial space. A suitable basis for the piecewise polynomial interpolation is a nodal basis defined as follows.

Definition 4. *Let $\mathcal{G} = \{(x_i, y_i)\}_{i=0}^N$ be a set of points in $\mathbb{R}^2$, and Ω be the convex hull of $\mathcal{G}$. Assume that $\mathcal{T}$ be a conforming decomposition of Ω into triangles or quadrilaterals. Then, a basis $\{\phi_i\}_{i=0}^N$ of piecewise polynomial space $U = span\{\phi_i\}_{i=0}^N$ is called a* **nodal basis** *of U with respect to $\mathcal{G}$ if and only if $\phi_j(x_i, y_i) = \delta_{ij}$ for $i, j = 0, \cdots, N$. The piecewise polynomial space U is a* **finite element space**, *and $\{\phi_i\}_{i=0}^N$ are also called* **finite element basis functions**.

Let $\{\phi_i\}_{i=0}^N$ be the set of piecewise linear or bilinear nodal basis functions with respect to the set of points $\mathcal{G}$ [29]. Assume that we are given the values $\{z_i\}_{i=0}^N$ of a function at $\mathcal{G}$. Then, the piecewise linear interpolant p of the given data is obtained by

$$p(x) = \sum_{i=0}^N z_i \phi_i(x).$$

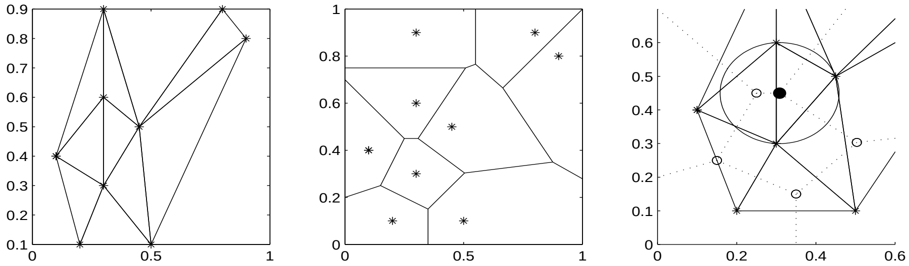


Fig. 1: The Delaunay triangulation of the set $\mathcal{G}_1$ (left), the corresponding Voronoi diagram (middle) and the circumcircle of a triangle with Delaunay triangulation and Voronoi diagram

The nearest neighbor interpolation of a scattered data $\mathcal{G} = \{(x_i, y_i)\}_{i=0}^{N}$ can be done by generating the Voronoi diagram of the set $\mathcal{G}$. Let χ_i be the characteristic function of the Voronoi cell corresponding to the point (x_i, y_i), $i = 0, \cdots, N$. Then, the nearest neighbor interpolant p of the given data is obtained by

$$p(x) = \sum_{i=0}^{N} z_i \chi_i(x).$$

Higher order interpolation can be defined in a similar way, see [27]–[29] for more details.

In order to interpolate a digital image, we associate a suitable set of points with the image called a tensor product partition defined as follows.

Definition 5. *Assume that* $\mathcal{S}_x = \{a = x_1 < \cdots < x_n = b\}$ *is a set of points in the closed interval* $[a, b]$ *and* $\mathcal{S}_y = \{c = y_1 < \cdots < y_m = d.\}$ *that of* $[c, d]$. *Then the set of points* $\mathcal{S} = \{(x_i, y_j)\}_{i=1,j=1}^{n,m}$ *is called a* **tensor product partition** *of the rectangular region* $[a, b] \times [c, d]$.

Associated with an intensity image I of size $m \times n$, we define a tensor product partition $\mathcal{S}$ of the square $[0, 1] \times [0, 1]$ as

$$\mathcal{S} = \{(a_i, b_j)\}_{i=1,j=1}^{n,m} \quad \text{with} \quad a_i = \frac{i-1}{n-1},\ b_j = \frac{j-1}{m-1}, \tag{1}$$

and an **image function** $I_f : \mathcal{S} \to \mathbb{R}^k$, $k \in \mathbb{N}$. We note that for color images, we have $k = 3$, and $I_f(a_i, b_j)$ denotes a pixel color at the point (a_i, b_j) having three values representing red, green and blue. For intensity images, we have $k = 1$, and $I_f(a_i, b_j)$ denotes the intensity of the image pixel or gray level at the point (a_i, b_j). In case of intensity images, the number $I_f(a_i, b_j)$ can be of class double, in which case it contains the values in $[0, 1]$, or of class $\mathrm{uint8}$, in which case the values are in $[0, 255]$. For both classes, the smallest intensity represents black, and the largest intensity represents white.

The impulsive noise for an original image I can be defined using the associated image function I_f. If $\tilde{I}$ is the image obtained by polluting the original image I by some impulsive noise of density $d\%$, then the image function $\tilde{I}_f$ for the corrupted image $\tilde{I}$ is defined as

$$\tilde{I}_f(a_i, b_j) = \begin{cases} I_f(a_i, b_j) & \text{with probability of} \quad 1 - \frac{d}{100} \\ \eta(a_i, b_j) & \text{with probability of} \quad \frac{d}{100}, \end{cases}$$

where $\eta(a_i, b_j)$ is the impulse at (a_i, b_j), and $1 \le i \le n$ and $1 \le j \le m$. The salt and pepper noise is obtained when $\eta(a_i, b_j)$ takes the largest and smallest pixel values [9].

Assume that an intensity image I of size $m \times n$ is corrupted with salt and pepper noise. Let the tensor product partition as given in (1) and the image function $I_f : \mathcal{S} \to \mathbb{R}$ be associated with the image. Denoting the set of points having corrupted and non-corrupted image pixels by $\mathcal{S}^n$ and $\mathcal{S}^p$, respectively with $\mathcal{S} = \mathcal{S}^n \cup \mathcal{S}^p$, we define a function $I_f^p : \mathcal{S}^p \to \mathbb{R}$ as the restriction of the function I_f to the set $\mathcal{S}^p$. As the positions of the noisy pixels are random, the points in $\mathcal{S}^p$ have no structure. Since the impulses of the salt and pepper noise are white or black pixels, $\mathcal{S}^n$ consists of all white and black pixels, and $\mathcal{S}^p$ consists of the rest pixels. However, for a general impulsive noise, an efficient impulse detector should be used to identify the sets $\mathcal{S}^n$ and $\mathcal{S}^p$, see, e.g., [6], [20]–[24].

