การเติมข้อมูลในหลายมิติที่ไม่สมบูรณ์โดยอาศัยเทคนิคโครงข่ายประสาทเทียม และการเปรียบเทียบความคล้ายของกลุ่มข้อมูล

นายสถิตย์ ประสมพันธ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรดุษฎีบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2554

IMPUTING INCOMPLETE MULTI-DIMENSIONAL DATA USING NEURAL NETWORK AND CLUSTERING SIMILARITY COMPARISON

Mr.Sathit Prasomphan

A Dissertation Submitted in Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy

Program in Computer Science

Department of Mathematics and Computer Science

Faculty of Science

Chulalongkorn University

Academic Year 2011

Thesis Title	IMPUTING INCOMPLETE MULTI-DIMENSIONAL DATA USING NEURAL NETWORK AND CLUSTERING SIMILARITY COMPARISON
By	Mr.Sathit Prasomphan
Field of Study	Computer Science
Thesis Advisor	Professor Chidchanok Lursinsap, Ph.D.
Thesis Co-advisor	Assistant Professor Sirapat Chiewchanwattana, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Doctoral Degree

..... Dean of the Faculty of Science
(Professor Supot Hannongbua, Dr.rer.nat.)

THESIS COMMITTEE

..... Chairman
(Associate Professor Peraphon Sophatsathit, Ph.D.)

..... Thesis Advisor
(Professor Chidchanok Lursinsap, Ph.D.)

..... Thesis Co-advisor
(Assistant Professor Sirapat Chiewchanwattana, Ph.D.)

..... Examiner
(Suphakant Phimoltares, Ph.D.)

..... External Examiner
(Khamron Sunat, Ph.D.)

..... External Examiner
(Chularat Tanprasert, Ph.D.)

สถิตย์ ประสมพันธ์ : การเติมข้อมูลในหลายมิติที่ไม่สมบูรณ์โดยอาศัยเทคนิคโครงข่ายประสาทเทียม และการเปรียบเทียบความคล้ายของกลุ่มข้อมูล. (IMPUTING INCOMPLETE MULTI-DIMENSIONAL DATA USING NEURAL NETWORK AND CLUSTERING SIMILARITY COMPARISON) อ. ที่ปรึกษาวิทยานิพนธ์หลัก : ศ. ดร. ชิด ชนก เหลือสินทรัพย์, อ. ที่ปรึกษาวิทยานิพนธ์ร่วม ผศ. ดร.สิรภัทร เชี่ยวชาญวัฒนา, 118 หน้า.

วิทยานิพนธ์ฉบับนี้นำเสนอวิธีการเติมข้อมูลที่สูญหายในข้อมูลหลายมิติ โดยแบ่งลักษณะของข้อมูลที่ใช้ออกเป็นสองกลุ่มคือ กลุ่มที่หนึ่งเป็นการเติมข้อมูลอนุกรมเวลาที่ไม่สมบูรณ์ โดยอาศัยข้อมูลเกรเดียนท์ของข้อมูลรอบข้างของบริเวณที่หายไป แนวคิดหลักของวิธีนี้คือ ข้อมูลที่หายไปจะมีเกรเดียนท์อยู่ในบริเวณเกรเดียนท์หนึ่งในสามประเภทต่อไปนี้คือ เกรเดียนท์ที่เป็นบวก เกรเดียนท์ที่เป็นลบ และเกรเดียนท์ที่เป็นศูนย์ เมื่อได้ประเภทของข้อมูลที่สูญหายแล้วจะใช้วิธีการสุ่มแบบบูตสแทรปสำหรับการเติมข้อมูล ส่วนกลุ่มที่สองคือ การเติมข้อมูลในหลายมิติที่ไม่สมบูรณ์โดยการทดลองกับข้อมูลรูปภาพ โดยอาศัยลักษณะของการสูญหายของข้อมูลมาใช้ในการเติมข้อมูล โดยที่ กรณีที่ข้อมูลมีการสูญหายในลักษณะสุ่มและมีกระจายตัวแบบสม่ำเสอ วิธีการแก้ปัญหาคือ การใช้แบบจำลองโครงข่ายประสาทเทียม โดยใช้เฉพาะข้อมูลรอบข้างของบริเวณที่สูญหายภายใต้รัศมีที่กำหนดเพื่อสร้างพื้นผิวสำหรับบริเวณที่สูญหาย กรณีที่ข้อมูลที่สูญหายอยู่ในลักษณะรูปร่างแบบต่าง ๆ วิธีการแก้ปัญหาคือ การแบ่งพื้นที่ที่สูญหายเป็นหน้าต่าง หลังจากนั้นจะนำบริเวณดังกล่าวไปเปรียบเทียบกับทุก ๆ บริเวณของรูปภาพเพื่อหาบริเวณที่มีความคล้ายกับบริเวณที่สูญหายมากที่สุด จากผลการทดลองสามารถสรุปได้ว่า เมื่อเติมข้อมูลโดยวิธีที่นำเสนอกับรูปภาพที่มีรูปแบบของการสูญหายแบบต่าง ๆ จะสามารถเพิ่มความถูกต้องของการเติมข้อมูลเมื่อเปรียบเทียบกับวิธีอื่น ๆ

ภาควิชา คณิตศาสตร์และวิทยาการคอมพิวเตอร์ ลายมือชื่อนิสิต.....

สาขาวิชา วิทยาการคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก.....

ปีการศึกษา 2554 ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ร่วม.....

5073884923: MAJOR COMPUTER SCIENCE

KEYWORDS : IMPUTATION TECHNIQUE / NEURAL NETWORK / CLUSTERING SIMILARITY COMPARISON / MULTI-DIMENSIONAL DATA / TIME-SERIES DATA

SATHIT PRASOMPHAN : IMPUTING INCOMPLETE MULTI-DIMENSIONAL DATA USING NEURAL NETWORK AND CLUSTERING SIMILARITY COMPARISON. ADVISOR: PROF. CHIDCHANOK LURSINSAP, PH.D., CO-ADVISOR : ASST. PROF. SIRAPAT CHIEWCHANWATTANA, PH.D., 118 pp.

This dissertation presented a method to fill in missing data in multi-dimensional data. These data are divided into two categories. The first one is incomplete time series data. The algorithm for imputing the missing time-series data is based on the gradient of the area surrounding the missing data. The missing information which is the gradient of a data falls in one of the following three categories: positive gradient, negative gradient, and zero gradient. When a group of missing data belongs to one of three categories, the missing data are imputed with bootstrapping method. The second type is filling in the incomplete multi-dimensional data in an image. To impute the missing image, the characteristics of missing image are used. If missing data are randomly and fine scattered, an artificial neural network model is used to create an approximated surface to cover those missing data. But if the missing data are clustered in forms of an empty shape, then a similarity pattern searching and filling is performed. The missing data areas are divided into a set of equal size of windows. This windowed area will be compared with every other non-missing data area of the image area to find the most similar area with the missing area. The experimental results concluded that our proposed algorithms are outperformed the other tradition methods in several cases.

Department : Mathematics and Computer Science Student's Signature .....

Field of Study : Computer Science Advisor's Signature .....

Academic Year : 2011 Co-advisor's Signature .....

Acknowledgements

First and foremost I offer my sincerest gratitude to my advisor, Professor Dr. Chidchanok Lursinsap, for kindly providing guidance throughout the research. His comment is very helpful to me to overcome the necessary difficult problems and also advice me to the discipline to be a good reseacher. I would like to show my gratitude to my co-advisor, Assistance Prof.Dr. Sirapat Chiewchanwattana, who give a good consults in time series theories. It is an honor for me to say thank you to Prof.Dr.Shigeru Mase, who give a good suggestion in the field of Geostatistics and providing the facilities during research at Mase's laboratories, Tokyo Institute of Technology, Japan. Also, I would like to thank all of the dissertation committee for their advices and guidances.

I would like to thank the Thai Government who grants the research scholarship. I would like to express my gratitude to King Mongkut's University of Technology North Bangkok for their support during my education.

