THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY

DIGITAL IMAGE INPAINTING USING HIGH DIMENSIONAL MODEL REPRESENTATION BASED METHODS

Ph.D. Thesis

EFSUN KARACA

İSTANBUL, 2018

THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
COMPUTER ENGINEERING

DIGITAL IMAGE INPAINTING USING HIGH DIMENSIONAL MODEL REPRESENTATION BASED METHODS

Ph.D. Thesis

EFSUN KARACA

Supervisor: ASSOC. PROF. DR. M. ALPER TUNGA

İSTANBUL, 2018

THE REPUBLIC OF TURKEY
BAHÇEŞEHİR UNIVERSITY
The Graduate School of Natural and Applied Sciences
Computer Engineering

Title of the Ph.D. Thesis	:	Digital Image Inpainting Using High Dimensional Model Representation Based Methods
Name/Last Name of the Student	:	Efsun KARACA
Date of Thesis Defense	:	January 25, 2018

The thesis has been approved by The Graduate School of Natural and Applied Sciences.

Prof. Dr. Nafiz Arıca
Graduate School Director

I certify that this thesis meets all the requirements as a thesis for the degree of Doctor of Philosophy.

Assist. Prof. Dr. Tarkan Aydın
Program Coordinator

This is to certify that we have read this thesis and that we find it fully adequate in scope, quality and content, as a thesis for the degree of Doctor of Philosophy in Computer Engineering Department.

Examining Commitee Members:	Signature
Assoc. Prof. Dr. M. Alper Tunga (Supervisor)	: ....
Prof. Dr. Adem Karahoca	: ....
Assoc. Prof. Dr. Ahmet Kırış	: ....
Assoc. Prof. Dr. Devrim Ünay	: ....
Assist. Prof. Dr. Cemal Okan Şakar	: ....

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor Assoc. Prof. Dr. M. Alper Tunga for the continuous support of my Ph.D study and related research, for his patience, motivation, and immense knowledge.

Besides my advisor, I would like to thank the rest of my thesis committee: Prof. Dr. Adem Karahoca, Assoc. Prof. Dr. Devrim Ünay, Assoc. Prof. Dr. Ahmet Kırış and Assist. Prof. Dr. Cemal Okan Şakar for their insightful comments and encouragement.

I would like to also thank the Scientific and Technological Research Council of Turkey (TUBITAK) for providing me financial supports throughout my PhD studies with the National Graduate Scholarship Programme (BIDEB2211) and The Starting R&D Projects Funding Programme (3001 - Grant No: 115E424).

I thank my friend and colleague Assist. Prof. Dr. Selçuk Keskin for his support throughout the whole process. I also thank my friend Özge Yücel Kasap for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last few years.

Last but not the least, I would like to thank my family: my parents and my sisters for supporting me spiritually throughout writing this thesis and my life in general. I would like to extend my sincerest thanks and appreciation to my husband, Ertunç Erdil. His unconditional love, patience, continual support and guidance helped me in all the time of research and writing, and enabled me to complete this thesis.

İstanbul, 2018 Efsun KARACA

ABSTRACT

DIGITAL IMAGE INPAINTING USING HIGH DIMENSIONAL MODEL REPRESENTATION BASED METHODS

Efsun Karaca

Computer Engineering
Supervisor: Assoc. Prof. Dr. M. Alper Tunga

January 2018, 74 Pages

Image inpainting is the process of filling missing or fixing corrupted regions in an image. The intensity values of the pixels in the 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.

In this thesis, we first propose a method to inpaint rectangular missing regions in grayscale images. Then, just by adding one more term into the High Dimensional Model Representation method, we propose a method to inpaint rectangular regions in color images. Experimental results show that the proposed method produces better results than the well-known and pioneering total variation-based image inpainting method in the literature. However, these methods can be used only in rectangular missing regions and if the missing region grows, the accuracy of the inpainting results decrease. Therefore, we propose a new hierarchical image inpainting approach to solve the trade-off between satisfying the orthogonality condition and accuracy of the inpainting because of the increasing size of the region to be inpainted. 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. This procedure decomposes missing regions into smaller ones and performs 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, especially in the ones containing more structural regions. All of these methods suffer from finding the underlying texture and pattern in the missing region. In this thesis, we propose a texture and pattern preserving interpolation-based algorithm for inpainting missing regions in color images. First, the proposed approach produces candidate inpainting results by interpolating to the observed data at the different neighborhoods of the missing region using High Dimensional Model Representation with Lagrange interpolation. Later, a final inpainting decision is given among the candidates for each pixel in the missing region for a texture and pattern preserving inpainting. This is achieved by combining the information obtained from co-occurrence matrix and from a patch found in the image that fits best to the missing region using normalized cross correlation. We evaluate the performance of the proposed approach on various color images that include different texture and pattern. We also compare the proposed approach with the state-of-the-art inpainting methods in the literature. Experimental results demonstrate the potential of the proposed approach.

Keywords: Digital image inpainting, High Dimensional Model Representation, Lagrange Interpolation.

ÖZET

YÜKSEK BOYUTLU MODEL GÖSTERİLİM TABANLI YÖNTEMLERLE SAYISAL İMGE İÇBOYAMA

Efsun Karaca

Bilgisayar Mühendisliği
Tez Danışmanı: Doç. Dr. M. Alper Tunga

Ocak 2018, 74 Sayfa

İmge içboyama, imgelerdeki eksik kısımların tamamlanması ya da bozuk kısımların düzeltilmesi süreci olarak tanımlanır. İmgede bulunan eksik kısımların yeğinlik değerlerinin, çevrede bulunan yeğinlik değerleriyle ilişkili olması beklenir. Problemin çözümünde yüksek başarım gösterebilecek aradeğerleme tabanlı yöntemler, aradeğerlenecek verinin boyutu arttıkça verimsiz hale gelir.

Bu tezde ilk olarak gri tonlamalı imgelerdeki dikdörtgensel kayıp alanların tamamlanması için bir yöntem önerilmiştir. Daha sonra, Yüksek Boyutlu Model Gösterilim yöntemine yalnızca bir terim daha ekleyerek, aynı işlemin renkli imgelerde de kullanılmasını sağlayan yeni bir yöntem önerilmektedir. Deneysel sonuçlar, önerilen yöntemin iyi bir yöntem olarak bilinen ve alanında öncü yöntemlerden olan toplam varyasyon tabanlı imge içboyama yönteminden daha iyi sonuçlar ürettiğini göstermiştir. Ancak bu yöntemler yalnızca dikdörtgensel alanlarda kullanılabilmektedir ve kayıp alan büyüdükçe başarı oranının düştüğü gözlemlenmiştir. Bu bağlamda, hem diklik koşulu kısıtını aşmak, hem de içboyamanın doğruluğunu arttırabilmek için hiyerarşik bir yapıyla içboyama yapan yeni bir imge içboyama yaklaşımı önerilmiştir. Bu yaklaşımın her iterasyonunda imgeyi yatay ve dikey düzlemde arayarak, aranan yöndeki en yakın komşuluklarının yeğinlik değerlerinin bilindiği en küçük kayıp alan bulunmaktadır. Bu işlem kayıp alanı daha küçük kayıp alanlara bölmekte ve hiyerarşik olarak en küçük kayıp alandan başlayarak içboyama işlemini gerçekleştirmektedir. Deneysel sonuçlar, önerilen yöntemin, çoğu test kümesinde, özellikle daha yapısal bölgelere sahip olan kayıp alanların olduğu test örneklerinde, varyasyon ve örneklem tabanlı içboyama yöntemlerine göre daha iyi sonuçlar verdiğini göstermiştir. Tüm bu önerilen yöntemler kayıp alanın altında yatan doku ve deseni bulmakta zorlandığı gözlemlenmiştir. Bu tezde, renkli imgelerdeki kayıp alanları tamamlarken, doku ve deseni koruyan aradeğerleme tabanlı bir imge içboyama yöntemi önerilmektedir. Önerilen yöntem ilk olarak, Yüksek Boyutlu Model Gösterilim ile Lagrange aradeğerleme yöntemlerini kullanır ve kayıp alanın farklı komşuluklarında gözlenen verileri aradeğerleyerek aday içboyama sonuçları üretir. Daha sonra, kayıp alandaki desen ve dokuyu korumayan bir içboyama yapabilmek için en uygun aday içboyama sonucunu seçilir. Bu işlem, ortak oluşum matrisi ve normalleştirilmiş çapraz korelasyon yöntemleri kullanılarak ve imge içerisinde kayıp alana en çok benzeyen yama bulunarak yapılır. Önerilen yöntemin performansı farklı desen ve dokular içeren birçok farklı renkli imgede test edilmiştir. Aynı zamanda önerilen yöntemin sonuçları literatürde alanında öncü olarak bilinen farklı yöntemler ile karşılaştırılmıştır. Deneysel sonuçlar önerilen yöntemin potansiyelini gösterir niteliktedir.

Anahtar Kelimeler: Sayısal İmge İçboyama, Yüksek Boyutlu Model Gösterilim, Lagrange Aradeğerleme.

CONTENTS

TABLES

FIGURES

ABBREVIATIONS

BPFA	:	Beta-Bernoulli Process Factor Analysis
CPU	:	Central Processing Unit
GSR	:	Group-based Sparse Representation
HDMR	:	High Dimensional Model Representation
HSR	:	Hybrid Sparse Representation
MSE	:	Mean Squared Error
NLTV	:	Non-Local Total Variation
PSNR	:	Peak Signal-to-Noise Ratio
RGB	:	Red-Green-Blue
SAIST	:	Spatially Adaptive Iterative Singular-value Thresholding
TPI-HDMR	:	Texture Preserving Inpainting using HDMR
TV	:	Total Variation

SYMBOLS

Angle	:	θ
Co-occurence Matrix	:	C
Extracted Patch around Missing Region	:	P
Image	:	$\mathbf{I}$
Inpainting Estimate	:	$\hat{M}$
Inpainting Result of TPI-HDMR Method	:	$M^{\star}$
Missing Region	:	M
Most Similar Patch to P	:	$\hat{P}$
Normalized Cross Correlation between $\hat{M}$ and M	:	c
Number of occurences for pixels in Co-occurence Matrix	:	v
Set of Neighboring Pixels	:	$\mathcal{X}$

1. INTRODUCTION

Image inpainting is the process of filling missing or fixing corrupted regions in an image. The intensity values of the pixels in the 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.

Image inpainting techniques are used in many problems such as repairing damaged photos, removing an object from a given image, completing missing regions (Bertalmio et al., 2000), solving red eye problem (Yoo & Park, 2009) and image deblurring (Chan et al., 2005). Image inpainting is a challenging problem since most of the images contain both structural and textural regions that lead to complicated patterns (Bertalmio et al., 2003). In the literature, there are image inpainting approaches which only focus on inpainting textural regions (Heeger & Bergen, 1995; Simoncelli & Portilla, 1998; Efros & Leung, 1999) as well as the ones that works only on structural regions (Bertalmio et al., 2000; Ballester et al., 2001; Bertalmio et al., 2001). 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 (Bertalmio et al., 2003).

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 (Chhabra & Birchha, 2014). Partial Differential Equation (PDE) based inpainting methods was first proposed by Bertalmio et al. (Bertalmio et al., 2000). Later, Chan and Shen proposed two PDE-based methods: Curvature Driven Diffusion (CDD) (Chan & Shen, 2001) and Total Variation (TV) (Shen & Chan, 2002). 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 (Criminisi et al., 2004). 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 (Guleryuz, 2006; Takeda et al., 2007; Zhang et al., 2010; Li, 2011; Zhou et al., 2012; Dong et al., 2013; Zhang et al., 2014).

There are many works proposed in the literature for image inpainting. One of the pioneering image inpainting methods, Total Variation (TV), was proposed by (Shen & Chan, 2002). TV is a partial differential equation based inpainting method that optimizes an energy function designed for maintaining the intensity distribution of the surrounding area. Then, (Zhang et al., 2010) proposed a method called Non-Local Total Variation (NLTV). NLTV extends TV by adding a term to the energy function that considers non-local constraints for inpainting. Both TV and NLTV produces good inpainting results in only smooth regions. However, as the region to be inpainted includes complex pattern and texture, the obtained results get blurry and worse. Exemplar-based image inpainting techniques are proposed for inpainting larger missing regions that include texture and pattern. These methods find the most probable patch within the image for inpainting the missing region. Finally, the patch is returned as an inpainting solution. These methods suffer if illumination varies in different parts of the image (Criminisi et al., 2004). Takeda et al. (2007) adapted and expanded kernel regression for different applications in image processing such as image denoising, upscaling, and interpolation. Although, the method produces promising inpainting results in missing regions with smooth intensities, the performance of the method in textural missing region are rather limited. Hybrid Sparse Representation (HSR) method uses the strengths of local and nonlocal sparse representations by Bayesian model averaging where the usage of local smoothness and nonlocal similarity have allowed exploitation of sparsity priors for image recovery applications (Li, 2011). Beta-Bernoulli Process Factor Analysis (BPFA) model uses several hierarchical Bayesian models to learn dictionaries for analysis of imagery with applications in inpainting (Zhou et al., 2012). The method requires a large training set for an effective learning which may not be available or expensive to obtain in many applications. Spatially Adaptive Iterative Singular-value Thresholding (SAIST) is an image restoration algorithm which connects low-rank methods with simultaneous sparse coding and provides a conceptually simple interpretation from a bilateral variance estimation perspective (Dong et al., 2013). Both BFPA and SAIST suffer from two problems: 1) they have to solve a large-scale optimization problem with high computational complexity in dictionary learning, 2) each patch is considered independently in dictionary learning and sparse coding, which ignores the relationship among patches, resulting in inaccurate sparse coding coefficients. Group-based Sparse Representation (GSR) deals with these problems by introducing the concept of group as the basic unit of sparse representation to capture the patch relations and to reduce the computational complexity (Zhang et al., 2014). Karaca & Tunga (2016a) proposes an interpolation-based image inpainting approach using Lagrange interpolation. The proposed method works well if the missing region is part of a smooth background. As in the other interpolation-based algorithms (Ballester et al., 2001; Karaca & Tunga, 2016b), the method proposed by Karaca & Tunga (2016a) is not able to capture and preserve underlying pattern and texture in the region to be inpainted.

The image inpainting process requires having a prior knowledge about the part to be completed and estimates the missing region accordingly. The apparent part of the image may give us information about the structure of whole image. Conventional interpolation techniques requires a lot of computation power when dimension of the data increases. This motivates us to apply divide-and-conquer methods to reduce the computational complexity and CPU time that is needed to interpolate an image.

High Dimensional Model Representation (HDMR) is a divide-and-conquer algorithm to represent a multivariate function in terms of less-variate functions and it partitions a high dimensional data into a number of sets of lower dimensional data such as univariate, bivariate and trivariate ones. Reducing the complexity of the multivariate interpolation problem to univariate, bivariate and trivariate interpolations enables us to apply interpolation methods in a more efficient way both in terms of computational complexity of the problem and required CPU time.

In this thesis, we propose to complete missing parts in a given image using interpolation based methods in a fast and efficient way with the help of HDMR. This thesis is structured as follows. Section 2 introduces HDMR Method. We explain the HDMR method and show how data is partitioned through HDMR. Then we illustrate the image representation by applying HDMR to grayscale and color image data. Performance evaluation method is alsow shown in this section. HDMR based inpainting methods are exlained in Section 3. We first show how to inpaint rectangular missing regions in grayscale and color images. Then, we also show how to inpaint complicated missing regions using a new HDMR based hierarchical approach. The major contribution of this thesis is exlained in Section 4. This section examines the Texture Preserving Inpainting using HDMR (TPI-HDMR) method in detail. The findings of the proposed method are given in Section 5. Finally, the thesis concludes in Section 6 with the conclusion and discussion.

2. METHODS

2.1 HIGH DIMENSIONAL MODEL REPRESENTATION METHOD

In this section, we first give mathematical background of the High Dimensional Model Representation (HDMR) method. Then, we provide the formulation of Lagrange interpolation with HDMR.

HDMR is a divide-and-conquer method which divides a multivariate function into less-variate functions (Sobol, 1993; Tunga & Demiralp, 2008). For a given multivariate $\mathcal{F}$ function, the HDMR expansion is given as follows:

$$\begin{aligned}\mathcal{F}(x_1,\ldots,x_N) = f_0 + \sum_{i_1=1}^{N} f_{i_1}(x_{i_1}) + \sum_{\substack{i_1,i_2=1\\ i_1<i_2}}^{N} f_{i_1 i_2}(x_{i_1},x_{i_2}) + \sum_{\substack{i_1,i_2,i_3=1\\ i_1<i_2<i_3}}^{N} f_{i_1 i_2 i_3}(x_{i_1},x_{i_2},x_{i_3}) \\ + \cdots + f_{12\ldots N}(x_1,\ldots,x_N)\end{aligned} \tag{2.1}$$

where f_0, $f_{i_1}(x_{i_1})$, $f_{i_1 i_2}(x_{i_1},x_{i_2})$, $f_{i_1 i_2 i_3}(x_{i_1},x_{i_2},x_{i_3})$ and $f_{12\ldots N}(x_1,\ldots,x_N)$ represent constant term, univariate terms, bivariate terms, trivariate terms and N-variate terms, respectively (Sobol, 1993). These terms are determined uniquely using the following vanishing conditions

$$\int_{a_1}^{b_1} \mathrm{d}x_1 \ldots \int_{a_N}^{b_N} \mathrm{d}x_N W(x_1,\ldots,x_N) f_i(x_i) = 0, \;\; 1 \le i \le N \tag{2.2}$$

where

$$W(x_1,\ldots,x_N) = \prod_{j=1}^{N} W_j(x_j), \;\; x_j \in [a_j,b_j], \;\; 1 \le j \le N \tag{2.3}$$

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 (2.3) should satisfy the following normalization condition (Tunga & Demiralp, 2008)

$$\int_{a_j}^{b_j} \mathrm{d}x_j W_j(x_j) = 1, \;\; 1 \leq j \leq N. \tag{2.4}$$

The vanishing condition given in Equation (2.2) corresponds to the following orthogonality condition via an inner product

$$\langle f_{i_1 i_2 \ldots i_k}, f_{i_1 i_2 \ldots i_l} \rangle = 0, \;\; 1 \leq k \neq l \leq N. \tag{2.5}$$

The right-hand side components of Equation (2.1) must satisfy these orthogonality conditions. Using the properties of the weight function and the orthogonality condition, terms in Equation (2.1) can be obtained. To achieve this, both sides of Equation (2.1) 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.

Using the properties of the weight function and the orthogonality conditions, the constant term of the HDMR expansion can be obtained as follows

$$f_0 = \int_{a_1}^{b_1} dx_1 \ldots \int_{a_N}^{b_N} dx_N W(x_1, \ldots, x_N) f(x_1, \ldots, x_N). \tag{2.6}$$

Univariate, bivariate and trivariate terms can be found in a similar manner as given in Equations (2.7), (2.8) and (2.9).

$$\begin{aligned} f_m\left(\xi_m^{(k_m)}\right) = &\int_{a_1}^{b_1} dx_1 W(x_1) \ldots \int_{a_{m-1}}^{b_{m-1}} dx_{m-1} W(x_{m-1}) \int_{a_{m+1}}^{b_{m+1}} dx_{m+1} W(x_{m+1}) \\ &\times \cdots \times \int_{a_N}^{b_N} dx_N W(x_N) f(x_1, \ldots, x_N) - f_0 \end{aligned} \tag{2.7}$$

$$
\begin{aligned}
f_{m_1 m_2}\left(\xi_{m_1}^{(k_{m_1})},\xi_{m_2}^{(k_{m_2})}\right) = &\int_{a_1}^{b_1} dx_1 W(x_1) \dots \int_{a_{m_1-1}}^{b_{m_1-1}} dx_{m_1-1} W(x_{m_1-1}) \\
&\times \dots \times \int_{a_{m_1+1}}^{b_{m_1+1}} dx_{m_1+1} W(x_{m_1+1}) \dots \int_{a_{m_2-1}}^{b_{m_2-1}} dx_{m_2-1} W(x_{m_2-1}) \\
&\times \dots \times \int_{a_{m_2+1}}^{b_{m_2+1}} dx_{m_2+1} W(x_{m_2+1}) \dots \int_{a_N}^{b_N} dx_N W(x_N) f(x_1,\dots,x_N) \\
&- f_0 - f_m\left(\xi_m^{(k_m)}\right)
\end{aligned}
\tag{2.8}
$$

$$
\begin{aligned}
f_{m_1 m_2 m_3}\left(\xi_{m_1}^{(k_{m_1})},\xi_{m_2}^{(k_{m_2})},\xi_{m_3}^{(k_{m_3})}\right) = &\int_{a_1}^{b_1} dx_1 W(x_1) \dots \int_{a_{m_1-1}}^{b_{m_1-1}} dx_{m_1-1} W(x_{m_1-1}) \\
&\times \dots \times \int_{a_{m_1+1}}^{b_{m_1+1}} dx_{m_1+1} W(x_{m_1+1}) \dots \int_{a_{m_2-1}}^{b_{m_2-1}} dx_{m_2-1} W(x_{m_2-1}) \\
&\times \dots \times \int_{a_{m_2+1}}^{b_{m_2+1}} dx_{m_2+1} W(x_{m_2+1}) \dots \int_{a_{m_3-1}}^{b_{m_3-1}} dx_{m_3-1} W(x_{m_3-1}) \\
&\times \dots \times \int_{a_{m_3+1}}^{b_{m_3+1}} dx_{m_3+1} W(x_{m_3+1}) \dots \int_{a_N}^{b_N} dx_N W(x_N) f(x_1,\dots,x_N) \\
&- f_0 - f_m\left(\xi_m^{(k_m)}\right) - f_{m_1 m_2}\left(\xi_{m_1}^{(k_{m_1})},\xi_{m_2}^{(k_{m_2})}\right).
\end{aligned}
\tag{2.9}
$$

2.2 DATA PARTITIONING THROUGH HDMR

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

$$\mathcal{D} \equiv \mathcal{D}_1 \times \mathcal{D}_2 \times \dots \times \mathcal{D}_N \tag{2.10}$$

where

$$\mathcal{D}_i \equiv \left\{\xi_i^{(k_i)}\right\}_{k_i=1}^{k_i=n_i} = \left\{\xi_i^{(1)},\dots,\xi_i^{(n_i)}\right\} \tag{2.11}$$

and $\xi_i^{(k_i)}$ represents the ${k_i}^{th}$ value of i^{th} independent variable (Tunga & Demiralp, 2008).

In our approach, we choose the weight function as

$$W_j(x_j) = \sum_{k_j=1}^{n_j} \alpha_{k_j}^{(j)} \delta\left(x_j - \xi_j^{(k_j)}\right), x_j \in [a_j, b_j], 1 \leq j \leq N \tag{2.12}$$

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.

An exact $\mathcal{F}$ function passing through all the data points can be found by using all right-hand side terms in Equation (2.1). If the integrations in Equations (2.6), (2.7), (2.8) and (2.9) are performed, constant, univariate, bivariate and trivariate terms given in Equation (2.1) can be obtained for cartesian set $\mathcal{D}$ as in Equations (2.13), (2.14), (2.15) and (2.16), respectively. Higher variate terms can also be written in a similar manner.

$$f_0 = \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \cdots \sum_{k_N=1}^{n_N} \left(\prod_{i=1}^{N} \alpha_{k_i}^{(i)} \right) f\left(\xi_1^{(k_1)}, \ldots, \xi_N^{(k_N)}\right) \tag{2.13}$$

$$\begin{aligned} & f_m\left(\xi_m^{(k_m)}\right) = \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \cdots \sum_{k_{m-1}=1}^{n_{m-1}} \sum_{k_{m+1}=1}^{n_{m+1}} \cdots \sum_{k_N=1}^{n_N} \left(\prod_{i=1}^{N} \alpha_{k_i}^{(i)} \right) \\ & \times f\left(\xi_1^{(k_1)}, \ldots, \xi_{m-1}^{(k_{m-1})}, \xi_m^{(k_m)}, \xi_{m+1}^{(k_{m+1})}, \ldots, \xi_N^{(k_N)}\right) - f_0, \\ & \xi_m^{(k_m)} \in \mathcal{D}_m, \;\; 1 \leq k_m \leq n_m, \;\; 1 \leq m \leq N \end{aligned} \tag{2.14}$$

$$\begin{aligned} & f_{m_1 m_2}\left(\xi_{m_1}^{(k_{m_1})}, \xi_{m_2}^{(k_{m_2})}\right) = \\ & \sum_{k_1=1}^{n_1} \sum_{k_2=1}^{n_2} \cdots \sum_{k_{m_1-1}=1}^{n_{m_1-1}} \sum_{k_{m_1+1}=1}^{n_{m_1+1}} \cdots \sum_{k_{m_2-1}=1}^{n_{m_2-1}} \sum_{k_{m_2+1}=1}^{n_{m_2+1}} \cdots \sum_{k_N=1}^{n_N} \left(\prod_{\substack{i=1 \\ i \neq m_1 \wedge i \neq m_2}}^{N} \alpha_{k_i}^{(i)} \right) \\ & \times f\left(\xi_1^{(k_1)}, \ldots, \xi_{m_1-1}^{(k_{m_1-1})}, \xi_{m_1}^{(k_{m_1})}, \xi_{m_1+1}^{(k_{m_1+1})}, \ldots, \xi_{m_2-1}^{(k_{m_2-1})}, \xi_{m_2}^{(k_{m_2})}, \xi_{m_2+1}^{(k_{m_2+1})}, \ldots, \xi_N^{(k_N)}\right) \\ & - f_{m_1}\left(\xi_{m_1}^{(k_{m_1})}\right) - f_{m_2}\left(\xi_{m_2}^{(k_{m_2})}\right) - f_0, \;\; \xi_{m_1}^{(k_{m_1})} \in \mathcal{D}_{m_1}, \;\; \xi_{m_2}^{(k_{m_2})} \in \mathcal{D}_{m_2}, \\ & 1 \leq k_{m_1} \leq n_{m_1}, \;\; 1 \leq k_{m_2} \leq n_{m_2}, \;\; 1 \leq m_1, m_2 \leq N \end{aligned} \tag{2.15}$$

$$\begin{aligned}
&f_{m_1m_2m_3}\left(\xi_{m_1}^{(k_{m_1})},\xi_{m_2}^{(k_{m_2})},\xi_{m_3}^{(k_{m_3})}\right)\\
&=\sum_{k_1=1}^{n_1}\sum_{k_2=1}^{n_2}\sum_{k_3=1}^{n_3}\cdots\sum_{k_{m_1-1}=1}^{n_{m_1-1}}\sum_{k_{m_1+1}=1}^{n_{m_1+1}}\cdots\sum_{k_{m_2-1}=1}^{n_{m_2-1}}\sum_{k_{m_2+1}=1}^{n_{m_2+1}}\cdots\sum_{k_{m_3-1}=1}^{n_{m_3-1}}\sum_{k_{m_3+1}=1}^{n_{m_3+1}}\cdots\sum_{k_N=1}^{n_N}\\
&\times\left(\prod_{\substack{i=1\\ i\neq m_1\wedge i\neq m_2\wedge i\neq m_3}}^{N}\alpha_{k_i}^{(i)}\right)f\left(\xi_1^{(k_1)},\ldots,\xi_{m_1-1}^{(k_{m_1-1})},\xi_{m_1}^{(k_{m_1})},\xi_{m_1+1}^{(k_{m_1+1})},\ldots,\xi_{m_2-1}^{(k_{m_2-1})},\xi_{m_2}^{(k_{m_2})},\right.\\
&\qquad\qquad\left.\xi_{m_2+1}^{(k_{m_2+1})},\ldots,\xi_{m_3-1}^{(k_{m_3-1})},\xi_{m_3}^{(k_{m_3})},\xi_{m_3+1}^{(k_{m_3+1})},\ldots,\xi_N^{(k_N)}\right)\\
&-f_{m_1m_2}\left(\xi_{m_1}^{(k_{m_1})},\xi_{m_2}^{(k_{m_2})}\right)-f_{m_1m_3}\left(\xi_{m_1}^{(k_{m_1})},\xi_{m_3}^{(k_{m_3})}\right)-f_{m_2m_3}\left(\xi_{m_2}^{(k_{m_2})},\xi_{m_3}^{(k_{m_3})}\right)-f_{m_1}\left(\xi_{m_1}^{(k_{m_1})}\right)\\
&-f_{m_2}\left(\xi_{m_2}^{(k_{m_2})}\right)-f_{m_3}\left(\xi_{m_3}^{(k_{m_3})}\right)-f_0,\;\xi_{m_1}^{(k_{m_1})}\in\mathcal{D}_{m_1},\;\xi_{m_2}^{(k_{m_2})}\in\mathcal{D}_{m_2},\;\xi_{m_3}^{(k_{m_3})}\in\mathcal{D}_{m_3},\\
&\;1\le k_{m_1}\le n_{m_1},\;1\le k_{m_2}\le n_{m_2},\;1\le k_{m_3}\le n_{m_3},\;1\le m_1,m_2,m_3\le N
\end{aligned}\tag{2.16}$$

where $\alpha_{k_i}^{(i)}$ is a constant which specifies the contribution level of each node to the model in which we set equally in our experiments. Once the univariate, bivariate and trivariate terms are obtained, the corresponding components of the Lagrange interpolation can be found as follows

$$P_m(x_m)=\sum_{k_m=1}^{n_m}L_{k_m}(x_m)f_m\left(\xi_m^{(k_m)}\right),\;\xi_m^{(k_m)}\in\mathcal{D}_m,\;1\le m\le N\tag{2.17}$$

$$\begin{aligned}
P_{m_1m_2}(x_{m_1},x_{m_2})=\sum_{k_{m_1}=1}^{n_{m_1}}\sum_{k_{m_2}=1}^{n_{m_2}}L_{k_{m_1}}(x_{m_1})L_{k_{m_2}}(x_{m_2})f_{m_1m_2}\left(\xi_{m_1}^{(k_{m_1})},\xi_{m_2}^{(k_{m_2})}\right),\\
\xi_{m_1}^{(k_{m_1})}\in\mathcal{D}_{m_1},\;\xi_{m_2}^{(k_{m_2})}\in\mathcal{D}_{m_2},\;1\le m_1,m_2\le N
\end{aligned}\tag{2.18}$$

$$\begin{aligned}
&P_{m_1m_2m_3}(x_{m_1},x_{m_2},x_{m_3})=\\
&\sum_{k_{m_1}=1}^{n_{m_1}}\sum_{k_{m_2}=1}^{n_{m_2}}\sum_{k_{m_3}=1}^{n_{m_3}}L_{k_{m_1}}(x_{m_1})L_{k_{m_2}}(x_{m_2})L_{k_{m_3}}(x_{m_3})f_{m_1m_2m_3}\left(\xi_{m_1}^{(k_{m_1})},\xi_{m_2}^{(k_{m_2})},\xi_{m_3}^{(k_{m_3})}\right),\\
&\xi_{m_1}^{(k_{m_1})}\in\mathcal{D}_{m_1},\;\xi_{m_2}^{(k_{m_2})}\in\mathcal{D}_{m_2},\;\xi_{m_3}^{(k_{m_3})}\in\mathcal{D}_{m_3},\;1\le m_1,m_2,m_3\le N
\end{aligned}\tag{2.19}$$

where $L_{k_m}(x_m)$ is the Lagrange polynomial. Finally, the polynomial that approximates to $\mathcal{F}$ using up to trivariate terms is obtained as follows

$$L_{k_m}(x_m) = \prod_{\substack{i=1 \\ i \neq k_m}}^{n_m} \frac{\left(x_m - \xi_m^{(i)}\right)}{\left(\xi_m^{(k_m)} - \xi_m^{(i)}\right)} \tag{2.20}$$

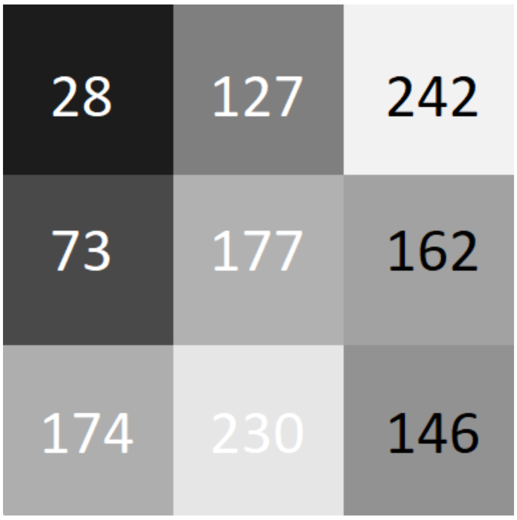
$$\begin{aligned} \mathcal{F}(x_1, \ldots, x_N) \approx f_0 + \sum_{m=1}^{N} P_m(x_m) + \sum_{\substack{m_1, m_2 = 1 \\ m_1 < m_2}}^{N} P_{m_1 m_2}(x_{m_1}, x_{m_2}) \\ + \sum_{\substack{m_1, m_2, m_3 = 1 \\ m_1 < m_2 < m_3}}^{N} P_{m_1 m_2 m_3}(x_{m_1}, x_{m_2}, x_{m_3}). \end{aligned} \tag{2.21}$$

2.3 IMAGE REPRESENTATION THROUGH HDMR

In this section, we have tested the HDMR method on small-sized grayscale and color images to illustrate the results. It can be shown from the results that using up to bivariate terms is enough to represent grayscale images, whereas, we need to also add the trivariate terms to the HDMR equation, Equation (2.1), to represent color images.