The idea of denoising the image corrupted with only impulsive noise is to find an interpolant which interpolates the discrete function I_f^p. Let $p : \bar{\Omega} \to \mathbb{R}$ be a piecewise polynomial interpolant based on the Delaunay triangulation of the scattered points in $\mathcal{S}^p$. Thus, $p(x, y) = I_f^p(x, y),\ (x, y) \in \mathcal{S}^p$. If a point $(x, y) \in \mathcal{S}$ but $(x, y) \notin \bar{\Omega}$, we need to extrapolate for this point. In this case, we find a point $(\hat{x}, \hat{y})$ in $\mathcal{S}^p$ nearest to (x, y) and set $p(x, y) = I_f^p(\hat{x}, \hat{y})$ using nearest neighbor extrapolation [31]. Another way of doing the extrapolation is to reflect some pixels near to the boundary of Ω across the boundary of the unit square and cover the whole unit square with Delaunay triangulation [9]. We refer to [18] for another approach to remove high density salt and pepper noise.

III. Finite Element Smoothing

When an image is corrupted only with impulsive noise, and it is possible to detect noisy and pure pixels, scattered data interpolation based on Delaunay triangulation provides an efficient approach to remove the impulsive noise from the image. However, in many practical situations the image is corrupted with impulsive noise as well as Gaussian noise. If a noisy image is transmitted over faulty communication lines, the received image might be corrupted with the mixture of Gaussian and impulsive noise. If the image is corrupted with the mixture of Gaussian and impulsive noise, it is necessary to smooth the image pixels as well as to remove impulses from them. In [22], [32], some methods are proposed to remove the mixture of Gaussian and impulsive noise. The filter proposed in [22] is based on a local image statistic, whereas the one proposed in [32] is based on fuzzy method. Here, we use a smoothing technique based on the finite element method to remove the mixture of Gaussian and impulsive noise.

Let $\tilde{I}$ be the image corrupted with the mixture of Gaussian and impulsive noise, and I the original image. We associate the tensor product partition $\mathcal{S}$ as defined in (1) with both images, and image functions $I_f : \mathcal{S} \to \mathbb{R}$ and $\tilde{I}_f : \mathcal{S} \to \mathbb{R}$ with the images I and $\tilde{I}$, respectively. As before, let $\mathcal{S}^n$ and $\mathcal{S}^p$ be the set of points corrupted and non-corrupted with impulsive noise, respectively, and $\mathcal{S}^p = \{(x_i, y_i)\}_{i=0}^N$. Since the image is corrupted with the mixture of Gaussian and impulsive noise, after removing the impulsive pixels from the noisy image we have

$$\tilde{I}_f(x_i, y_i) = I_f(x_i, y_i) + n_i, \; (x_i, y_i) \in \mathcal{S}^p,$$

where n_i are drawn from zero-mean Gaussian distribution. Hence we have to do two things at once. We should find the missing pixels of the image and reduce the Gaussian noise of the pixels in $\mathcal{S}^p$. Mathematically, we want to find a function defined on $[0, 1] \times [0, 1]$, possibly smooth, which approximates the unknown image function I_f at the points in $\mathcal{S}$. The smoothness of the function will reduce the Gaussian noise. This is equivalent to fitting a smooth surface to the scattered data in $\mathcal{S}^p$, where the smoothness and approximation is suitably balanced.

Radial basis functions and thin plate splines are often used to interpolate and smooth scattered data [15], [16]. Here, we consider an approach based on the bivariate L-spline analyzed in [16] which is a minimizer of the functional

$$F(u) = \sum_{i=1}^{N} (u(x_i, y_i) - z_i)^2 + \lambda \int_\Omega (L\, u(x, y))^2 \, dx \, dy \quad (2)$$

over a space of function, where L is a partial differential operator, and λ is a positive constant. The approach of functional minimization is quite popular in the context of image processing based on partial differential equations [13], [14], [19]. As the approach is based on minimizing the energy defined by a functional, it is often called an energy method or variational approach [18], [19]. A simple approach is obtained by replacing the operator L by the gradient operator. In this case, the approach is closely connected with the steady state solution of linear diffusion problem [13], [14] with some appropriate boundary condition. On the other hand, the thin plate spline smoothing is obtained by replacing $(L\, u)^2$ by $u_{xx}^2 + 2u_{xy}^2 + u_{yy}^2$, see [15], [16]. As $\mathcal{S}^p = \{(x_i, y_i) : 1 \leq i \leq N\}$, the functional F depends only on the set of pure pixels and the set of corrupted pixels is completely discarded as in finite element interpolation.

Since interpolation methods based on radial basis functions or thin plate splines are very expensive, we replace u with a function belonging to the finite element space on the structured grid formed by using the tensor product partition $\mathcal{S}$, where the pixel values of the image are to be computed. An example of such a grid along with the scattered points is shown in the left graph of Figure 2 using triangular elements, and a function belonging to the finite element space V having zero boundary condition is shown in the right.

The finite element thin plate spline presented in [33] is closely related to our approach. In [33] the second order derivative in the thin plate spline formulation is replaced with a first order condition to use a low order finite element method by using a variational theorem, where the basis functions are only continuous, not even continuously differentiable. Similar ideas for spline smoothing using finite element methods are exploited in [17], [34]. Our approach is motivated by the fact that although the given data is completely unstructured, we want to reconstruct the image in a structured grid. Working with the finite element thin plate spline as in [33] or the Laplace operator for L as in [34], a mixed finite element method is to be employed to work with only continuous finite element space [17], [29], [33], [34]. If a mixed method is used for the discretization, a solver adapted to the saddle point matrix is to be used. Furthermore, the finite element thin plate spline presented in [17] or the bivariate L-spline proposed in [34] being based on higher order partial differential operators produce much smoother solution. Since the underlying image function might not be a smooth function, they do not necessarily produce good solutions.