I would like to thank to the Advanced Virtual and Intelligent Computing (AVIC) Center for providing matherials during my research. I would also thank you to all my colleages at the Advanced Virtual and Intelligent Computing (AVIC) Center who give me a suggestion of my research. I am greatful to the Mase's Laboratory, Tokyo Institute of Technology, Japan. for providing facilities and material support during conducting research in Japan. I would also thank to all my colleage at Mase's Laboratory espicially Mr.Shinsuke Mita who was very kindly to me and give the good suggestion to live in Japan.

I would like to express my gratitude to my parents who give me love and support during my education. Also I would like to thank all of my sisters and brothers who encorage me to continue my research when I fell give up.

Finally, I would like to thank all whose direct and indirect support helped me completing my dissertation.

Contents

Chapter Page

Chapter Page

List of Tables

List of Figures

Figure Page

Figure Page

Figure Page

CHAPTER I

INTRODUCTION

1.1 Statement of Problems

Imputing incomplete data is one of the most important problems in many fields such as data mining, medical, statistics, image processing, bio-informatics, data modeling, time series [1][2][3][4]. These problems become an important issue if the missing values are required for data analysis. Previously, many techniques were proposed for imputing incomplete missing data. Several data set may have a difference characteristic of the missing data such as the missing data in time series data is different from the missing data in image data set. From this reason, the imputation algorithms should be carefully selected depends on its characteristics of each data set. Accordingly, in this dissertation, the incomplete data set was divided into two types: time series data set and image data set. Since there are different algorithms for each data set, however, there are some common similar algorithms between two groups of data set were used. In this research, the imputation techniques for incomplete multi-dimensional data were focused. The proposed ideas in this dissertation are based on the semi-supervised learning by using feedforward neural network to train data in each group to extract the relationship between inputs and output of the unknown data. Moreover, the similarity measurements between two groups of data were used for giving more accurate in the imputed values.

1.2 Objective

The main objectives of this dissertation are the following:

1. To propose a new fill-in technique for incomplete multi-dimensional data in n-dimensional spaces, where n is number of dimensions.
2. To achieve the accuracy of incomplete multi-dimensional data at least 70% under 70 % missing rate for n=2.
3. To achieve the accuracy of incomplete multi-dimensional data at least 70% under 50 % missing rate for n>2.

1.3 Scope of Work

In this dissertation, the scope of work is constrained as follows:

1. This research focus on multi-dimensional data.
2. The mechanism of missing data is MCAR(Missing Completely At Random).
3. Assume that each output data x_p are formulated by an unknown function of input data x_i for $1 <= i <= p-1$, as $x_p = f(x_1, x_2, \ldots, x_{p-1})$, where p is dimension of data set.
4. The characteristics of multi-dimensional data used to this research having unknown distribution.
5. All data set are numerical data.
6. The complex domain data set such as image dataset is considered.

1.4 Dissertation advantages

1. A new fill-in technique for incomplete multi-dimensional data in n-dimensional spaces is proposed.
2. This technique can be applied for real world application which has missing data such as missing data in timeseries dataset and image dataset.

1.5 Dissertation contributions

The significant contributions of the missing values imputation discussed in this dissertation are:

1. New time series imputation based on the regional-gradient guided bootstrapping algorithms which used the gradient of time series data to consider when performs the imputation processes.
2. New image imputation based on the similarity comparison of RGB values of an image inside the window area between the missing area and non-missing area.
3. Applicability of the proposed algorithms to image inpainting problem and image restoration problem.

4. A feasible similarity measurement concept between two groups of data.

5. Ability to preserve the shape of image after imputation.

1.6 Outline of the thesis

The remaining of this research is organized as follows. In chapter 2, the literature reviews of time-series imputation, image imputation, and clustering similarity measurement are proposed. In chapter 3, the proposed method for imputing incomplete time series data by using regional-gradient guided bootstrapping algorithms are presented. In chapter 4, the missing data imputation based on the hybrid of adjustable neural networks and similarity measurement between two clusters are introduced. The imputing incomplete SLC-off imagery based on neural networks and similarity measurement between two clusters is proposed in chapter 5. The experimental results are described in chapter 6. Finally, the conclusions are presented in chapter 7.

CHAPTER II

REVIEWS OF IMPUTATION ALGORITHMS

2.1 Time-series imputation algorithms

Imputing incomplete data is one of the most important problems in many fields such as data mining, medical, statistics, image processing, bio-informatics, data modeling, time series, etc [1][2][3][4]. These problems become an important issue if the missing values are required for data analysis. Previously, many techniques are proposed for imputing incomplete missing data. In [2][3], the procedure for imputing missing data was classified into three categories: ignore base procedure, parameter estimation, and imputation procedure. The method of *Ignore the missing data* is a well known method for imputing incomplete data but it has a serious consideration when being applied to the missing value. In this method, the error evaluations are difficult to perform.

Different algorithms were proposed for solving the problem of ignoring the missing data. The most well known algorithms is *parameter estimation procedure* such as maximum likelihood algorithms [5] and the expectation and maximization algorithms (EM) [6]. EM algorithm provides a good framework for imputing missing data but the limitation of this algorithm is similar to other methods in terms of parameter estimation. They must know the prior distribution of data which will affect the performance and efficiency of imputation. Moreover, the time complexity of EM algorithm or other algorithms for parameter estimation can be high if there are many missing values. The method of imputation approach replaces the missing data by substituting the suitable values. The examples of this method in this category is case substitution, mean, or mode imputation, hot deck and cold deck imputation, prediction model [6][7][8], K-Nearest Neighborhood method (KNN)[4], and Varies Windows Similarity Measure(VWSM) algorithm [1] etc. The concept of hot deck imputation used the value of observed data called donor to impute a record have missing value called recipient. The advantage of this method is the occurring values are used for imputation and they are not used the distribution assumption [6]. In K-Nearest Neighbor method, the missing data are imputed by the observed data with minimum distance from the missing value. The performance of this method depends on the suitable distance measure[4]. The concept of MI interpolation is to impute the missing data by using a suitable value and repeating this procedure M times [5][3][6]. For VWSM algorithm, they used the assumption of the similarity characteristic of the cyclical of the data set [1].

The procedure finds the cycles which are similar to the cycle having missing value and, then, imputes the missing sample from the complete subsequence. From these methods, VWSM outperforms than other methods. However, this approach has some consideration before being used to impute any missing data because it is time consuming. So, in this dissertation, this problem will be solved by using concept of the slopes of nearest neighboring data and re-sampling the estimated data with replacement based on bootstrapping concept. It is not necessary to find the similar subsequence for imputing the missing data but used only the value nearest to missing value to be imputed value. The characteristics summary of each method[9] is presented in Tables 2.1- 2.2.

2.2 Image imputation algorithms

Due to malfunctions during the acquisition process, the problem of incomplete data has dramatically increased in many fields of data set acquisition, for example, image data sets, geostatistics data sets, and time-series data sets. The application of imputation methods for reconstructing damaged areas have been extensively studied. Most of the conventional methods use the nearest pixels for restoring the damaged areas. In recent works, several different imputation methods have been proposed. Takahiro et al.[13] used an algorithm based on a kernel principal component analysis(PCA)-based projection onto a convex set. In this algorithm, a non-linear eigenspace was constructed from each kind of texture and the optimal subspace for the target local texture was introduced into the constraints of the POCS algorithms. A limitation of this algorithm is that the size of the local image and the number of clusters are set manually. It is desirable ifhese values can be adaptively determined from the observed image. Amano and Sato[14] proposed a method for image imputation for the case of the where the missing area in an image is surrounded by characters by using the eigenspace. They used the BPLP method based on the self-correlation in the image using only one image. However, parameters such as the size of the clipping window and the step width of window were experimentally determined. These parameters may affect the precision of interpolation. Grover et al.[15] introduced a technique for filling in a missing area by using texture filling based on a Laplace equation with Dirichlet boundary condition. The main asset of this method is that it uses a bounded search window. The method is able to fill in a missing area from outside to inside. Hui et al.[16] proposed a regularization based approach to recover degraded images by enforcing the analysis-based sparity prior of images in tight frame domain. Telea[17] proposed an image inpainting algorithm based on the fast marching

Table 2.1: Characteristics summary of the imputation method which used machine learning techniques.