2.3.1 Representing a Grayscale Image through HDMR

Figure 2.1: Test Image



28	127	242
73	177	162
174	230	146

For a given $X \times Y \times Z$ image $\mathcal{F}$, 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 = 1 for grayscale images), respectively. Then, the sets that are used to create the cartesian product given in Equation (2.10) can be written as follows:

$$\mathcal{D}_1 = \{1, 2, \ldots, X\},\ \mathcal{D}_2 = \{1, 2, \ldots, Y\},\ \mathcal{D}_3 = \{1, 2, \ldots, Z\} \tag{2.22}$$

There are 9 pixels in the image data given in Figure 2.1. This data has 3 independent variables: row numbers, x_1, column numbers, x_2, and the color values, x_3.

$$x_1 = \{1, 2, 3\}, \qquad x_2 = \{1, 2, 3\}, \qquad x_3 = \{1\} \tag{2.23}$$

Also, $f(x_1, x_2, x_3)$ is the corresponding intensity value in the image. Note that there is just 1 color channel in a grayscale image. Thus, we can eliminate x_3 from the cartesian product set and assume there are 2 independent variables, x_1 and x_2. The Cartesian product set of this image has 9 nodes. Each node is characterized by 2 parameters as shown in Equation (2.24).

$$\begin{aligned} f(1,1) &= 28, & f(1,2) &= 127, & f(1,3) &= 242, \\ f(2,1) &= 73, & f(2,2) &= 177, & f(2,3) &= 162, \\ f(3,1) &= 174, & f(3,2) &= 230, & f(3,3) &= 146. \end{aligned} \tag{2.24}$$

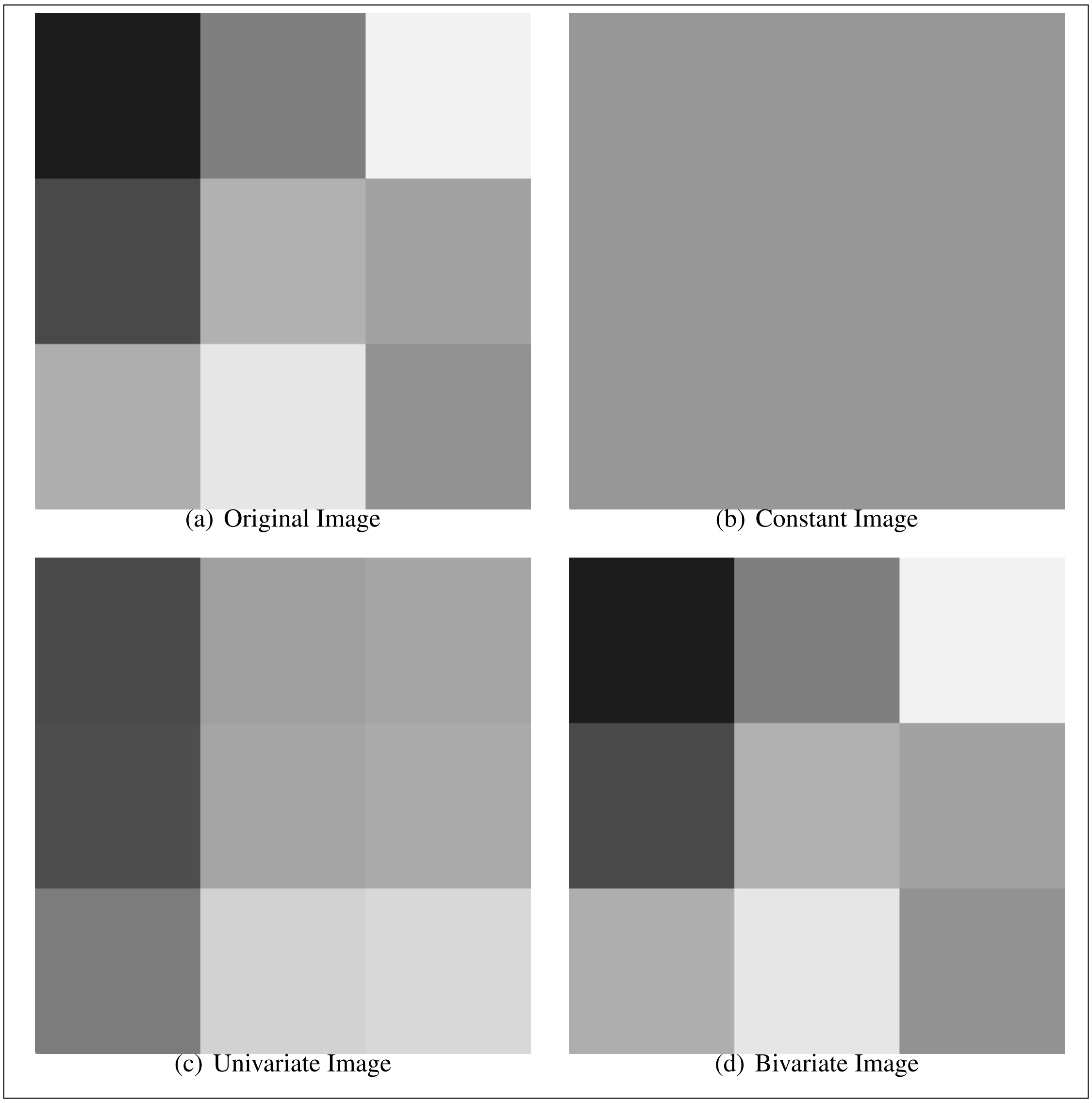
The α parameters appearing in the weight function are taken same. Using the Equation (2.1), this multivariate data is partitioned into its constant, univariate and bivariate components. The constant term can be obtained by using the relation given in Equation (2.13) as follows

$$f_0 = 151 \tag{2.25}$$

If we create a new image with the same size of our original image and fill every one of the pixels with the value of constant term, it can be clearly seen that the constant term is not enough to represent the original image as shown in Figure 2.2(b).

Thus, we have also added the univariate terms in order to represent our multivariate data. The univariate terms can be obtained by using the relation given in Equation (2.14) as follows

Figure 2.2: The image obtained by the HDMR Equation using the constant term



(a) Original Image

(b) Constant Image

(c) Univariate Image

(d) Bivariate Image

$$\begin{array}{lll} f_1\left(\xi_1^{(1)}\right) = -18.6667, & f_1\left(\xi_1^{(2)}\right) = -13.6667, & f_1\left(\xi_1^{(3)}\right) = 32.3333 \\ f_2\left(\xi_2^{(1)}\right) = -59,3333, & f_2\left(\xi_2^{(2)}\right) = 27.0000, & f_2\left(\xi_2^{(3)}\right) = 32,3333 \end{array} \tag{2.26}$$

where

$$\begin{array}{lll} \xi_1^{(1)} = 1, & \xi_1^{(2)} = 2, & \xi_1^{(3)} = 3, \\ \xi_2^{(1)} = 1, & \xi_2^{(2)} = 2, & \xi_2^{(3)} = 3. \end{array} \tag{2.27}$$

When all of the pixels are filled using the constant and univariate terms, it can also be seen that univariate terms are not enough to represent the image as well as shown in Figure 2.2(c). The relation to find the intensity value of the upper-left pixel in the image using up to univariate terms is given below. All of the intensity values can be found in a similar manner.

$$f(\xi_1^{(1)}, \xi_2^{(1)}) = f_0 + f_1\left(\xi_1^{(1)}\right) + f_2\left(\xi_2^{(1)}\right) = 73.0000 \tag{2.28}$$

The bivariate terms can be obtained by using the relation given in Equation (2.15) as follows

$$\begin{gathered} f_{12}\left(\xi_1^{(1)}, \xi_2^{(1)}\right) = -45.0000,\ f_{12}\left(\xi_1^{(1)}, \xi_2^{(2)}\right) = -32.3333,\ f_{12}\left(\xi_1^{(1)}, \xi_2^{(3)}\right) = 77.3333, \\ f_{12}\left(\xi_1^{(2)}, \xi_2^{(1)}\right) = -5.0000,\ \ f_{12}\left(\xi_1^{(2)}, \xi_2^{(2)}\right) = 12.6667,\ \ f_{12}\left(\xi_1^{(2)}, \xi_2^{(3)}\right) = -7.6667, \\ f_{12}\left(\xi_1^{(3)}, \xi_2^{(1)}\right) = 77.3333,\ \ f_{12}\left(\xi_1^{(3)}, \xi_2^{(2)}\right) = -7.6667,\ \ f_{12}\left(\xi_1^{(3)}, \xi_2^{(3)}\right) = -69.6667. \end{gathered} \tag{2.29}$$

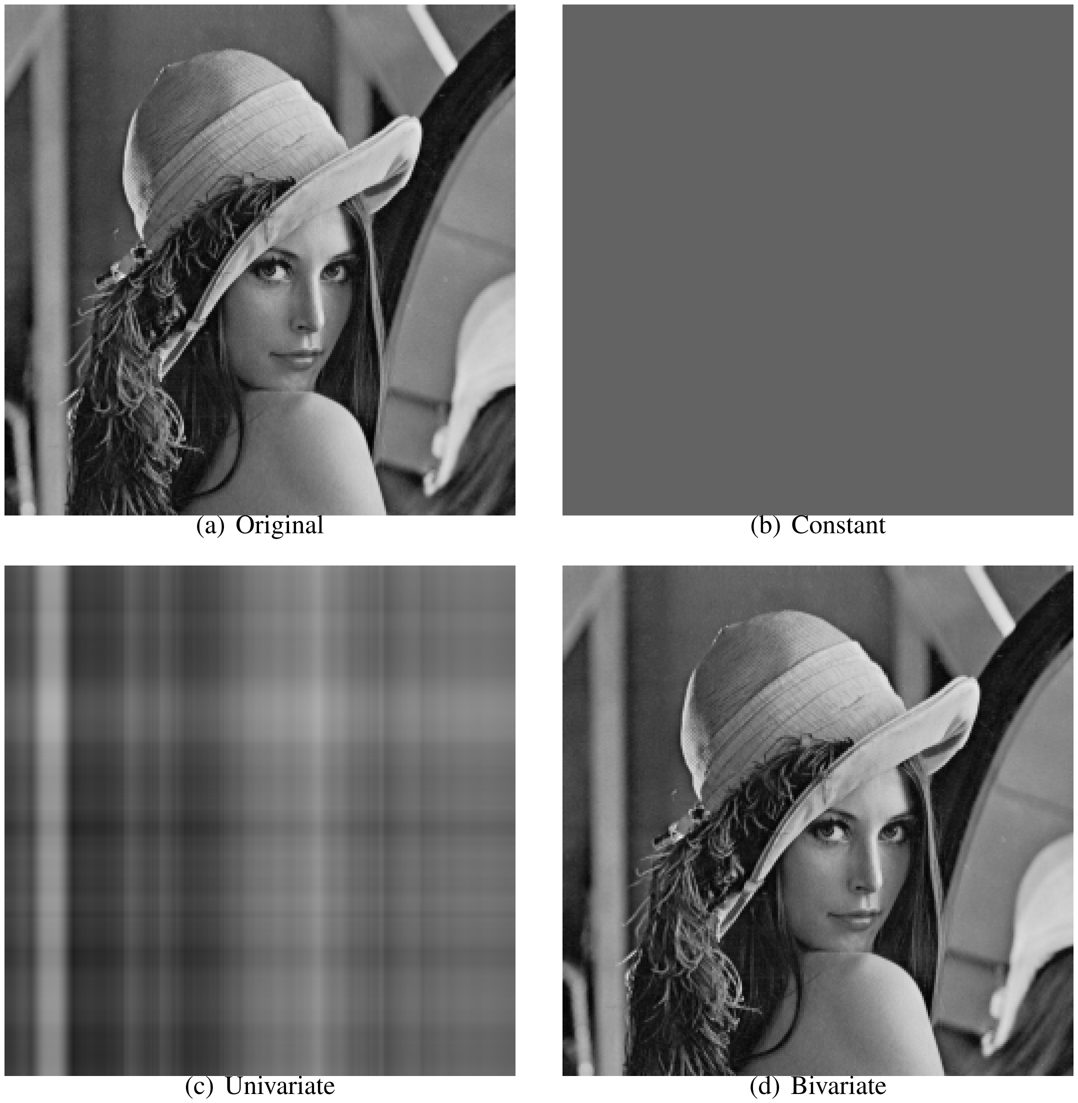
When all of the pixels are filled using the constant, univariate and bivariate terms, it can again be seen that bivariate terms are also not enough to represent the original image as shown in Figure 2.2(d). The relation to find the intensity value of the upper-left pixel in the image using up to bivariate terms is given below.

$$f(\xi_1^{(1)}, \xi_2^{(1)}) = f_0 + f_1\left(\xi_1^{(1)}\right) + f_2\left(\xi_2^{(1)}\right) + f_{12}\left(\xi_1^{(1)}, \xi_2^{(1)}\right) = 28 \tag{2.30}$$

This test shows us that, using up to bivariate terms is enough to represent a grayscale image.

We also show the same procedure for a real grayscale image as shown below. In Figure 2.3, we show interpolation results on an image by using the terms up to constant, univariate and bivariate terms in the HDMR expansion. In Figure 2.4, we also show absolute differences between these images and the original image. As shown from the results in Figure 2.4, using the terms up to bivariate terms perfectly interpolates to the original image, i.e., all pixels of the image in Figure 2.4(c) is zero.

Figure 2.3: Interpolated images by using the terms up to (b) constant term, (c) univariate terms and (d) bivariate terms.



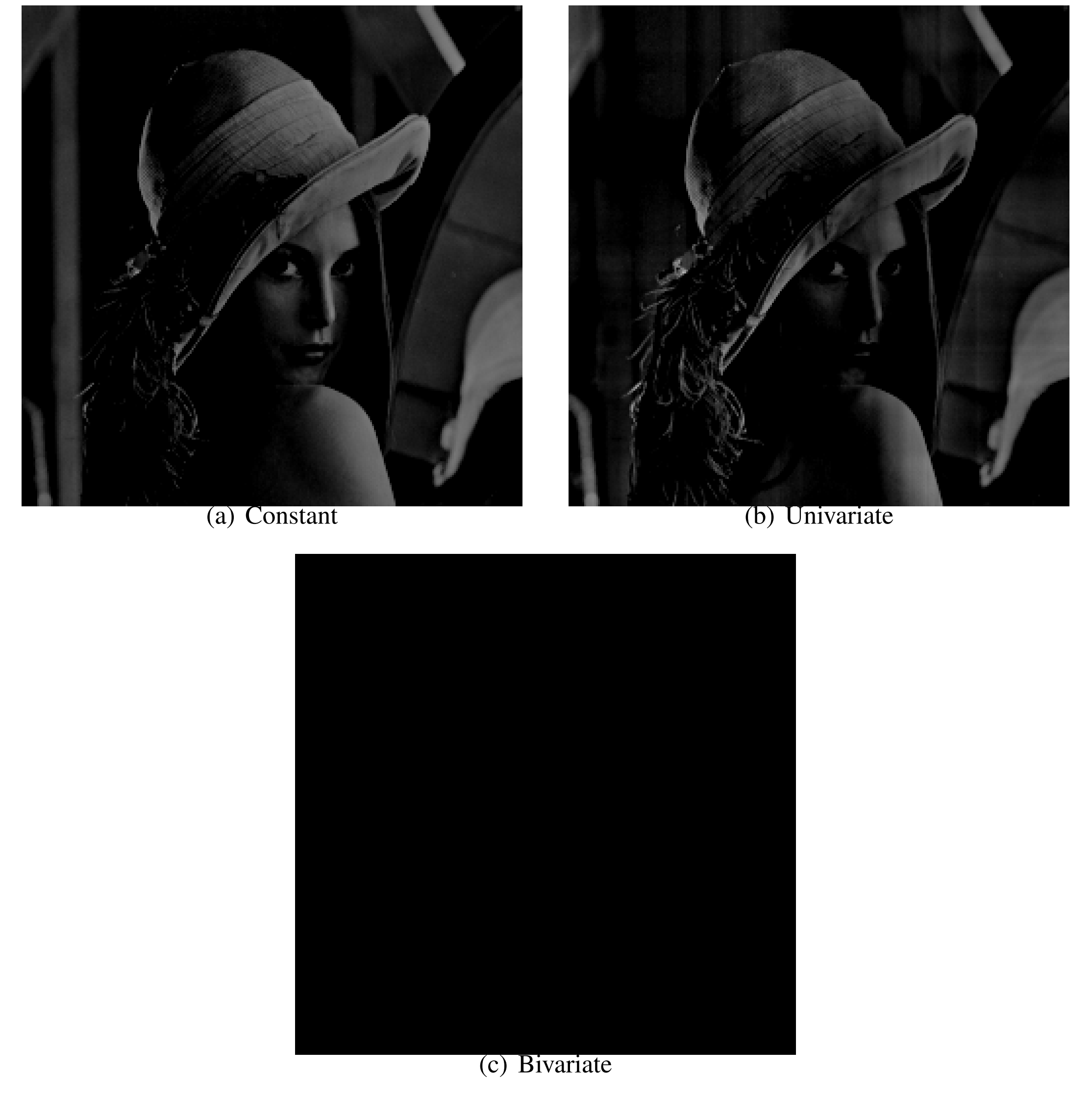
(a) Original

(b) Constant

(c) Univariate

(d) Bivariate

Figure 2.4: Absolute differences between the original image and the image obtained by using the terms up to (b) constant term, (c) univariate terms, (d) bivariate terms and (e) trivariate terms.



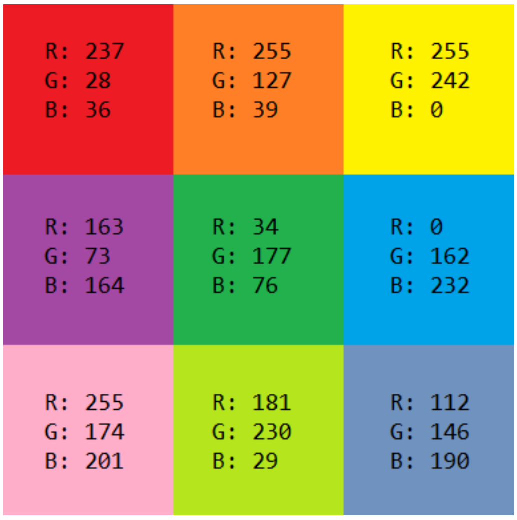
(a) Constant

(b) Univariate

(c) Bivariate

2.3.2 Representing a Color Image through HDMR

Figure 2.5: Test Image



R: 237 G: 28 B: 36	R: 255 G: 127 B: 39	R: 255 G: 242 B: 0
R: 163 G: 73 B: 164	R: 34 G: 177 B: 76	R: 0 G: 162 B: 232
R: 255 G: 174 B: 201	R: 181 G: 230 B: 29	R: 112 G: 146 B: 190

For a given $X \times Y \times Z$ image $\mathcal{F}$, 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 = 3 for color images), respectively. Then, the sets that are used to create the cartesian product given in Equation (2.10) can be written as follows:

$$\mathcal{D}_1 = \{1, 2, \dots, X\},\ \mathcal{D}_2 = \{1, 2, \dots, Y\},\ \mathcal{D}_3 = \{1, 2, \dots, Z\} \tag{2.31}$$

There are 9 pixels in the image data given in Figure 2.5. This data has 3 independent variables: row numbers, x_1, column numbers, x_2, and the color values, x_3 (R=1, G=2, B=3).

$$x_1 = \{1, 2, 3\}, \qquad x_2 = \{1, 2, 3\}, \qquad x_3 = \{1, 2, 3\} \tag{2.32}$$

Also, $f(x_1, x_2, x_3)$ is the corresponding intensity value in the image. The Cartesian product set of this image has 27 nodes. Each node is characterized by 3 parameters as shown in Equation (2.33).

$$\begin{aligned} f(1,1,1) &= 237, & f(1,1,2) &= 28, & f(1,1,3) &= 36,\\ f(1,2,1) &= 255, & f(1,2,2) &= 127, & f(1,2,3) &= 39,\\ f(1,3,1) &= 255, & f(1,3,2) &= 242, & f(1,3,3) &= 0,\\ f(2,1,1) &= 163, & f(2,1,2) &= 73, & f(2,1,3) &= 164,\\ f(2,2,1) &= 34, & f(2,2,2) &= 177, & f(2,2,3) &= 76,\\ f(2,3,1) &= 0, & f(2,3,2) &= 162, & f(2,3,3) &= 232,\\ f(3,1,1) &= 255, & f(3,1,2) &= 174, & f(3,1,3) &= 201,\\ f(3,2,1) &= 181, & f(3,2,2) &= 230, & f(3,2,3) &= 29,\\ f(3,3,1) &= 112, & f(3,3,2) &= 146, & f(3,3,3) &= 190. \end{aligned} \tag{2.33}$$

The α parameters appearing in the weight function are taken same. Using the Equation (2.1), this multivariate data is partitioned into its constant, univariate, bivariate and trivariate components. The constant term can be obtained by using the relation given in Equation (2.13) as follows

$$f_0 = 141.4074 \tag{2.34}$$

If we create a new image with the same size of our original image and fill every one of the pixels with the value of constant term, it can be clearly seen that the constant term is not enough to represent the original image as shown in Figure 2.6(b).

Thus, we have also added the univariate terms in order to represent our multivariate data. The univariate terms can be obtained by using the relation given in Equation (2.14) as follows

$$\begin{aligned} f_1\left(\xi_1^{(1)}\right) &= -5.9630, & f_1\left(\xi_1^{(2)}\right) &= -21.2963, & f_1\left(\xi_1^{(3)}\right) &= 27.2593\\ f_2\left(\xi_2^{(1)}\right) &= 6.4815, & f_2\left(\xi_2^{(2)}\right) &= -13.8519, & f_2\left(\xi_2^{(3)}\right) &= 7.3704\\ f_3\left(\xi_3^{(1)}\right) &= 24.3704, & f_3\left(\xi_3^{(2)}\right) &= 9.5926, & f_3\left(\xi_3^{(3)}\right) &= -33.9630 \end{aligned} \tag{2.35}$$

where

$$\begin{aligned}
&\xi_1^{(1)} = 1, \qquad && \xi_1^{(2)} = 2, \qquad && \xi_1^{(3)} = 3, \\
&\xi_2^{(1)} = 1, && \xi_2^{(2)} = 2, && \xi_2^{(3)} = 3, \\
&\xi_3^{(1)} = 1, && \xi_3^{(2)} = 2, && \xi_3^{(3)} = 3.
\end{aligned} \tag{2.36}$$

When all of the pixels are filled using the constant and univariate terms, it can also be seen that univariate terms are not enough to represent the image as well as shown in Figure 2.6(c). The relation to find the intensity value in the red color channel of the upper-left pixel in the image using up to univariate terms is given below. All of the intensity values for every color channels of each pixel can be found in a similar manner.

$$f(\xi_1^{(1)}, \xi_2^{(1)}, \xi_3^{(1)}) = f_0 + f_1\left(\xi_1^{(1)}\right) + f_2\left(\xi_2^{(1)}\right) + f_3\left(\xi_3^{(1)}\right) = 166.2963 \tag{2.37}$$

The bivariate terms can be obtained by using the relation given in Equation (2.15) as follows

$$\begin{aligned}
f_{12}\left(\xi_1^{(1)}, \xi_2^{(1)}\right) &= -41.5926, & f_{12}\left(\xi_1^{(1)}, \xi_2^{(2)}\right) &= 18.7407, & f_{12}\left(\xi_1^{(1)}, \xi_2^{(3)}\right) &= 22.8519, \\
f_{12}\left(\xi_1^{(2)}, \xi_2^{(1)}\right) &= 6.7407, & f_{12}\left(\xi_1^{(2)}, \xi_2^{(2)}\right) &= -10.5926, & f_{12}\left(\xi_1^{(2)}, \xi_2^{(3)}\right) &= 3.8519, \\
f_{12}\left(\xi_1^{(3)}, \xi_2^{(1)}\right) &= 34.8519, & f_{12}\left(\xi_1^{(3)}, \xi_2^{(2)}\right) &= -8.1481, & f_{12}\left(\xi_1^{(3)}, \xi_2^{(3)}\right) &= -26.7037, \\
f_{13}\left(\xi_1^{(1)}, \xi_3^{(1)}\right) &= 89.1852, & f_{13}\left(\xi_1^{(1)}, \xi_3^{(2)}\right) &= -12.7037, & f_{13}\left(\xi_1^{(1)}, \xi_3^{(3)}\right) &= -76.4815, \\
f_{13}\left(\xi_1^{(2)}, \xi_3^{(1)}\right) &= -78.8148, & f_{13}\left(\xi_1^{(2)}, \xi_3^{(2)}\right) &= 7.6296, & f_{13}\left(\xi_1^{(2)}, \xi_3^{(3)}\right) &= 71.1852, \\
f_{13}\left(\xi_1^{(3)}, \xi_3^{(1)}\right) &= -10.3704, & f_{13}\left(\xi_1^{(3)}, \xi_3^{(2)}\right) &= 5.0741, & f_{13}\left(\xi_1^{(3)}, \xi_3^{(3)}\right) &= 5.2963, \\
f_{23}\left(\xi_2^{(1)}, \xi_3^{(1)}\right) &= 46.0741, & f_{23}\left(\xi_2^{(1)}, \xi_3^{(2)}\right) &= -65.8148, & f_{23}\left(\xi_2^{(1)}, \xi_3^{(3)}\right) &= 19.7407, \\
f_{23}\left(\xi_2^{(2)}, \xi_3^{(1)}\right) &= 4.7407, & f_{23}\left(\xi_2^{(2)}, \xi_3^{(2)}\right) &= 40.8519, & f_{23}\left(\xi_2^{(2)}, \xi_3^{(3)}\right) &= -45.5926, \\
f_{23}\left(\xi_2^{(3)}, \xi_3^{(1)}\right) &= -50.8148, & f_{23}\left(\xi_2^{(3)}, \xi_3^{(2)}\right) &= 24.9630, & f_{23}\left(\xi_2^{(3)}, \xi_3^{(3)}\right) &= 25.8519.
\end{aligned} \tag{2.38}$$

When all of the pixels are filled using the constant, univariate and bivariate terms, it can again be seen that bivariate terms are also not enough to represent the original image as shown in Figure 2.6(d). The relation to find the intensity value in the red color channel of the upper-left pixel in the image using up to bivariate terms is given below (Note that since the range of the intensity values are between 0 and 255, it is seen as 255 on the screen if it's found greater than 255 and seen as 0 if it's found less than 0). All of the intensity values for every color channels of each pixel can be found in a similar manner.