On the one hand, we want to use a simple finite element method leading to a positive-definite formulation, which can be easier to solve than the saddle point system. On the other hand, we do not want to have a much smooth solution. Therefore, we replace Lu with $\|\nabla u\|$, where ∇ is the gradient operator and $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^2$. In the following, the Euclidean norm of $x \in \mathbb{R}^n$ will be denoted by $\|x\|$.

Let $C^0(\Omega)$ be the space of continuous functions in Ω. Let $\mathcal{T}$ be a structured decomposition of the rectangular domain Ω into rectangles or triangles, and

$$V = \{u \in C^0(\Omega) |\, u_{|T} \in \mathcal{P}(T), \, T \in \mathcal{T}\} \quad (3)$$

be a finite element space, where $\mathcal{P}(T)$ is the linear or bilinear polynomial space on T depending on T being a triangle or rectangle. Now, our discrete problem is to minimize the functional (2) over the function space V so that the discrete problem can be written as

$$\min_{u \in V} \sum_{i=0}^{N} (u(x_i, y_i) - z_i)^2 + \lambda \int_\Omega \|\nabla u\|^2 \, dx \, dy. \quad (4)$$

Denoting function values of u at the measurement points by

$$Pu = (u(x_0, y_0), u(x_1, y_1), \cdots, u(x_N, y_N))^T,$$

we introduce a functional

$$J_\lambda(u) = \|Pu\|^2 + \lambda \int_\Omega \|\nabla u\|^2 \, dx \, dy - 2\, (Pu)^T \mathbf{z},$$

where $\mathbf{z}$ is a column vector having i-th entry as z_i for $i = 0, \cdots, N$. It is easy to see that the minimization problem (4) is equivalent to

$$\min_{u \in V} J_\lambda(u).$$

For the mathematical analysis of the problem (4) we need the following definition.

Definition 6. *Let H be a Hilbert space with inner product $\langle, \cdot, \rangle_H$, and W a subspace of H. Let $a : W \times W \to \mathbb{R}$ be a*

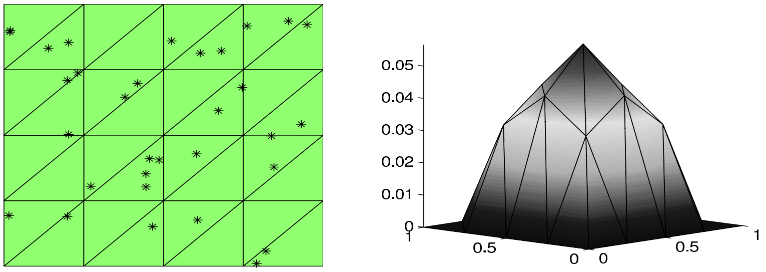


Fig. 2: A structured triangular grid formed by using a tensor product partition and a set of scattered points (left) and a finite element function having zero boundary condition (right)

symmetric bilinear form and $f : W \to \mathbb{R}$ a linear form. The problem of finding $u \in W$ such that

$$a(u, v) = f(v), \quad v \in W \tag{5}$$

is called an **abstract variational problem***.*

The crucial tools for the analysis are the following theorems. We refer to [28], [29] for proofs.

Theorem 1. *Let H be a Hilbert space with inner product $\langle,\cdot,\rangle_H$, and W a subspace of H. Let $a : W \times W \to \mathbb{R}$ be a symmetric bilinear form satisfying $a(v, v) \geq 0$ for all $v \in W$. And $f : W \to \mathbb{R}$ a linear form. Then the abstract variational problem* (5) *is equivalent to the problem of finding $u \in W$ which minimizes the functional $a(u, u) - 2f(u)$ in W.*

The existence of the unique solution is shown by using the Lax-Milgram lemma.

Theorem 2. *Let H be a Hilbert space with inner product $\langle,\cdot,\rangle_H$, and W a subspace of H. Let the linear form $f : W \to \mathbb{R}$ be continuous and the bilinear form $a : W \times W \to \mathbb{R}$ be continuous and coercive with respect to the norm $\|\cdot\|_H$ induced by the inner product $\langle,\cdot,\rangle_H$. Then, the abstract variational problem* (5) *admits a unique solution which depends continuously on the data.*

In order to put our problem into the settings of the previous theorems, we define a P-inner product $\langle\cdot,\cdot\rangle_P$ with $\langle u, v\rangle_P = (Pu)^T Pv + \lambda \int_\Omega \nabla u^T \nabla v \, dx \, dy$. The following lemma shows that the P-inner product is an inner product on the vector space V given by (3).

Lemma 1. *Let $\lambda > 0$ and $\mathcal{G} \subset \bar{\Omega}$ be non-empty. Then, the P-inner product defined above is an inner product on the vector space V so that V is a Hilbert space over $\mathbb{R}$.*

Proof: In order to show that the P-inner product is indeed an inner product, we have to prove the following properties of P-inner product:

(1) $\langle v, v\rangle_P \geq 0$, and $\langle v, v\rangle_P = 0$ if and only if $v = 0, \quad v \in V$
(2) $\langle v + w, z\rangle_P = \langle v, z\rangle_P + \langle w, z\rangle_P, \quad v, w, z \in V$
(3) $\langle v, az\rangle_P = a\langle v, z\rangle_P, \quad v \in V,\ a \in \mathbb{R}$
(4) $\langle v, w\rangle_P = \langle w, v\rangle_P, \quad v, w \in V$

It is trivial to show that the P-inner product satisfies the second, third and fourth properties. It is also obvious that $\langle v, v\rangle_P \geq 0$. It remains to show that the P-inner product is positive-definite. Since u is continuous, $\langle \nabla u, \nabla u\rangle = 0$ if and only if u is constant. On the other hand, $\langle Pu, Pu\rangle = 0$ and u is constant in Ω if and only if $u = 0$ in Ω. ■

The P-norm of an element $u \in V$ induced by the inner product $\langle\cdot,\cdot\rangle_P$ is given by $\|u\|_P^2 = \|Pu\|^2 + \lambda \int_\Omega \|\nabla u\|^2 dx \, dy$.