Method	Method summary	Advantage	Disadvantage	Data set	Complexity
ISOM-DH[10]	Use ICA and SOM.	Classify data into similar group.	Time consuming. Must known data set distribution.	The data sets of main economic indicators for major retail enterprises from Beijing Statistic Annual	N/A
TS-SOM[11]	Create an imputation model by probabilistic distribution.	Use the similar cluster and regression.	Time consuming. Use several SOMs to create a tree structure.	Danish Labour Force Survey Data set	$O(Nlog(N))$
SOM-FCM[11]	Incomplete data are translated into fuzzy data. After that generate fuzzy maps	Provide more information	Relative fuzzy cluster might not be significant.	Students course evaluation at a Canadian university.	N/A
FCM[12]	Use the fuzzy C-means clustering of incomplete data	Provide more information	Time consuming	Artificial Two-Dimensional (2-D) Clusters, IRIS Data.	$O(N)$
VWSM[1]	Find the cycles which are similar to the cycle having missing value and then imputes the missing sample from the complete subsequence.	More accuracy	Time consuming. Only 2-dimensions.	MackeyGlass chaotic time-series data, the monthly sunspots data, the daily gauge height at Ban Luang gauging station, and the daily air temperature at Nakhon Ratchasima province, Thailand	$O(N^2)$
RGGB[2]	Use neural network and similarity measure	Use observed data to impute. Not concern the distribution of data. Use local training. Use nearest data to impute incomplete data.	Time consuming.	MackeyGlass chaotic time-series data, the monthly sunspots data, the daily gauge height at Ban Luang gauging station	$O(N)$

Table 2.2: Characteristics summary of imputation method which used statistical techniques.

Method	Method summary	Advantage	Disadvantage	Data set	Complexity
Variance	Use a variance of data set.	Give true distribution and variance of data set.	Must know the prior distribution.	breast cancer	$O(N)$
EM	Use prior distribution to test the missing data and impute until maximum likelihood rich.	Give true distribution and variance of data set.	Must know the prior distribution estimation. Time Consuming.	breast cancer	$O(N)$
Hotdeck	Use the value of observed data called donor to impute a record having missing value called recipient.	Use observed data to impute	Less accuracy. Not used distribution assumption.	breast cancer	$O(N)$
KNN	Use the observed data with minimum distance from the missing value	Use observed data to impute.	Depends on distance measure.	breast cancer	$O(N^2)$
MI	Use a suitable value and repeatly impute M times	Use observed data	Require parameter estimation	breast cancer	$O(N^2)$

method[17]. The limitation of this algorithm is the blurring produced when inpainting points are thicker than 10-15 pixels[17]. Huan et al.[18] introduced an image inpainting algorithm based on a two-step process. In the first step, the filling order of the missing pixels was determined by using the fast marching method. In the second step, a block of textures was computed by using a search process and an SSD measurement[17]. Criminisi et al.[19] developed an exemplar-based inpainting method which used the magnitude of the gradient and the observed pixels of image to define the filling order in the target region. The missing region of the image was filled with source patch blocks ordered by priorities. This method is an efficient imputation method which is able to preserve the linear structure in the missing area. However, the filling order in the method is random and unreliable and it seems to have the phenomenon of growing garbage[18]. Bertalmio et al.[20] proposed an image inpainting method based on smooth propagation of information from the surrounding areas in the isophotes direction. One of the main problems with their algorithm occurs with the reproduction of large textured regions.

One of the main problems of the methods discussed above are that the algorithms for the imputation of missing pixels requires a suitable selection of the nearest pixels. Other problems occur if the missing image has a large size or if there are randomly missing pixels. In this paper, an imputation technique is developed which focusses on characteristics (e.g., the shape) of the damaged areas inside a picture. If the damaged area is small and surronded by known pixels, then a neural network can be developed which uses only nearest-neighbor pixels for training and imputing. The number of sample nodes used for training is automatically adjusted depending on the data available in the neighborhood. Therefore, the number of sample nodes used in training are not necessarily equal for each missing pixel. The number of data points used during the training process depends on characteristics of the missing window and its neighhorhood.

If the missing area is large, then we have found that the neural network method cannot be used for imputation. Instead, we impute the pixels in the missing area by finding a similar area in the original image by using a clustering method.

2.3 Clustering similarity measurement

The next issue which concerned in the proposed method is the clustering similarity measure for comparing the target clusters having missing values. Clustering is a method for classifying a data set into groups of similar data sets. The process for clustering data set is based on clustering similarity measure. There are several clustering similarity measure

between two clusters. Torres[21] proposed a similarity measure by using the Euclidean distance between cluster centroids and pearson correlation. Bae[22] proposed the new measure name Attribute Distribution Clustering Orthogonality(ADCO) which consider the density profile for each attribute. This method considered distribution information of data points in each attribute, and the shape of each cluster. Dong et,.al[23] proposed new similarity measure by using a cosine similarity-based negative selection algorithms for time series detection. All of these methods were used for similarity measure between two clusters. In image processing, the clustering similarity measure which is well-known for comparing is Ward-Walfowitz test of randomness. The concept of this technique uses a run test calculated from a consecutive sequence of identical label between two clusters. The number of runs is used as the test statistics[24].

CHAPTER III

IMPUTATION OF INCOMPLETE TIME-SERIES DATA

3.1 Problem Formulation

3.1.1 Time series data

A time series data considered in this study concerns a sequence of data whose values can be written as a function of times. These data can be plotted in terms of time sequences as shown by the example in Fig. 3.1. The vertical axis is the value of each data while the horizontal axis is the time. Let T be a set of time series data denoted by $T = (x_1, x_2, \ldots, x_n)$. Each x_t is the value of data at time t. There are three types of gradient in any time series data. The first type is the gradient having positive value. The second type is the gradient having negative value. The last type of gradient is the gradient having zero value. Obviously, a missing data must belong to one of these three types of gradient and it may lay in between the same or different types of gradient. For example, if a missing data actually exists in the first type of gradient, then it must lay in between two neighboring gradients with positive values. The proposed imputation techniques will be based on these observations.

3.1.2 Bootstrap method

Bootstrap is a statistical algorithm proposed by Efron[25]. The main objective is to estimate the natural mean and variance of collected data by iteratively re-sampling the collected data. This technique can be applied to any dimension [25]. Bootstrap method composed of two types: non-parametric bootstrap and parametric bootstrap[25]. In non-parametric bootstrap, data are sampled with unknown distribution of parameter estimator but in parametric bootstrap, the distribution of parameter estimator must be known in advance [25]. In this dissertation, the concept of bootstrap re-sampling was applied for testing the variance of data set when already imputed with the nearest neighbors of missing data to get the confidence interval of the imputed missing value.

3.2 Proposed Imputation

The following four possible cases were considered for any missing data. In each case, a set of bootstrapped data are generated and used to impute the missing data.

1. A missing data is in between two positive gradients. Area B in Fig. 3.1 shows an example of this case. The missing data are on the dotted line.
2. A missing data is in between two negative gradients. Area A in Fig. 3.1 shows an example of this case. The missing data are on the dotted line.
3. A missing data is in between a positive gradient and a negative gradient. Fig. 3.3 shows an example of this case. The missing data are on the dotted line.
4. A missing data is in between a negative gradient and a positive gradient. Fig. 3.2 shows an example of this case. The missing data are on the dotted line.

Since the considered data are in a sequence of time, the whole data sequence can be partitioned into three consecutive groups. The first group is the sequence of data to the left of missing data sequence. The second group is the missing data sequence laying next to the first sequence. The last group is the data sequence to the right of the second sequence. With of loss of generality, assume that groups 1 and 3 have no missing data. Only group 2 has missing data. To define the boundary of missing region, four locations of time steps are defined as follows. Let

t_b: the rightmost time location in the first group adjacent to the considered missing data in the second group.

t_{b-1}: the one-unit time location to the left of t_b.

t_f: the leftmost time location in the third group adjacent to the considered missing data in the second group.

t_{f+1}: the one-unit time location to the right of t_f.

t_m: the time location of missing data.