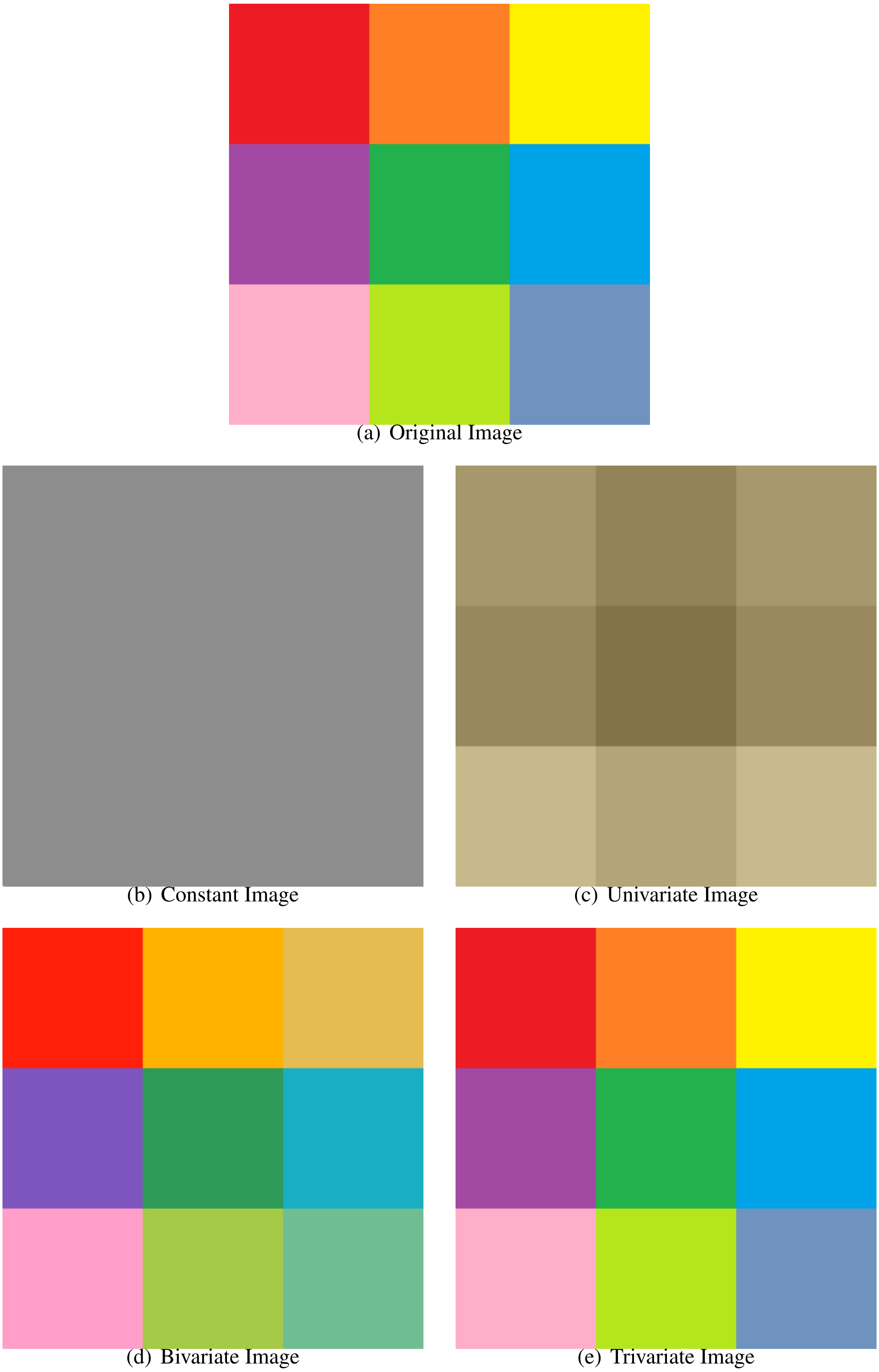
$$\begin{aligned} f(\xi_1^{(1)},\xi_2^{(1)},\xi_3^{(1)}) = f_0 + & f_1\left(\xi_1^{(1)}\right) + f_2\left(\xi_2^{(1)}\right) + f_3\left(\xi_3^{(1)}\right) + f_{12}\left(\xi_1^{(1)},\xi_2^{(1)}\right) \\ & + f_{13}\left(\xi_1^{(1)},\xi_3^{(1)}\right) + f_{23}\left(\xi_2^{(1)},\xi_3^{(1)}\right) = 259.9630 \end{aligned} \tag{2.39}$$

Finally, we decided to add the trivariate terms as well to represent the original image. The trivariate terms can be obtained by using the relation given in Equation (2.16) as follows

$$\begin{aligned} f_{123}\left(\xi_1^{(1)},\xi_2^{(1)},\xi_3^{(1)}\right) &= -22.9630, & f_{123}\left(\xi_1^{(1)},\xi_2^{(1)},\xi_3^{(2)}\right) &= -3.4074, \\ f_{123}\left(\xi_1^{(1)},\xi_2^{(1)},\xi_3^{(3)}\right) &= 26.3704, & f_{123}\left(\xi_1^{(1)},\xi_2^{(2)},\xi_3^{(1)}\right) &= -3.6296, \\ f_{123}\left(\xi_1^{(1)},\xi_2^{(2)},\xi_3^{(2)}\right) &= -51.0741, & f_{123}\left(\xi_1^{(1)},\xi_2^{(2)},\xi_3^{(3)}\right) &= 54.7037, \\ f_{123}\left(\xi_1^{(1)},\xi_2^{(3)},\xi_3^{(1)}\right) &= 26.5926, & f_{123}\left(\xi_1^{(1)},\xi_2^{(3)},\xi_3^{(2)}\right) &= 54.4815, \\ f_{123}\left(\xi_1^{(1)},\xi_2^{(3)},\xi_3^{(3)}\right) &= -81.0741, & f_{123}\left(\xi_1^{(2)},\xi_2^{(1)},\xi_3^{(1)}\right) &= 38.0370, \\ f_{123}\left(\xi_1^{(2)},\xi_2^{(1)},\xi_3^{(2)}\right) &= -11.7407, & f_{123}\left(\xi_1^{(2)},\xi_2^{(1)},\xi_3^{(3)}\right) &= -26.2963, \\ f_{123}\left(\xi_1^{(2)},\xi_2^{(2)},\xi_3^{(1)}\right) &= -11.9630, & f_{123}\left(\xi_1^{(2)},\xi_2^{(2)},\xi_3^{(2)}\right) &= 23.2593, \\ f_{123}\left(\xi_1^{(2)},\xi_2^{(2)},\xi_3^{(3)}\right) &= -11.2963, & f_{123}\left(\xi_1^{(2)},\xi_2^{(3)},\xi_3^{(1)}\right) &= -26.0741, \\ f_{123}\left(\xi_1^{(2)},\xi_2^{(3)},\xi_3^{(2)}\right) &= -11.5185, & f_{123}\left(\xi_1^{(2)},\xi_2^{(3)},\xi_3^{(3)}\right) &= 37.5926, \\ f_{123}\left(\xi_1^{(3)},\xi_2^{(1)},\xi_3^{(1)}\right) &= -15.0741, & f_{123}\left(\xi_1^{(3)},\xi_2^{(1)},\xi_3^{(2)}\right) &= 15.1481, \\ f_{123}\left(\xi_1^{(3)},\xi_2^{(1)},\xi_3^{(3)}\right) &= -0.0741, & f_{123}\left(\xi_1^{(3)},\xi_2^{(2)},\xi_3^{(1)}\right) &= 15.5926, \\ f_{123}\left(\xi_1^{(3)},\xi_2^{(2)},\xi_3^{(2)}\right) &= 27.8148, & f_{123}\left(\xi_1^{(3)},\xi_2^{(2)},\xi_3^{(3)}\right) &= -43.4074, \\ f_{123}\left(\xi_1^{(3)},\xi_2^{(3)},\xi_3^{(1)}\right) &= -0.5185, & f_{123}\left(\xi_1^{(3)},\xi_2^{(3)},\xi_3^{(2)}\right) &= -42.9630, \\ f_{123}\left(\xi_1^{(3)},\xi_2^{(3)},\xi_3^{(3)}\right) &= 43.4815. \end{aligned} \tag{2.40}$$

The relation to find the intensity value in the red color channel of the upper-left pixel in the image using up to trivariate terms is given below. All of the intensity values for every color channels of each pixel can be found in a similar manner. When all of the pixel intensity values are found, the image in Figure 2.6(e) can be illustrated, which is exactly the same with our original image.

Figure 2.6: The Image obtained by the HDMR Equation using the constant term



(a) Original Image

(b) Constant Image

(c) Univariate Image

(d) Bivariate Image

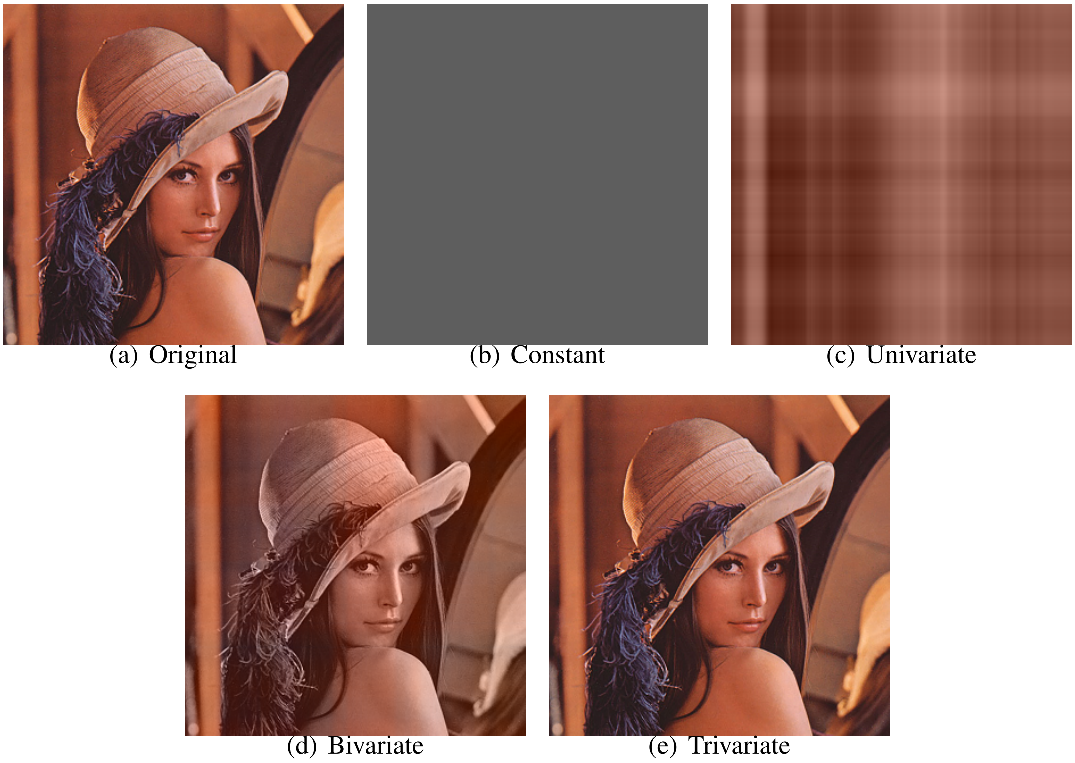
(e) Trivariate Image

$$f(\xi_1^{(1)},\xi_2^{(1)},\xi_3^{(1)}) = f_0 + f_1\left(\xi_1^{(1)}\right) + f_2\left(\xi_2^{(1)}\right) + f_3\left(\xi_3^{(1)}\right) + f_{12}\left(\xi_1^{(1)},\xi_2^{(1)}\right) + f_{13}\left(\xi_1^{(1)},\xi_3^{(1)}\right) + f_{23}\left(\xi_2^{(1)},\xi_3^{(1)}\right) + f_{123}\left(\xi_1^{(1)},\xi_2^{(1)},\xi_3^{(1)}\right) = 237 \tag{2.41}$$

This test shows us that, using up to trivariate terms is enough to represent a color image.

We also show the same procedure for a real color image as shown below. In Figure 2.7, we show interpolation results on an image by using the terms up to constant, univariate, bivariate and trivariate terms in the HDMR expansion. In Figure 2.8, we also show absolute differences between these images and the original image. As shown from the results in Figure 2.8, using the terms up to trivariate terms perfectly interpolates to the original image, i.e., all pixels of the image in Figure 2.8(d) is zero.

Figure 2.7: Interpolated images by using the terms up to (b) constant term, (c) univariate terms, (d) bivariate terms and (e) trivariate terms.



(a) Original

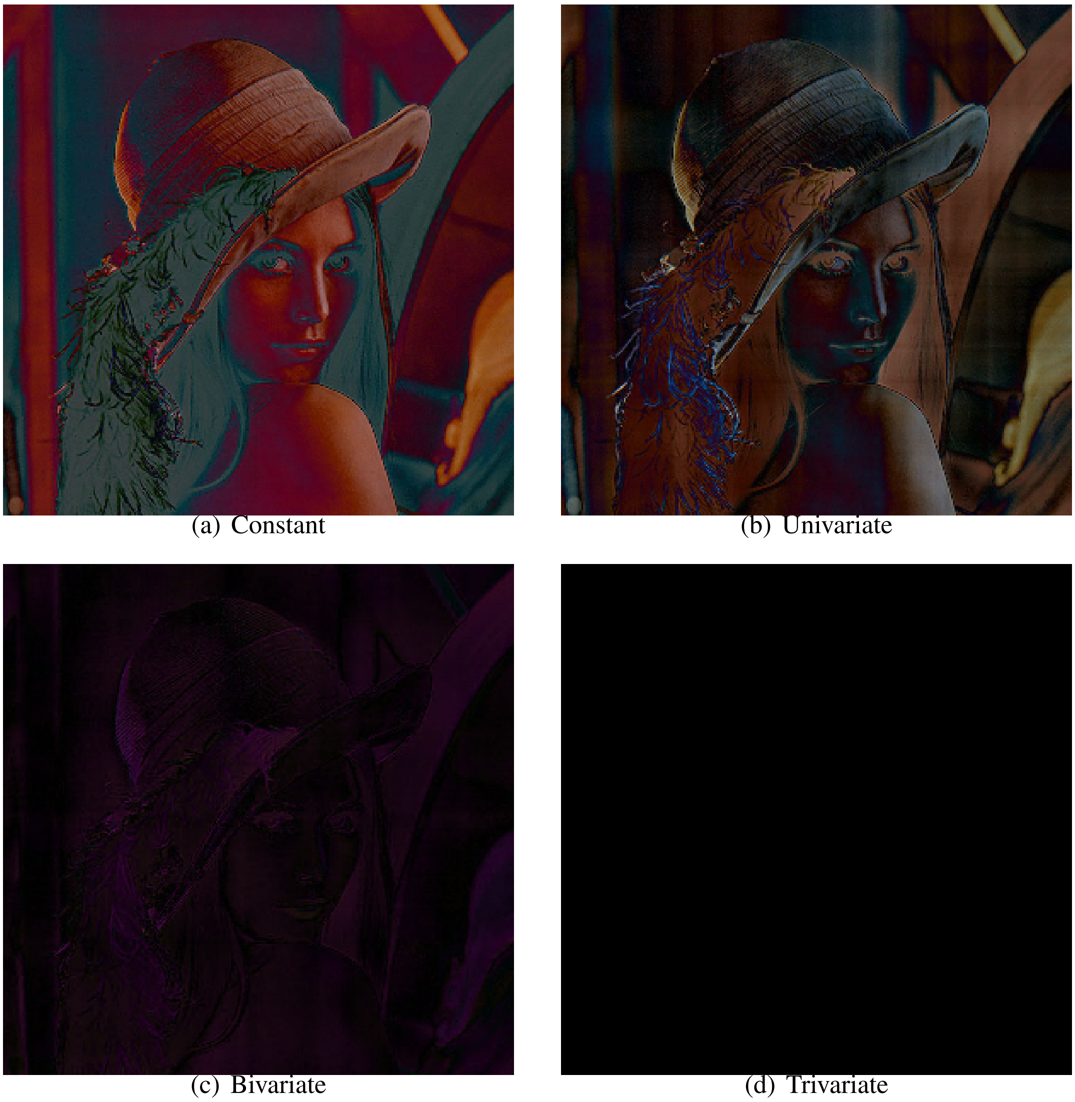
(b) Constant

(c) Univariate

(d) Bivariate

(e) Trivariate

Figure 2.8: Absolute differences between the original image and the image obtained by using the terms up to (b) constant term, (c) univariate terms, (d) bivariate terms and (e) trivariate terms.


(a) Constant

(b) Univariate

(c) Bivariate

(d) Trivariate

2.3.3 Performance Evaluation

We evaluate the performance of the method by comparing the obtained images using up to constant, univariate, bivariate and trivariate terms with the original images using peak signal-to-noise ratio (PSNR) (Huynh-Thu & Ghanbari, 2008). PSNR is most easily defined via the mean squared error (MSE) which is computed as follows

$$PSNR = 20.\log(MAX_I) - 10.\log(MSE) \tag{2.42}$$

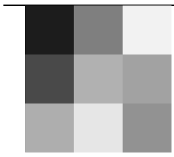
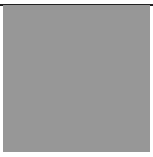
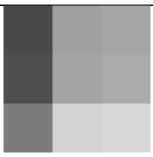
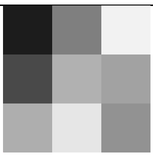
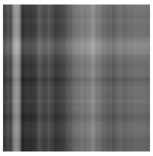
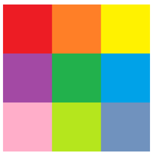
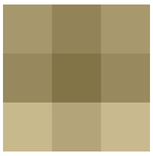
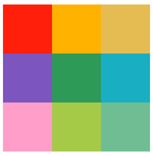
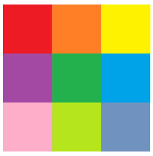
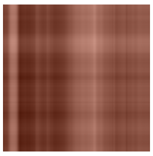
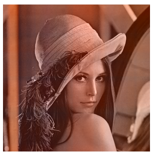
where

$$MSE = \frac{1}{XYZ}\sum_{i=1}^{X}\sum_{j=1}^{Y}\sum_{k=1}^{Z}(I(i,j,k) - \hat{I}(i,j,k))^2. \tag{2.43}$$

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

PSNR results that shows the comparement between the original test images in Figures 2.2(a) and 2.6(a) with the images obtained by using up to constant, univariate, bivariate and trivariate terms are given in Table 2.1.

Table 2.1: PSNR results of the test images in Figures 2.2 and 2.6.

Original	Constant	Univariate	Bivariate	Trivariate
				
PSNR	11.9142	15.3596	Inf	
				
PSNR	13.7620	15.0814	Inf	
				
PSNR	9.5684	10.2983	18.0555	Inf
				
PSNR	13.3846	16.1911	25.3227	Inf

3. HDMR-BASED IMAGE INPAINTING METHODS

Image inpainting is the process of filling missing or fixing corrupted regions in an image. The intensity values of the pixels in the missing area are expected to be associated with the pixels in the 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. In this chapter, we propose a new image inpainting method. We show the development of the method through the thesis process step by step in subsections. We first propose a method to inpaint rectangular regions in grayscale images using HDMR which is explained in detail in Section 3.1. Then, we propose the same method to inpaint rectangular regions in color images (see Section 3.2). But in real life problems, when you try to remove an object from an image, or when you try to remove a scratch from an image, it generally won't be in a rectangular shape. Thus, we propose a new method to inpaint any shape of missing regions in a color image (see Section 3.3). But this method also has a constraint as it cannot be succeed in textural images. Finally, we propose a final texture and pattern preserving interpolation-based method to inpaint color images. This Texture and Pattern Preserving Inpainting using HDMR (TPI-HDMR) method is explained in detail in Section 4. Note that the methods which work for a color image can also work for all grayscale images as well.

3.1 INPAINTING RECTANGULAR MISSING REGIONS IN GRAYSCALE IMAGES

The proposed method in this section represents a high dimensional data in lower dimensions using High Dimensional Model Representation Method (HDMR) and performs image inpainting for rectangular regions in grayscale images with Lagrange interpolation. Proposed approach works only when the missing region is a rectangle due to a constraint that comes from the structure of the HDMR method. Experimental results show that the proposed method produces better results than the well-known and pioneering total variation-based image inpainting method in the literature in many test cases.

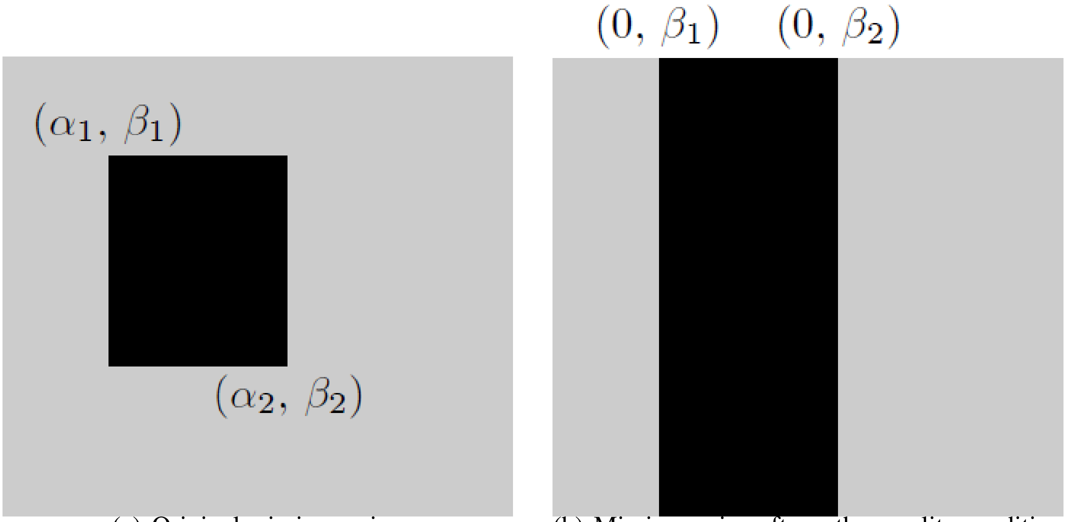
It has been shown in Chapter 2.1 that a given grayscale image can be exactly obtained with HDMR by using at most bivariate terms in the Equation (2.1). Therefore, in our approach for grayscale image inpainting, we use constant, univariate and bivariate terms in Equation (2.1).

As it is mentioned in Chapter 2, orthogonality condition must be satisfied to apply HDMR to a data set (Tunga and Demiralp, 2009). Orthogonality condition requires that all the values of $\mathcal{F}$ to be known for all points in $\mathcal{D}$. In image inpainting, since there are some pixel coordinates in $\mathcal{D}$ 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 $\mathcal{D}$ using new $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$ sets as shown in Equation (3.1).

$$\mathcal{D}_1 = \{1, 2, \ldots, X\},\ \mathcal{D}_2 = \{1, 2, \ldots \beta_1 - 1, \beta_2 + 1, \ldots, Y\},\ \mathcal{D}_3 = \{1, 2, \ldots, Z\} \quad (3.1)$$

Let us assume that the intensity values in the black region, between the coordinates $(\alpha_1, \beta_1) - (\alpha_2, \beta_2)$, shown in Figure 3.1(a) are missing. Due to the orthogonality condition, when applying HDMR, the image inpainting problem in Figure 3.1(a) turns into inpainting the image shown in Figure 3.1(b). Note the significant increase of the missing region with the changes that we made to satisfy the orthogonality condition.

Figure 3.1: Missing Region Illustration



(a) Original missing region

(b) Missing region after orthogonality condition is satisfied

We perform experiments on 4 different grayscale test images shown in Figure 3.2. We design 12 different test settings by using each test image with 3 different masks with different sizes of rectangular missing regions shown in Figure 3.3. Note that black regions in each mask represent the missing region in the corresponding test setting. We compare our approach with total variation inpainting method (Shen & Chan, 2002) which is a pioneering inpainting approach in the literature.

Figure 3.2: Original Images

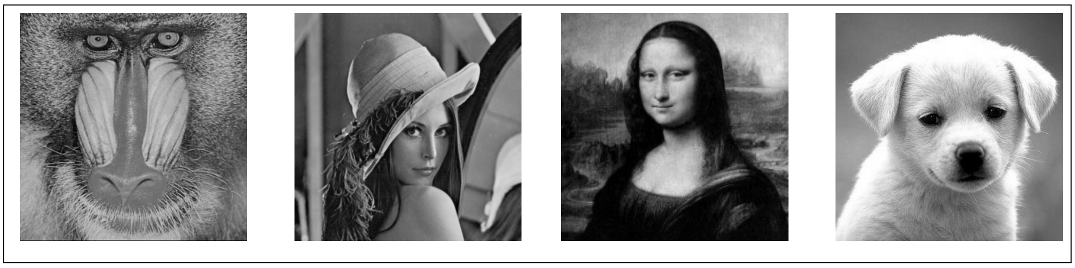
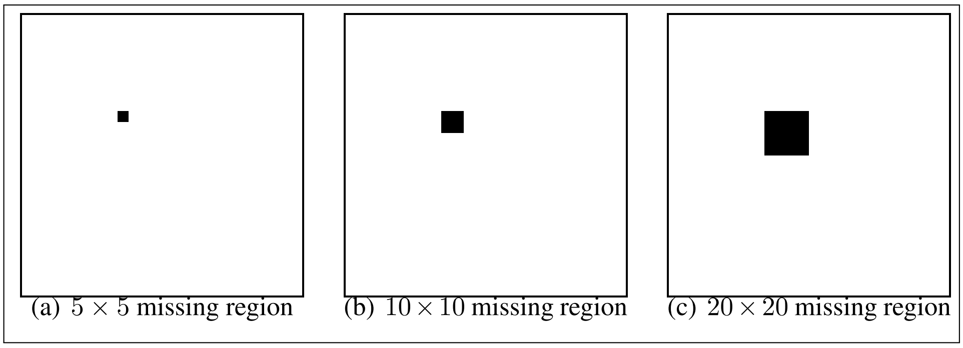


Figure 3.3: Missing Regions


(a) 5 × 5 missing region (b) 10 × 10 missing region (c) 20 × 20 missing region

Proposed method inpaints the missing regions using the nearest neighboring known pixel intensity values. We compare our results with TV inpainting method (Shen & Chan, 2002). We obtain quantitative results by comparing inpainting results of each method with the original images using PSNR (Huynh-Thu & Ghanbari, 2008) which is explained in Section 2.3.3.

We present visual results with their corresponding PSNR values for 4 test images with 3 different sizes of masks in each. Figures 3.4, 3.5 and 3.6 show the results in images with 5×5, 10×10 and 20×20 missing regions, respectively. When the missing region is smaller, as in the 5×5 mask, there is no significant difference between the results of compared methods. But, when the size of the missing region grows, our method produces better results than TV inpainting method (Shen & Chan, 2002) in most of the test cases. Our approach achieves the better PSNR values in 3 of the test cases whereas, the TV approach (Shen & Chan, 2002) achieves only 1 in both masks with sizes of 10×10 and 20×20.

Figure 3.4: Visual results with their corresponding PSNR values for test images with 5×5 missing region for our proposed method and TV inpainting method

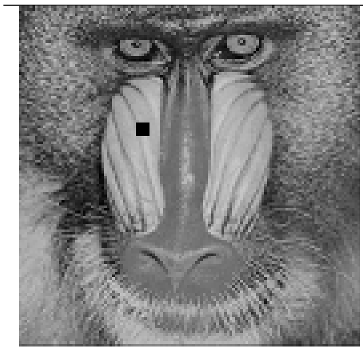
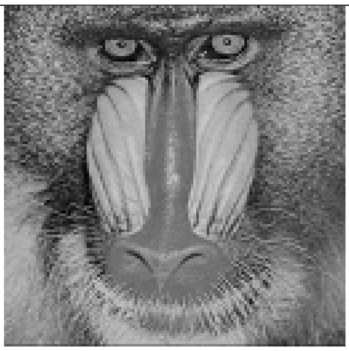
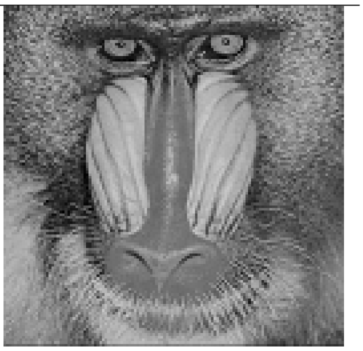
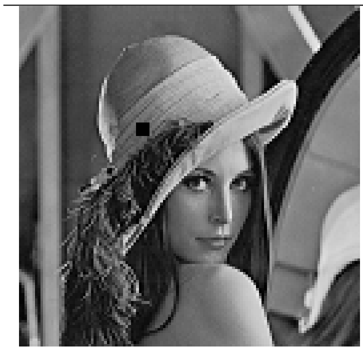
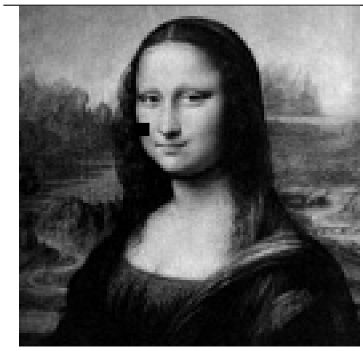
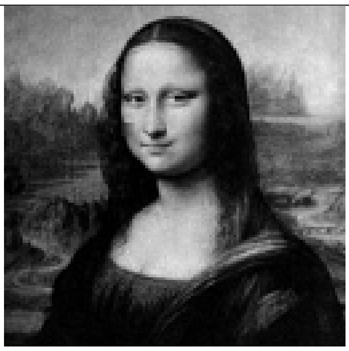
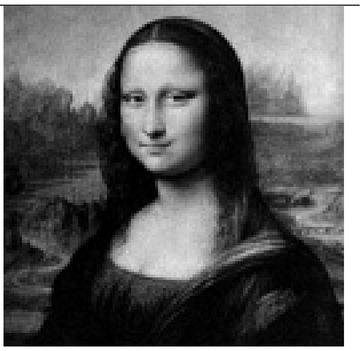
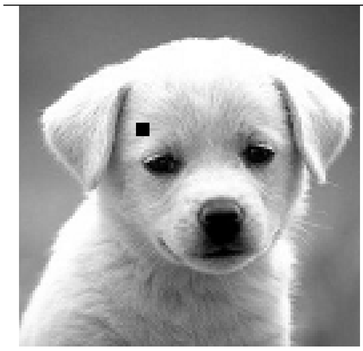
Test Images	**HDMR**	**TV**
		
PSNR	**59.6155**	58.7767
		
PSNR	56.6332	**58.4680**
		
PSNR	44.4300	**47.7881**
		
PSNR	**58.8794**	56.6201

Figure 3.5: **Visual results with their corresponding PSNR values for test images with 10×10 missing region for our proposed method and TV inpainting method**

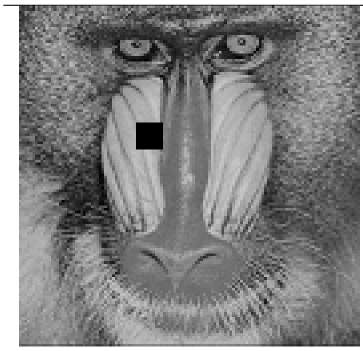
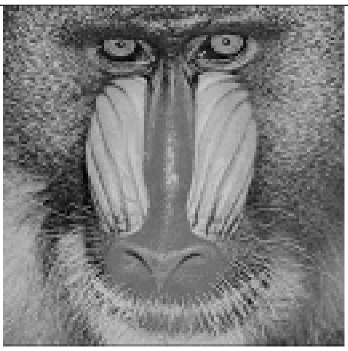
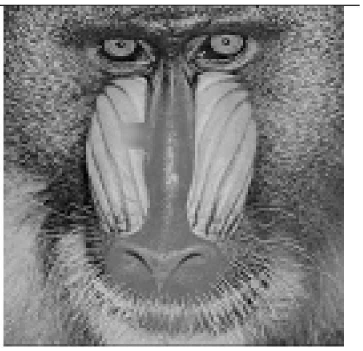
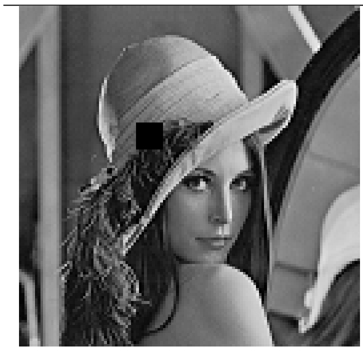
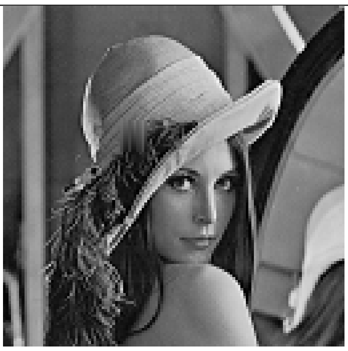
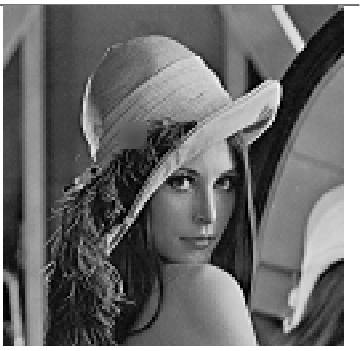
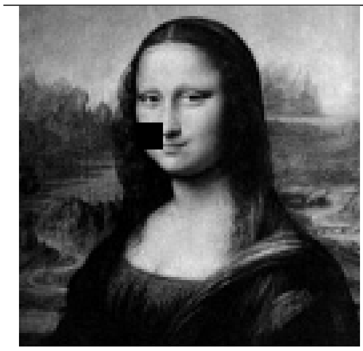
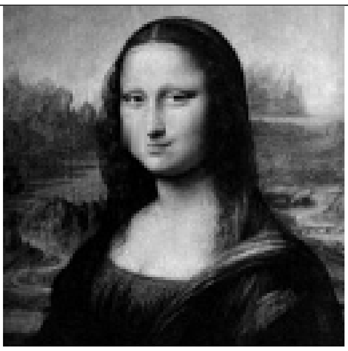
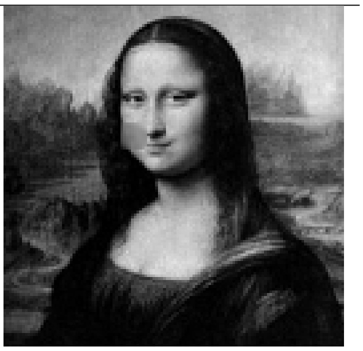
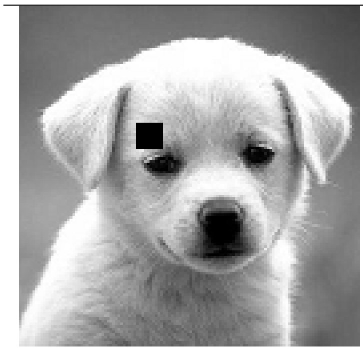
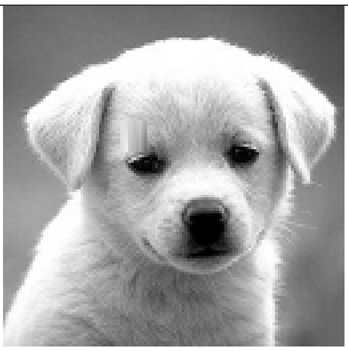
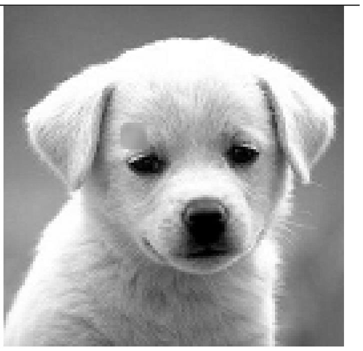
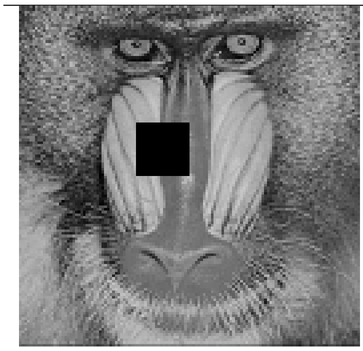
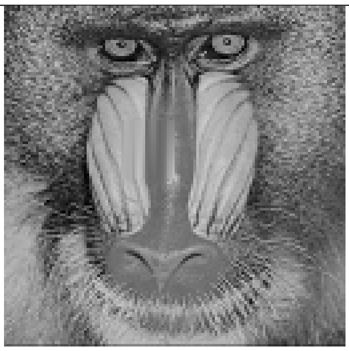
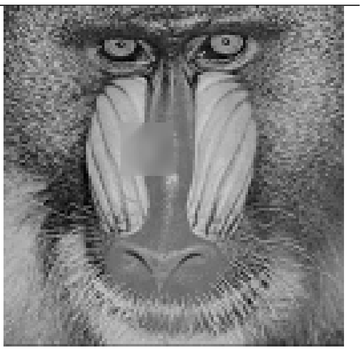
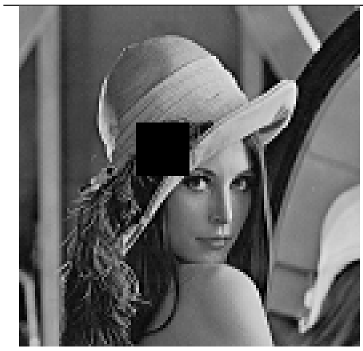
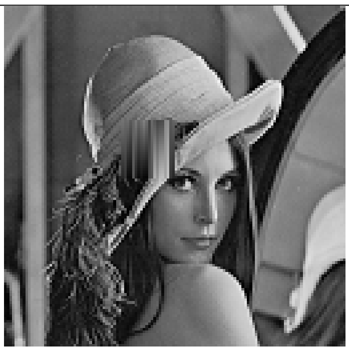
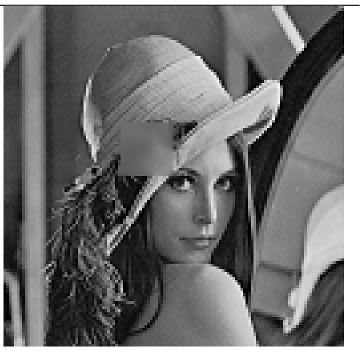
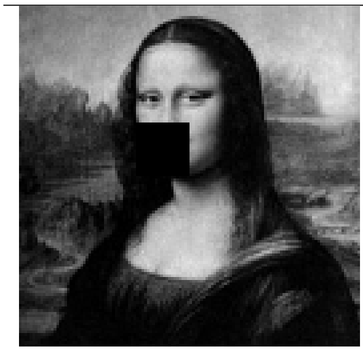
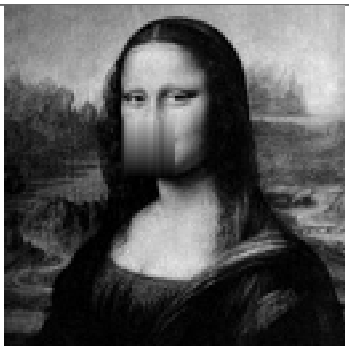
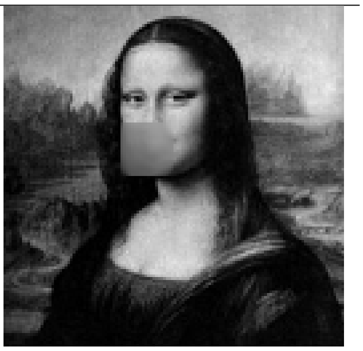
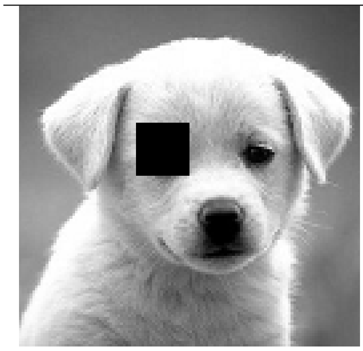
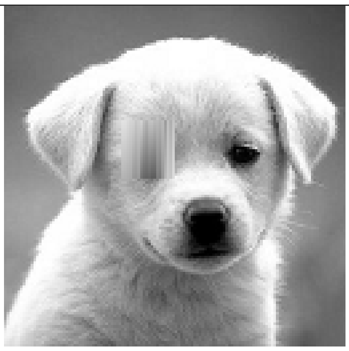
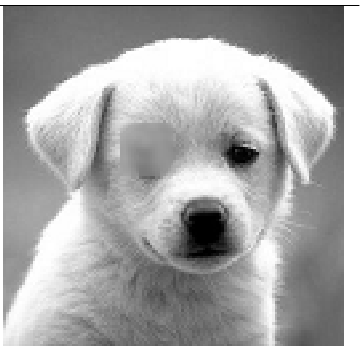
Test Images	HDMR	TV
		