The bilinear form $a(\cdot,\cdot)$ and the linear form $f(\cdot)$ associated with the functional $J_\lambda(\cdot)$ are given by

$$a(u, v) = (Pu)^T Pv + \lambda \int_\Omega \nabla u^T \nabla v \, dx \, dy, \text{ and } f(v) = (Pv)^T \mathbf{z}$$

so that our minimization problem (4) is equivalent to the variational problem to find $u \in V$ such that

$$a(u, v) = f(v), \quad v \in V. \tag{6}$$

Since the bilinear form $a(\cdot,\cdot)$ and the linear form $f(\cdot)$ as defined above satisfy all the properties of Theorem 1, the unique minimizer is the solution of the variational problem (5). Furthermore, the following corollary holds.

Corollary 1. *The variational problem* (6) *admits a unique solution which depends continuously on the data.*

Proof: Since $u, v \in V$, it follows that $|a(u, v)| \leq \|u\|_P \|v\|_P$ and $|f(v)| \leq C\|v\|_P$. Hence the bilinear form $a(\cdot,\cdot)$ and the linear form $f(\cdot)$ are continuous with respect to the norm $\|\cdot\|_P$. Moreover, from the definition of P-norm $a(u, u) = \|u\|_P$ so that $a(\cdot,\cdot)$ is coercive with respect to the norm $\|\cdot\|_P$. Hence, our variational problem (6) has a unique solution. From the definition of the P-inner product, we have

$$a(v, v) = \|v\|_P^2,\ v \in V,$$

and thus, for the solution $u \in V$, $\|u\|_P^2 = f(u)$. ■

Remark 1. *When there is no Gaussian noise, we can basically find the solution of the variational problem: Find $u \in V$ such that*

$$\int_\Omega \nabla u^T \nabla v \, dx \, dy = 0,\ v \in V$$

subject to the conditions

$$u(x_i, y_i) = z_i,\ i = 0, \cdots, N.$$

In the context of image inpainting, this method is called *variational or harmonic inpainting [2], and produces a smooth solution. A penalty formulation of this problem is given by Problem* (4) *where λ is a very small penalty parameter.*

Each finite element basis function is associated with a point in the tensor product grid $\mathcal{S}$, and there are $m\,n$ points in the grid, so there are $m\,n$ finite element basis functions. Let $\{\phi_i\}_{i=1}^{m\,n}$ be the set of nodal finite element basis functions which spans the finite element space V. Then, the solution $u \in V$ can be written as

$$u(x,y) = \sum_{i=1}^{m\,n} u_i \phi_i(x,y).$$

Let $\mathbf{u} = (u_1, \cdots, u_{mn})^T \in \mathbb{R}^{m\,n}$ and K be the finite element stiffness matrix for the Laplace operator having (i,j)-th entry as $\int_\Omega \nabla\phi_i^T \nabla\phi_j \, dx\, dy$. Then, the algebraic formulation of our problem is to find the solution to the linear system

$$(A^T A + \lambda K)\mathbf{u} = A^T \mathbf{z}, \tag{7}$$

where the matrix A is of size $N \times m\,n$ and its (i,j)-th entry is $\phi_i(x_j, y_j)$.

Since the stiffness matrix K is to be computed in a structured mesh and there are many efficient approaches to compute it, our approach is very efficient and simple. Furthermore, the domain Ω and the structured mesh on it can be the same for different problems.

Remark 2. *If the image is to be recovered in the original tensor product partition $\mathcal{S}$, all points in $\mathcal{G}$ also belong to $\mathcal{S}$. Under the assumption that the set of finite element basis functions*

$$\{\phi_1, \cdots, \phi_{mn}\}$$

forms a nodal basis of V, the matrix $A^T A$ is a diagonal matrix of size $m\,n \times m\,n$ with a diagonal entry one at the position of a point in $\mathcal{G}$ and zero elsewhere.

IV. Numerical Results

This section is devoted to some examples of image denoising based on the techniques developed above. Assuming that we know the image before being corrupted with the noise, we use peak signal-to-noise ratio (PSNR) to compare our results with the results obtained by using the standard median filter and wavelets [35], [36]. Let I and $\hat{I}$ be the original image before corruption and the image recovered after removing the noise, respectively. The peak signal-to-noise ratio (PSNR) is defined as

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20 \cdot \log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right),$$

where MAX_I is the maximum pixel value of the image, and MSE is the mean square error, i.e.,

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n} \|I_{ij} - \hat{I}_{ij}\|^2.$$

In Figure 3, we show an example of applying the piecewise interpolation to remove the salt and pepper noise. The first picture (from the left) of Figure 3 shows the original image and the second one shows the noisy image with noise density 50%. The third is the image reconstructed by using the linear interpolation, and the fourth one is the image denoised by using the median filter. In this example, the nearest neighbor and cubic interpolation and finite element smoothing produce the image which are visually equivalent to the one produced by the linear interpolation. Although we can easily see the superiority of the finite element method, the median filter also recovers the image well. However, if the noise density is increased, the median filter does not recover even the main feature of the image. This has been explained in Figure 4, where the image is corrupted with noise density 95%. In Figure 4, the picture at the top left hand side shows the noisy image, the top middle picture shows the image reconstructed by using the nearest neighbor interpolation and the image recovered by the linear interpolation is shown in the top right hand side. On the bottom row of Figure 4, the picture on the left hand side shows the image reconstructed by using cubic interpolation, the middle picture is the one recovered by using the finite element smoothing and the image at the right hand side is reconstructed by using the median filter. We can see that the finite element interpolation and smoothing show very good performance here, and the median filter produces a meaningless image. Although we use $\lambda = 10^{-10}$ for the finite element smoothing, the recovered image with this method is smoother than with that of finite element interpolation. We have shown the peak signal-to-noise ration (PSNR) for the original and recovered images in Table I using two test images. Both images are corrupted with only salt and pepper noise with different noise densities as shown in Table I. For both images, the linear and cubic interpolation shows the best behavior and the nearest neighbor interpolation almost performs equally. Similarly, finite element smoothing using linear and bilinear elements also produce results equivalent to those produced by finite element interpolation. However, median filter performs well in case of low noise density, whereas at high noise density (above 50%), the performance of the median filter degrades considerably. Here, we have applied median filter with different window sizes for different noise densities.