Fig. 3.1, 3.2, and 3.3 denote the pictorial meanings of these time locations in each case. The number of missing data may be more than one. The proposed imputing procedure is summarized in the following **ALGORITHM 1**. Let $\mathbf{X}$ be a set of time locations with non-missing and missing data and $|\mathbf{X}| = n$. For each missing data, the procedure

starts by identifying one of the four cases of missing data. Then, the missing data is imputed by using the proposed bootstrapping concept.

ALGORITHM 1: Regional-Gradient-Guided Imputation

1. Count number of missing data, calculate percentage
   of the missing data, and normalize each non-missing data
   x_t in range [0-1] by this equation,
   $x_t = (x_t - \min(\mathbf{X}))/(\max(\mathbf{X}) - \min(\mathbf{X}))$.
2. **for** $t = 1$ **to** n **do**
3. **If** a missing data x_t exists **then**
4. **begin**
5. Find non-missing data $x_{t_{b-1}}$, x_{t_b}, x_{t_f}, $x_{t_{f+1}}$.
6. Calculate boundary regions in terms of
   $\tan\alpha_1 = \frac{x_{t_b} - x_{t_{b-1}}}{t_b - t_{b-1}}$ and
   $\tan\alpha_2 = \frac{x_{t_{f+1}} - x_{t_f}}{t_{f+1} - t_f}$.
7. **If** $\tan\alpha_1 \geq 0$ **and** $\tan\alpha_2 \geq 0$ **or**
   $\tan\alpha_1 \leq 0$ **and** $\tan\alpha_2 \leq 0$ **then**
8. Impute the missing data by **ALGORITHM 2**.
9. **If** $\tan\alpha_1 > 0$ **and** $\tan\alpha_2 < 0$ **or**
   $\tan\alpha_1 < 0$ **and** $\tan\alpha_2 > 0$ **then**
10. Impute the missing data by **ALGORITHM 3**.
11. **end**
12. **end**

Based on the location of a missing data with respect to the neighboring non-missing data regions, two imputation separated procedures, i.e. **ALGORITHM 2** and **ALGORITHM 3** are introduced. **ALGORITHM 2** covers cases 1 and 2 while **ALGORITHM 3** covers cases 3 and 4. Note that cases 1 and 2 are the inverse of one another. Therefore, the imputation procedure for each case can be combined as illustrated in **ALGORITHM 2**. Similarly, cases 3 and 4 can also be combined as provided in **ALGORITHM 3**. The detail of each algorithm is the following.

ALGORITHM 2: Cases 1 and 2 Imputation

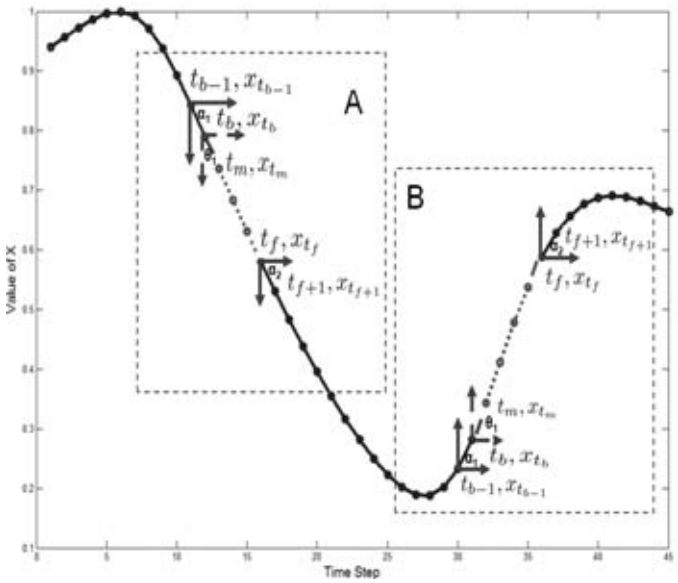


Figure 3.1: An example of missing data occurring at the positive gradient region (area B) and the negative gradient region (area A). The dotted lines denote the missing data locations and the thick lines denotes the non-missing data locations.

1. Let $x_{t_{b-1}}$, x_{t_b}, x_{t_f}, and $x_{t_{f+1}}$ be the values of non-missing data at t_{b-1}, t_b, t_f, and t_{f+1}, respectively.
2. Let $x_{t_m}, x_{t_{m+1}}, \ldots, x_{t_{mc}}, \ldots, x_{t_{f-1}}$ be the missing data at $t_m, t_{m+1}, \ldots, t_{mc}, \ldots, t_{f-1}$.
3. Calculate $tan\theta_1$ between x_{t_b} and x_{t_f} by $\tan\theta_1 = \frac{x_{t_f} - x_{t_b}}{t_f - t_b}$
4. **for** $j = t_m$ **to** t_{f-1} **do**
5. $x_j = x_{t_b} + (j - t_b) * \tan\theta_1$
6. Let $D = \{d_1, d_2, \ldots, d_s\}$ be a set of randomly sampled values in between x_j and x_{t_b}.
7. Set x_j to b_k from step 7 having highest occurrence frequency among all elements in D.
8. Move the non-missing location forward by setting $t_b = j$.
9. **end**

ALGORITHM 3: Cases 3 and 4 Imputation

1. Let $x_{t_{b-1}}$, x_{t_b}, x_{t_f}, $x_{t_{f+1}}$ be the values of non-missing

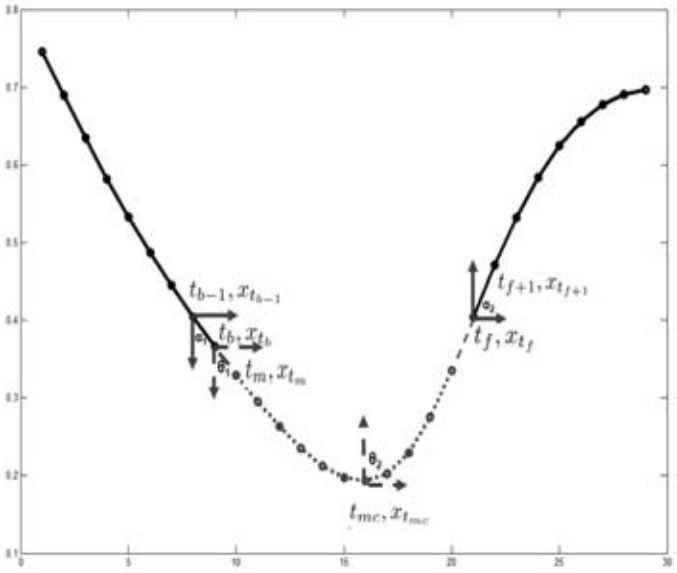


Figure 3.2: An example of missing data with $\tan\alpha_1$ and $\tan\alpha_2$ of negative gradient and positive gradient, respectively. The missing data are on the dotted line.

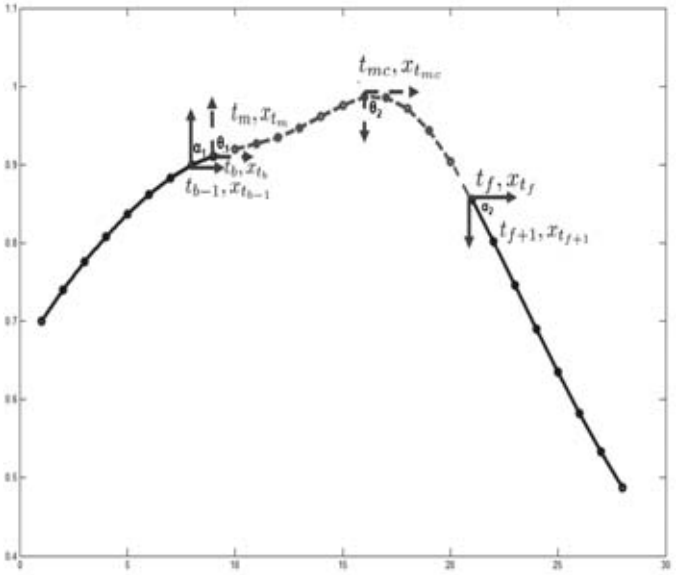


Figure 3.3: An example of missing data with $\tan\alpha_1$ and $\tan\alpha_2$ of positive gradient and negative gradient, respectively. The missing data are on the dotted line.

data at t_{b-1}, t_b, t_f, and t_{f+1}, respectively.