PSNR	**55.4292**	40.9879
		
PSNR	42.0705	**43.0258**
		
PSNR	**41.7881**	39.2955
		
PSNR	**42.5529**	38.8301

Figure 3.6: Visual results with their corresponding PSNR values for test images with 20×20 missing region for our proposed method and TV inpainting method

Test Images	**HDMR**	**TV**
		
PSNR	**43.4788**	37.0732
		
PSNR	**29.1677**	28.6785
		
PSNR	**29.6029**	29.3509
		
PSNR	28.7023	**28.8453**

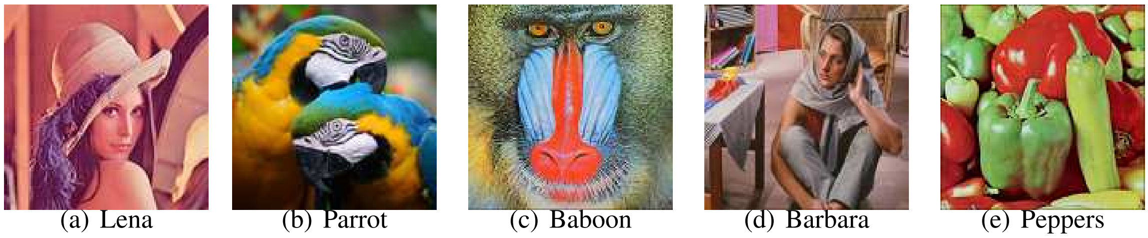
3.2 INPAINTING RECTANGULAR MISSING REGIONS IN COLOR IMAGES

In this work, we propose an image inpainting method which represents a high dimensional data in lower dimensions using High Dimensional Model Representation Method (HDMR) and performs image inpainting for rectangular regions in color images using Lagrange interpolation. Proposed approach works only on rectangular regions due to a constraint that comes from the structure of the HDMR method. Experimental results show that the proposed method produces better results than the well-known and pioneering total variation-based image inpainting method in the literature.

We used at most the bivariate terms in the method proposed in Chapter 3.1 which can inpaint only grayscale images. It has been shown in Chapter 2.3 that we also need the trivariate terms in the Equation (2.1) to represent a given color image. Therefore, in our approach for color image inpainting, we use constant, univariate, bivariate and trivariate terms in Equation (2.1).

We perform experiments on 5 different color test images shown in Figure 3.7. We design 15 different test settings by using each test image with 3 different masks with different sizes of rectangular missing regions shown in Figure 3.14. Note that black regions in each mask represent the missing region in the corresponding test setting. We compare our approach with total variation inpainting method (Shen et al., 2002) which is a pioneering inpainting approach in the literature.

Figure 3.7: Original Images

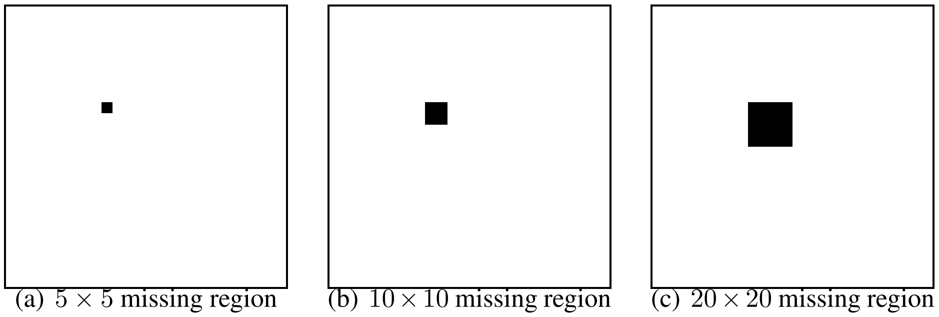


(a) Lena (b) Parrot (c) Baboon (d) Barbara (e) Peppers

Proposed method inpaints the missing regions using the nearest neighboring known pixel intensity values. We compare our results with TV inpainting method (Shen & Chan, 2002). We obtain quantitative results by comparing inpainting results of each method with the original images using PSNR. Note that higher values of PSNR indicate better inpainting results.

We present visual results with their corresponding PSNR values for 4 test images with 3

Figure 3.8: Missing Regions


(a) 5×5 missing region
(b) 10×10 missing region
(c) 20×20 missing region

different sizes of masks in each. Figures 3.9, 3.10, 3.11, 3.12, and 3.13 shows the results in images with 5×5, 10×10 and 20×20 missing regions for test images 3.7(a), 3.7(b), 3.7(c), 3.7(d) and 3.7(e), respectively. When the missing region is smaller, as in the 5×5 mask, there is no significant difference between the results of compared methods. But, when the size of the missing region grows, our method produces better results than TV inpainting method (Shen & Chan, 2002) in most of the test cases. Our approach achieves the better PSNR values in 3 of the test cases whereas, the TV approach (Shen & Chan, 2002) achieves only 1 in both masks with sizes of 10×10 and 20×20.

Figure 3.9: Visual results with their corresponding PSNR values for Test image 3.7(a) with 5×5, 10×10 and 20×20 missing regions (from top to bottom respectively) for our proposed method and TV inpainting method

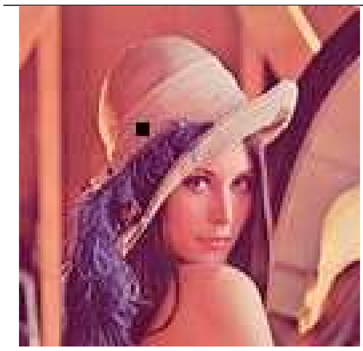
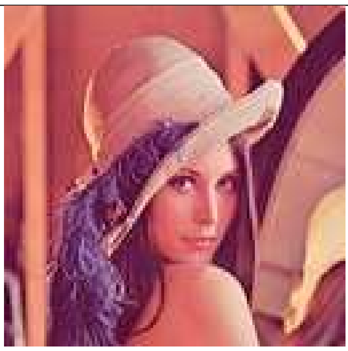
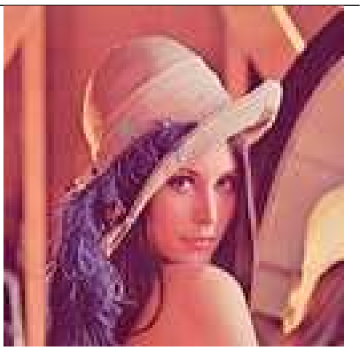
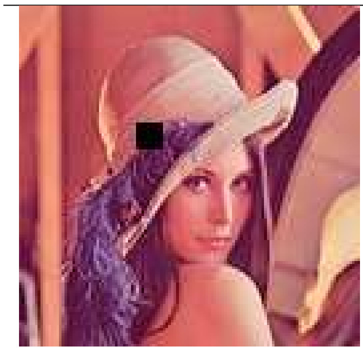
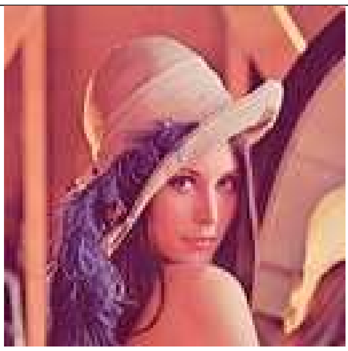
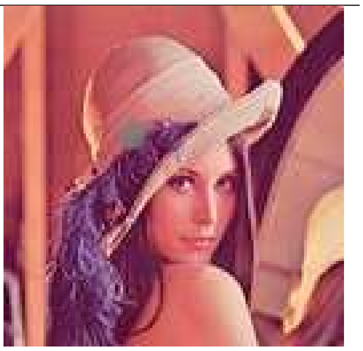
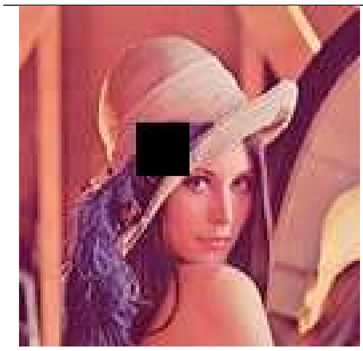
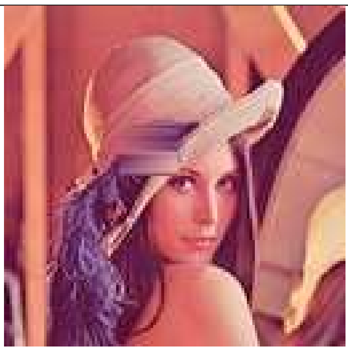
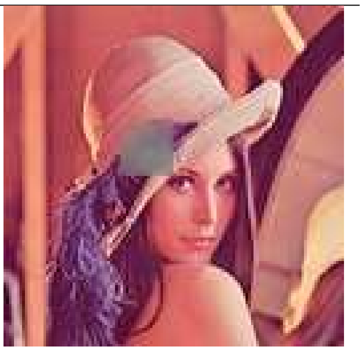
Test Images	**HDMR**	**TV**
		
PSNR	**76.8989**	66.4288
		
PSNR	**70.7002**	66.4236
		
PSNR	**63.2760**	62.8325

Figure 3.10: Visual results with their corresponding PSNR values for Test image 3.7(b) with 5×5, 10×10 and 20×20 missing regions (from top to bottom respectively) for our proposed method and TV inpainting method

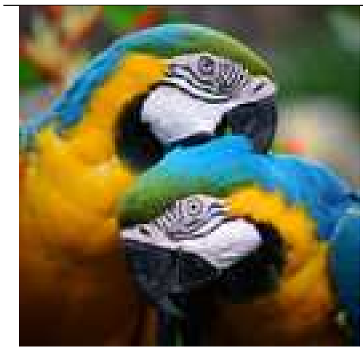
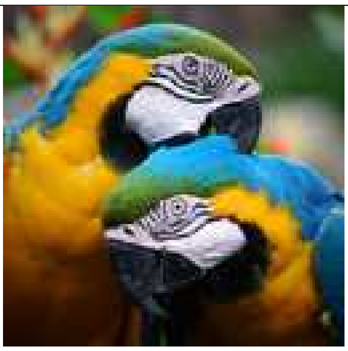
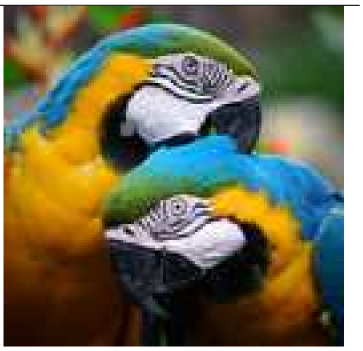
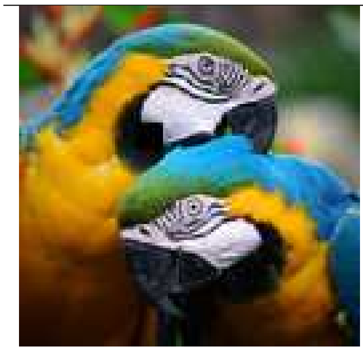
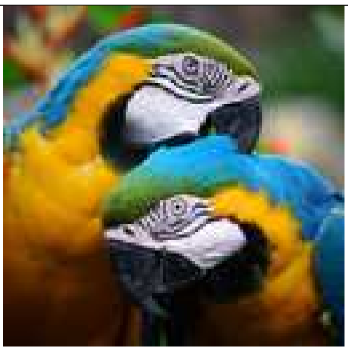
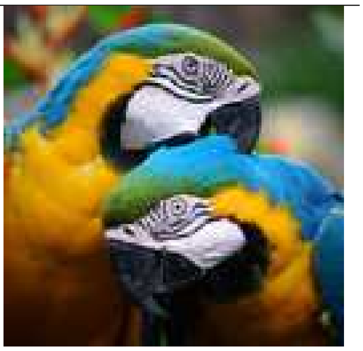
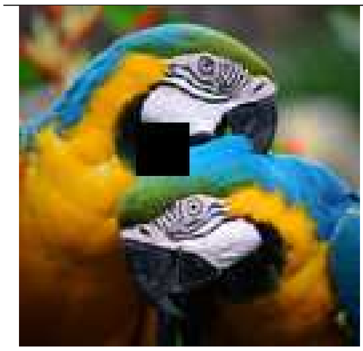
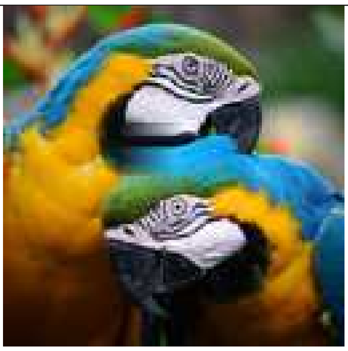
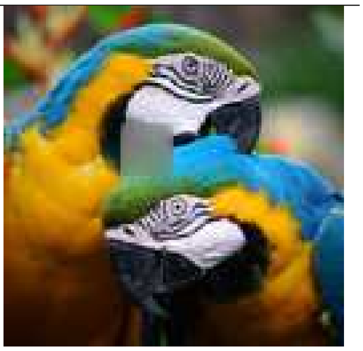
Test Images	HDMR	TV
		
PSNR	**71.8615**	59.2500
		
PSNR	**64.8924**	57.9459
		
PSNR	**62.1457**	58.5664

Figure 3.11: Visual results with their corresponding PSNR values for Test image 3.7(c) with 5×5, 10×10 and 20×20 missing regions (from top to bottom respectively) for our proposed method and TV inpainting method

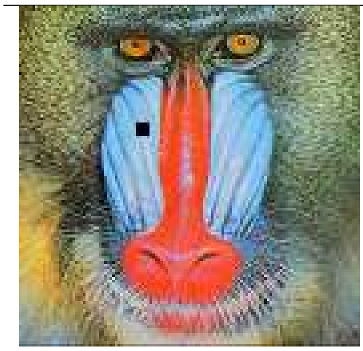
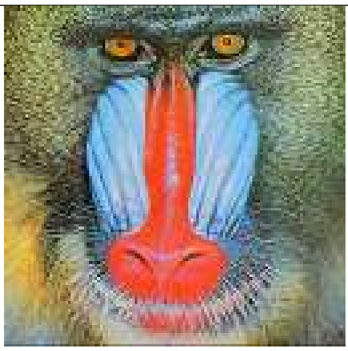
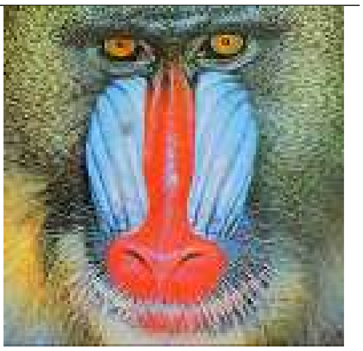
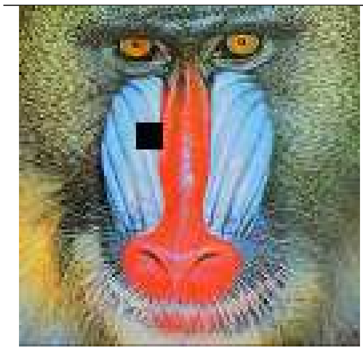
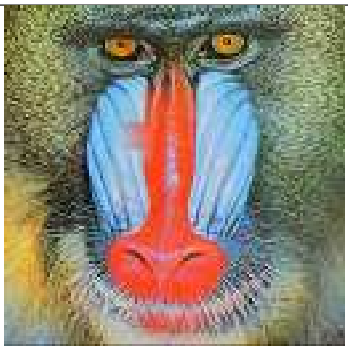
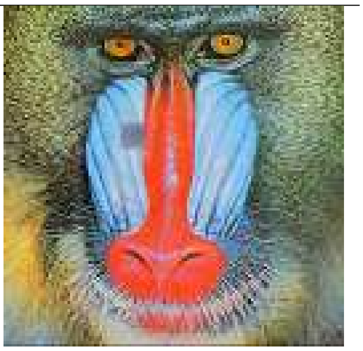
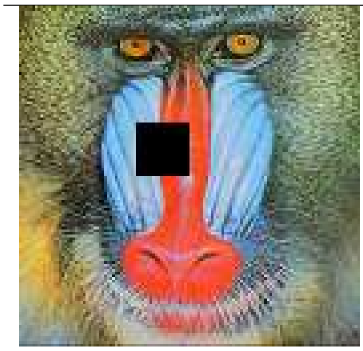
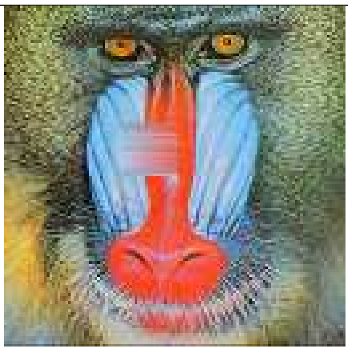
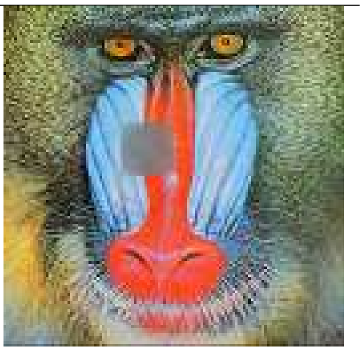
Test Images	**HDMR**	**TV**
		
PSNR	**77.0999**	64.2065
		
PSNR	**61.3470**	61.4093
		
PSNR	**61.8201**	60.1751

Figure 3.12: Visual results with their corresponding PSNR values for Test image 3.7(d) with 5×5, 10×10 and 20×20 missing regions (from top to bottom respectively) for our proposed method and TV inpainting method

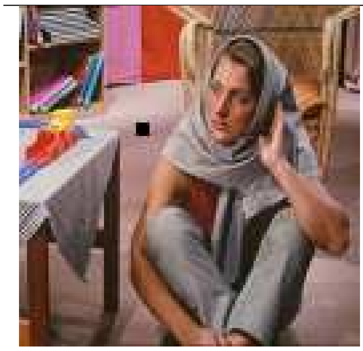
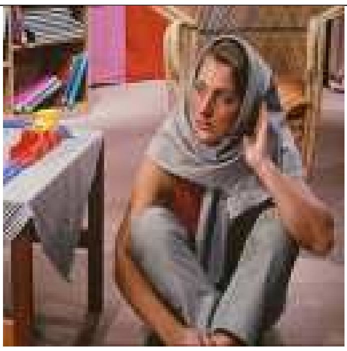
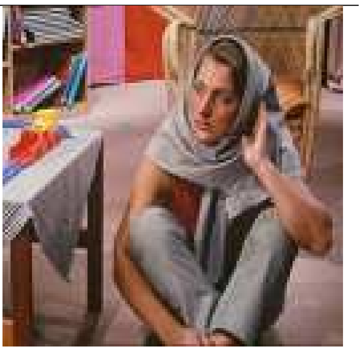
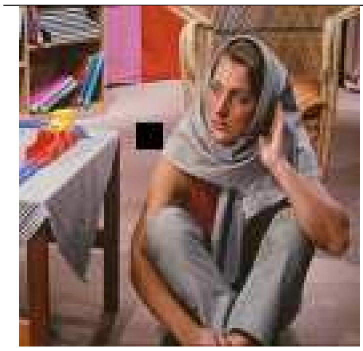
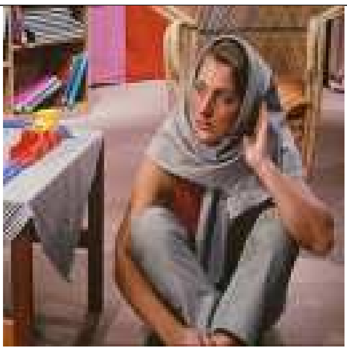
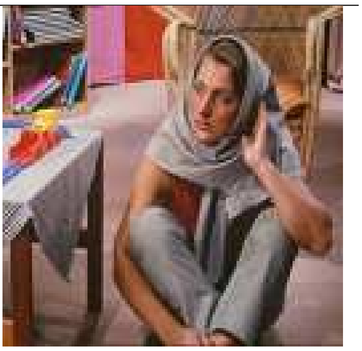
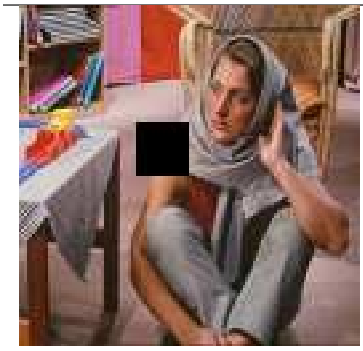
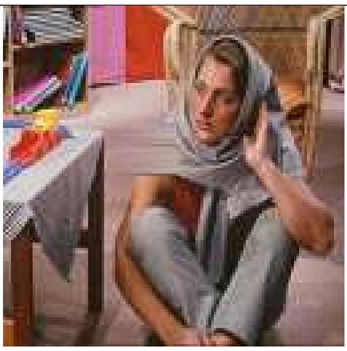
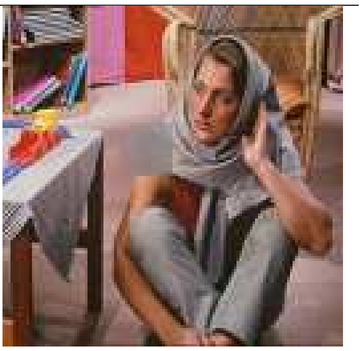
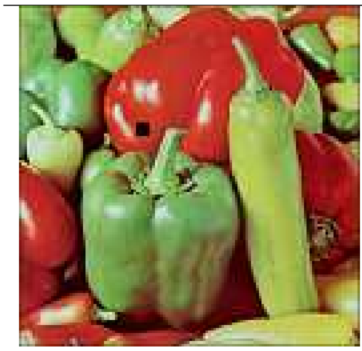
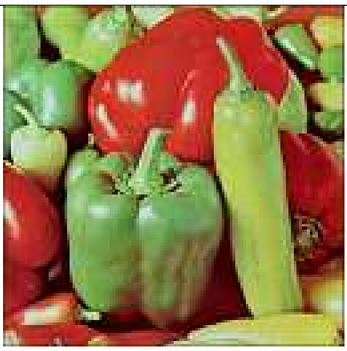
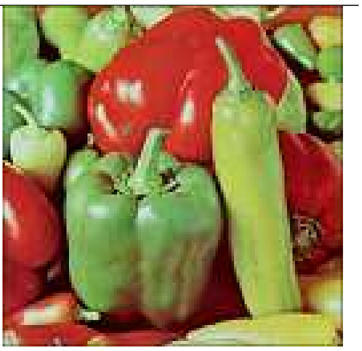
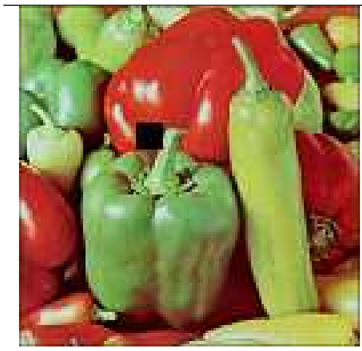
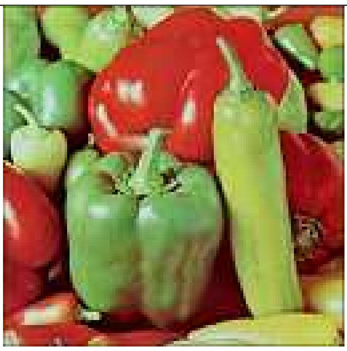
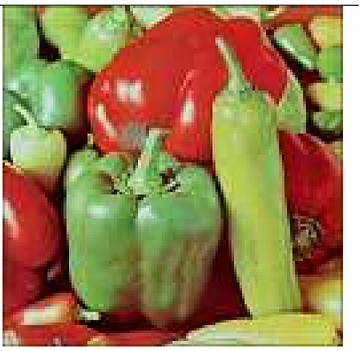
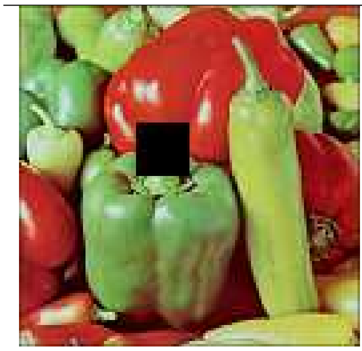
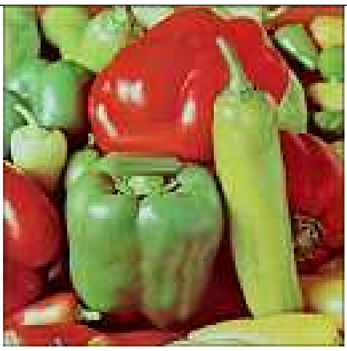
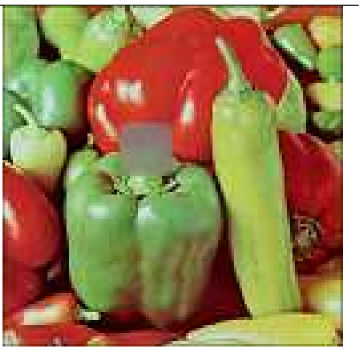
Test Images	HDMR	TV
		
PSNR	**88.7938**	71.4969
		
PSNR	**82.3690**	75.0978
		
PSNR	**69.4557**	66.8703

Figure 3.13: Visual results with their corresponding PSNR values for Test image 3.7(e) with 5×5, 10×10 and 20×20 missing regions (from top to bottom respectively) for our proposed method and TV inpainting method

Test Images	**HDMR**	**TV**
		
PSNR	**81.6617**	64.4975
		
PSNR	**62.0729**	60.5547
		
PSNR	**61.9500**	57.7178

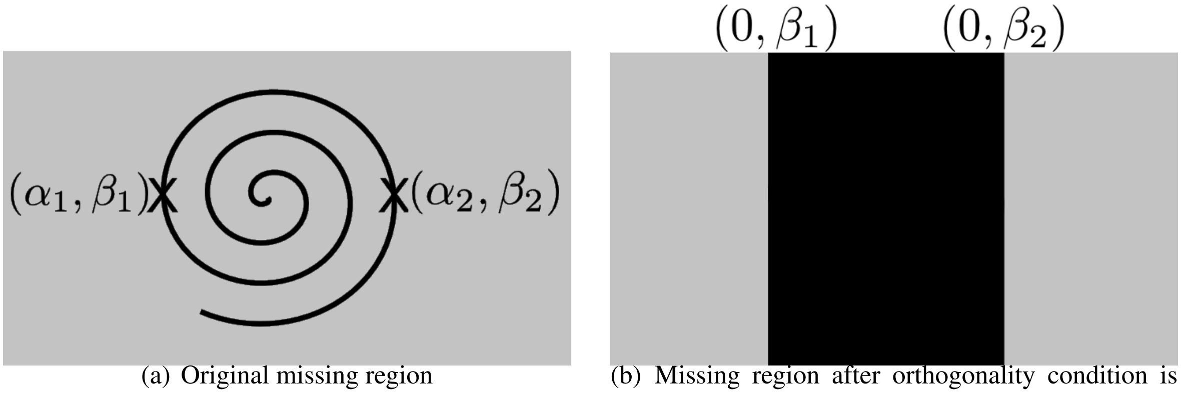
3.3 A NEW HIERARCHICAL APPROACH TO INPAINT IMAGES WITH COMPLICATED MISSING REGIONS

In this section, we present a new interpolation-based image inpainting approach which is based on 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 (Tunga & Demiralp, 2009). In order to deal with high dimensional data, we use HDMR (Sobol, 1993) 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 (Tunga & Demiralp, 2008, 2009; Karahoca & Tunga, 2015; Alış & Rabitz, 2001). However, in image inpainting, HDMR brings some difficulties due to the orthogonality condition that comes from the derivation of the HDMR equation (Tunga & Demiralp, 2008). 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.

