Interpolation-Based Image Inpainting in Color Images Using High Dimensional Model Representation

Efsun Karaca*, M. Alper TUNGA†
Software Engineering Department, Faculty of Engineering and Natural Sciences
Bahcesehir University, Istanbul, TURKEY
Email: *efsun.karaca@eng.bau.edu.tr, †alper.tunga@eng.bau.edu.tr

***Abstract*—Image inpainting is the process of filling missing or fixing corrupted regions in a given image. The intensity values of the pixels in missing area are expected to be associated with the pixels in the surrounding area. Interpolation-based methods that can solve the problem with a high accuracy may become inefficient when the dimension of the data increases. We solve this problem by representing images with lower dimensions using High Dimensional Model Representation method. We then perform Lagrange interpolation on the lower dimensional data to find the intensity values of the missing pixels. In order to use High Dimensional Model Representation method and to improve the accuracy of Lagrange interpolation, we also propose a procedure that decompose missing regions into smaller ones and perform inpainting hierarchically starting from the smallest region. Experimental results demonstrate that the proposed method produces better results than the variational and exemplar-based inpainting approaches in most of the test images.**

I. Introduction

Image inpainting techniques are used in many problems such as repairing damaged photos, removing an object from a given image, completing missing regions [1], solving red eye problem [2] and image deblurring [3]. Image inpainting is a challenging problem since most of the images contain both structural and textural regions that lead to complicated patterns [4]. In the literature, there are image inpainting approaches which only focus on inpainting textural regions [5]–[7] as well as the ones that works only on structural regions [1], [8], [9]. There are also hybrid approaches that decompose a given image into structural and textural components, and apply structural inpainting to structural component and texture synthesis to textural component [4].

Texture synthesis algorithms are one of the oldest image inpainting techniques. These methods inpaint missing regions by exploiting the pixel intensity values of its neighbouring regions. In these methods, texture is synthesized pixel by pixel. They search similar pixels from neighbourhoods and inpaints the missing region by sampling and copying the intensity values of the most similar pixels [10]. Partial Differential Equation (PDE) based inpainting methods was first proposed by Bertalmio et al. [1]. Later, Chan and Shen proposed two PDE-based methods: Curvature Driven Diffusion (CDD) [11] and Total Variation (TV) [12]. These methods basically aim to complete missing regions by maintaining the structure of the surrounding area. Thus, these methods provide good results in small regions. However, as the region to be inpainted grows, the obtained results get blurry and worse. Exemplar-based image inpainting techniques can be used efficiently in larger missing regions. These algorithms differ from the texture synthesis based algorithms with their patch size. Similar patches instead of pixels are used to sample and copy to inpaint the missing regions [13]. In these methods, filling order of the pixels in the missing region and pre-determined sampling patch size plays an important role on accuracy of these methods. Other related works on image inpainting can be found in [14]–[20].

In this paper, we present a new interpolation-based image inpainting approach which is based on High Dimensional Model Representation (HDMR) and Lagrange interpolation. We consider image inpainting as an interpolation problem in which unknown pixel intensities are estimated by performing interpolation through known pixel intensities in the surrounding region. However, applying interpolation to a high dimensional data set is not a trivial task, even for $3D$ data as in color images, due to computational difficulties [21]. In order to deal with high dimensional data, we use HDMR [22] method and represent high dimensional data with lower dimensions. Then, we perform Lagrange interpolation through the outputs of HDMR for image inpainting. HDMR and Lagrange interpolation have already been successfully applied to high dimensional data in other applications in the literature [21], [23]–[25]. However, in image inpainting, HDMR brings some difficulties due to the orthogonality condition that comes from the derivation of the HDMR equation [23]. In order to satisfy the orthogonality condition for image inpainting using HDMR, pixels in the corresponding row or column of the missing region must also be considered as missing. We deal with this problem with a hierarchical approach in which we decompose missing regions into smaller regions and start inpainting from the smallest one.

The major contribution of this paper is a new image inpainting algorithm that is based on HDMR and Lagrange interpolation. To the best of our knowledge, this is the first image inpainting approach that is developed using HDMR and any kind of interpolation technique in the literature. We perform experiments on variety of test images and missing regions combinations. We also compare the accuracy of our approach

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

with two pioneering approaches: total variation inpainting [12] and exemplar-based inpainting [13]. Experimental results demonstrate that our approach produces better results than both approaches in most of the test images, especially in the ones containing more structural region.