2. Let $x_{t_m}, x_{t_{m+1}}, \ldots, x_{t_{f-1}}$ be the missing value at $t_m, t_{m+1}, \ldots, t_{f-1}$.
3. Let t_{mc} be the middle location among $t_m, t_{m+1}, \ldots, t_{f-1}$.
4. Compute $x_{t_{mc}}$ from the intersection of gradients from the values of $\tan\alpha_1$ and $\tan\alpha_2$ in **ALGORITHM 1**.
5. Calculate $tan\theta_1$ between t_b and t_{mc} by
$$\tan\theta_1 = \frac{x_{t_{mc}} - x_{t_b}}{t_{mc} - t_b}$$
6. **for** $j = t_m$ **to** t_{mc} **do**
7. $x_j = x_{t_b} + (j\text{-}t_b)^* tan\theta_1$
8. Let $E = \{e_1, e_2, \ldots, e_s\}$ be a set of randomly sampled values in between x_{t_b} and x_j.
9. Set x_j to b_k from step 7 having highest occurrence frequency among all elements in E.
10. Move the non-missing location forward by setting $t_b = j$.
11. **end**
12. Calculate $tan\theta_2$ between t_{mc} and t_f
$$\tan\theta_2 = \frac{x_{t_f} - x_{t_{mc}}}{t_f - t_{mc}}$$
13. **for** $j = t_{mc+1}$ **to** t_{f-1} **do**
14. $x_j = x_{t_{mc}} + (j - t_{mc})^* tan\theta_1$
15. Let $F = \{f_1, f_2, \ldots, f_s\}$ be a set of randomly sampled values in between $x_{t_{mc}}$ and x_j.
16. Set x_j to c_k from step 15 having highest occurrence frequency among all elements in F.
17. Move the non-missing location forward by setting $t_{mc} = j$.
18. **end**

CHAPTER IV

MISSING DATA IMPUTATION BASED ON THE HYBRID OF ADJUSTABLE NEURAL NETWORKS AND SIMILARITY COMPARISON BETWEEN TWO CLUSTERS.

4.1 Introduction

In this chapter, an imputation technique is developed which focusses on characteristics (e.g., the shape) of the damaged areas inside a picture. If the damaged area is small and surronded by known pixels, then a neural network can be developed which uses only nearest-neighbor pixels for training and imputing. The number of sample nodes used for training is automatically adjusted depending on the data available in the neighborhood. Therefore, the number of sample nodes used in training are not necessarily equal for each missing pixel. The number of data points used during the training process depends on characteristics of the missing window and its neighhorhood.

If the missing area is large, then we have found that the neural network method cannot be used for imputation. Instead, we impute the pixels in the missing area by finding a similar area in the original image by using a clustering method. In this dissertation, two methods for this similarity test are used, namely a local test and a global test. Information from the similar areas is then used to impute the missing area.

4.2 Problem Formulation

Multi-dimensional data for a pixel are a set of multi-dimensional attributes of the pixel. An example of a multi-dimensional data set is shown in Fig. 4.1 where the position, RGB, and intensity data are given for each pixel.

Let $\mathbf{M}$ be a matrix of a multi-dimensional data set

$$\mathbf{M} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n], \tag{4.1}$$

where $\mathbf{M} \in n \times c$, $p_i \in \mathbb{R}^c$ and $1 \leq i \leq n$.

$$\mathbf{p}_i = [x_{i1}\ x_{i2}\ \ldots\ x_{ic}]^T \tag{4.2}$$

is the attribute vector for pixel i, where c is number of attributes and n is the number of pixels.

We assume that a multi-dimensional data set is composed of two parts, which are called input attributes and output attributes. An example of a multi-dimensional data set in a real world application is a set of vectors $\mathbf{p}_i = (x_i, y_i, z_i)$ for pixel i, $1 \leq i \leq n$, where (x_i, y_i) gives the position of pixel i and z_i gives the intensity of pixel i. In this case, (x_i, y_i) are called input attributes and z_i is called an output attribute. The relationship between these two parts of a multi-dimensional data set is

$$z_i = f(x_i, y_i) \tag{4.3}$$

where the output attribute z_i of pixel i (the intensity) is regarded as a function of the input attributes (x_i, y_i) (the position) of pixel i.

In this dissertation, we assume that missing data consists only of missing output data, i.e., that values of z_i are unknown for some set of pixels but that corresponding (x_i, y_i) values are always known. If the output data of a pixel i is missing, then its attribute vector is written as $\mathbf{p}_i^m$ and it will be called a target vector for imputation.

We consider the following problems for imputation of missing data:

1. How to impute missing multi-dimensional data?
2. How to select the number of nearest data points for training neural networks to tentatively fill missing data for small areas of missing data?
3. How to cluster incomplete multi-dimensional data?
4. How to measure the similarity between two clusters and how to measure the similarity between two manifolds?
5. How to impute incomplete multi-dimensional data by using clustering similarity comparison techniques?

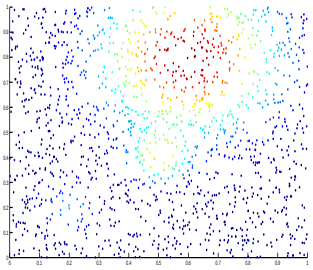


Figure 4.1: Characteristic of multi-dimensional data set in multi-dimensional spaces

4.3 The hybrid imputation of missing image using neural network and the similarity measurement based on the characteristics of damaged area

In this research, the missing values inside a damaged image were imputed using a hybrid imputation method based on a neural network and a similarity measurement based on the characteristics of missing areas. The processes used for imputing the missing pixels are shown in Fig. 4.2. The first step in the method is to set the size and shape of the mask template window to be used for the imputation starting with 3x3 pixels. The missing pixels are then detected and the marching method[26][18] is used to define an ordering in which the missing pixels will be imputed. The first step in the imputation of a missing pixel is to check the surrounding pixels using the mask template window. If a missing pixel is surrounded by observed pixels as shown in Fig. 4.3, then a neural network can be used to impute the missing value. However, if that area cannot be imputed with the neural network because of the large size of the damaged area then the similarity technique is used for imputing missing pixels. The imputation based on the similarity measurement can be done by the following process. First, find a direction of the pixels in an image using the Hough transform[27][28]. Then, check whether or not the damaged area should be imputed by global imputation. If the mean square error by using global imputation technique is greater than a predefined threshold, then the global imputation algorithm is not suitable for imputing missing values. In this case, an imputation technique for the damaged area based on a local imputation technique is used instead of the global imputation technique. The details of neural network based imputation and the similarity measurement based imputation are described in the following section.