We perform experiments on variety of test images and missing regions combinations. We also compare the accuracy of our approach with two pioneering approaches: total variation inpainting (Shen & Chan, 2002) and exemplar-based inpainting (Criminisi et al., 2004). 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.

As it is mentioned in the previous sections, when applying HDMR, the image inpainting problem in Figure 3.14(a) turns into inpainting the image shown in Figure 3.14(b). 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.

Figure 3.14: Missing Regions

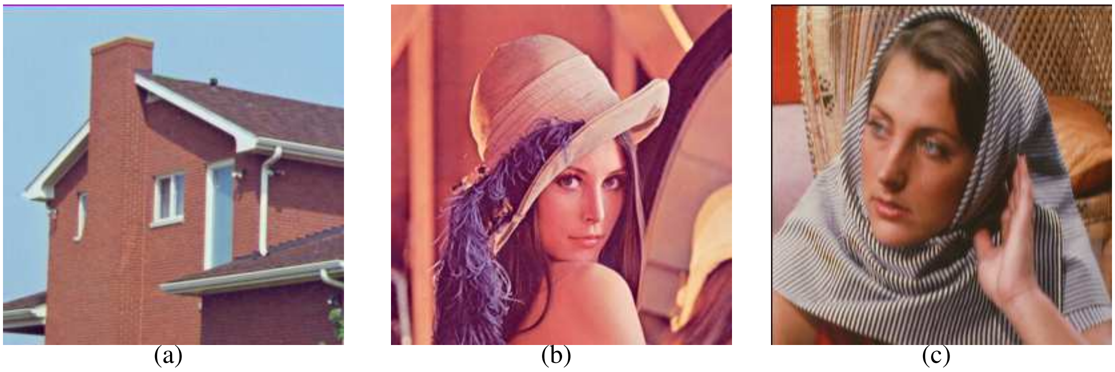


(a) Original missing region

(b) Missing region after orthogonality condition is satisfied

We perform experiments on 3 different test images shown in Figure 3.15. We design 15 different test settings by using each test image with 5 different masks shown in Figure 3.16. 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 (Shen & Chan, 2002) and exemplar-based inpainting (Criminisi et al., 2004).

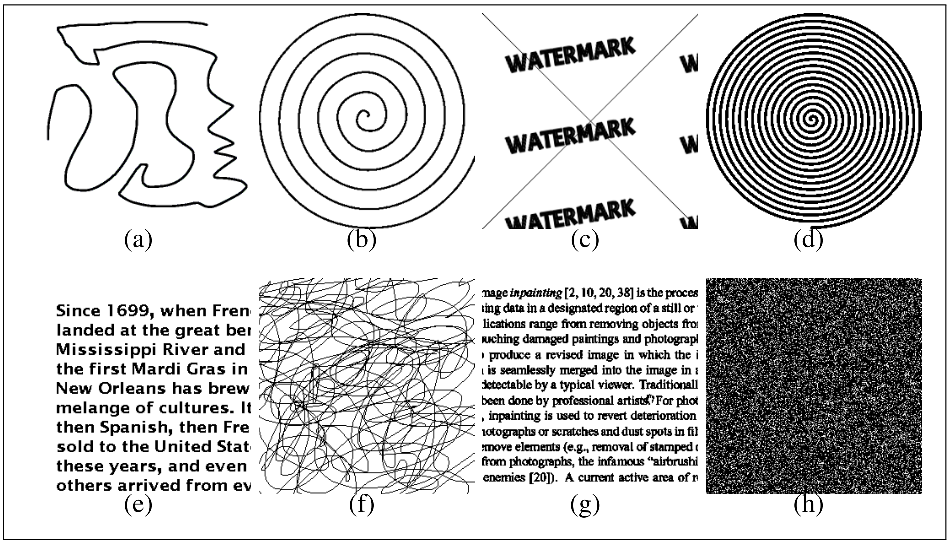
Figure 3.15: Original Images



(a) (b) (c)

We obtain quantitative results by comparing inpainting results of each method with the original images using PSNR. Note that higher values of PSNR indicate better inpainting results.

Figure 3.16: Missing Regions



(a) (b) (c) (d)

(e) (f) (g) (h)

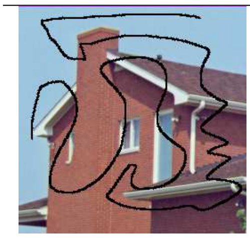
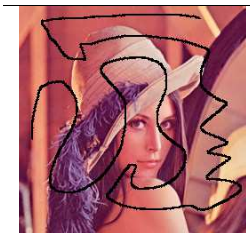
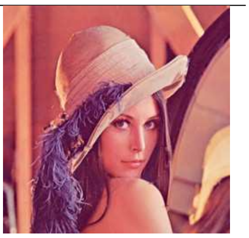
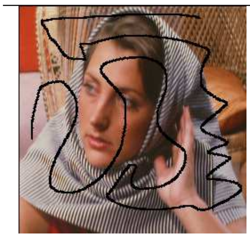
Figures 3.17, 3.18, 3.19, 3.20, and 3.21 shows the visual results and corresponding PSNR results in all test images with missing regions shown in Figures 3.16(a), 3.16(b), 3.16(c), 3.16(d), 3.16(e), 3.16(f), 3.16(g) and 3.16(h), respectively.

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 (Criminisi et al., 2004) produces the best result with the mask shown in Figure 3.16(a) 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 (Shen & Chan, 2002).

It can be seen in Figure 3.24 that Exemplar-based method cannot produce an output for this mask that has more than 80% missing pixels in it. Since exemplar-based method tries to find a similar patch to the missing region, it's impossible to find patches bigger than 1×1 in this image, thus, it cannot produce an output.

Test images given in Figures 3.15(a) and 3.15(b) 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 3.15(c) contains many textural regions like the scarf of the lady and the chair in the background. Since, the examplar-based approach in (Criminisi et al., 2004) 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 (Criminisi et al.,

Figure 3.17: Visual results with their corresponding PSNR values for all test images in 3.15 with a missing region given in Figure 3.16(a) for our proposed method, total variation (TV) inpainting method and exemplar-based inpainting method

Test Images	TV	Exemplar	HDMR
			
PSNR	37.9607	**38.2588**	38.0464
			
PSNR	35.9661	33.6417	**36.4427**
			
PSNR	32.6628	**33.1591**	32.6184

2004).

Figure 3.18: Visual results with their corresponding PSNR values for all test images in 3.15 with a missing region given in Figure 3.16(b) for our proposed method, total variation (TV) inpainting method and exemplar-based inpainting method

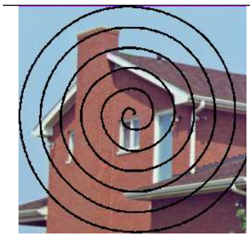
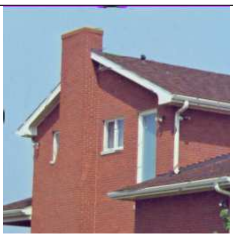
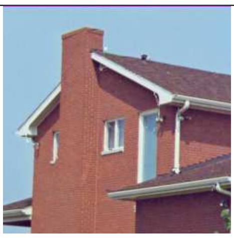
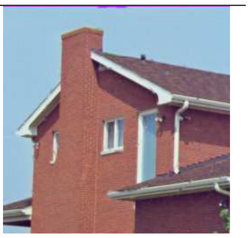
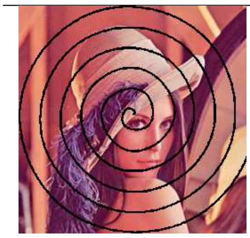
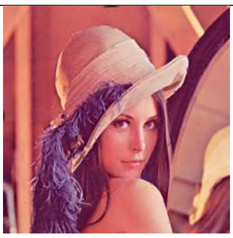
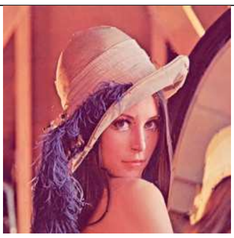
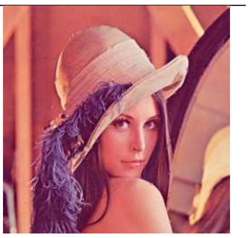
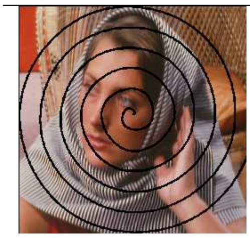
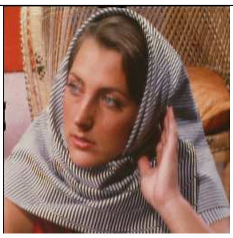
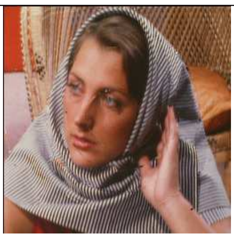
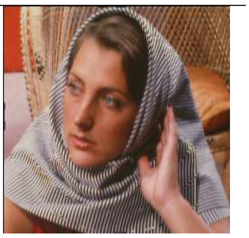
Test Images	TV	Exemplar	HDMR
			
PSNR	33.0644	35.4030	**35.9883**
			
PSNR	35.5414	33.3323	**37.1111**
			
PSNR	29.5369	**31.8904**	31.1452

Figure 3.19: **Visual results with their corresponding PSNR values for all test images in 3.15 with a missing region given in Figure 3.16(c) for our proposed method, total variation (TV) inpainting method and exemplar-based inpainting method**

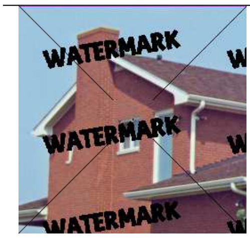
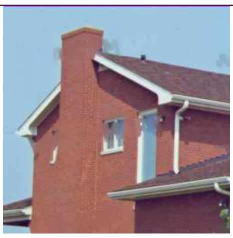
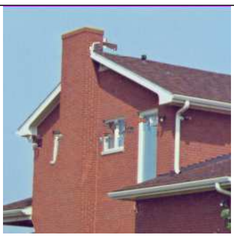
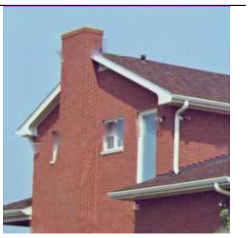
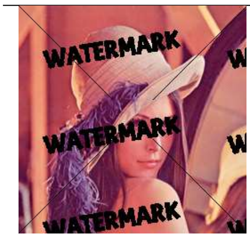
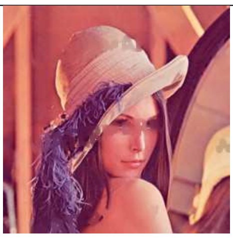
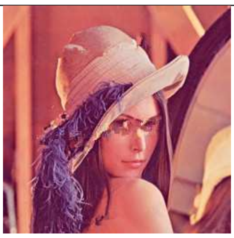
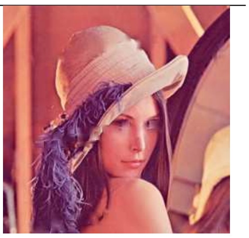
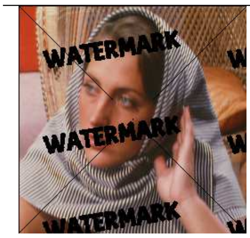
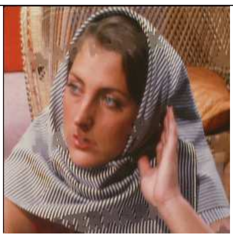
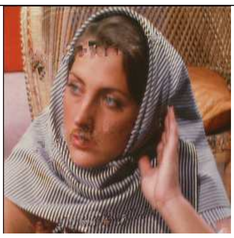
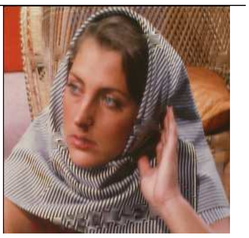
Test Images	TV	Exemplar	HDMR
			
PSNR	31.2465	30.0557	**32.4042**
			
PSNR	30.1658	28.9228	**31.3753**
			
PSNR	28.3516	27.4568	**28.3986**

Figure 3.20: Visual results with their corresponding PSNR values for all test images in 3.15 with a missing region given in Figure 3.16(d) for our proposed method, total variation (TV) inpainting method and exemplar-based inpainting method

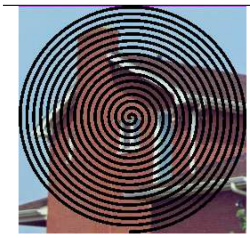
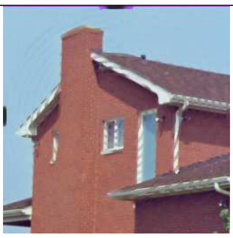
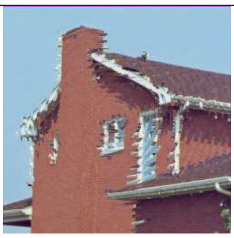
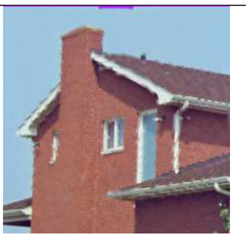
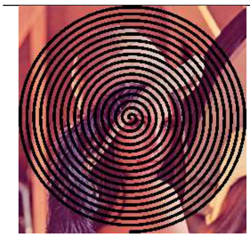
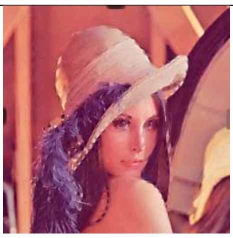
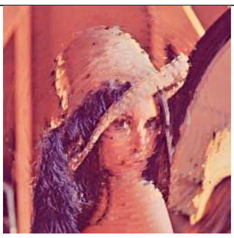
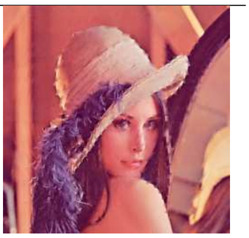
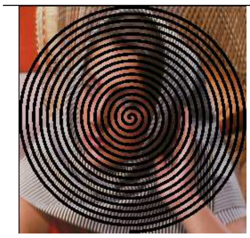
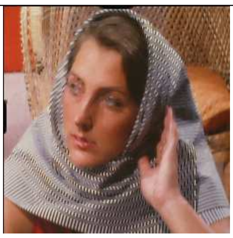
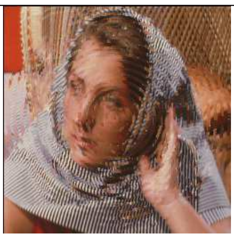
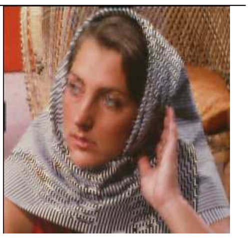
Test Images	**TV**	**Exemplar**	**HDMR**
			
PSNR	33.3071	32.0079	**34.2790**
			
PSNR	31.2359	28.7694	**31.4559**
			
PSNR	26.1509	**28.7713**	27.4096

Figure 3.21: Visual results with their corresponding PSNR values for all test images in 3.15 with a missing region given in Figure 3.16(e) for our proposed method, total variation (TV) inpainting method and exemplar-based inpainting method

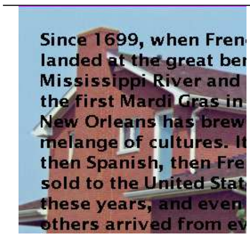
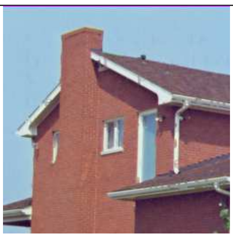
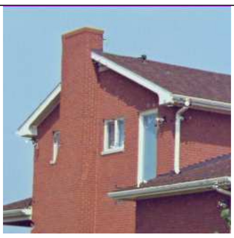
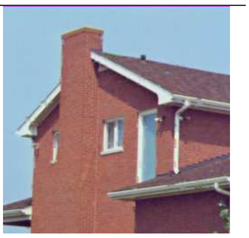
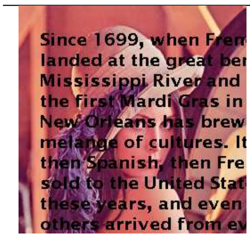
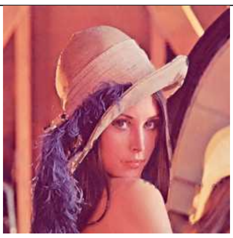
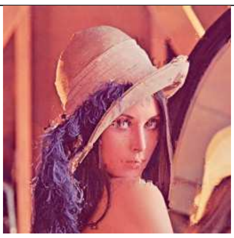
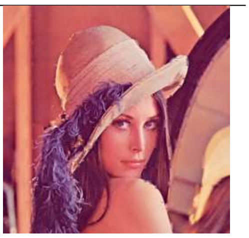
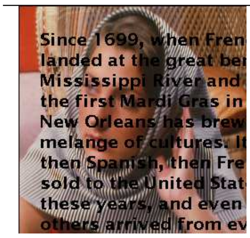
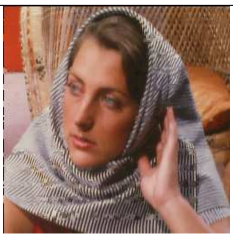
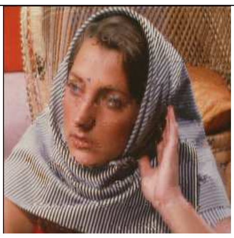
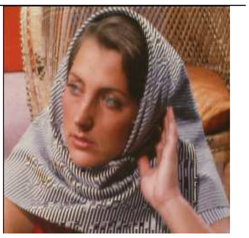
Test Images	TV	Exemplar	HDMR
 Since 1699, when Fren landed at the great ber Mississippi River and the first Mardi Gras in New Orleans has brew melange of cultures. It then Spanish, then Fre sold to the United Stat these years, and even others arrived from ev			
PSNR	27.1687	24.2372	**29.7899**
 Since 1699, when Fren landed at the great ber Mississippi River and the first Mardi Gras in New Orleans has brew melange of cultures. It then Spanish, then Fre sold to the United Stat these years, and even others arrived from ev			
PSNR	27.8587	24.3407	**28.6769**
 Since 1699, when Fren landed at the great ber Mississippi River and the first Mardi Gras in New Orleans has brew melange of cultures. It then Spanish, then Fre sold to the United Stat these years, and even others arrived from ev			
PSNR	23.9638	22.3270	**24.5221**

Figure 3.22: Visual results with their corresponding PSNR values for all test images in 3.15 with a missing region given in Figure 3.16(f) for our proposed method, total variation (TV) inpainting method and exemplar-based inpainting method

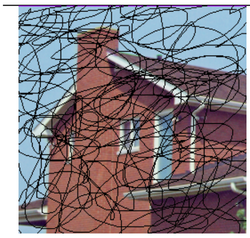
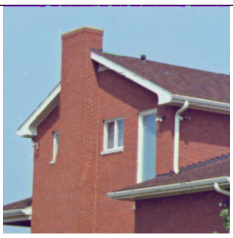
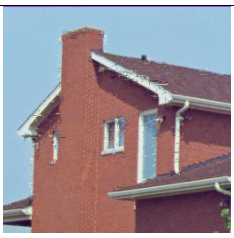
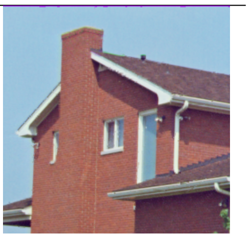
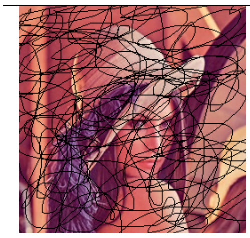
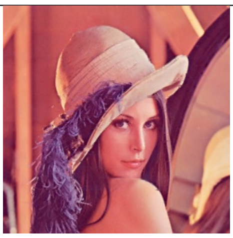
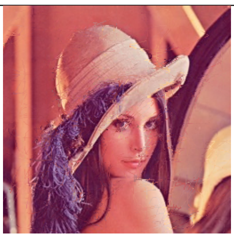
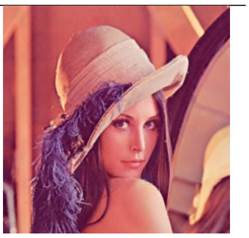
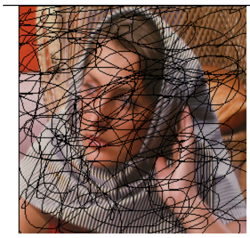
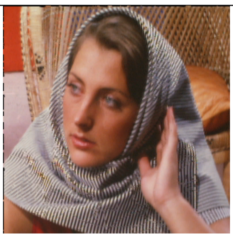
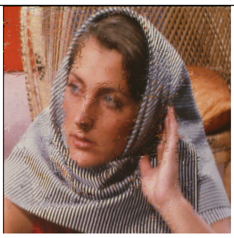
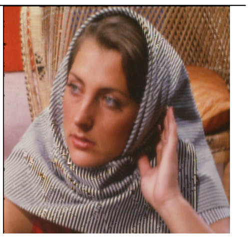
Test Images	TV	Exemplar	HDMR
			
PSNR	35.8511	29.8883	**36.5968**
			
PSNR	34.9561	28.9600	**35.2738**
			
PSNR	28.1698	26.7474	**29.4976**

Figure 3.23: **Visual results with their corresponding PSNR values for all test images in 3.15 with a missing region given in Figure 3.16(g) for our proposed method, total variation (TV) inpainting method and exemplar-based inpainting method**

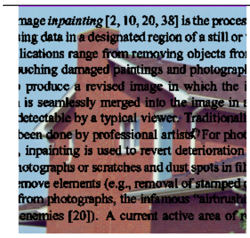
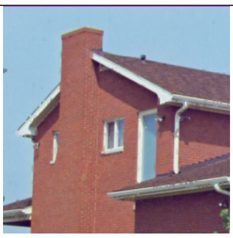
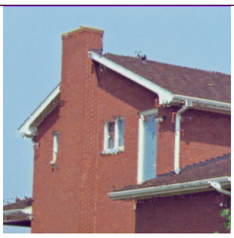
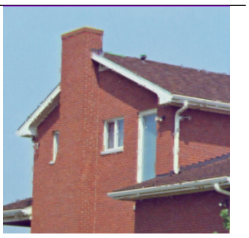
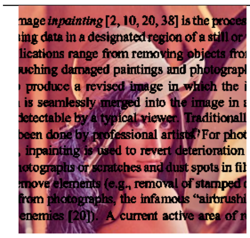
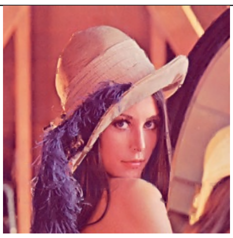
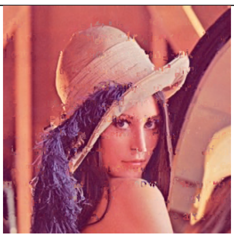
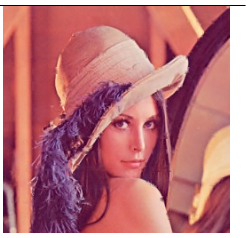
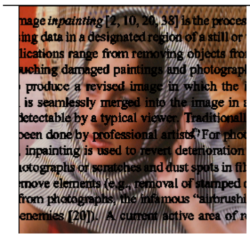
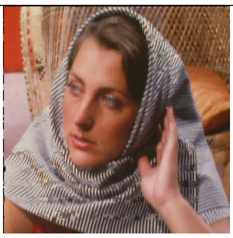
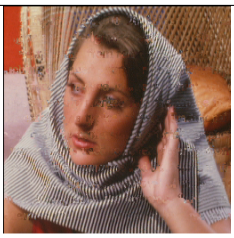
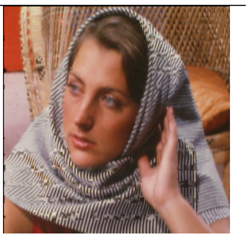
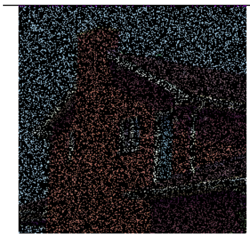
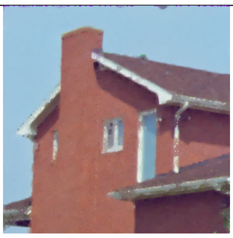
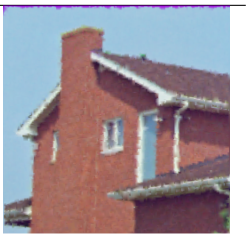
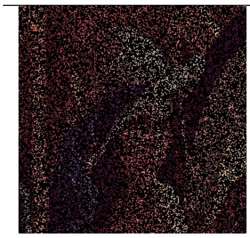
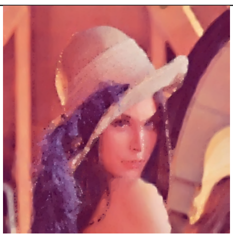
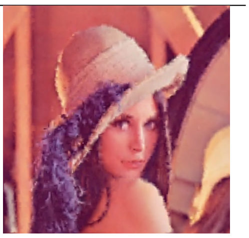
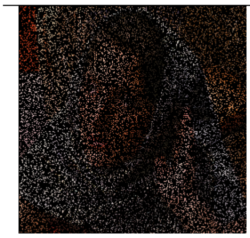
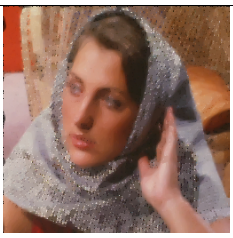
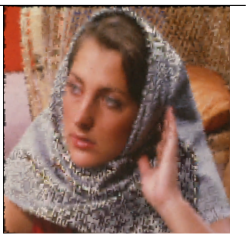
Test Images	TV	Exemplar	HDMR
			
PSNR	33.3136	29.8015	**34.0606**
			
PSNR	31.8898	27.6677	**32.1718**
			
PSNR	26.2808	26.3168	**26.4956**

Figure 3.24: **Visual results with their corresponding PSNR values for all test images in 3.15 with a missing region given in Figure 3.16(h) for our proposed method, total variation (TV) inpainting method and exemplar-based inpainting method**

Test Images	TV	Exemplar	HDMR
		NO OUTPUT	
PSNR	25.8512	-	**27.0324**
		NO OUTPUT	
PSNR	25.5556	-	**26.2642**
		NO OUTPUT	
PSNR	20.9099	-	**21.2415**

4. TEXTURE AND PATTERN PRESERVING INPAINTING USING HDMR METHOD

Interpolation-based methods that can solve the problem with a high accuracy may become inefficient when the dimension of the data increases. Also, they suffer from finding the underlying texture and pattern in the missing region. In this study, we propose a new texture and pattern preserving interpolation-based algorithm for inpainting missing regions. First, the proposed approach produces candidate inpainting results by interpolating to the observed data at the different neighborhoods of the missing region using HDMR with Lagrange interpolation. Later, a final inpainting decision is given among the candidates for each pixel in the missing region for a texture and pattern preserving inpainting. This is achieved by combining the information obtained from co-occurrence matrix and from a patch found in the image that fits best to the missing region. We evaluate the performance of the proposed approach on various color images that include different texture and pattern. We also compare the proposed approach with the state-of-the-art inpainting methods in the literature. Experimental results demonstrate the potential of the proposed approach.

Interpolation-based methods can produce inpainting results with high accuracy if a part of a smooth region is missing as shown in Figure 4.1. However, in many cases, the underlying structure of the missing region can contain complicated texture and pattern. Such structures cannot be captured by interpolating to whole surrounding pixels of the missing region. If we have prior knowledge about the direction of texture and pattern in the missing region, interpolating to the observed neighboring pixels in only that direction can help to retain the underlying structure.

We illustrate the aforementioned difficulty of using interpolation for texture and pattern preserving inpainting in Figure 4.2. Let us consider the part of a zebra body shown in Figure 4.2(a) where a small region (shown by green) on the vertical black pattern of the zebra body is missing. If we perform Lagrange interpolation only using the pixels on the left and the right parts of the missing region, we lose the vertical black texture in zebra after inpainting (see Figure 4.2(b)). However, if we use the pixels on the upper and the lower parts of the missing region for interpolation, the interpolation can complete the missing region quite well as shown in Figure 4.2(c). Inpainting results using the neighboring observed pixels on the upper-right and the lower-left, and on the upper-left and the lower-right of the missing region are also shown in Figure 4.2(b) and 4.2(d),

Figure 4.1: An example that shows interpolation works well for inpainting smooth regions. (a) The input image with a missing region (shown with black pixels), (b) inpainting result obtained using interpolation.

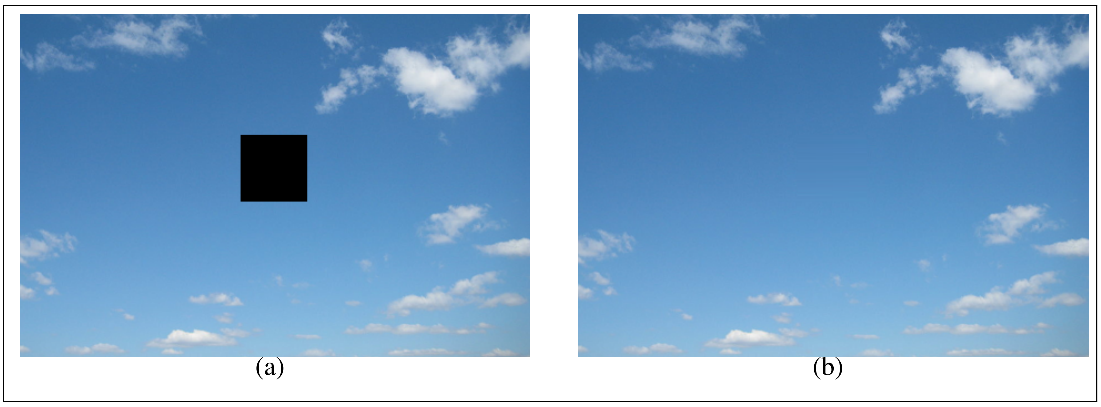
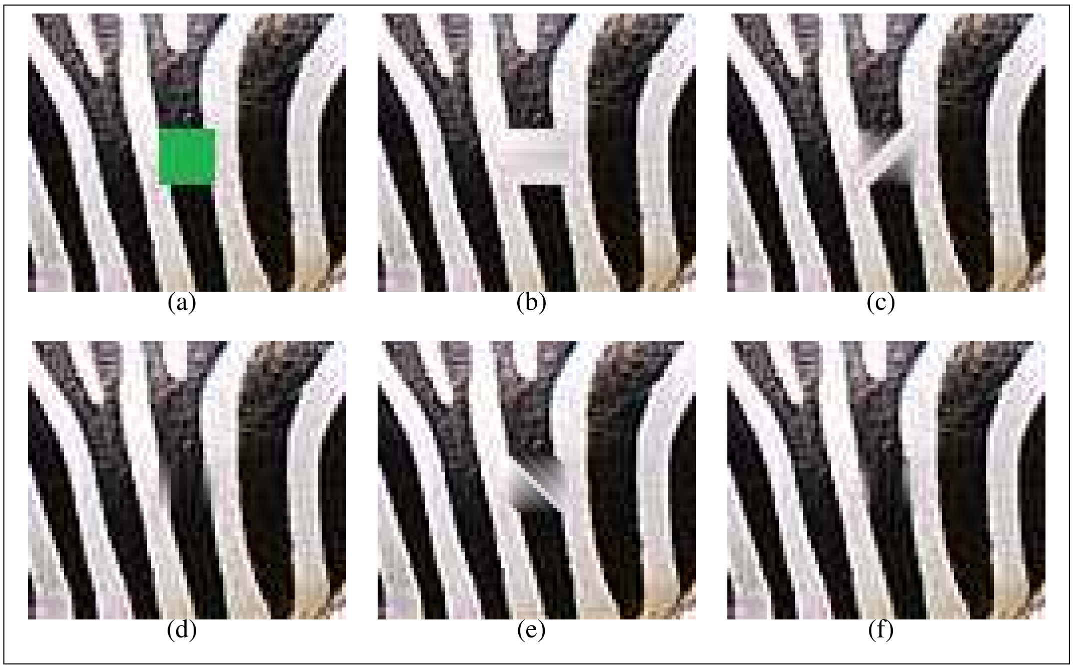


Figure 4.2: An example that demonstrates the motivation of the proposed method. (a) The input image with a missing region (shown by green). Candidate inpainting results using the neighboring pixels on (b) the left and the right (0°), (c) the upper-right and the lower-left (45°), (d) the upper and the lower (90°), (e) the upper-left and the lower-right (135°) parts of the missing region, respectively. (f) Inpainting result of the proposed method.



respectively.