II. Proposed Method

In this section, we first give mathematical background of the HDMR method. Then, we provide the formulation of Lagrange interpolation with HDMR. Finally, we introduce our proposed image inpainting method.

A. High Dimensional Model Representation

HDMR is a divide-and-conquer method which divides a multivariate function into less-variate functions [22], [23]. For a given multivariate f function, the HDMR expansion is given as follows:

$$f(x_1,\ldots,x_N) = f_0 + \sum_{i_1=1}^{N} f_{i_1}(x_{i_1}) + \sum_{\substack{i_1,i_2=1 \\ i_1<i_2}}^{N} f_{i_1 i_2}(x_{i_1},x_{i_2}) + \cdots + f_{12\ldots N}(x_1,\ldots,x_N) \tag{1}$$

where f_0, $f_{i_1}(x_{i_1})$, $f_{i_1 i_2}(x_{i_1},x_{i_2})$ and $f_{12\ldots N}(x_1,\ldots,x_N)$ represent constant term, univariate terms, bivariate terms and N-variate terms, respectively. These terms are determined uniquely using the following vanishing conditions

$$\int_{a_1}^{b_1} \mathrm{d}x_1 \ldots \int_{a_N}^{b_N} \mathrm{d}x_N W(x_1,\ldots,x_N) f_i(x_i) = 0, \;\; 1 \le i \le N \tag{2}$$

where

$$W(x_1,\ldots,x_N) = \prod_{j=1}^{N} W_j(x_j), \quad x_j \in [a_j,b_j], \quad 1 \le j \le N \tag{3}$$

and a_j and b_j are the lower and the upper bounds of data points in the j^{th} dimension, respectively. Also, the weight function of each dimension, $W_j(x_j)$, in Equation (3) should satisfy the following normalization condition [23]

$$\int_{a_j}^{b_j} \mathrm{d}x_j W_j(x_j) = 1, \quad 1 \le j \le N. \tag{4}$$

The vanishing condition given in Equation (2) corresponds to the following orthogonality condition via an inner product

$$\langle f_{i_1 i_2 \ldots i_k}, f_{i_1 i_2 \ldots i_l} \rangle = 0, \quad 1 \le k \ne l \le N. \tag{5}$$

The right-hand side components of Equation (1) must satisfy these orthogonality conditions. Using the properties of the weight function and the orthogonality condition, terms in Equation (1) can be obtained. To achieve this, both sides of Equation (1) are multiplied by the weight functions $\big($multiplied by $W_1(x_1)W_2(x_2)\ldots W_N(x_N)$ for constant term, $W_1(x_1)W_2(x_2)\ldots W_{i-1}(x_{i-1})W_{i+1}(x_{i+1})\ldots W_N(x_N)$ for univariate terms and so on$\big)$ and are integrated over whole Euclidean space defined by independent variables except x_i.

In a real application, since f function is unknown, the cartesian product of the independent variables $x_1,\ldots,x_N$ defined in Euclidean space and the known function values of the nodes in the cartesian product set are considered to approximate f. The cartesian product set can be written as follows:

$$\mathcal{D} \equiv \mathcal{D}_1 \times \mathcal{D}_2 \times \cdots \times \mathcal{D}_N \tag{6}$$

where

$$\mathcal{D}_i \equiv \left\{\xi_i^{(k_i)}\right\}_{k_i=1}^{k_i=n_i} = \left\{\xi_i^{(1)},\ldots,\xi_i^{(n_i)}\right\} \tag{7}$$

and $\xi_i^{(k_i)}$ represents the $k_i{}^{th}$ value of i^{th} independent variable. In our approach, we choose the weight function as

$$W_j(x_j) = \sum_{k_j=1}^{n_j} \alpha_{k_j}^{(j)} \delta\Big(x_j - \xi_j^{(k_j)}\Big), x_j \in [a_j,b_j], 1 \le j \le N \tag{8}$$

where $\delta(.)$ is the Dirac delta function and $\alpha_{k_j}^{(j)}$ is a constant which specifies the contribution level of each node to the model in which we set $\alpha_{k_j}^{(j)} = 1/N$ for all nodes, in our experiments.