4.4 Adjustable neural network

The method is based on the assumption that the properties of a missing pixel should be closer to the properties of nearby observed pixels than to properties of further distant

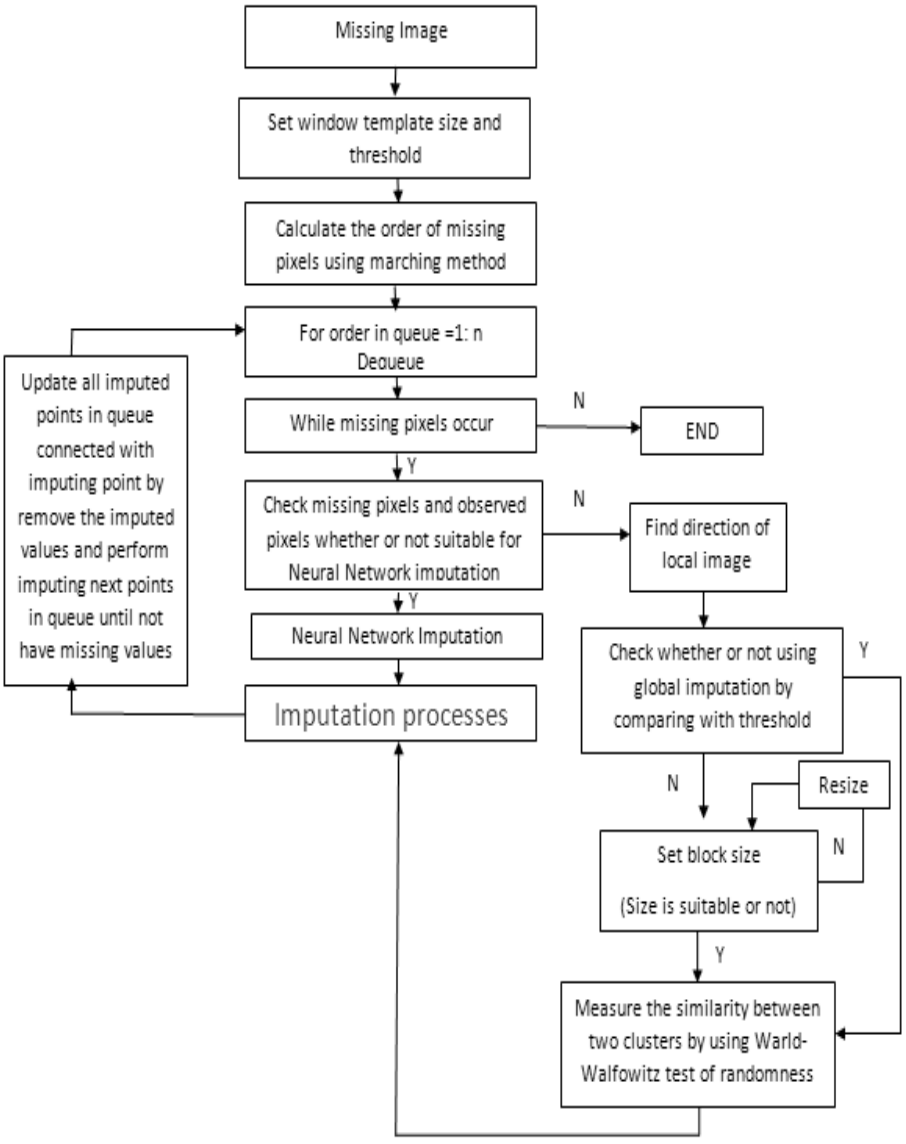


Figure 4.2: The framework of hybrid imputation of missing pixels using neural networks and similarity comparison.

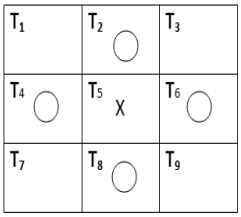


Figure 4.3: An example of 3×3 mask templete window.

pixels. In this section, a new method is introduced for imputing missing values based on an adjustable neural network for the primary imputation step. The neural network process for imputing missing values can be described as follows: Let $\mathbf{I}$ be a matrix of image data set with $n \times c$ pixels. Let $\mathbf{p}_i$ be a vector for pixel i at position x_i, y_i. Let (x_i, y_i) be the input attribute of vector $\mathbf{p}_i$. If the output data of a pixel i is missing then its attribute vector is written as $\mathbf{p}_i^m$. If a missing pixel occurs in a position (x_i, y_i) then set the mask template window with size of $w \times w$ pixels, denoted with T. The size of this window is calculated by the experiment because this size can cover a large number of observed data surrounding the damaged area. Next, superimpose position (x_c, y_c) of mask template window at position x_i, y_i of the missing value $\mathbf{p}_i^m$, where (x_c, y_c) is a center of mask template window and (x_i, y_i) is the position of missing pixel. After the suitable w value for window size are selected, the observed position pattern inside the window mask template is searched.

Fig. 4.3 shows an example of mask window template that will be used in the imputation process that is using data in T_2, T_4, T_6, T_8 positions for becoming a training sample. After the training sampling have been found, the intensity of these pixels are used as a target. Consider the following case. If a pattern of missing pixels inside mask window template occurs as shown in Fig. 4.3, the training data set will have only four inputs. Another case is if a pattern of missing pixel inside mask window template have more than four observed values in window mask template such as the occurring of observed data at positions T_6, T_7, T_8, and T_9, use this data to be a training sample in the adjustable neural network. So, not only to have only four points but also to have more than four points for the training process. Number of patterns to be used in the training process depends on the observed data in the mask window template, for example 4, 5, 6, 7, and 8 patterns.

ALGORITHM 4: Neural network training.

1. Let $\mathbf{p}_i$ be a vector for pixel i at position (x_i, y_i),
   $\mathbf{p}_i = [x_i \ y_i \ z_i]^T$ for pixel i, $1 \leq i \leq n$ where (x_i, y_i) gives the position
   of pixel i and z_i gives the intensity of pixel i.
   $\mathbf{p}_i^m$ denote output data of pixel i is missing.
2. Let $\mathbf{m}_i$ be an index vector to denote whether or not data at position (x_i, y_i) is missing.
3. Let m be a number of missing pixels and n be a number of pixels in an image.
4. Let k be a number of nearest neighbors of vector $\mathbf{p}_i^m$.
5. Let (x_i, y_i) be an input attribute of pixel i,
   z_i be an output attribute of pixel i.
6. **for** i=1 **to** n **do**
7. **if** a missing data $\mathbf{m}_i$ exists **then**
8. Let $\mathbf{K}$ be a set of nearest neighbors of vector $\mathbf{p}_i$ for pixel i
   at position (x_i, y_i) denoted by $(x_j, y_j) \in \mathbf{K}$.
9. Find k nearest neighbors of vector $\mathbf{p}_i$ based on
   minimum Euclidean distance by
   $$\underset{(x_j, y_j) \in \mathbf{K}}{\operatorname{argmin}} \ (\Sigma_{j=1}^{n}(x_i - x_j)^2 + (y_i - y_j)^2)$$
10. Suppose k nearest neighbors of missing data $\mathbf{p}_i^m$
    are $\mathbf{p}_m^1, \mathbf{p}_m^2, ..., \mathbf{p}_m^k, \ \mathbf{p}_m^d \in \mathbb{R}^n$.
11. Use the following data set to be a training pattern:
    (x_m^t, y_m^t) is input pattern.
    z_m^t is target output pattern.
    where t is an index of nearest neighbors of a missing pixel, $1 \leq t \leq k$.
12. Train only on the training set by setting the stopping criteria
    and the network parameters.
13. Stop training as soon as the error equals the predefined mean square error.
14. Use the weights in the previous step as the imputation step
    by using the following data,
    (x_m, y_m) is input pattern.
    z_m is desired output.
15. Use z_m to impute a missing value $\mathbf{p}_i^m$.
16. **end**
17. **end**

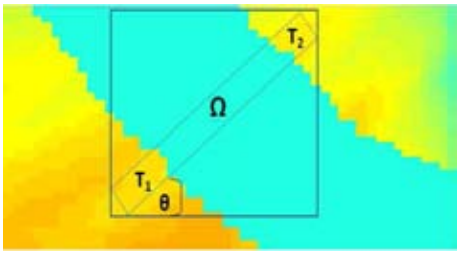


Figure 4.4: The mask template window using in global imputation which rotate depends on the angle of image.

4.5 Similarity measurement between two clusters.

The similarity measurement could be performed by measuring the similarity of cluster shape and cluster distribution of the two groups. This technique can be used when either the large size of damaged area occurred or the neural network cannot be used to perform the imputation of damaged image.

4.5.1 Global imputation processes.

In this section, the imputation algorithm was proposed by using global imputation technique to impute the large damaged areas inside an image. Suppose there is a missing value in vector $\mathbf{p}_i^m$, where $\mathbf{p}_i = [x_i \ y_i \ z_i]^T$. Search for the next observed values which have the same direction compared to the whole image and located on the opposite side of the pixel $\mathbf{p}_i^m$ by equation(4.4) and equation(4.5).

$$x_{i+1} = x_i + 1 \tag{4.4}$$

$$y_{i+1} = tan\theta(x_{i+1} - x_i) + y_i \tag{4.5}$$

Calculate the (x_{i+1}, y_{i+1}) position until reaching the observed values in the opposite side of damaged region. The scanning direction is performed by scanning through a straight line. The structure of mask template window is shown in Fig. 4.4.