Performing interpolation using the observed pixels in the direction of the underlying structure is not trivial, since we do not know the direction of the texture and the pattern in advance. The problem becomes even more complex than the example in Figure 4.2 when the underlying structure of the missing region consists of more complicated patterns (e.g., different textures and patterns in different directions) as in many natural images. This motivates us to exploit interpolation results obtained from different directions to develop an interpolation-based texture and pattern preserving algorithm for image inpainting.

The Texture and Pattern Preserving Inpainting using HDMR (TPI-HDMR) approach performs interpolation by using the observed data at different directions of the missing region and generates candidate inpainting results for each direction. Then, the algorithm selects the best intensities for each pixel of the missing region among the candidate inpainting results such that the underlying texture and pattern are preserved. The inpainting result of our approach on the image in Figure 4.2(a) is shown in Figure 4.2(f).

4.1 TPI-HDMR ALGORITHM

In this section, we introduce our interpolation-based texture and pattern preserving approach for inpainting color images. The overall algorithm is given in Algorithm 1. The algorithm consists of two major parts:
1) generating candidate inpainting results using HDMR with Lagrange interpolation (between lines 1 - 4 in Algorithm 1),
2) giving an inpainting decision among the candidates for each missing pixel by preserving the underlying texture and pattern (between lines 5 - 36 in Algorithm 1).

In the following two subsections, we explain these major parts of the proposed approach in details. Also, in the last subsection, we numerically demonstrate how the proposed algorithm works on a toy example.

4.1.1 Generating Candidate Inpainting Results

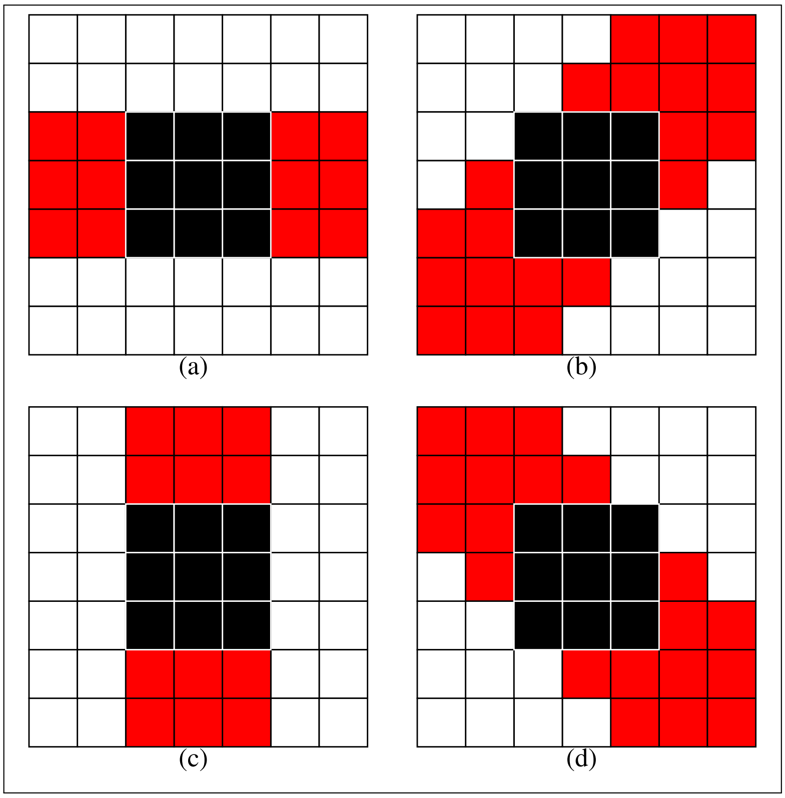
Let us assume that we are given an image I with a missing region M. Also, let us assume that we interpolate to the neighboring pixels of M from n different directions at angles $\theta = \{\theta_1, \theta_2, \dots, \theta_n\}$ to capture textures and patterns in different directions. As a result of each interpolation, the algorithm generates an inpainting estimate, M_{θ_i}, for a particular angle θ_i ($1 \leq i \leq n$).

Algorithm 1 Texture and Pattern Preserving Inpainting using HDMR (TPI-HDMR)

```
                                                  ▷ Lines 1-5 are detailed in Section 4.1.1
 1: for $i = 1 \rightarrow n$ do                                        ▷ $n$ : # of angles
 2:     Find a function, $\mathcal{F}_{\theta_i}$, by interpolating using HDMR to the neighboring pixels of
    the missing region, $M$, from the angle, $\theta_i$.
 3:     Find a candidate inpainting, $M_{\theta_i}$, by $M_{\theta_i} = \mathcal{F}_{\theta_i}(M)$.
 4: end for
 5: Extract a patch, $P$, around $M$.
                                                  ▷ Lines 6-36 are detailed in Section 4.1.2
 6: Compute co-occurrence matrix $C$ using the observed pixels in $P$.
 7: for each pixel $x$ in $M$ do
 8:     $sum = 0$
 9:     Find the set of neighboring known pixels, $\mathcal{X}$, of $x$.
10:     for $i = 1 \rightarrow n$ do
11:         $v_{\theta_i}(x) = 0$
12:         for each pixel $x' \in \mathcal{X}$ do
13:             $v_{\theta_i}(x) = v_{\theta_i}(x) + C(M_{\theta_i}(x), I(x'))$
14:         end for
15:         $sum = sum + v_{\theta_i}(x)$
16:     end for
17:     for $i = 1 \rightarrow n$ do
18:         $v_{\theta_i}(x) = v_{\theta_i}(x)/sum$                              ▷ Normalization
19:     end for
20: end for
                                                  ▷ Lines 21-22 are illustrated in Figure 4.4
21: Find a patch, $\hat{P}$, in $I$ that is the most similar to $P$ using normalized cross correlation.
22: Take the region in $\hat{P}$ that corresponds to the missing region as an inpainting estimate
    $\hat{M}$.
23: Initialize $M^{\star}$ such that it has the same size with $M$
24: for each pixel $x$ in $M$ do
25:     $sum = 0$
26:     for $i = 1 \rightarrow n$ do
27:         $c_{\theta_i}(x) = 1/\|\hat{M}(x) - M_{\theta_i}(x)\|$
28:         $sum = sum + c_{\theta_i}(x)$
29:     end for
30:     for $i = 1 \rightarrow n$ do
31:         $c_{\theta_i}(x) = c_{\theta_i}(x)/sum$                              ▷ Normalization
32:     end for
33:     $\mathcal{L}(x) = v(x) + \gamma.c(x)$                    ▷ $\gamma$ is a hyperparameter.
34:     Find the index, $i_{max}$, where the $\mathcal{L}(x)$ has the maximum value.
35:     $M^{\star}(x) = M_{\theta_{i_{max}}}(x)$
36: end for
37: return $M^{\star}$.
```

In our experiments, we empirically set $n = 4$ and $\theta = \{0°, 45°, 90°, 135°\}$. The choice of the neighboring pixels to be interpolated for each angle are illustrated by red in Figure 4.3.

Figure 4.3: Pixels used in interpolation at different angles. (a) $\theta_1 = 0°$, (b) $\theta_2 = 45°$, (c) $\theta_3 = 90°$, (d) $\theta_4 = 135°$. Note that black indicates missing pixels and red indicates pixels to be interpolated.



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 (Tunga & Demiralp, 2009; Altın, Emine Mine and Tunga, Burcu, 2014). Therefore, in this study, we use HDMR together with Lagrange interpolation to accelerate the interpolation process. For a given multivariate $\mathcal{F}$ function, the HDMR expansion is given in Equation 2.1. An exact $\mathcal{F}$ function passing through all the data points can be found by using all right-hand side terms in Equation (2.1).

In our experiments (see Section 2.3.2, we empirically found sufficient enough to use up to trivariate terms to interpolate the observed intensities. Applying the interpolation procedure mentioned in Chapter 2 to the neighboring pixels in each direction results n different functions, $\mathcal{F}_{\theta_i}$, where $1 \leq i \leq n$. Finally, each candidate inpainting, M_{θ_i}, is obtained by

$M_{\theta_i} = \mathcal{F}_{\theta_i}(M)$.

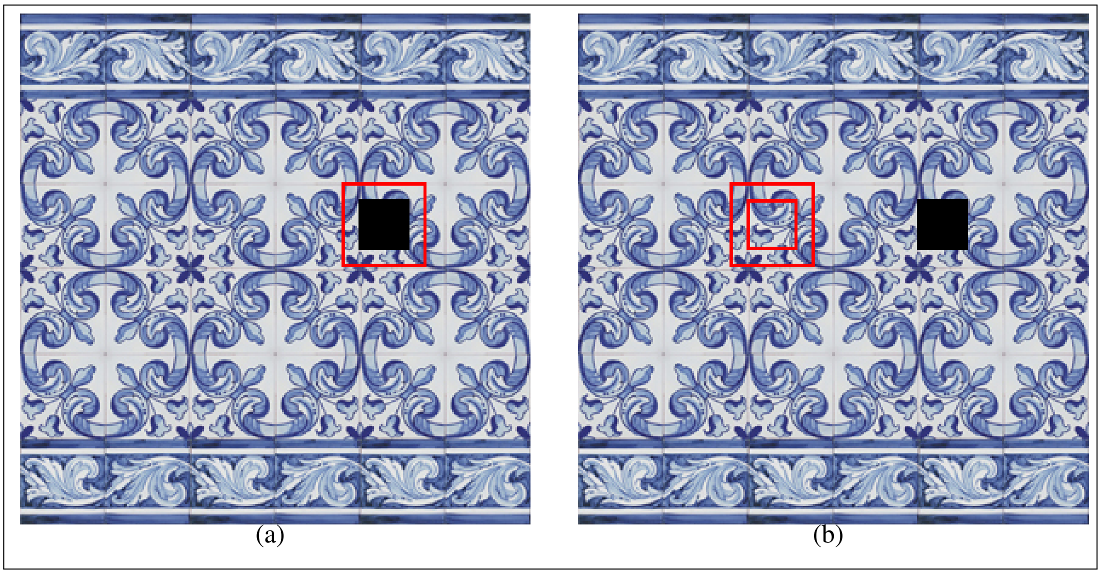
4.1.2 Texture Preserving Inpainting

We obtain n candidate inpainting results, M_{θ_i}, $1 \leq i \leq n$, by interpolating to the observed pixels at the different angles of the missing region, M, as introduced in the previous section. In other words, we have n candidate inpainting results for each pixel in the missing region. In this section, we explain how to choose one of the n candidate inpainting results of each pixel to obtain a texture and pattern preserving final inpainting. In order to achieve this, we combine the information obtained from the co-occurrence matrix (Haralick et al., 1973) and from a patch found in the image that fits best to the missing region.

Co-occurrence matrix stores the number of occurrences of neighboring pixels in an image and has been widely used to measure the texture of images (Haralick et al., 1973; Avci, 2007). In order to compute co-occurrence matrix, C, we extract a patch, P, around M (see red rectangular region in Figure 4.4(a)) and compute C using the observed pixels in P. We assume that the neighboring pixels that have higher number of occurrences in C are more likely to become a neighbor in the completed part. Therefore, for each candidate inpainting $M_{\theta_i}(x)$ of each pixel $x \in M$, we compute total number of occurrences ($v_{\theta_i}(x)$) of $M_{\theta_i}(x)$ with its neighboring observed pixels (see between lines 5 - 20 in Algorithm 1). Note that the higher values of $v_{\theta_i}(x)$ indicate that the inpainting result $M_{\theta_i}(x)$ preserves the texture with higher probability. We normalize and use this v_{θ_i} by combining it with another measure for the underlying pattern and texture to give a final inpainting decision as explained in the following paragraph.

Let us assume that there is a patch, $\hat{P}$, that is similar to the patch, P, in an image I. Then, the region $\hat{M}$ inside $\hat{P}$ that correspond to M can be a good candidate for inpainting (see Figure 4.4(b)). We find the most similar patch $\hat{P}$ to P using the normalized cross correlation (Bracewell, Ronald Newbold and Bracewell, Ronald N, 1986). We compute the absolute difference between $\hat{M}(x)$ and candidate inpainting results, $M_{\theta_i}(x)$, for each pixel x. Note that the higher values of $\frac{1}{||\hat{M}(x)-M_{\theta_i}(x)||}$ are expected. Finally, we normalize these values to obtain $c_{\theta_i}(x)$. Once we compute $v_{\theta_i}(x)$ and $c_{\theta_i}(x)$ for each pixel x in the missing region, we find the index of the candidate inpainting, i_{max}, where $v(x) + \gamma.c(x)$ is the maximum. Then, the final inpainting, $M^{\star}(x)$, of pixel x become $M_{\theta_{i_{max}}}(x)$ (see between lines 21 - 36 in Algorithm 1). Note that γ is the hyperparameter that controls the contribution of the information that comes from the co-occurrence matrix and a similar patch found by correlation coefficient. For example, one can set γ to zero if there is no similar pattern to the missing region within the image. Then, the final decision is given based on the information that is obtained from the co-occurrence matrix.

Figure 4.4: Illustrative example of finding P, $\hat{P}$, and $\hat{M}$. (a) Extracting a patch P (shown by red) around M (the black region). (b) Finding a patch $\hat{P}$ that is the most similar to P. Note that the inner red rectangular patch in (b) corresponds to $\hat{M}$.

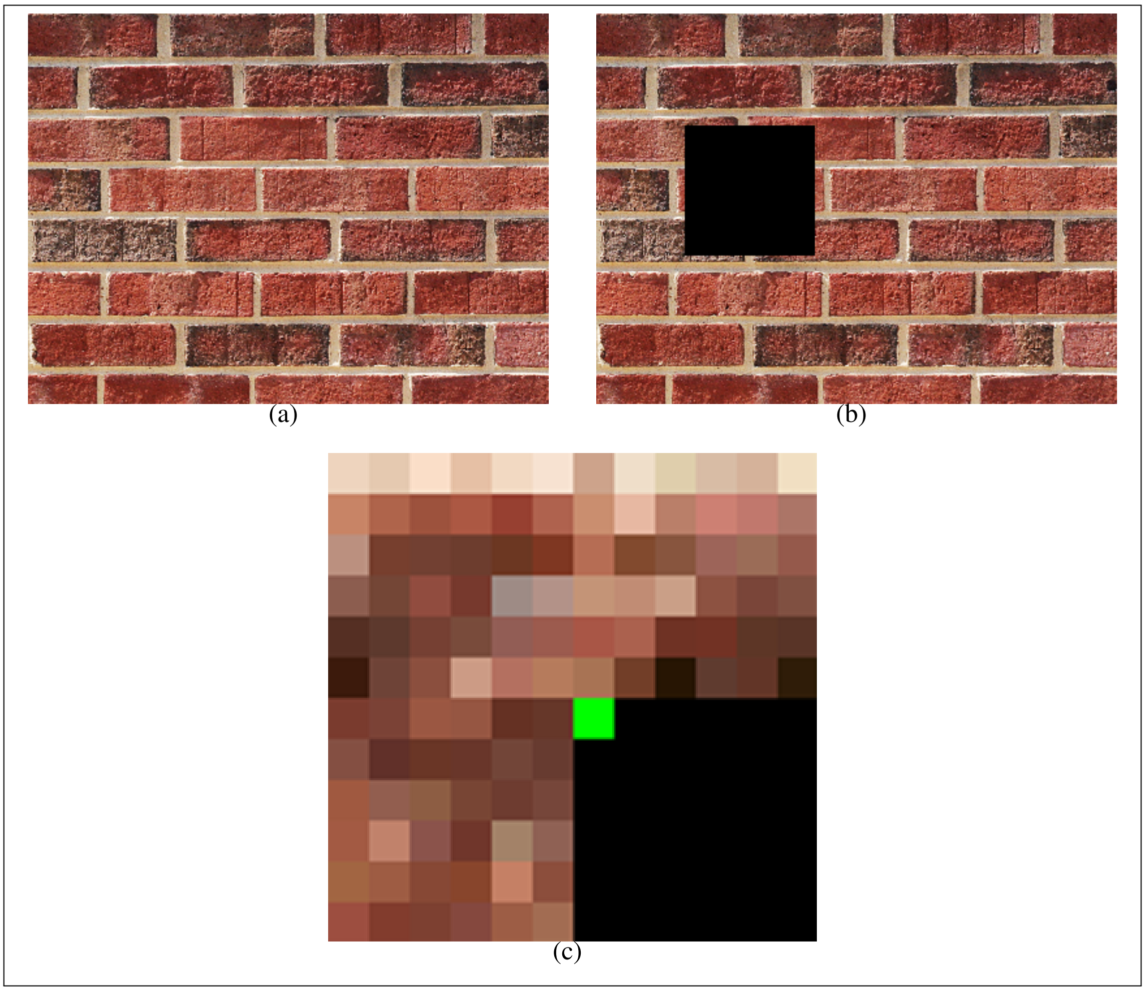


4.1.3 Toy Example Demonstration

In this section, we present a toy example demonstration to achieve a better understanding of the proposed algorithm. We use a $400 \times 300 \times 3$ brick wall image in our toy example with a $100 \times 100 \times 3$ missing region as shown in Figure 4.5. In this toy example, we numerically show how the proposed method works to inpaint a particular pixel, ϕ, shown by green in Figure 4.6(d). Note that the image in Figure 4.6(d) is obtained by zooming into the upper-left corner of M in the input image shown in Figure 4.6(b).

Let us assume that we have $n = 4$ and $\theta = \{0°, 45°, 90°, 135°\}$ in this toy example. As stated in lines between 1 - 4 in Algorithm 1, we first find 4 candidate inpainting results, $M_{\theta_1}, M_{\theta_2}, M_{\theta_3}, M_{\theta_4}$ by interpolating to the neighboring pixels of the missing region (see Figure 4.3) using HDMR. Candidate inpainting results $M_{\theta_1}, M_{\theta_2}, M_{\theta_3}, M_{\theta_4}$ for all pixels in the missing region are shown in Figure 4.6. RGB values of the candidate inpainting

Figure 4.5: (a) Original image. (b) The input image, I. Note that the black region in I corresponds to the missing region, M. (c) Zoomed upper-left corner of the missing region. Note that the pixel in green, $M(\phi)$, is to be inpainted in this toy example.



(a)

(b)

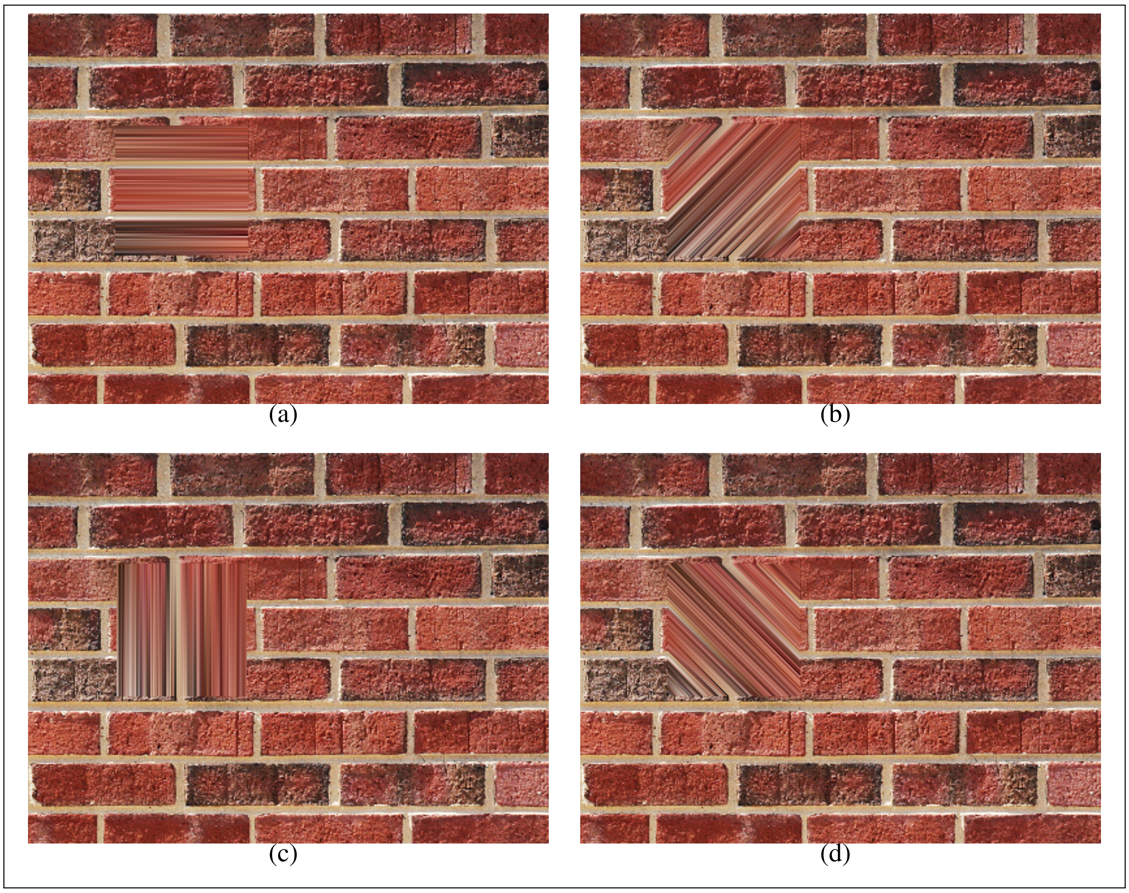
(c)

results for the green pixel ϕ (shown in Figure 4.6(d)) are as follows

$$\begin{aligned} M_{\theta_1}(\phi) &= [103\ 55\ 41] \quad & M_{\theta_2}(\phi) &= [109\ 61\ 44] \\ M_{\theta_3}(\phi) &= [168\ 116\ 85] \quad & M_{\theta_4}(\phi) &= [182\ 123\ 92] \end{aligned}$$

After finding the candidate inpainting results, we extract a patch, P, around the missing region M (see line 5 in Algorithm 1) as shown in Figure 4.7(a) and compute the co-occurence matrix, C, using the observed pixels in P (see line 6 in Algorithm 1). Then, we find the set of neighboring pixels, $\mathcal{X}$, of pixel ϕ as stated in line 9 in Algorithm 1. Finally, we compute total number of occurrences (shown by $v_{\theta_i}(x)$ in Algorithm 1 where $x = \phi$

Figure 4.6: Candidate inpainting results. (a) M_{0°, (b) M_{45°, (c) M_{90°, (d) M_{135°.



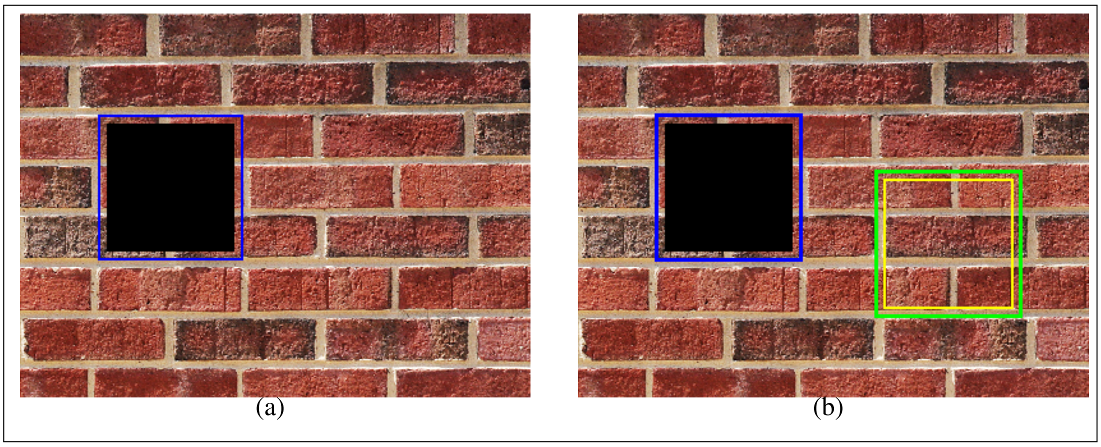
in this example) for each candidate inpainting $M_{\theta_i}(\phi)$ and $I(x')$ for each pixel $x' \in \mathcal{X}$ (see lines 10 - 16 in Algorithm 1). The normalized $v_{\theta_i}(\phi)$ values are as follows

$$v_{\theta_i}(\phi) = [0.1005\ 0.2211\ 0.3229\ 0.3555]$$

In line 21 of Algorithm 1, we find a patch, $\hat{P}$, in I that is the most similar to P using normalized cross correlation. Then, we take the region inside $\hat{P}$ that corresponds to the missing region as an inpainting estimate $\hat{M}$ (see line 22 in Algorithm 1). Regions that corresponds to $\hat{P}$ and $\hat{M}$ are shown in Figure 4.7 for this toy example. Once we find the inpainting estimate $\hat{M}(\phi)$, we compute the reciprocal absolute differences between $\hat{M}(\phi)$ and $M_{\theta_i}(\phi)$ to obtain $c_{\theta_i}(\phi)$ where $1 \leq i \leq n$ (see lines 26 - 29 in Algorithm 1). The normalized $c_{\theta_i}(\phi)$ values are as follows

$$c_{\theta_i}(\phi) = [0.0770\ 0.0850\ 0.5830\ 0.2550]$$

Figure 4.7: (a) Blue pixels around the missing region show the boundaries for the extracted patch, P. (b) The patch, $\hat{P}$, which is the most similar patch to P in I, is the region that lies in the green rectangle. The region inside the yellow rectangle is an inpainting estimate $\hat{M}$ for the missing region M.



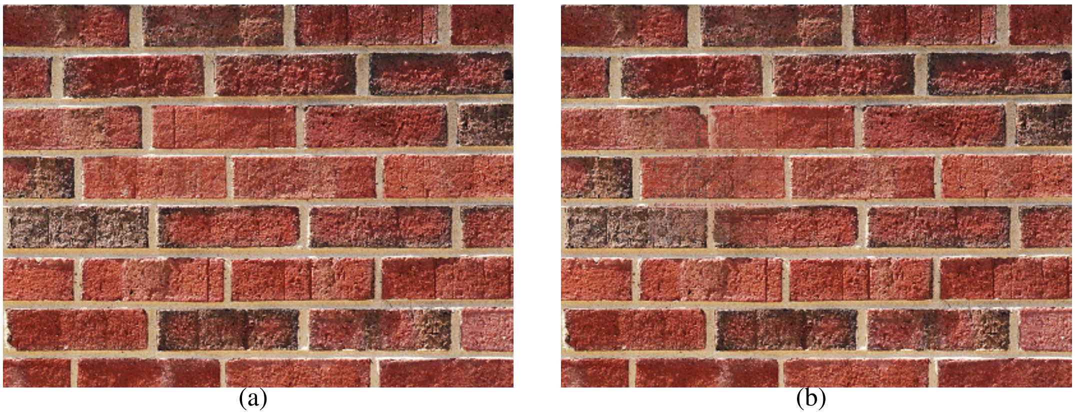
Finally, we obtain $\mathcal{L}_{\theta_i}(\phi)$ by $v_{\theta_i}(\phi) + \gamma.c_{\theta_i}(\phi)$ which we set $\gamma = 1$ for this toy example. Then, the resulting $\mathcal{L}_{\theta_i}(\phi)$ is found as follows

$$\mathcal{L}_{\theta_i}(\phi) = [0.1775\ 0.3061\ 0.9059\ 0.6105].$$

For this example, we find $i_{max} = 3$ since $\mathcal{L}_{\theta_i}(\phi)$ has its highest value at the 3^{rd} index (see line 34). To this end, the RGB values for the final inpainting $M^{\star}(\phi)$ of the green pixel ϕ becomes $M_{\theta_3}(\phi)$ which we already found above as $[168\ 116\ 85]$.

When we repeat the same procedure and inpaint each pixel $x \in M$, using the most suitable candidate, $M_{\theta_{i_{max}}}$, $M^{\star}$ is obtained as shown in Figure 4.8(b).

Figure 4.8: (a) Original Image. (b) Inpainted Image.



5. FINDINGS

In this section, we present experimental results of our proposed image inpainting approach. We perform experiments on 10 different $150 \times 150 \times 3$ and 10 different $300 \times 300 \times 3$ test images that contain complicated texture and pattern. Each test image in $150 \times 150 \times 3$ set contains a $20 \times 20 \times 3$ missing region whereas $300 \times 300 \times 3$ images contain $40 \times 40 \times 3$ missing region as shown in Figure 5.1. Note that black regions in each test image are missing. We choose the sizes and locations of the missing regions such that the underlying region contains complicated pattern and texture.

We compare our approach with 5 state-of-the-art inpainting approaches in the literature: Total Variation (TV) inpainting (Shen & Chan, 2002), Spatially Adaptive Iterative Singular-value Thresholding (SAIST) inpainting (Dong et al., 2013), exemplar-based inpainting (Criminisi et al., 2004), Hybrid Sparse Representation (HSR) inpainting (Li, 2011), Group-based Sparse Representation (GSR) inpainting (Zhang et al., 2014). We obtain quantitative results by comparing inpainting results of each method with the original images using PSNR. Note that higher values of PSNR indicate better inpainting results.

Visual inpainting results of all algorithms together with their corresponding PSNR values are presented in Figures between 5.2 and 5.21. Note that the presented images are the zoomed in versions of the test images around the missing regions to achieve better visualization.

Both visual and quantitative results demonstrate that the proposed method produces better results than the other methods in all test cases. TV inpainting method (Shen & Chan, 2002) produces blurry inpainting results in all of the test images. SAIST inpainting method (Dong et al., 2013) is originally designed for inpainting grayscale images. Therefore, it cannot produce good inpainting results in all color channels. The inconsistencies due to this deficiency of SAIST (Dong et al., 2013) can be seen from the corresponding visual results given in Figures 5.2, 5.3, 5.4, 5.8, 5.10, 5.16, 5.18 and 5.20. Although, SAIST (Dong et al., 2013) quantitatively produces slightly better inpainting results in the remaining test images, the blurriness effect in the inpainted regions can easily be observed in the visual results. Similar to SAIST (Dong et al., 2013), GSR inpainting method (Zhang et al., 2014) suffers from blurry results as well. However, it still generates inpainting results with high PSNR, since it can capture overall intensity values in the missing region. Note that, in Test Image 1, 3, 5, 10, 12, 13, 14 and 19, GSR (Zhang et al.,

2014) has the second highest PSNR values after the proposed method as shown in Figures 5.2, 5.4, 5.6, 5.11, 5.13, 5.14, 5.15 and 5.20, respectively, although visual investigation of the results are not promising. The exemplar-based inpainting (Criminisi et al., 2004) method is able to capture the underlying texture in most test images since it copies a real patch with a similar texture within the image. However, it suffers from sustaining the correct pattern. For example, in Test Image 1, the inpainted region contains the texture of a wooden surface, however, the copied patch does not perfectly retain the underlying rectangular pattern. The same problem can be observed in other test images as well (e.g. on the border of the circular region in Test Image 2 (see Figure 5.3), on the vertical straw in Test Image 4 (see Figure 5.5) and on the pattern that is composed by the brick orders in Test Image 7 (see Figure 5.8)). HSR inpainting method (Li, 2011) produces good visual inpainting results in most of the test images, however, it does not have a mechanism that exploits repeating patterns. This leads to completing the missing region with a pattern that is inconsistent with the repeating structure such as in Figures 5.5, 5.8 and 5.18. In the remaining test images, it exhibits a behaviour of finding the correct pattern with some blur effect. Also, it worths to note that the results of many of these methods get worsens as the image sizes increase.