Finally, constant and univariate terms given in Equation (1) can be written for cartesian set D as in Equation (9) and Equation (10), respectively. Higher variate terms can also be written in a similar manner.

$$f_0 = \sum_{k_1=1}^{n_1}\sum_{k_2=1}^{n_2}\cdots\sum_{k_N=1}^{n_N}\left(\prod_{i=1}^{N}\alpha_{k_i}^{(i)}\right) f\Big(\xi_1^{(k_1)},\ldots,\xi_N^{(k_N)}\Big) \tag{9}$$

$$f_m\Big(\xi_m^{(k_m)}\Big) = \sum_{k_1=1}^{n_1}\sum_{k_2=1}^{n_2}\cdots\sum_{k_{m-1}=1}^{n_{m-1}}\sum_{k_{m+1}=1}^{n_{m+1}}\cdots\sum_{k_N=1}^{n_N}\left(\prod_{i=1}^{N}\alpha_{k_i}^{(i)}\right) \times f\Big(\xi_1^{(k_1)},\ldots,\xi_{m-1}^{(k_{m-1})},\xi_m^{(k_m)},\xi_{m+1}^{(k_{m+1})},\ldots,\xi_N^{(k_N)}\Big) - f_0, \quad \xi_m^{(k_m)} \in \mathcal{D}_m, \quad 1 \le k_m \le n_m, \quad 1 \le m \le N \tag{10}$$

B. Lagrange Interpolation using HDMR

An exact f function passing through all the data points can be found by using all right-hand side terms in Equation (1). It has been shown in the literature that using the bivariate terms as the highest variate terms is sufficient for representing most of the multivariate functions [23]. The formula of the polynomial that is obtained by finding the terms in Equation (1) is given as follows:

$$f(x_1,\ldots,x_N) \approx f_0 + \sum_{m=1}^{N} P_m(x_m) + \sum_{\substack{m_1,m_2=1 \\ m_1<m_2}}^{N} P_{m_1 m_2}(x_{m_1},x_{m_2}) \tag{11}$$

where

$$P_m(x_m) = \sum_{k_m=1}^{n_m} L_{k_m}(x_m) f_m\Big(\xi_m^{(k_m)}\Big), \quad \xi_m^{(k_m)} \in \mathcal{D}_m, \quad 1 \le m \le N. \tag{12}$$

Note that $P_{m_1 m_2}(x_{m_1}, x_{m_2})$ can be computed similarly and $L_{k_m}(x_m)$ is the Lagrange polynomial which is given in Equation (13).

$$L_{k_m}(x_m) = \prod_{\substack{i=1 \\ i \neq k_m}}^{n_m} \frac{\left(x_m - \xi_m^{(i)}\right)}{\left(\xi_m^{(k_m)} - \xi_m^{(i)}\right)}, \xi_m^{(k_m)} \in \mathcal{D}_m, \quad 1 \leq k_m \leq n_m, \quad 1 \leq m \leq N \tag{13}$$

C. Image Inpainting using HDMR and Lagrange Interpolation

In this section, we introduce our image inpainting algorithm based on HDMR and Lagrange interpolation.

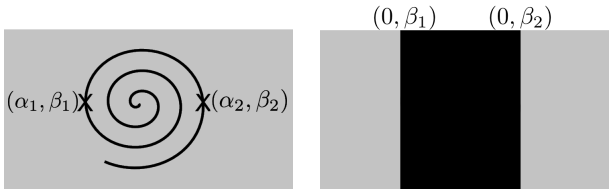
For a given $X \times Y \times Z$ image f, let $f(x, y, z)$ be the intensity value at x, y, z coordinates. Here, X, Y and Z represents the number of rows, columns and color channels ($Z = 2$ for grayscale, $Z = 3$ for color images), respectively. Then, the sets that are used to create the cartesian product given in Equation (6) can be written as follows:

$$\mathcal{D}_1 = \{1, 2, \ldots, X\}, \mathcal{D}_2 = \{1, 2, \ldots, Y\}, \mathcal{D}_3 = \{1, 2, \ldots, Z\}. \tag{14}$$

It has been shown in the literature that a given grayscale image can be exactly obtained with HDMR by using at most bivariate terms in the Equation (1) [26]. Therefore, in our approach for color image inpainting, we use constant, univariate, bivariate and trivariate terms in Equation (1).