The mask template window composes of the observed values in both two sides and the missing area locates at the center of the two observed areas. The mask template window is referred as the target mask window which composes of area T_1 , area T_2 and a missing area (Ω), denoted with $\Omega_m^T = T_1 \cup \Omega \cup T_2$ in θ direction. The size of T_1 and

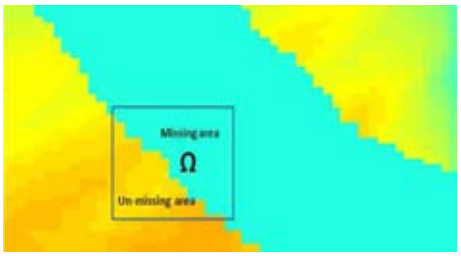


Figure 4.5: The mask template window using in local imputation which depends on the observed values.

T_2 areas is equal to the length of missing area. So, the size of T_1 is $\Omega/2$ and the size of T_2 is $\Omega/2$. After the target mask window Ω_m^T is calculated, search inside the searching window to find the most similar area(ψ) compare to the target mask window by using Wald-Wafowitz test. After the most similar area which is compared to the missing area with the predefined threshold is found, use this reference area to impute the missing area.

4.5.2 Local imputation process.

If the damaged area cannot be imputed with the global imputation technique due to the mean square error of the target mask window and the most similar area are greater than the threshold, then the local similarity measurement will be used. In this step, the target mask window which is mostly covered by the observed values is used for comparing with undamaged areas. The processes for local imputation is starting with calculated percentage of missing values and observed values inside the pre-defined target mask window.

$$\zeta = \frac{n(observed\_pixels) \times 100}{n(observed\_pixels) + n(missing\_pixels)}\% \tag{4.6}$$

If the ratio ζ is greater than 60%, use the target mask window for finding the similar area. If the ratio ζ of the target mask window is less than the pre-defined threshold, the size of mask template window will be resized untill the ratio are accepted. After that, using the target mask window to find the most similar area for reconstructing damaged area.

The algorithms of global based imputation and local based imputation can be described by the following algorithms.

ALGORITHM 5: Global Imputation algorithms

1. Let $I(x_i, y_i)$ be an image represented in the from of matrix of $r \times c$
   having some missing pixels.
   The position of any pixel is denoted by its coordinate (x_i, y_i).
2. Use marching method[29] to keep the order of the missing pixels to be imputed
   into a queue. A queue is defined by
   $\mathbf{Q} = \{(x_{a_i}, y_{a_i}, z_{a_i}) | 1 \leq i \leq m\}$
   where m is a number of missing pixels in an image.
3. Calculate the azimuth angle of sub-image by using Hough algorithm.
4. Use a matrix $FillArea(x_i, y_i)$ to denote whether or not the position (x_i, y_i) is missing.
5. Set threshold (α) for comparing whether or not using the global imputation algorithm.
6. Let vector $\mathbf{p}_m = [x_m \ y_m \ z_m]^T$ for a missing pixel m at row x_m column y_m
   which is dequeue from a queue.
7. Let l be index of missing pixel dequeue from $\mathbf{Q}$.
8. **for** $l = 1$ **to** m **do**
9. Calculate the next un-missing pixel position locate on the opposite side
   of the missing area by using the following variables.
   x_{start} is the starting position of a missing pixel $\mathbf{p}_m$ on $x-$axis
   y_{start} is the starting position of a missing pixel $\mathbf{p}_m$ on $y-$axis
   x_{end} is the next observed pixel on $x-$ axis
   y_{end} is the next observed pixel on $y-$ axis in θ direction calculated from
   $y_{end} = y_{start} + tan\theta(x_{end} - x_{start})$
10. Create the mask template window at the following position,
    $\Omega_m^T = \{(x_{startT}, y_{startT}), (x_{startT}, y_{startT+1}), ..., (x_{endT}, y_{endT})\}$
    where,
    Ω_m^T is a mask template window of missing pixel m.
    x_{startT} is the start x-position of mask template calculated from.
    $x_{startT} = x_{start}$ - (size of missing area)/2
    y_{startT} is the start y-position of mask template calculated from.
    $y_{startT} = y_{start}$ - (size of missing area)/2
11. Compare the similar area between Ω_m^T and the observed area called ψ_i^R
    by using ward-walfowitz test in ALGORITHM 7.
12. Find the most similar area ψ_m^R for imputing Ω_m^T by ALGORITHM 8.
13. Compare the mean square error(mse) between ψ_m^R and Ω_m^T denoted with
    $mse = \psi_m^R - \Omega_m^T$

14. **if** $mse < \alpha$ **then**
15. Use ψ_m^R area to impute Ω_m^T area that have a missing values.
16. **else**
17. Impute the missing data by ALGORITHM 6.
18. **end**
19. **end**

ALGORITHM 6: Local imputation

1. Let vector $\mathbf{p}_m = [x_m \ y_m \ z_m]^T$ for a missing pixel m at row x_m column y_m which is dequeue from a queue.
2. Let l be index of missing pixel dequeue from $\mathbf{Q}$.
3. **for** $l = 1$ **to** m **do**
4. Create an initial mask window template with size of 7×7 by the position x_m, y_m is in the center of mask template.
5 Check the percentage of available data in a mask window template with $\zeta = \frac{n(observed\_pixels)\times 100}{n(observed\_pixels)+n(missing\_pixels)}$
6. **if** the percentage of available data is greater than 60% **then**
7. Use Ω_m^T to compare the similarity with known area called ψ_i^R by ALGORITHM 7.
8. **else**
9. Resize the mask template window size by 1
10. Return to step 5.
11. **end**
12. Find the most similarity area ψ_m^R by ALGORITHM 8.
13. Use ψ_m^R area to impute Ω_m^T area which having a missing values.
14. **end**

4.5.3 Distribution based comparison between two clusters.

To measure the similarity between two sub-images, the density properties of these sub-images is used. In general, sub-image is called cluster. The target sub-image or the target cluster denoted by Ω_m^T. The reference sub-image or the reference cluster used to compare with the target cluster denoted with ψ_i^R. The similarity between two clusters

can be performed by comparing the similarity of manifold of two images. The similarity of two clusters can be checked by measuring the distribution of the pixels in these two clusters as follows. The method called Wald-Walfowitz test of randomness which is a two-sample test was adapted for this purpose. This idea is performed by comparing shape and distribution of two relevant clusters. The main concept is to check whether the probability candidate clusters originaly came from the same distribution with target cluster. Let Ω_m^T $=\{\mathbf{t}_1,\mathbf{t}_2,\mathbf{t}_3,...,\mathbf{t}_{n1}\}$ be a target cluster, $\mathbf{t}_i$ is a vector for pixel i in target cluster where some output attribute of $\mathbf{t}_i$ contains missing value. Let ψ_i^R $=\{\mathbf{r}_1,\mathbf{r}_2,\mathbf{r}_3,...,\mathbf{r}_{n2}\}$ be a reference cluster to be compared to checke whether the chance that these two clusters came from the same distribution. $\mathbf{r}_i$ is a vector for pixel i in a reference cluster.

The test is done by using a run which is defined as a consecutive sequence of identical labels. The number of runs is used as a statistical test. The distribution similarity measure between two clusters are calculated by applying Wald-Walfowitz test for multi-dimensional data sets as follows.