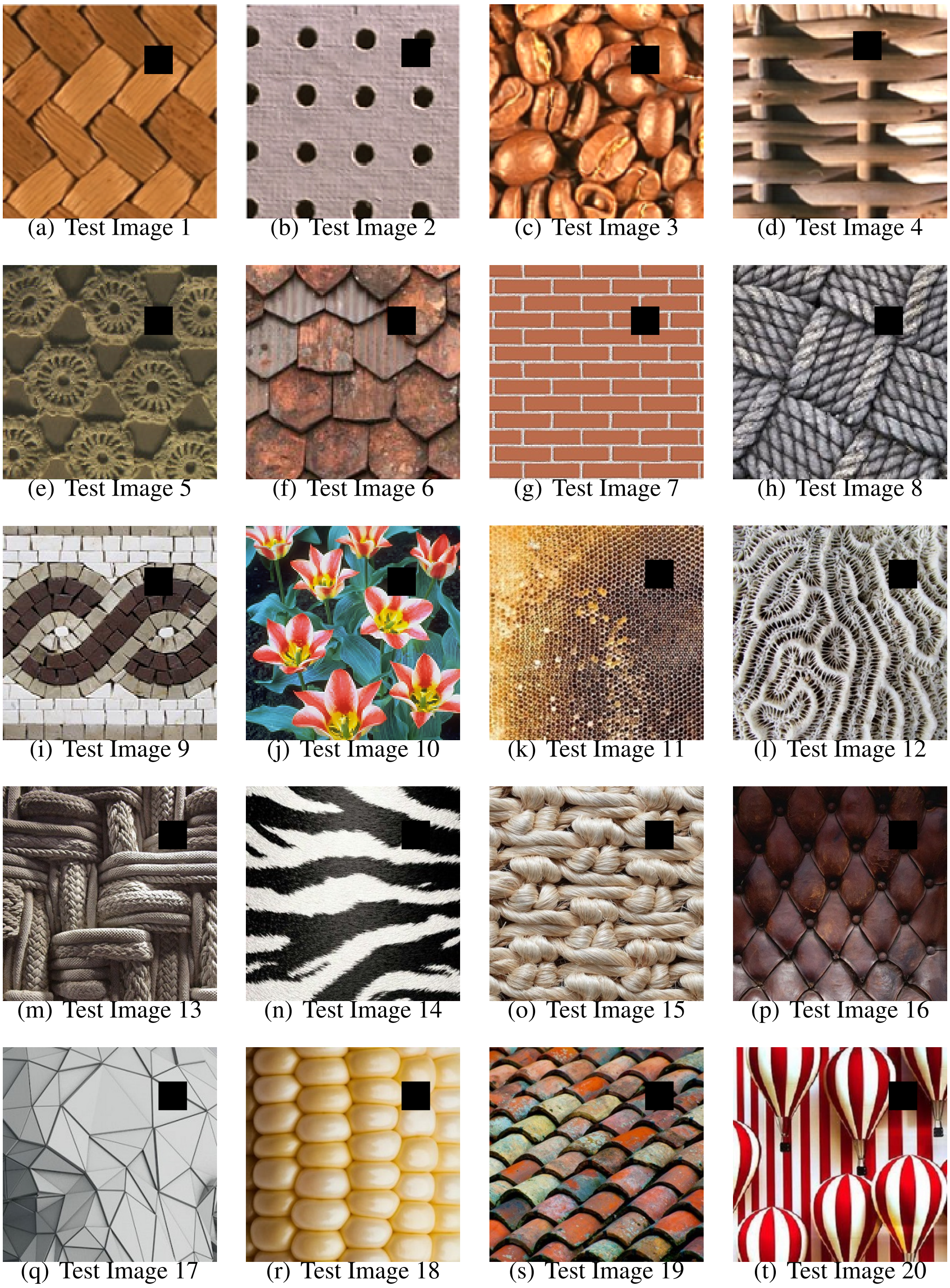
Proposed inpainting method produces better visual and quantitative results than the other methods in all test images by preserving the underlying pattern and texture in most of the images. All of the test images have similar patterns to be matched with the missing region, but these patches have different luminance, illuminance and intensity distributions. So, inpainting the image by copying the similar patch directly into the missing region does not provide satisfactory results as shown from the results of exemplar-based inpainting (Criminisi et al., 2004). We exploit similar patch to select one of the outputs of interpolations from different angles. Therefore, while the proposed method sustains the underlying texture and pattern, it also considers intensity changes with the aid of interpolation. Thus, the PSNR results and visual results are better than the compared methods and promising for further research.

We also perform execution time comparison between the proposed method and the other methods in the literature. We measure the average running time of the algorithms on both $150 \times 150 \times 3$ and $300 \times 300 \times 3$ images that we use in our experiments. The execution time results are shown in Table 5.1. According to the results, the proposed method is in the 3^{rd} order among all methods in terms of the running time.

Table 5.1: Average running time of the algorithms on both $150 \times 150 \times 3$ and $300 \times 300 \times 3$ images.

	Time (sec.)					
	TV	Exemplar	SAIST	HSR	GSR	Proposed
$150 \times 150 \times 3$	0.997	0.641	80.722	151.688	1734.097	44.173
$300 \times 300 \times 3$	1.648	4.059	366.940	524.474	6829.783	262.972

Figure 5.1: Test images that are used in our experiments. Note that Test Image 1-10 are $150 \times 150 \times 3$ with $20 \times 20 \times 3$ missing region whereas Test Image 11-20 are $300 \times 300 \times 3$ with $40 \times 40 \times 3$ missing region. Missing regions are indicated by black pixels.



(a) Test Image 1 (b) Test Image 2 (c) Test Image 3 (d) Test Image 4

(e) Test Image 5 (f) Test Image 6 (g) Test Image 7 (h) Test Image 8

(i) Test Image 9 (j) Test Image 10 (k) Test Image 11 (l) Test Image 12

(m) Test Image 13 (n) Test Image 14 (o) Test Image 15 (p) Test Image 16

(q) Test Image 17 (r) Test Image 18 (s) Test Image 19 (t) Test Image 20

Figure 5.2: Visual inpainting and corresponding PSNR results for Test Image 1 in Figure 5.1(a).

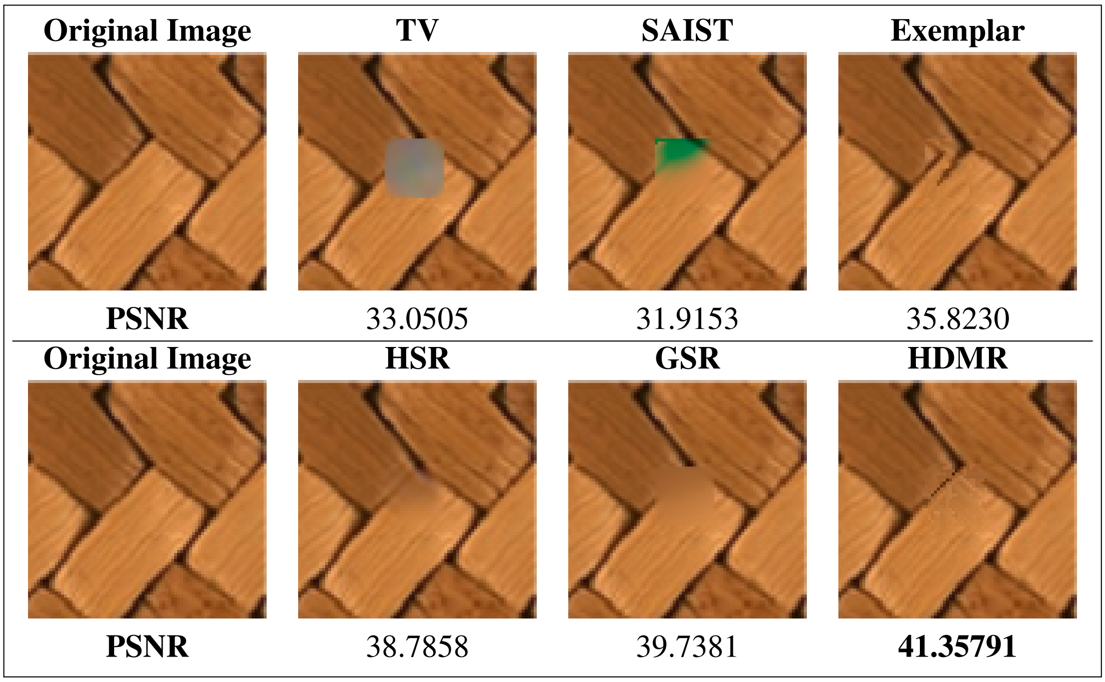


Figure 5.3: Visual inpainting and corresponding PSNR results for Test Image 2 in Figure 5.1(b).

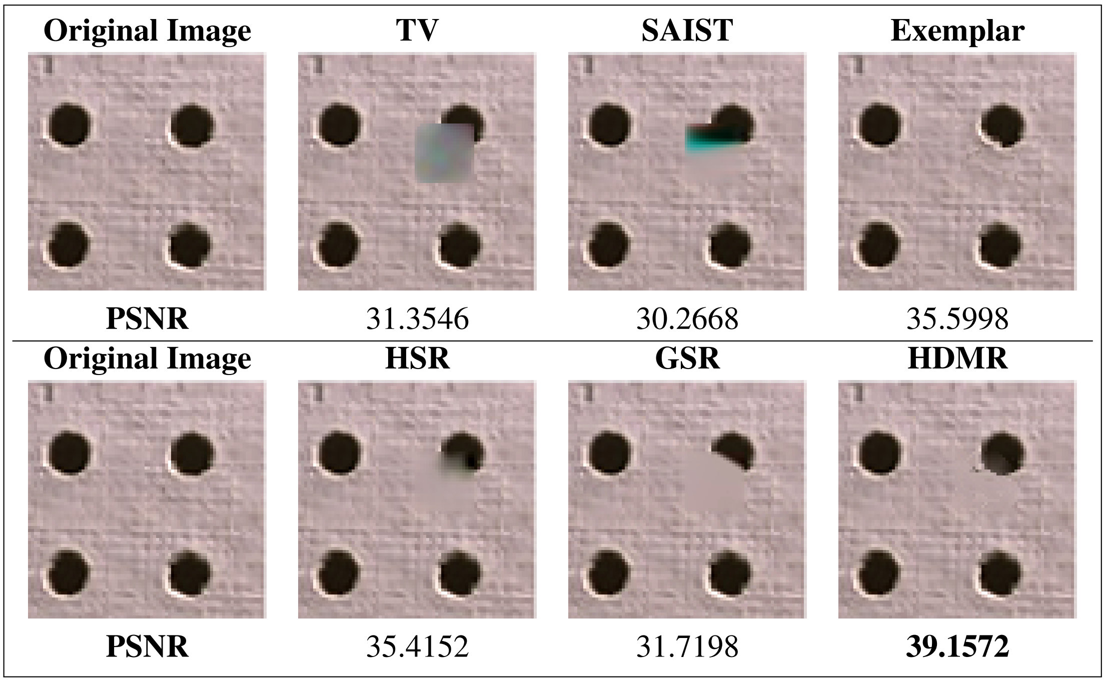
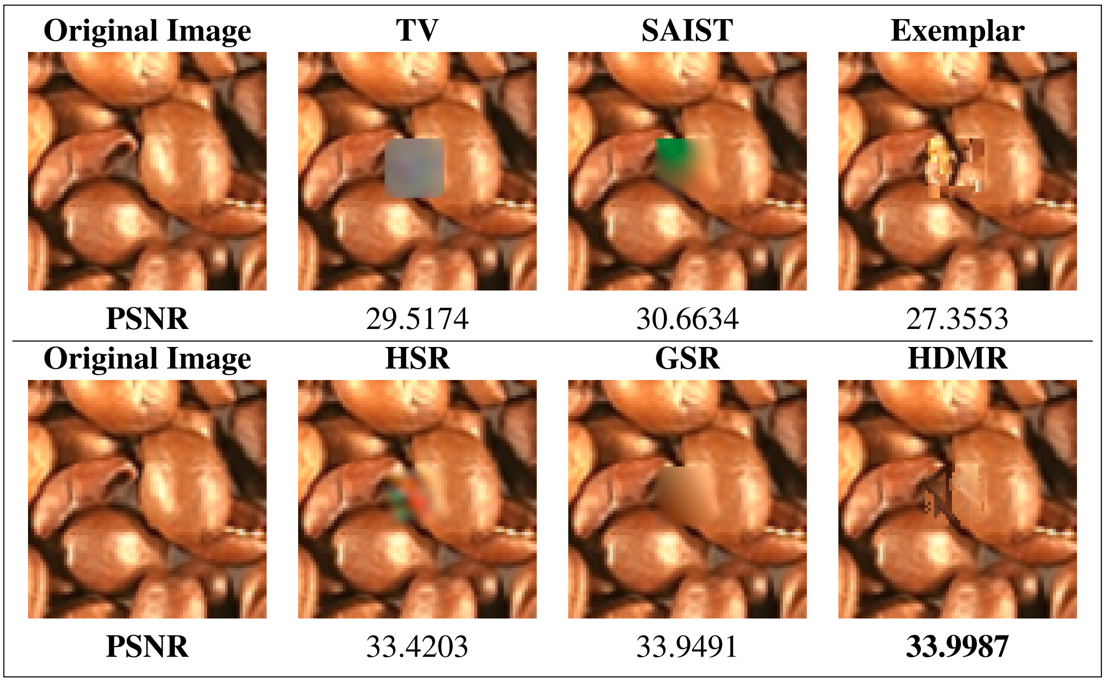
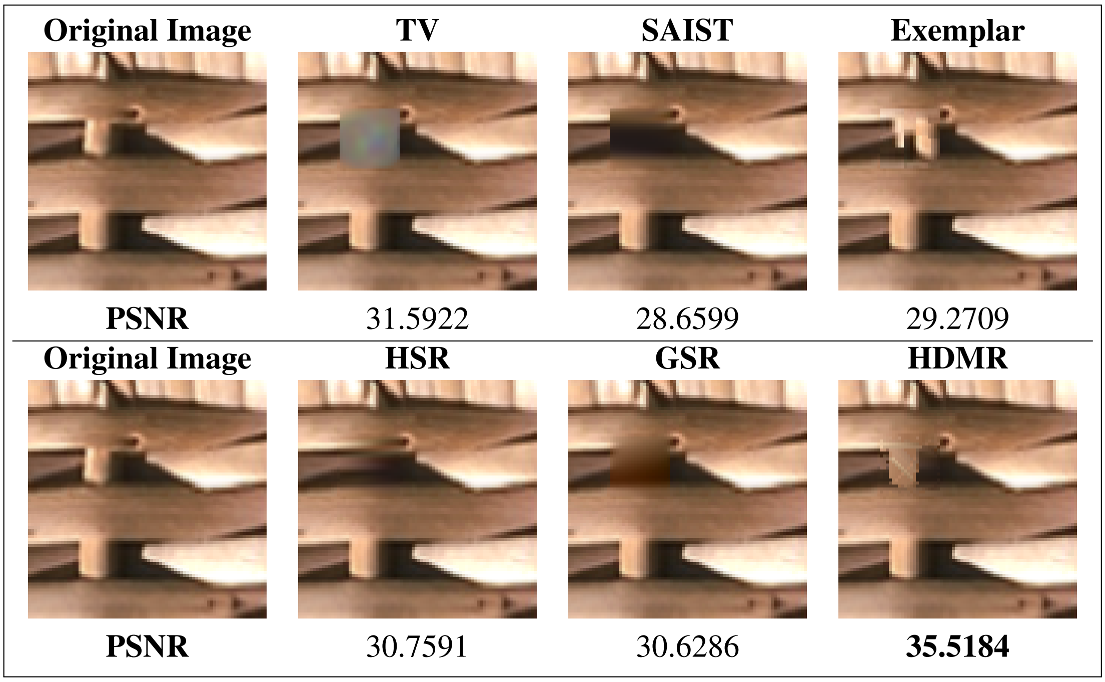


Figure 5.4: Visual inpainting and corresponding PSNR results for Test Image 3 in Figure 5.1(c).



Original Image	TV	SAIST	Exemplar
PSNR	29.5174	30.6634	27.3553
Original Image	**HSR**	**GSR**	**HDMR**
PSNR	33.4203	33.9491	**33.9987**

Figure 5.5: Visual inpainting and corresponding PSNR results for Test Image 4 in Figure 5.1(d).



Original Image	TV	SAIST	Exemplar
PSNR	31.5922	28.6599	29.2709
Original Image	**HSR**	**GSR**	**HDMR**
PSNR	30.7591	30.6286	**35.5184**

Figure 5.6: Visual inpainting and corresponding PSNR results for Test Image 5 in Figure 5.1(e).

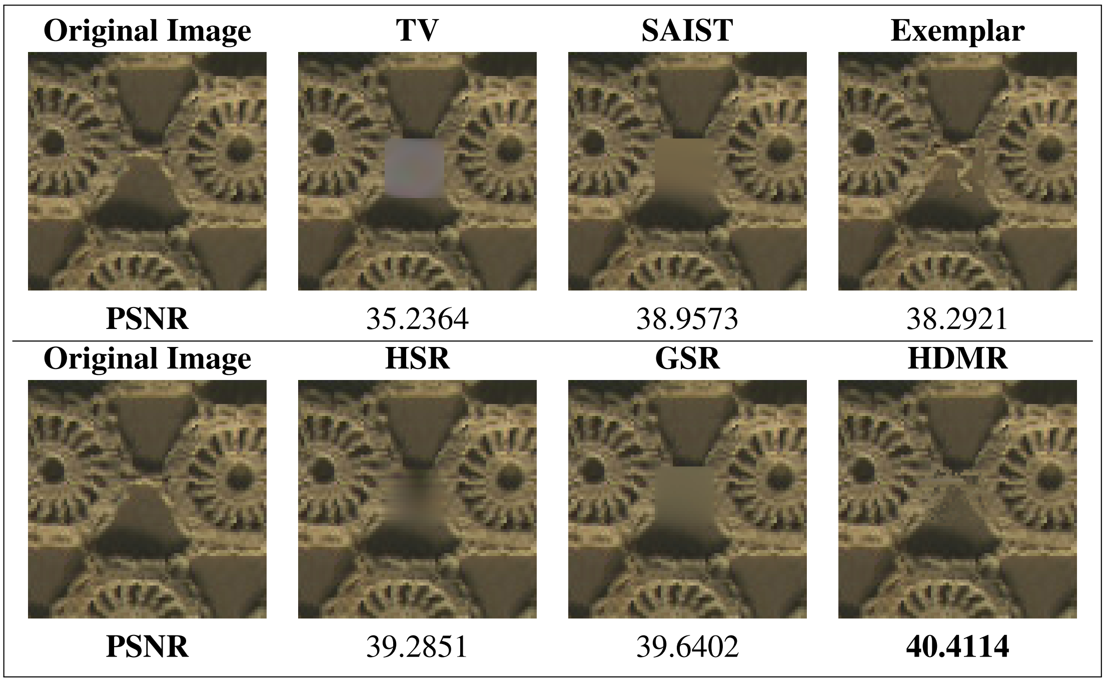


Figure 5.7: Visual inpainting and corresponding PSNR results for Test Image 6 in Figure 5.1(f).

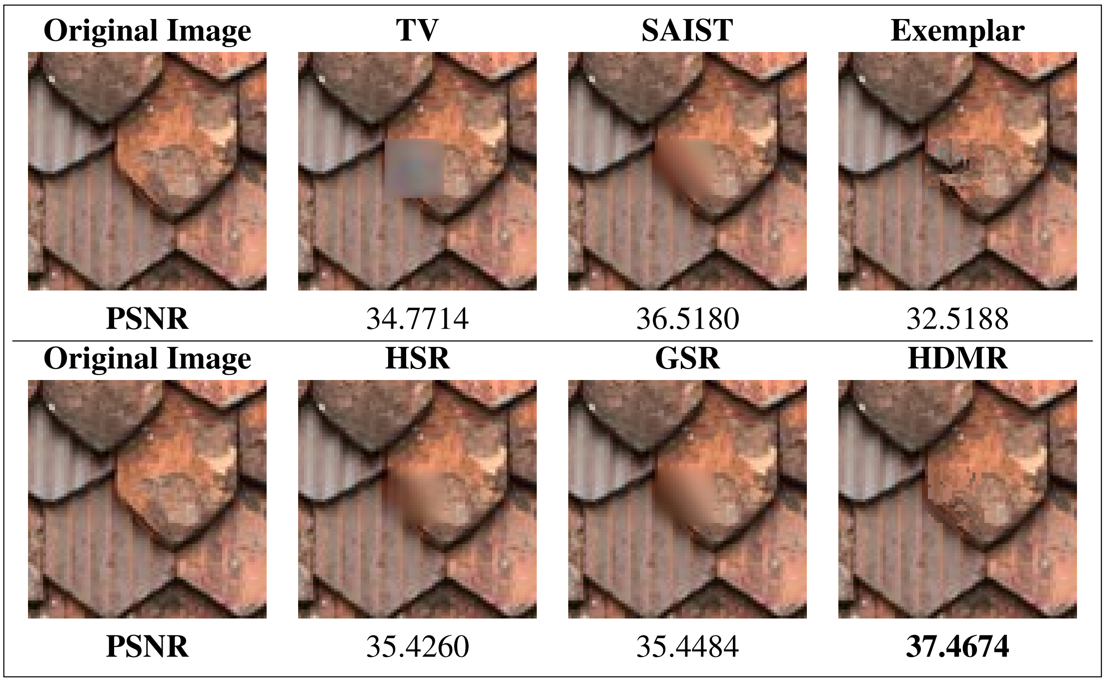


Figure 5.8: Visual inpainting and corresponding PSNR results for Test Image 7 in Figure 5.1(g).

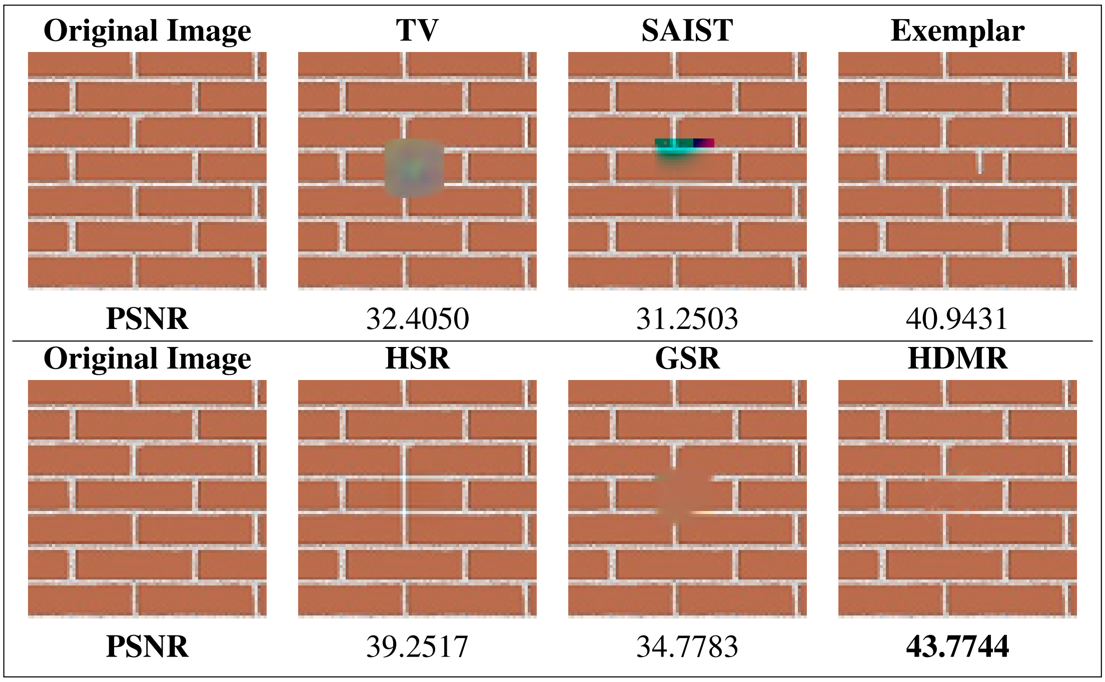


Figure 5.9: Visual inpainting and corresponding PSNR results for Test Image 8 in Figure 5.1(h).

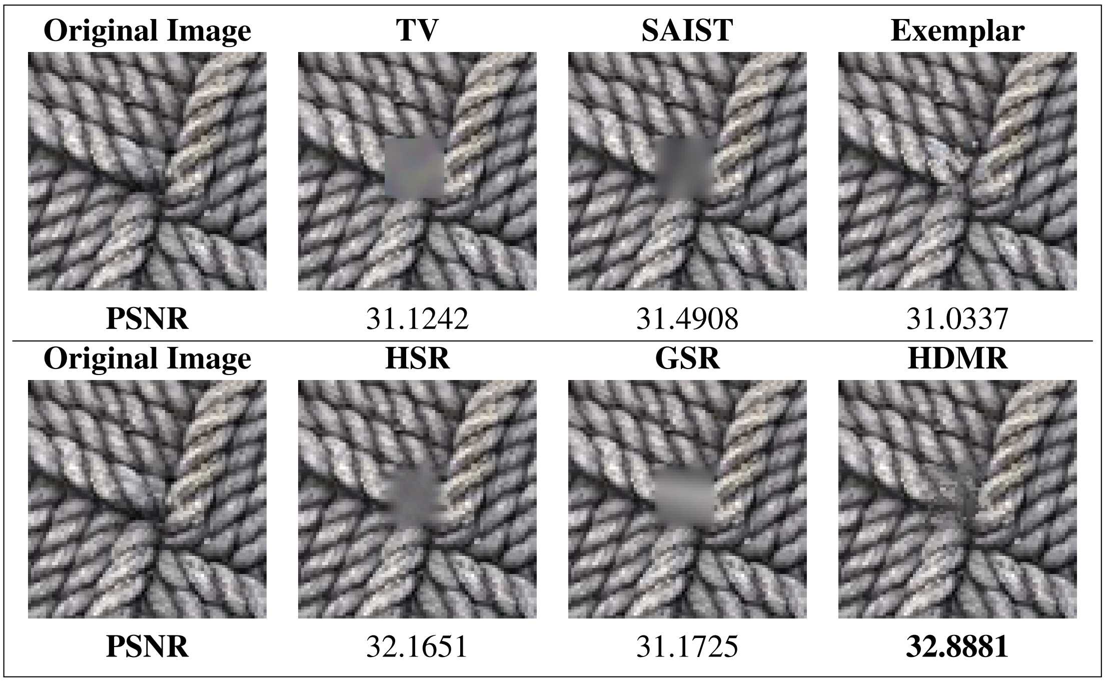


Figure 5.10: Visual inpainting and corresponding PSNR results for Test Image 9 in Figure 5.1(i).

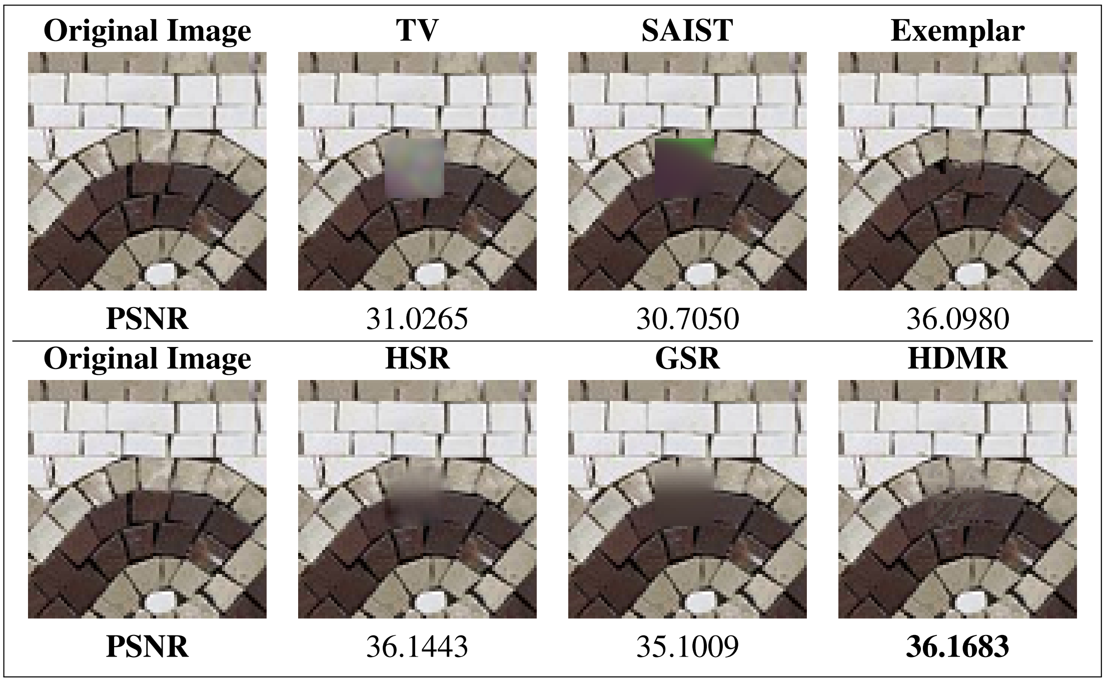


Figure 5.11: Visual inpainting and corresponding PSNR results for Test Image 10 in Figure 5.1(j).