As we mentioned in the previous section, orthogonality condition must be satisfied to apply HDMR to a data set [23]. Orthogonality condition requires that all the values of f to be known for all points in D. In image inpainting, since there are some pixel coordinates in D whose intensity values are unknown, the orthogonality condition is not satisfied. Therefore, we remove row indices corresponding to missing region (or column indices corresponding to missing region) from $\mathcal{D}_1$ (or $\mathcal{D}_2$) and construct new cartesian set D using new D_1, D_2 and D_3 sets. Let us assume that the intensity values in the black region shown in Figure 1a are missing. Coordinates of the leftmost and the rightmost pixels in the missing region are (α_1, β_1) and (α_2, β_2), respectively. 'X' indicates these pixels in Figure 1a. Let us also assume that we remove column indices corresponding to missing region from D_2. Then, the sets that construct the cartesian product D become $\mathcal{D}_1 = \{1, 2, \ldots, X\}$, $\mathcal{D}_2 = \{1, 2, \ldots, \beta_1 - 1, \beta_2 + 1, \ldots, Y\}$ and $\mathcal{D}_3 = \{1, 2, \ldots, Z\}$ and Lagrange interpolation followed by HDMR can be applied using new cartesian set D for inpainting.

When applying HDMR, the image inpainting problem in Figure 1a turns into inpainting the image shown in Figure 1b. Note the significant increase of the missing region with the changes that we made to satisfy the orthogonality condition. There is a trade-off between satisfying the orthogonality condition and accuracy of the inpainting because of the increasing size of the region to be inpainted. We use an hierarchical image inpainting procedure to solve this trade-off. In each iteration of this procedure, we search the image both vertically and horizontally to find the smallest missing region whose immediate neighbours are known in the search direction. A patch is created containing only the found missing region and its immediate known neighbouring pixels. Then, D is constructed with respect to indices of the patch. Once the HDMR and Lagrange interpolation is applied to find the missing pixel values in this patch, the found pixel values are put to their original location in the image.



(a) (b)

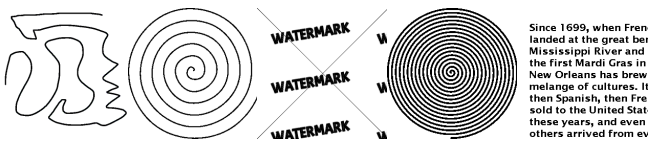
Fig. 1. (a) Original missing region, (b) missing region after orthogonality condition is satisfied.

III. Experimental Results

In this section, we present experimental results of our image inpainting approach. We perform experiments on 3 different test images shown in Figure 2. We design 15 different test settings by using each test image with 5 different masks shown in Figure 3. Note that black regions in each mask represent the missing region in the corresponding test setting. We compare our approach with two pioneering inpainting approaches in the literature: total variation inpainting [12] and exemplar-based inpainting [13].

(a) House (b) Lena (c) Barbara

Fig. 2. Test images.



(a) Scratch (b) Spiral-1 (c) Watermark (d) Spiral-2 (e) Text

Fig. 3. Masks representing missing regions.

We obtain quantitative results by comparing inpainting results of each method with the original images using peak signal-to-noise ratio (PSNR) which is computed as follows:

$$PSNR = 20.\log(MAX_I) - 10.\log(MSE) \tag{15}$$

where

$$MSE = \frac{1}{XYZ}\sum_{i=1}^{X}\sum_{j=1}^{Y}\sum_{k=1}^{Z}(I(i,j,k) - \hat{I}(i,j,k))^2. \quad (16)$$

MAX_I is the maximum possible pixel value of the image, I and $\hat{I}$ are original and inpainted images, respectively. Note that higher values in PSNR mean better results.

PSNR results for the test image given in Figure 2a are shown in Table I. Results demonstrate that the proposed inpainting approach produces better results than both state-of-the-art methods in 4 test cases. The exemplar-based method in [13] produces the best result with the mask shown in Figure 3a in terms of PSNR. In this test case, the result of our approach is very close to the best result and better than the result of the method in [12].

Mask	Method in [12]	Method in [13]	Proposed Method
Scratch	37.9607	**38.2588**	38.0464
Spiral-1	33.0644	35.4030	**35.9883**
Watermark	31.2465	30.0557	**32.4042**
Spiral-2	33.3071	32.0079	**34.2790**
Text	27.1687	24.2372	**29.7899**

TABLE I
PSNR RESULTS OF THE TEST IMAGE IN FIGURE 2A WITH 5 DIFFERENT MASKS.

Table II contains the PSNR values for the test image given in Figure 2b with 5 different masks. In this experiment, our approach produces the highest PSNR values in all test cases.