In our next example, we consider images corrupted with salt and pepper noise as well as Gaussian noise. Here, we compare two approaches: one is based on applying finite element interpolation to remove the salt and pepper noise and then wavelets to clear the image from the Gaussian noise, and the other approach is based on applying finite element smoothing to remove both noises at the same time. We have applied our methods to two images as before: Lena's image and Baboon's image. In both images, 60% of the pixels are corrupted with the salt and pepper noise and the remaining pixels also carry the Gaussian noise with zero mean and 0.05 variance. We have used standard MATLAB *imnoise* function to create both noises. We note that the variance 0.05 refers to the image where the pixel values are scaled to be in $[0,1]$. We have shown Lena's image carrying both noises in the first picture of Figure 6 and reconstructed image by removing only the salt and pepper noise using linear interpolation in the second picture. The third one shows the image after smoothing the previous image by using wavelets. We have used the standard

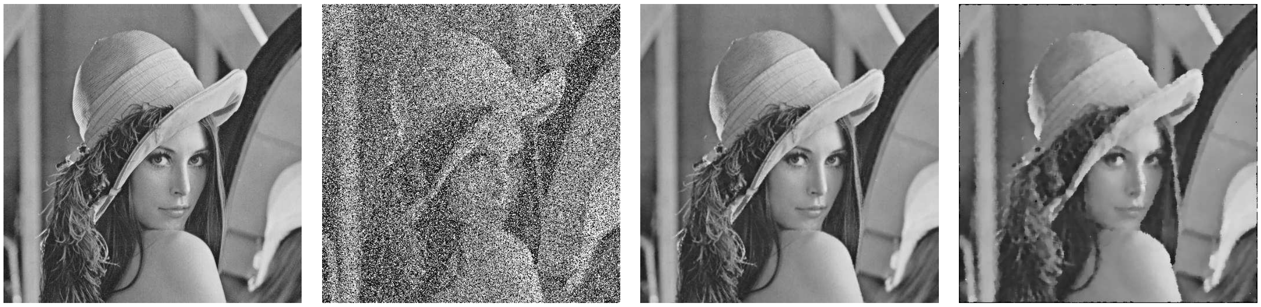


Fig. 3: Lena's image (first), noisy image (noise density 50%) (second), denoised with linear interpolation (third) and denoised with median filter (last)

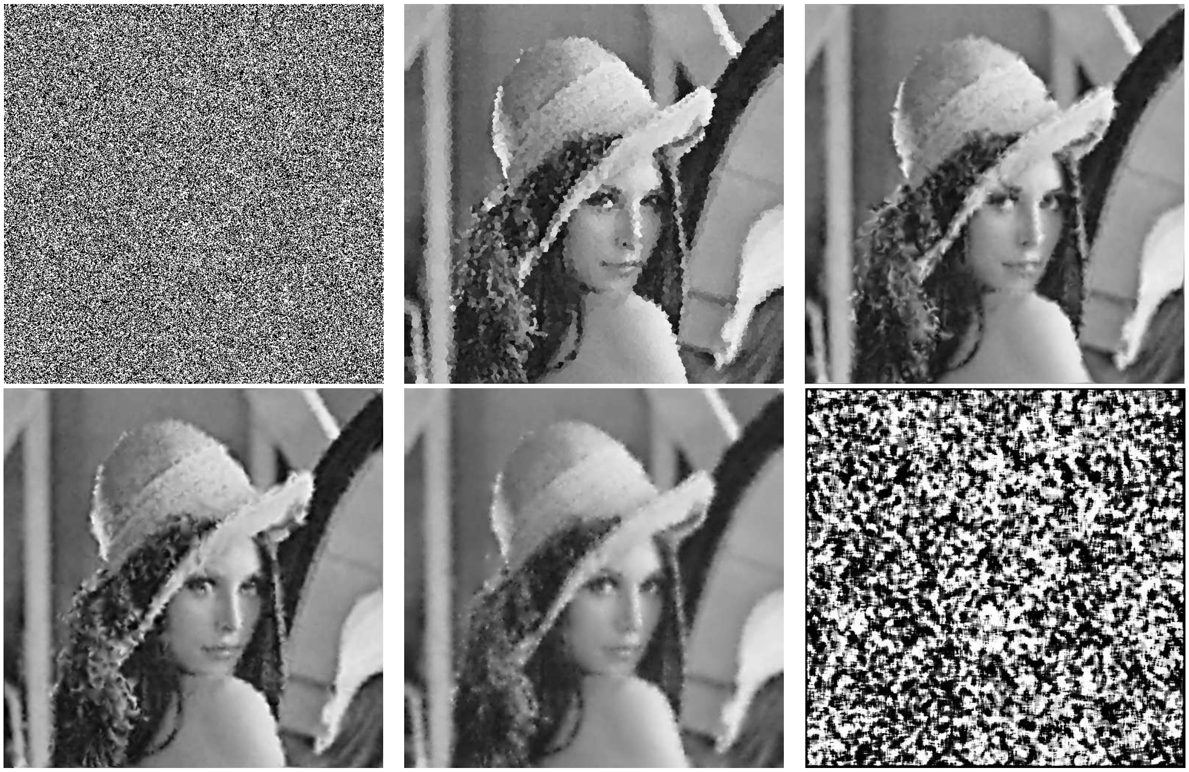


Fig. 4: Noisy image (95 % noise density) (top left), denoised by nearest neighbor interpolation (top middle), denoised by linear interpolation (top right), denoised by cubic interpolation (bottom left), denoised by linear finite element smoothing (bottom middle) and denoised by median filter (bottom right)

wavelet tool to clear the Gaussian noise by using global and soft thresholding, see [36], [37]. The image recovered by using finite element smoothing is shown in the last picture.

Figure 7 shows the same experiments done for Baboon's image using the same quantity of noise as for Lena's image. We can see that in Figures 6 and 7, wavelet method also smooth the image well. However, the images recovered by using the finite element smoothing are visually superior to those recovered by using finite element interpolation and wavelets. We have used a structured grid based on the tensor product partition as defined in (1) to compute the finite element space (3), where the number of vertices of the grid is equal to the number of image pixels. An example of such a grid having 25 vertices is shown in the left picture of Figure 2 using triangular elements. Similar grids can be defined for rectangular elements. Here, the positive constant λ is fixed to be 2.8. However, in the numerical results presented in Tables II–V, we use the estimated λ by using the generalized cross-validation [16] and stochastic trace estimator proposed in [38]. It is interesting to note that using the trace estimator in the generalized cross-validation to estimate λ gives a very good estimate of λ. We have plotted the exact error and generalized cross-validation function versus λ in two pictures of Figure 5 for Lena's image, where the exact error is scaled properly for the visualization purpose. In both pictures, the image is corrupted with Gaussian noise of variance 0.1, whereas 60% of the pixels are corrupted with impulsive noise in the left picture, and 70% of the pixels are corrupted in the right one.