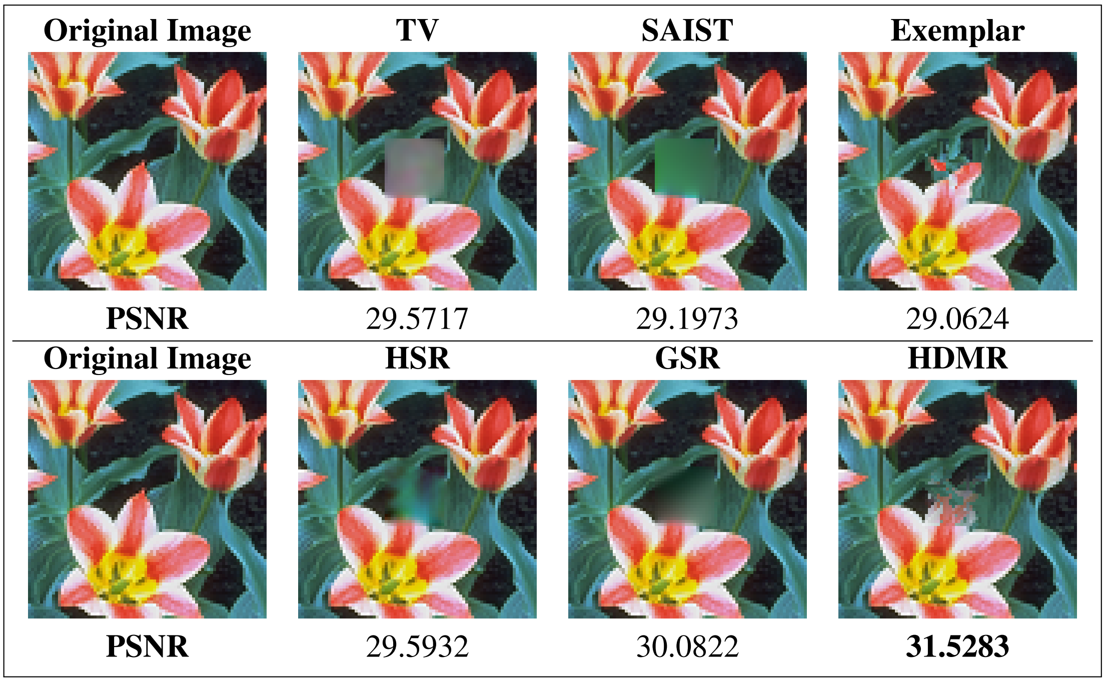
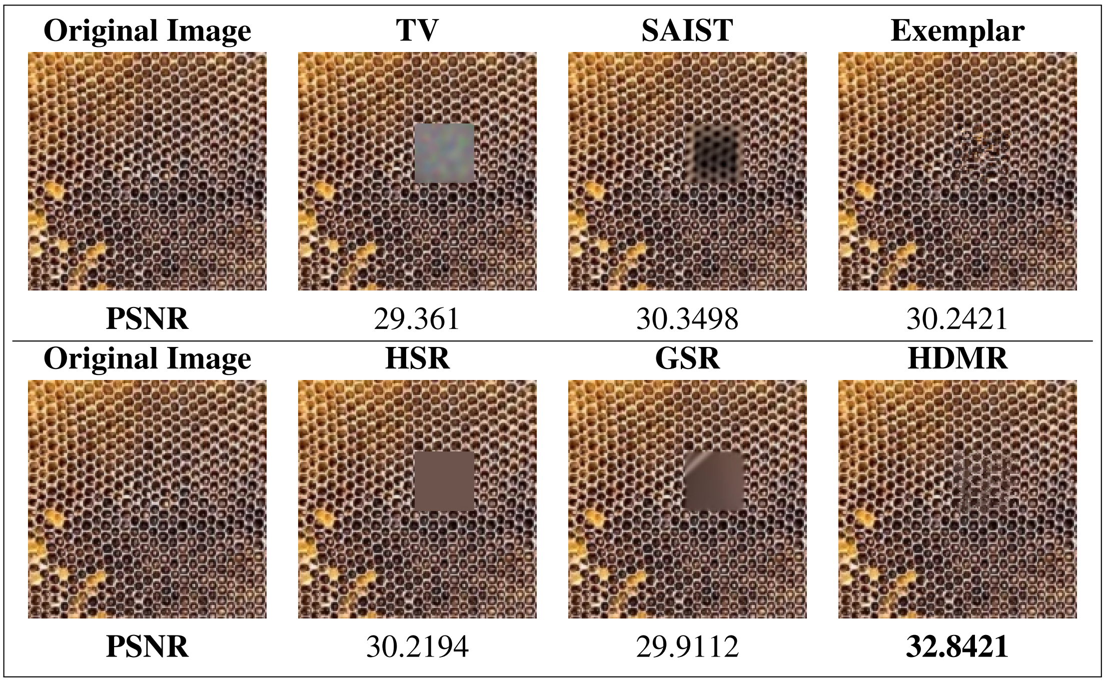
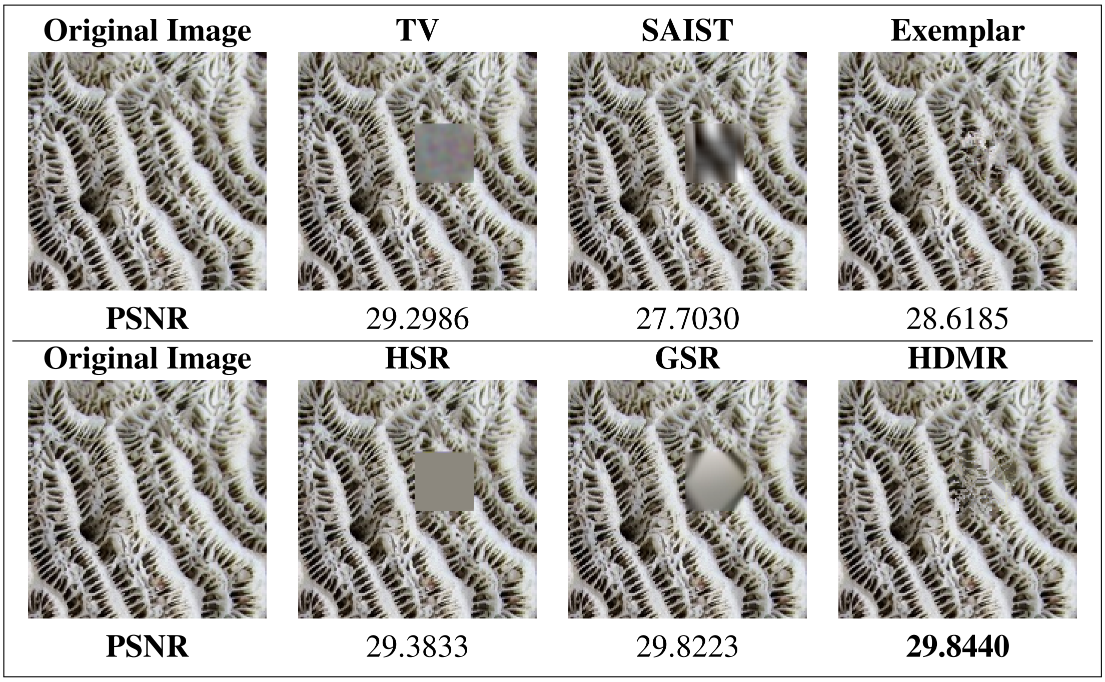


Figure 5.12: Visual inpainting and corresponding PSNR results for Test Image 11 in Figure 5.1(k).



Original Image	TV	SAIST	Exemplar
PSNR	29.361	30.3498	30.2421
Original Image	**HSR**	**GSR**	**HDMR**
PSNR	30.2194	29.9112	**32.8421**

Figure 5.13: Visual inpainting and corresponding PSNR results for Test Image 12 in Figure 5.1(l).



Original Image	TV	SAIST	Exemplar
PSNR	29.2986	27.7030	28.6185
Original Image	**HSR**	**GSR**	**HDMR**
PSNR	29.3833	29.8223	**29.8440**

Figure 5.14: Visual inpainting and corresponding PSNR results for Test Image 13 in Figure 5.1(m).

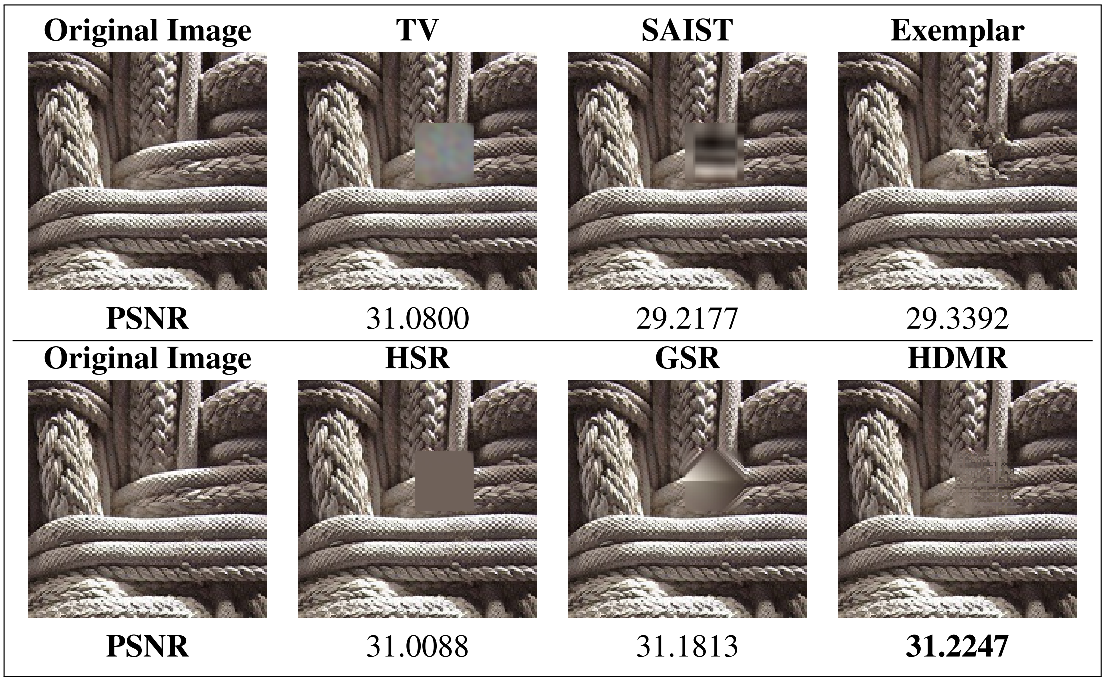


Figure 5.15: Visual inpainting and corresponding PSNR results for Test Image 14 in Figure 5.1(n).

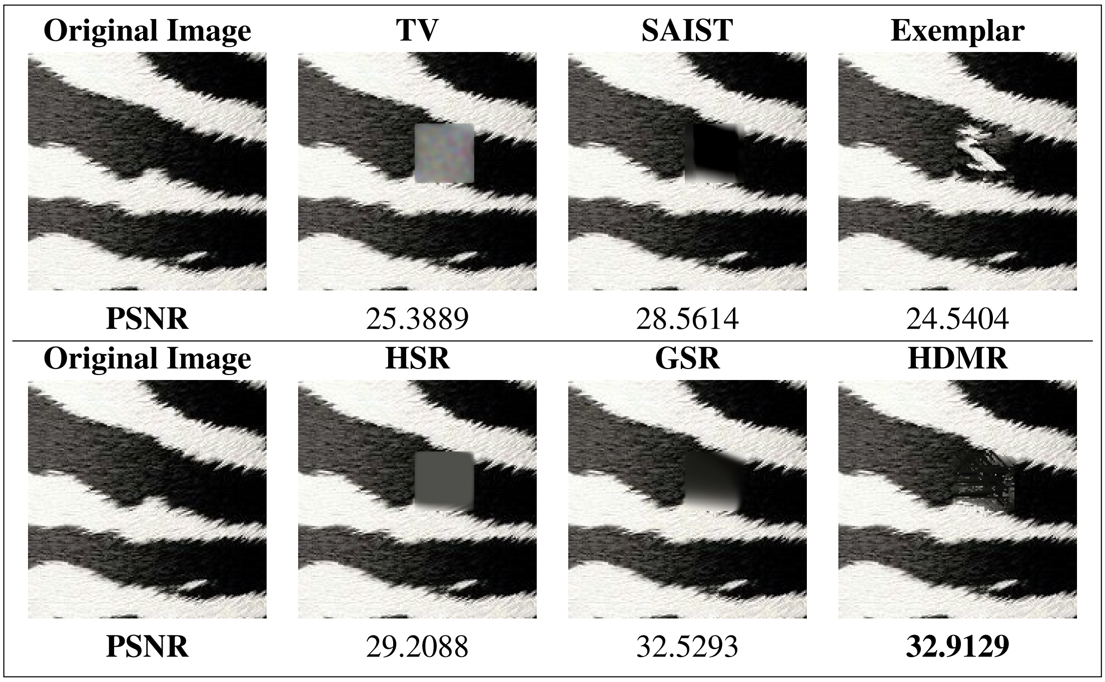


Figure 5.16: Visual inpainting and corresponding PSNR results for Test Image 15 in Figure 5.1(o).

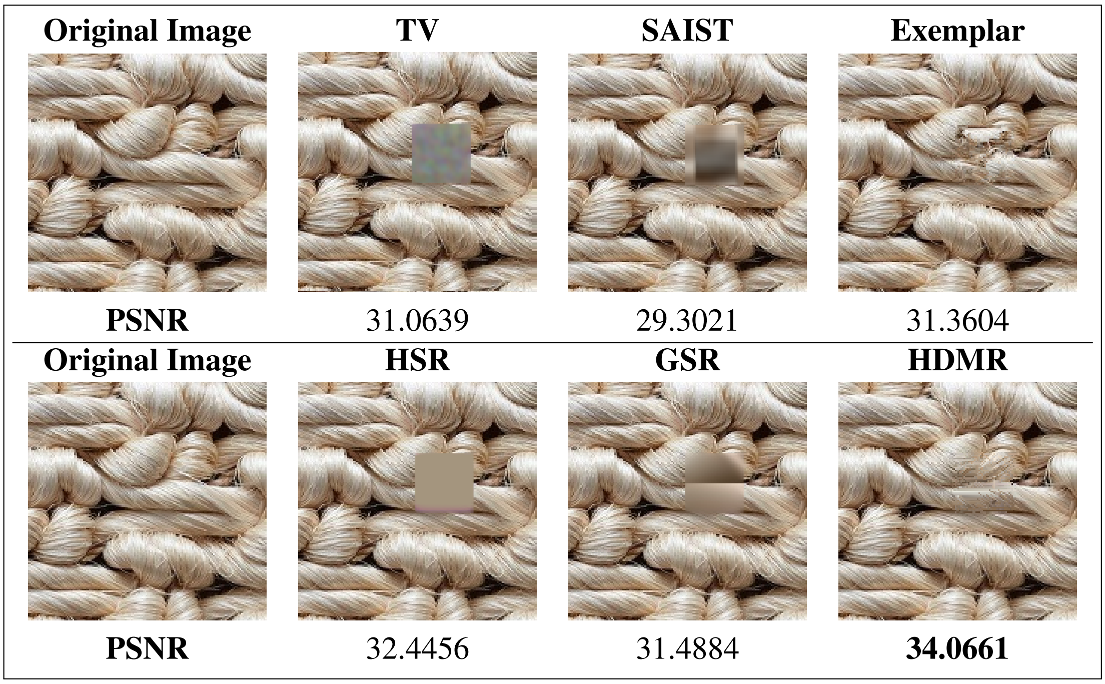


Figure 5.17: Visual inpainting and corresponding PSNR results for Test Image 16 in Figure 5.1(p).

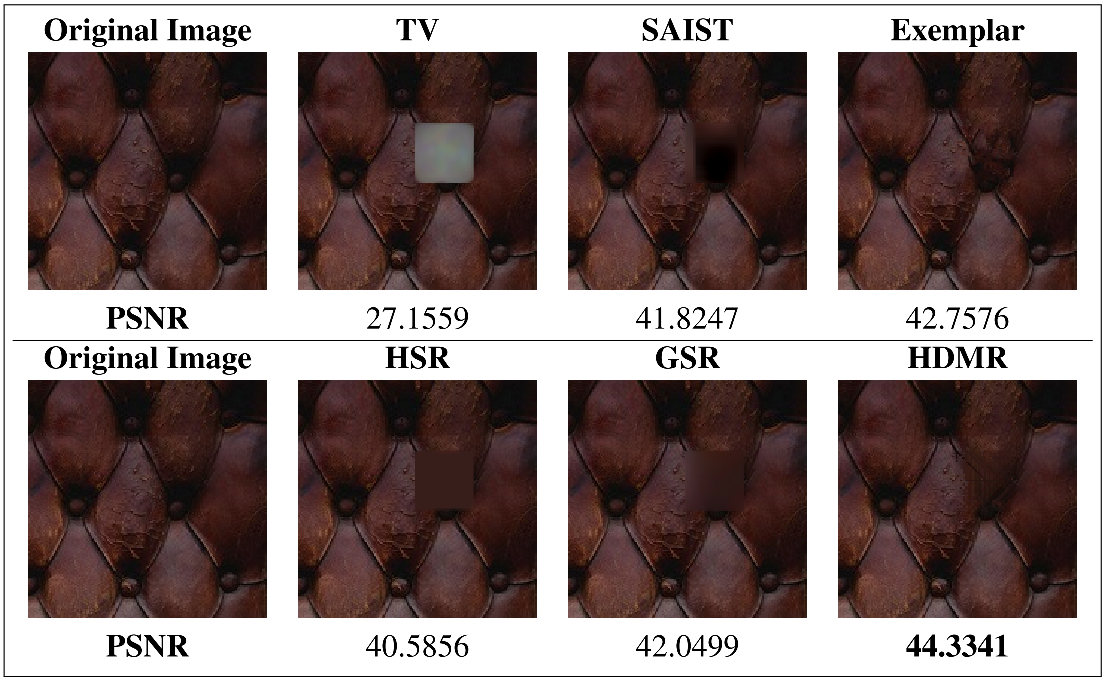


Figure 5.18: Visual inpainting and corresponding PSNR results for Test Image 17 in Figure 5.1(q).

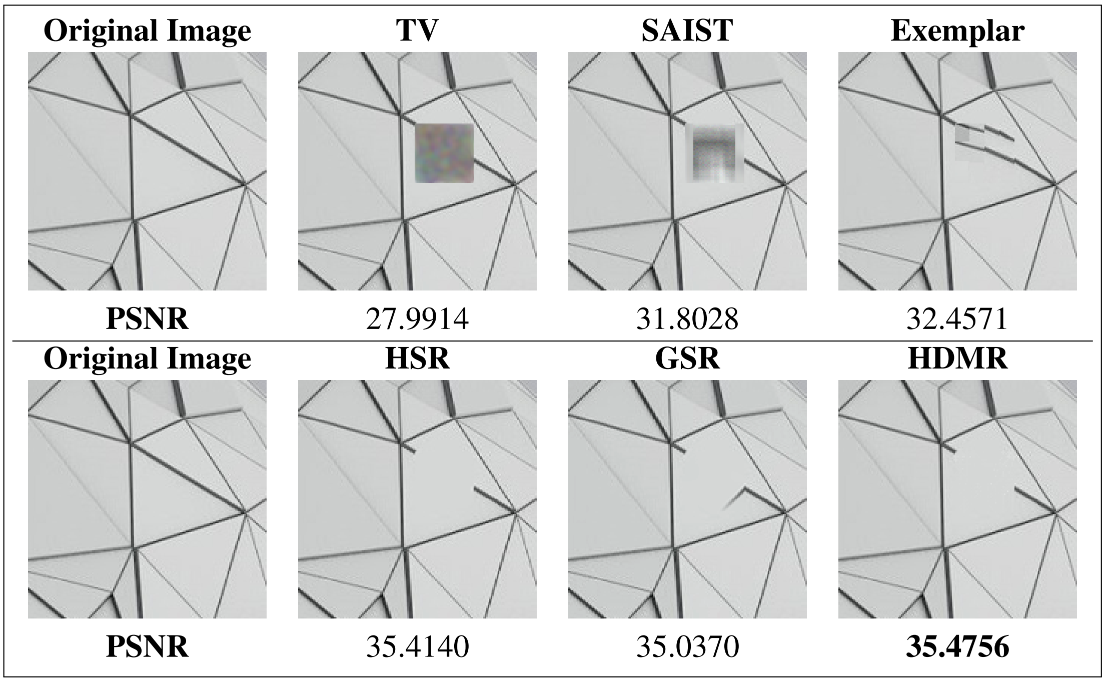


Figure 5.19: Visual inpainting and corresponding PSNR results for Test Image 18 in Figure 5.1(r).

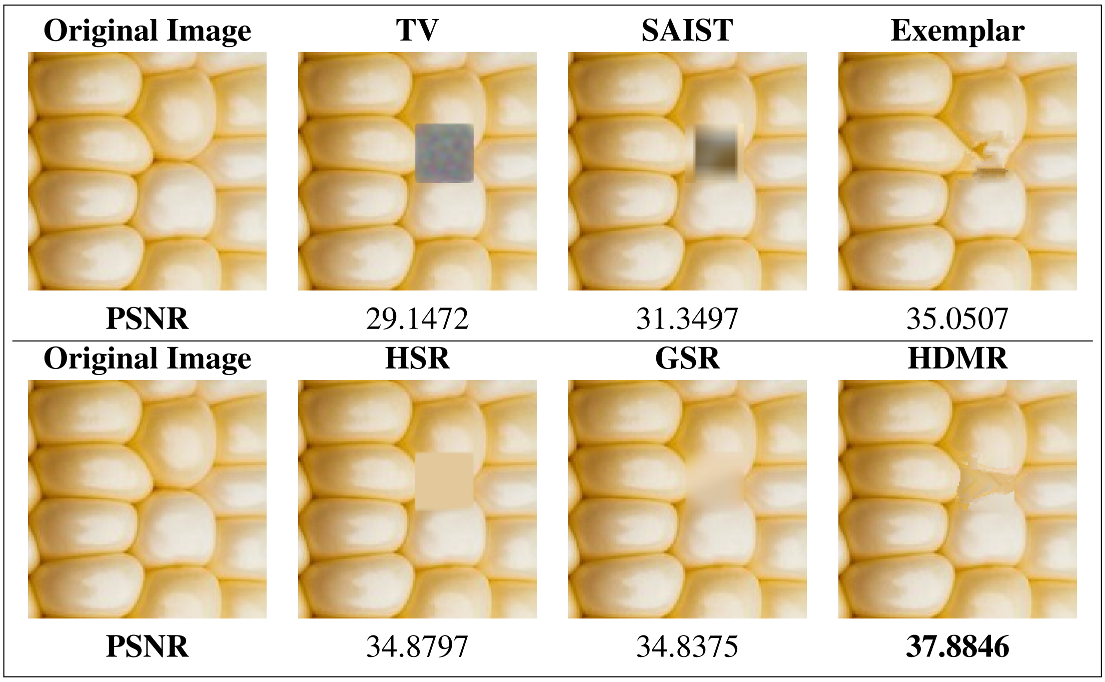


Figure 5.20: Visual inpainting and corresponding PSNR results for Test Image 19 in Figure 5.1(s).

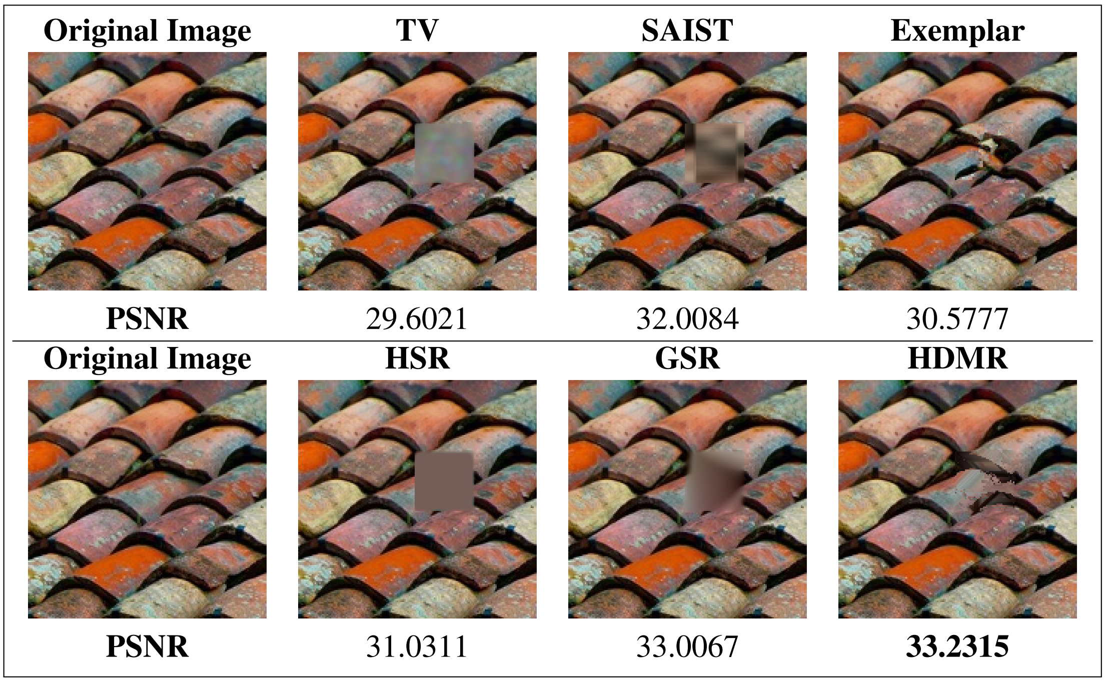
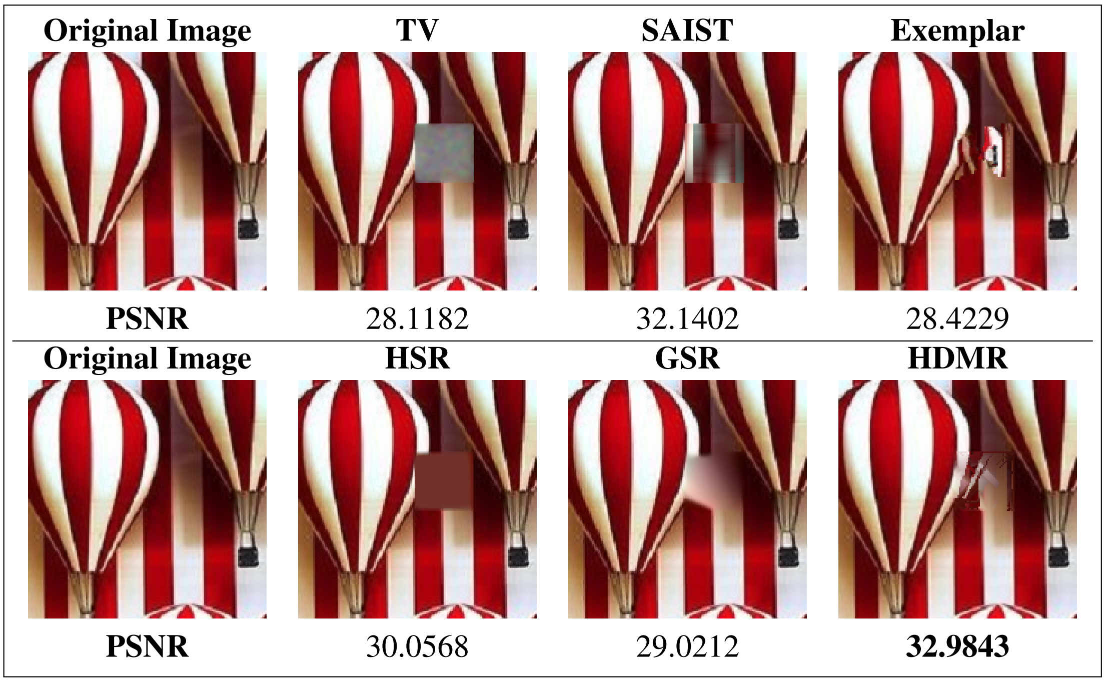


Figure 5.21: Visual inpainting and corresponding PSNR results for Test Image 20 in Figure 5.1(t).



6. CONCLUSION AND DISCUSSION

Image inpainting is the process of filling missing or fixing corrupted regions in a given image. The image inpainting process requires having a prior knowledge about the part to be completed and estimates the missing region accordingly. The apparent part of the image may give us information about the structure of whole image. Conventional interpolation techniques requires a lot of computation power when dimension of the data increases. This motivates us to apply HDMR and HDMR-based methods to reduce the computational complexity and CPU time that is needed to interpolate an image. HDMR is a divide-and-conquer algorithm to represent a multivariate function in terms of less-variate functions and it partitions a high dimensional data into a number of sets of lower dimensional data such as univariate, bivariate and trivariate ones. Reducing the complexity of the multivariate interpolation problem to univariate, bivariate and trivariate interpolations enables us to apply interpolation methods in a more efficient way both in terms of computational complexity of the problem and required CPU time. In this thesis, we propose a new inpainting approach using HDMR based methods.

We first propose a method which inpaints the rectangular missing regions in images using HDMR method and Lagrange interpolation. The results of this proposed method are compared with the results of TV-based inpainting method (Shen & Chan, 2002) which is a pioneering method in literature. The results demonstrate that our approach produces comparable results. This method works well if the missing region is part of a smooth background instead of a textural one. Also, 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. If the missing region is large, the assumed missing regions grow according to the given missing region as well. Thus, we propose a new procedure that decompose large missing regions into smaller ones and perform inpainting hierarchically starting from the smallest region. The results of the proposed method are compared with the results of TV-based (Shen & Chan, 2002) and exemplar-based (Criminisi et al., 2004) 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. But this method was also suffering from finding the underlying texture and pattern in the missing region.

In this thesis, we propose a Texture and Pattern Preserving HDMR (TPI-HDMR) method for inpainting missing regions in color images. To the best of our knowledge, an interpolation-based texture and pattern preserving inpainting algorithm have not been proposed in the literature. First, the proposed approach generates candidate inpainting results by interpolating to observed pixels at different angles of the missing region. Then, a texture and pattern preserving final inpainting decision is given for each pixel based on the information obtained using co-occurrence matrix and the normalized cross correlation. Experimental results demonstrate that the proposed algorithm achieves better inpainting results than the state-of-the-art. One possible future direction of the proposed method might be representing the problem with Bayesian formulation that involves a suitable likelihood term and a prior term for texture and pattern. Then, a solution that maximizes the resulting posterior density can be a good inpainting solution.

REFERENCES

Books

Bracewell, Ronald Newbold and Bracewell, Ronald N, 1986. *The Fourier transform and its applications*. vol. 31999. McGraw-Hill New York.

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

Periodicals

Alış, Ö. F. & Rabitz, H., 2001. Efficient implementation of high dimensional model representations. *Journal of Mathematical Chemistry*. **29(2)**, pp. 127–142.

Avci, E., 2007. An expert system based on wavelet neural network-adaptive norm entropy for scale invariant texture classification. *Expert Systems with Applications*. **32(3)**, pp. 919 – 926.

Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., & Verdera, J., 2001. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing*. **10(8)**, pp. 1200–1211.

Bertalmio, M., Vese, L., Sapiro, G., & Osher, S., 2003. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*. **12(8)**, pp. 882–889.

Chan, T. F. & Shen, J., 2001. Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*. **12(4)**, pp. 436–449.

Chan, T. F., Yip, A. M., & Park, F. E., 2005. Simultaneous total variation image inpainting and blind deconvolution. *International Journal of Imaging Systems and Technology*. **15(1)**, pp. 92–102.

Chhabra, J. K. & Birchha, M. V., 2014. Detailed survey on exemplar based image inpainting techniques. *International Journal of Computer Science and Information Technologies*. **5(5)**, pp. 6350–635.

Criminisi, A., Pérez, P., & Toyama, K., 2004. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*. **13(9)**, pp. 1200–1212.

Dong, W., Shi, G., & Li, X., 2013. Nonlocal image restoration with bilateral variance estimation: A low-rank approach. *IEEE transactions on image processing*. **22(2)**, pp. 700–711.

Guleryuz, O. G., 2006. Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part ii: adaptive algorithms. *IEEE Transactions on Image Processing*. **15(3)**, pp. 555–571.

Haralick, R. M., Shanmugam, K. et al., 1973. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*. **3(6)**, pp. 610–621.

Huynh-Thu, Q. & Ghanbari, M., 2008. Scope of validity of psnr in image/video quality assessment. *Electronics letters*. **44(13)**, pp. 800–801.

Karahoca, A. & Tunga, M. A., 2015. A polynomial based algorithm for detection of embolism. *Soft Computing*. **19(1)**, pp. 167–177.

Li, X., 2011. Image recovery via hybrid sparse representations: A deterministic annealing approach. *IEEE Journal of Selected Topics in Signal Processing*. **5(5)**, pp. 953–962.

Shen, J. & Chan, T. F., 2002. Mathematical models for local nontexture inpaintings. *SIAM Journal on Applied Mathematics*. **62(3)**, pp. 1019–1043.

Sobol, I., 1993. Sensitivity estimates for nonlinear mathematical models. *Math. Model. Comput. Exp. (MMCE)*. **1(1)**, pp. 407–414.

Takeda, H., Farsiu, S., & Milanfar, P., 2007. Kernel regression for image processing and reconstruction. *IEEE Transactions on image processing*. **16(2)**, pp. 349–366.

Tunga, M. A. & Demiralp, M., 2008. A new approach for data partitioning through high dimensional model representation. *International Journal of Computer Mathematics*. **85(12)**, pp. 1779–1792.

Yoo, S. & Park, R. H., 2009. Red-eye detection and correction using inpainting in digital photographs. *IEEE Transactions on Consumer Electronics*. **55(3)**, pp. 1006–1014.

Zhang, J., Zhao, D., & Gao, W., 2014. Group-based sparse representation for image restoration. *IEEE Transactions on Image Processing*. **23(8)**, pp. 3336–3351.

Zhang, X., Burger, M., Bresson, X., & Osher, S., 2010. Bregmanized nonlocal regularization for deconvolution and sparse reconstruction. *SIAM Journal on Imaging Sciences*. **3(3)**, pp. 253–276.

Zhou, M., Chen, H., Paisley, J., Ren, L., Li, L., Xing, Z., Dunson, D., Sapiro, G., & Carin, L., 2012. Nonparametric bayesian dictionary learning for analysis of noisy and incomplete images. *IEEE Transactions on Image Processing*. **21(1)**, pp. 130–144.

Other Publications

Altın, Emine Mine and Tunga, Burcu, 2014. High dimensional model representation in image processing. *10th International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE*, vol. 2014. pp. 55–64.

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

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

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

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

Karaca, E. & Tunga, M. A., 2016a. Interpolation-based image inpainting in color images using high dimensional model representation. *24th European Signal Processing Conference (EUSIPCO)*. IEEE, pp. 2425–2429.

Karaca, E. & Tunga, M. A., 2016b. A method for inpainting rectangular missing regions using high dimensional model representation and lagrange interpolation. *24th Signal Processing and Communication Application Conference (SIU)*. IEEE, pp. 1349–1352.

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

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