Mask	Method in [12]	Method in [13]	Proposed Method
Scratch	35.9661	33.6417	**36.4427**
Spiral-1	35.5414	33.3323	**37.1111**
Watermark	30.1658	28.9228	**31.3753**
Spiral-2	31.2359	28.7694	**31.4559**
Text	27.8587	24.3407	**28.6769**

TABLE II
PSNR RESULTS OF THE TEST IMAGE IN FIGURE 2B WITH 5 DIFFERENT MASKS.

We present PSNR results for test image in Figure 2c with 5 different masks in Table III. In this test setting, the approach in [13] produces the best results in 3 masks whereas our approach achieves the best PSNR values on the remaining 2 test cases.

Mask	Method in [12]	Method in [13]	Proposed Method
Scratch	32.6628	**33.1591**	32.6184
Spiral-1	29.5369	**31.8904**	31.1452
Watermark	28.3516	27.4568	**28.3986**
Spiral-2	26.1509	**28.7713**	27.4096
Text	23.9638	22.3270	**24.5221**

TABLE III
PSNR RESULTS OF THE TEST IMAGE IN FIGURE 2C WITH 5 DIFFERENT MASKS.

Test images given in Figures 2a and 2b contains more structural patterns relative to the textural ones. Therefore, our interpolation-based inpainting approach produces better results than the other two approaches in the literature in most of the test cases. The test image in Figure 2c contains many textural regions like the scarf of the lady and the chair in the background. Since, the examplar-based approach in [13] performs image inpainting by copying similar patterns, it is capable of inpainting textural images. Although, the proposed approach does not have the mechanism for inpainting textural images, PSNR values are very close to results of the method in [13].

Finally, we present some visual results in Figure 4. These results also demonstrate that the proposed approach achieves better inpainting performance than the other methods in most of the test cases.

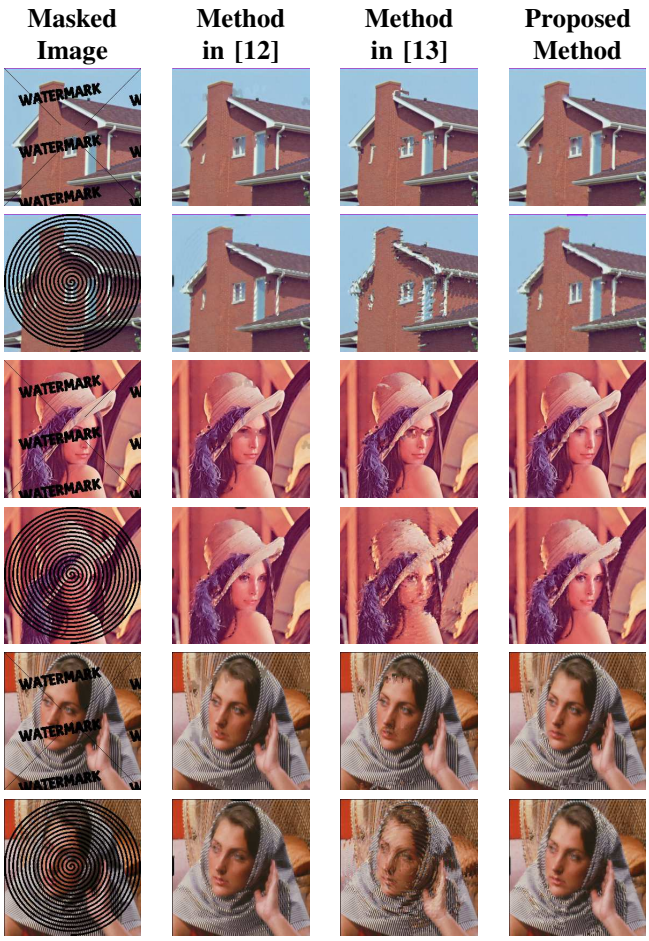


Fig. 4. Some visual inpainting results.

IV. CONCLUSION

In this work, we propose a new inpainting algorithm which uses HDMR method and Lagrange interpolation. The results of the proposed method are compared with the results of TV-based [12] and exemplar-based [13] inpainting methods which are two pioneering methods in literature. The results demonstrate that our approach produces better results than both inpainting methods in most of the test cases, especially in the ones containing more structural regions.