Finally, we show peak signal-to-noise ratio for the original images and the reconstructed images, where the images are

TABLE I
PSNR OF THE TEST IMAGES WITH VARIOUS NOISE DENSITIES (ONLY SALT AND PEPPER NOISE)

	PSNR for Lena's image				PSNR for Baboon's image			
	Noise Density				Noise Density			
Method of Denoising	30%	50%	70%	90%	30%	50%	70%	90%
Interp. (nearest-neigh.)	33.94	31.11	28.12	24.36	23.79	21.8	19.98	17.78
Interp. (linear)	38.08	34.85	30.75	26.45	26.68	23.97	21.67	19.26
Interp. (cubic)	39.19	35.32	32.07	26.95	26.12	23.93	21.39	18.78
Smoothing (triangle)	38.23	34.54	30.36	26.15	27.15	24.27	22.01	19.81
Smoothing (rectangle)	36.83	33.52	29.79	25.85	26.29	23.74	21.76	19.76
Median Filter	24.50	22.76	17.90	9.43	19.41	18.19	15.56	9.21

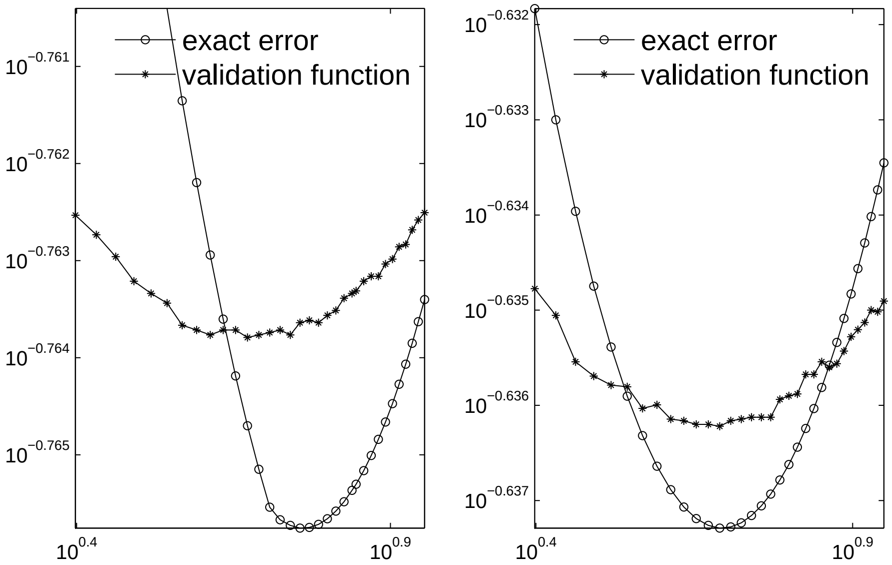


Fig. 5: Generalized cross-validation function and exact error versus λ for the mixture of Gaussian and impulsive noise with Gaussian noise variance: 0.1 (impulsive noise density: 60% (left) and 70% (right))

TABLE II
PSNR FOR THE MIXTURE OF GAUSSIAN AND SALT AND PEPPER NOISE, GAUSSIAN NOISE VAR: 0.05 AND IMAGE SIZE: 512×512

	PSNR for Lena's image				PSNR for Baboon's image			
	Noise Density				Noise Density			
Method of Denoising	50%	60%	70%	80%	50%	60%	70%	80%
FE Interp. and wavelets	22.24	21.86	20.95	19.46	18.57	18.26	17.76	17.05
FE Interp. and SURE	16.39	16.35	16.06	15.90	15.86	15.53	15.36	15.26
FE Smooth. (triangle)	22.42	22.05	21.59	20.91	18.82	18.57	18.46	18.38

corrupted with the mixture of Gaussian and salt and pepper noise in Tables II–V. The Gaussian noise with mean zero and variances 0.05 and 0.1 and salt and pepper noise with densities 50%, 60%, 70% and 80% are applied. We compare the finite element smoothing with two different methods based on wavelets. Both of these methods are applied after removing the salt and pepper noise from the image. One method referred to as "FE Interp. and wavelets" is the standard wavelet method to denoise an image by using global and soft thresholding, [36], [37], whereas the other method referred to as "FE Interp. and SURE" is based on SURE (Stein's unbiased risk estimate) and interscale orthonormal wavelet thresholding recently proposed in [39]. Since the method should be applied after getting the interpolated image, the noise variance estimator performs very badly in the second approach. Because of this reason, the performance of this method is worse than the standard wavelets. Therefore, we provide the results only for the case of Gaussian noise of variance 0.05. For the case of Gaussian noise of variance 0.1, we only compare our finite element smoothing approach with the standard wavelet method, which can be found in Tables IV–V.

In all Tables II–V, we can see a very good performance of the finite element smoothing to remove the mixture of Gaussian and impulsive noise. In particular, finite element smoothing performs better than wavelets in Tables IV–V, where the variance of the Gaussian noise is very high. The