The proposed method assumes that all columns or rows that include the pixels in the missing region are also missing due to the orthogonality constraint of HDMR. One possible future

research direction is to use indexing HDMR method which does not require orthogonality condition [27]. We are also planning to combine our approach with a texture synthesis method to achieve better accuracy in textural images. Finally, using HDMR method with different interpolation or regression methods worth investigating.

V. Acknowledgment

This work has been supported by the Scientific and Technological Research Council of Turkey (TUBITAK) under Grant 115E424.

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.

[2] S. Yoo and R. H. Park, "Red-eye detection and correction using inpainting in digital photographs," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 3, pp. 1006–1014, 2009.

[3] T. F. Chan, A. M. Yip, and F. E. Park, "Simultaneous total variation image inpainting and blind deconvolution," *International Journal of Imaging Systems and Technology*, vol. 15, no. 1, pp. 92–102, 2005.

[4] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 882–889, 2003.

[5] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. ACM, 1995, pp. 229–238.

[6] E. P. Simoncelli and J. Portilla, "Texture characterization via joint statistics of wavelet coefficient magnitudes," in *International Conference on Image Processing (ICIP)*. IEEE, 1998, pp. 62–66.

[7] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *International Conference on Computer Vision (ICCV)*. IEEE, 1999, pp. 1033–1038.

[8] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Transactions on Image Processing*, vol. 10, no. 8, pp. 1200–1211, 2001.

[9] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Computer Vision and Pattern Recognition (CVPR)*, vol. 1. IEEE, 2001, pp. I–355.

[10] J. K. Chhabra and M. V. Birchha, "Detailed survey on exemplar based image inpainting techniques," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 5, pp. 6350–635, 2014.

[11] T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *Journal of Visual Communication and Image Representation*, vol. 12, no. 4, pp. 436–449, 2001.

[12] J. Shen and T. F. Chan, "Mathematical models for local nontexture inpaintings," *SIAM Journal on Applied Mathematics*, vol. 62, no. 3, pp. 1019–1043, 2002.

[13] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[14] O. G. Guleryuz, "Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part ii: adaptive algorithms," *IEEE Transactions on Image Processing*, vol. 15, no. 3, pp. 555–571, 2006.

[15] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, 2007.

[16] X. Zhang, M. Burger, X. Bresson, and S. Osher, "Bregmanized nonlocal regularization for deconvolution and sparse reconstruction," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 253–276, 2010.

[17] X. Li, "Image recovery via hybrid sparse representations: A deterministic annealing approach," *Selected Topics in Signal Processing*, vol. 5, no. 5, pp. 953–962, 2011.

[18] M. Zhou, H. Chen, J. Paisley, L. Ren, L. Li, Z. Xing, D. Dunson, G. Sapiro, and L. Carin, "Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 130–144, 2012.

[19] W. Dong, G. Shi, and X. Li, "Nonlocal image restoration with bilateral variance estimation: a low-rank approach," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 700–711, 2013.

[20] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3336–3351, 2014.

[21] M. A. Tunga and M. Demiralp, "Computational complexity investigations for high-dimensional model representation algorithms used in multivariate interpolation problems," in *Advances in Numerical Methods*. Springer, 2009, pp. 15–29.

[22] I. Sobol, "Sensitivity estimates for nonlinear mathematical models," *Math. Model. Comput. Exp. (MMCE)*, vol. 1, no. 1, pp. 407–414, 1993.

[23] M. A. Tunga and M. Demiralp, "A new approach for data partitioning through high dimensional model representation," *International Journal of Computer Mathematics*, vol. 85, no. 12, pp. 1779–1792, 2008.

[24] A. Karahoca and M. A. Tunga, "A polynomial based algorithm for detection of embolism," *Soft Computing*, vol. 19, no. 1, pp. 167–177, 2015.

[25] Ö. F. Alış and H. Rabitz, "Efficient implementation of high dimensional model representations," *Journal of Mathematical Chemistry*, vol. 29, no. 2, pp. 127–142, 2001.

[26] E. M. Altn and B. Tunga, "High dimensional model representation in image processing," in *10th International Conference on Computational and Mathematical Methods in Science and Engineering (CMMSE)*, vol. 1, 2014, pp. 55–64.

[27] M. A. Tunga, "An approximation method to model multivariate interpolation problems: Indexing hdmr," *Mathematical and Computer Modelling*, vol. 53, no. 9, pp. 1970–1982, 2011.