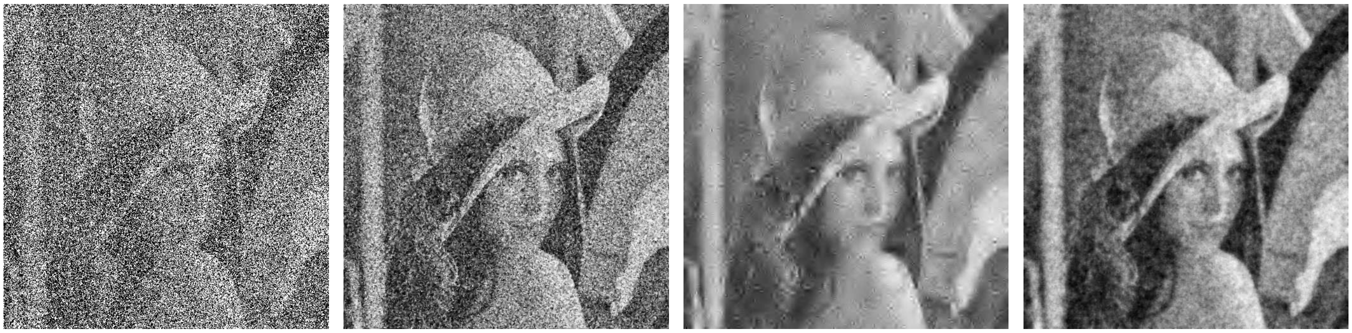


Fig. 6: Noisy image (first), image recovered by removing salt and pepper noise using finite element interpolation (second), image recovered by using wavelet smoothing in the second image (third) and denoised by linear finite element smoothing (last)

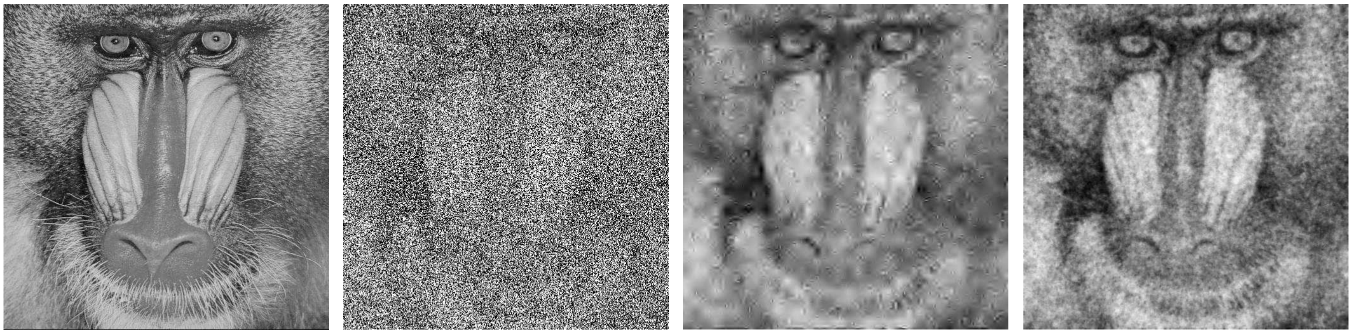


Fig. 7: Baboon's image (first), noisy image (second), image recovered by using finite element interpolation and wavelet smoothing (third), denoised by linear finite element smoothing (last)

TABLE III
PSNR for the mixture of Gaussian and salt and pepper noise, Gaussian noise var: 0.05 and image size: 256 × 256

	PSNR for Lena's image				PSNR for Baboon's image			
	Noise Density				Noise Density			
Method of Denoising	50%	60%	70%	80%	50%	60%	70%	80%
FE Interp. and wavelets	18.76	18.47	18.97	18.05	17.35	17.33	17.07	16.85
FE Interp. and SURE	16.25	15.88	15.67	15.71	15.13	14.84	14.57	14.31
FE Smooth. (triangle)	20.99	20.49	20.22	19.59	17.92	17.75	17.08	16.79

TABLE IV
PSNR for the mixture of Gaussian and salt and pepper noise, Gaussian noise var: 0.1 and image size: 512 × 512

	PSNR for Lena's image				PSNR for Baboon's image			
	Noise Density				Noise Density			
Method of Denoising	50%	60%	70%	80%	50%	60%	70%	80%
FE Interp. and wavelets	20.14	19.39	17.66	16.86	17.26	16.91	16.20	16.34
FE Smooth. (triangle)	20.80	20.33	20.27	19.87	18.08	18.0591	17.82	17.73

difference is more visible in Lena's image.

V. Conclusion

We have presented a method of removing the mixture of Gaussian and impulsive noise from images based on a finite element technique. The finite element method is applied to minimize a functional involving the gradient of a finite element function and is shown to be equivalent to a variational problem. We have proved the existence, uniqueness and stability of the variational problem by introducing a suitable inner product. Numerical results show that the approach based on finite element smoothing to remove the mixture of Gaussian and impulsive noise is quite efficient.

Acknowledgement

Support from EPSRC (EP/D062632/1) is gratefully acknowledged. I am grateful to the anonymous referees for their valuable suggestions to improve the quality of the earlier version of this work.

TABLE V

PSNR for the mixture of Gaussian and salt and pepper noise, Gaussian noise var: 0.1 and image size: 256×256

	PSNR for Lena's image				PSNR for Baboon's image			
	Noise Density				Noise Density			
Method of Denoising	50%	60%	70%	80%	50%	60%	70%	80%
FE Interp. and wavelets	17.71	16.55	16.17	16.09	16.57	15.57	15.54	15.41
FE Smooth. (triangle)	19.66	19.24	18.88	18.34	16.55	16.28	16.16	16.05

References

[1] K. Plataniotis and A. Venetsanopoulos, *Color Image Processing and Applications*, 1st ed. Springer, 2000.

[2] T. Chan and J. Shen, *Image Processing And Analysis: Variational, PDE, Wavelet, And Stochastic Methods*, 1st ed. SIAM, 2005.

[3] R. Gonzalez, R. Woods, and S. Eddins, *Digital Image Processing Using MATLAB*, 1st ed. Prentice Hall, 2004.

[4] T. Preusser and M. Rumpf, "An adaptive finite element method for large scale image processing," *Journal of Visual Communication and Image Representation*, vol. 11, pp. 183–195, 2000.

[5] M. Ferrant, A. Nabavi, B. Macq, F. Jolesz, R. Kikinis, and S. Warfield, "Registration of 3-d intraoperative MR images of the brain using a finite-element biomechanical model," *IEEE Transactions on Medical Imaging*, vol. 20, pp. 1384 – 1397, 2001.

[6] E. Besdok, "Impulsive noise suppression from images by using ANFIS interpolant and Lillietest," *EURASIP Journal on Applied Signal Processing*, vol. 2004, pp. 2423–2433, 2004.

[7] Z. Wang, F. Qi, and F. Zhou, "A discontinuous finite element method for image denoising," in *Image Analysis and Recognition*, ser. Lecture Notes in Computer Science. Springer Berlin/Heidelberg, 2006, vol. 4141.

[8] L. Demaret and A. Iske, "Adaptive image approximation by linear splines over locally optimal Delaunay triangulations," *IEEE Signal Processing Letters*, vol. 13, pp. 281–284, 2006.

[9] P. Civicioğlu and M. Alci, "Impulsive noise suppression from highly distorted images with triangular interpolants," *AEU - International Journal of Electronics and Communications*, vol. 58, pp. 311–318, 2004.

[10] F. R. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. Springer-Verlag, 1985.

[11] C. B. Barber, D. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software*, vol. 22, pp. 469–483, 1996.

[12] F. Aurenhammer and R. Klein, "Voronoi diagrams," in *Handbook of Computational Geometry*, J.-R. Sack and J. Urrutia, Eds. North-Holland, Amsterdam, Netherlands, 2000, pp. 201–290.

[13] J. Weickert, *Anisotropic Diffusion in Image Processing*. Stuttgart, Germany: Teubner-Verlag, 1998.

[14] G. Sapiro, *Geometric Partial Differential Equations and Image Processing*. Cambridge, UK: Cambridge University Press, 2001.

[15] J. Duchon, "Splines minimizing rotation-invariant semi-norms in Sobolev spaces," in *Constructive Theory of Functions of Several Variables, Lecture Notes in Mathematics*. Berlin: Springer-Verlag, 1977, vol. 571, pp. 85–100.

[16] G. Wahba, *Spline Models for Observational Data*, 1st ed., ser. Series in Applied Mathematic. SIAM, Philadelphia, 1990, vol. 59.

[17] S. Roberts, M. Hegland, and I. Altas, "Approximation of a thin plate spline smoother using continuous piecewise polynomial functions," *SIAM Journal on Numerical Analysis*, vol. 41, pp. 208–234, 2003.

[18] R. Chan, C. Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization," *IEEE Transaction on Image Processing*, vol. 14, pp. 1479–1485, 2005.

[19] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations (second edition)*, ser. Applied Mathematical Sciences. Springer-Verlag, 2006, vol. 147.

[20] P. Civicioğlu, M. Alci, and E. Besdok, "Impulsive noise suppression from images with the noise exclusive filter," *EURASIP Journal on Applied Signal Processing*, vol. 2004, pp. 2434–2440, 2004.

[21] I. Aizenberg and C. Butakoff, "Effective impulse detector based on rank-order criteria," *IEEE Signal Processing Letters*, vol. 11, pp. 363–366, 2004.

[22] R. Garnett, T. Huegerich, C. Chui, and W. He, "A universal noise removal algorithm with an impulse detector," *IEEE Transactions on Image Processing*, vol. 14, pp. 1747–1754, 2005.

[23] P. Civicioğlu, "Using uncorrupted neighborhoods of the pixels for impulsive noise suppression with ANFIS," *IEEE Transactions on Image Processing*, vol. 16, pp. 759–773, 2007.

[24] Y. Dong, R. Chan, and S. Xu, "A detection statistic for random-valued impulse noise," *IEEE Transactions on Image Processing*, vol. 16, pp. 1112–1120, 2007.

[25] R. Franke and G. Nielson, "Scattered data interpolation and applications: A tutorial and survey," in *Geometric Modelling: Methods and Their Application*, H. Hagen and D. Roller, Eds. Springer-Verlag, 1991, pp. 131–160.

[26] I. Amidror, "Scattered data interpolation methods for electronic imaging systems: a survey," *Journal of Electronic Imaging*, vol. 11, pp. 157–176, 2002.

[27] P. Ciarlet, *The Finite Element Method for Elliptic Problems*. North Holland, Amsterdam, 1978.

[28] S. Brenner and L. Scott, *The Mathematical Theory of Finite Element Methods*. Springer–Verlag, New York, 1994.

[29] A. Quarteroni and A. Valli, *Numerical approximation of partial differential equations*. Berlin: Springer–Verlag, 1994.

[30] D. Braess, *Finite Elements. Theory, fast solver, and applications in solid mechanics*. Cambridge University Press, Second Edition, 2001.

[31] B. Lamichhane and L. Rebollo-Neira, "Projection and interpolation based techniques for structured and impulsive noise filtering," in *New Research in Signal Processing*. New York: Nova Science Publisher, 2008.

[32] Q. Xu, L. Ma, M. Li, W. Wang, J. Cai, R. Brunelli, and S. Messelodi, "Fuzzy weighted average filtering for mixture noises," in *Third International Conference on Image and Graphics*, 2004, pp. 18–21.

[33] I. Altas, M. Hegland, and S. Roberts, "Finite element thin plate splines for surface fitting," in *Computational Techniques and Applications: CTAC97*, 1998, pp. 289–296.

[34] T. Ramsay, "Spline smoothing over difficult regions," *Journal of Royal Statistical Society. Series B (Statistical Methodology)*, vol. 64, pp. 307–319, 2002.

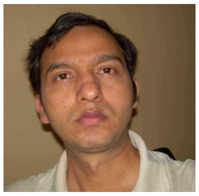
[35] J. Lim, *Two-Dimensional Signal and Image Processing*. Englewood Cliffs, NJ, Prentice Hall, 1990.

[36] D. Donoho, "De-noising by soft-thresholding," *IEEE Transaction on Information Theory*, vol. 41, pp. 613–627, 1995.

[37] D. Donoho and I. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, 1994.

[38] M. Hutchinson, "A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines," *Communications in Statisfics – Simulation and Computation*, vol. 18, pp. 1059–1076, 1989.

[39] F. Luisier, T. Blu, and M. Unser, "A new sure approach to image denoising: Interscale orthonormal wavelet thresholding," *IEEE Transactions on Image Processing*, vol. 16, pp. 593–606, 2007.



Bishnu P. Lamichhane received the M.Sc. degree in Industrial Mathematics from University of Kaiserslautern, Germany in 2001, and Ph.D. degree in applied mathematics from University of Stuttgart, Germany in 2006. The main part of this paper was completed when he was a research fellow at Aston University, UK.

He is now a postdoctoral research fellow at the Australian National University, Canberra, Australia. His research interests include finite element methods for differential equations, domain decomposition and variational approach in image processing.