1. Merge two clusters into one cluster denoted by matrix $\mathbf{A}$.
2. Calculate the minimum spanning tree between two clusters in matrix $\mathbf{A}$.
3. From the minimum spanning tree in step 2, calculate the statistical test by using a number of connected between two different groups in matrix $\mathbf{A}$ denoted with R. If there is the connected graph between two different groups then increase R by one. Otherwise, if the connection is in the same group of cluster, do nothing. Calculate R until all paths are already computed. Finally, increased R by one.
4. Calculate the statistical test value of R, as follows.

$$W = \frac{R - \mu}{\sigma} \tag{4.7}$$

$$\mu = \frac{2n_1n_2}{N} + 1 \tag{4.8}$$

$$\sigma = \frac{2n_1n_2(2n_1n_2 - N)}{N^2(N-1)} \tag{4.9}$$

5. Calculate p-value of W as same as calculating Z-statistics.
6. Compare the 95% confidence interval between p-value of W and α with the following hypothesis:
   (a) H_0: two clusters come from same distribution.

(b) H_1: two clusters come from different distribution.

With the above hypothesis, if p-value of W greater than α, then the null hypothesis H_0 are accepted. Otherwise reject the null hypothesis.

7. Add the reference cluster which is already accepted by the test statistics, and is from the same distribution with target cluster into the similarity clustering list(C_{list}).

ALGORITHM 7: Clustering Similarity Comparison

1. Let $I(x_i, y_i)$ be an image represented in the from of matrix of $r \times c$ having some missing pixels.
   The position of any pixel is denoted by its coordinate (x_i, y_i).
2. Let $\mathbf{M}$ be a matrix of missing value, $\mathbf{m}_i$ denoted whether or not output attribute in pixel i is missing.
3. Let Ω_m^T be a target mask window of missing pixel $\mathbf{m}$.
4. Let $\mathbf{C}_{list}$ be a list of clusters that most similar to Ω_m^T.
5. Let Ψ_i^R be a sub-window in an image with the same size of Ω_m^T where i is index of each sub-window. $1 \leq i \leq w$, where w is number of sub-windows.
6. Compare the similarity between Ω_m^T and Ψ_i^R.
7. **For** $i = 1$ **to** w **do**
8. Merge these two clusters: Ω_m^T and Ψ_i^R into matrix $\mathbf{A}$
9. Create minimum spanning tree of each data in matrix $\mathbf{A}$.
10. Count number of runs (R), where R come from number of connected paths between two clusters.
11. **If** there is a connected path between two different clusters **then**
    $R = R + 1$.
12. **If** there is a connected path between two similar clusters **then**
    do nothing.
13. Calculate W by
    $W = (R - \mu)/\sigma$
    $\mu = ((2n_1n_2)/N) + 1,$
    $\sigma = 2n_1n_2(2n_1n_2 - N)/N^2(N - 1)$
14. Calculate p-value of W.
15. **If** p-value $> \alpha_{0.05}$ **then**
    the null hypothesis H_0 are accepted,

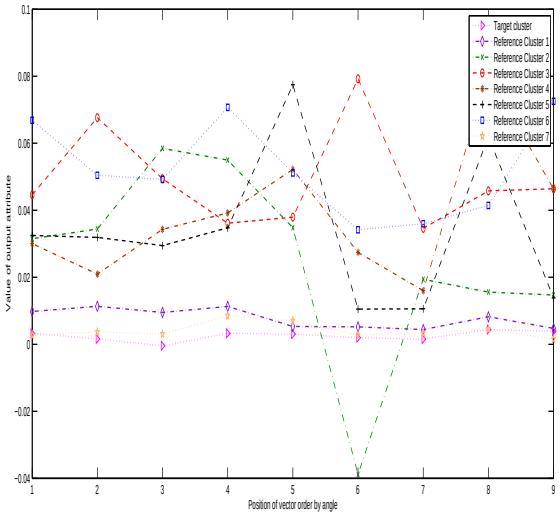


Figure 4.6: Two-dimensional graph for comparing the similarity between two clusters. Graph is drawn with two axes:x-axis is presented for vector position, y-axis is presented for the output attribute of vector.

otherwise reject null hypothesis.

16. Add this cluster into similarity list $\mathbf{C}_{list}$ by $\mathbf{C}_{list} = \mathbf{C}_{list} \cup \Psi_i^R$.
17. **end**
18. From the similarity list (C_{list}), find the most similarity cluster with Ω_m^T by using ALGORITHM 8.
19. Use Ψ_m^R which is the most similar to Ω_m^T to impute the missing area.
20. **end**

4.5.4 Calculating Euclidean distance between two clusters

After the similarity clustering list $\mathbf{C}_{list}$ of Ω_m^T are calculated, find the most similar cluster between target cluster (Ω_m^T) and the similarity clustering list $\mathbf{C}_{list}$ by using Euclidean distance between two clusters in the two dimensions graph. A two-dimensional graph is created for comparing the similarity between two clusters with two axes: x-axis is presented for vector position order by $cos\theta$, y-axis is presented for the output value of each pixels in this cluster. Each points in two clusters are numbered with respect to the most similar direction. After that, calculate the Euclidean distance between two clusters in this two dimensional graph. Repeat these steps for every cluster in the similarity cluster lists and select the most similar cluster compared to target cluster for imputing the missing values.

4.5.5 Imputing missing values in target cluster

After the most similar cluster with target cluster is selected denoted with Ψ_m^R, this reference cluster is used to impute the missing values. In this process, the missing values in target cluster are imputed with the following equation:

$$\mathbf{t}_m^T = \mathbf{r}_m^R - \frac{1}{2}(\mathbf{r}_{m-1}^R - \mathbf{t}_{m-1}^T) - \frac{1}{2}(\mathbf{r}_{m+1}^R - \mathbf{t}_{m+1}^T) \tag{4.10}$$

where

- $\mathbf{t}_m^T$ is the missing vector in target cluster.
- $\mathbf{r}_m^R$ is the reference cluster.
- $\mathbf{r}_{m-1}^R$ is the element at the left of missing data in reference cluster.
- $\mathbf{t}_{m-1}^T$ is the element at the left of missing data in target cluster.
- $\mathbf{r}_{m+1}^R$ is the element at the right of missing data in reference cluster.
- $\mathbf{t}_{m+1}^T$ is the element at the right of missing data in target cluster.

ALGORITHM 8: Two-dimension Similarity Measure.

1. Let $\mathbf{C}_{list} = \{\Psi_1, \Psi_2, ..., \Psi_w\}$ be a list of reference clusters having the same distribution with target cluster.
2. Each points in two clusters are numbered regarding to the most similar direction.
3. Create two dimension graph for comparing the similarity between two clusters where x-axis denoted with position of each vectors in cluster order by $cos\theta$, y-axis denoted with the value of output attribute at this vector.
4. Calculate the Euclidean distance between two cluster by using two dimensional graph.
5. Repeat steps 2-4 for every reference cluster in the similarity cluster lists.
6. Select the minimum Euclidean distance of the similarity cluster lists to impute the missing values.

CHAPTER V

IMPUTING INCOMPLETE SCAN LINE CORRECTOR (SLC)-OFF IMAGERY BASED ON NEURAL NETWORKS AND SIMILARITY MEASUREMENT BETWEEN TWO CLUSTERS.

5.1 Introduction

In this chapter, the imputation technique for the damaged areas inside a satellite image is focused. Satellite imagery is widely used in many fields such as agriculture, meteorology, geology, regional planning and forestry[30][31]. The source of satellites images is produced by different satellites; one of the most powerful satellites is Landsat 7 Enhanced Thematic Mapper Plus sensor or Landsat 7 ETM+. Unfortunately, in this machine, some sensors have malfunctioned since May 2003. The occurring malfunction of the Scan Line Corrector (SLC) in Landsat 7 ETM+ causes many missing pixels of images during the sensor's scanning process. This problem affects images acquired by that machine. It is referred to Landsat 7 ETM+ SLC-off problem. About 22% of scenes of images is lost. Many methods have been proposed for solving and recovering these missing pixels. Local Linear Histogram Matching (LLHM) technique[32] proposed by USGS was developed. In this method, they divided an image into two groups which are image to be filled called primary scene and image which does not have missing pixels called filled scene. The missing pixels are calculated from the image of Landsat 7 ETM+ SLC-on with lower resolution by calculating gain and standard values of their images. After that, the calculated values from filled scene are used to fill the primary scene. In this method, there are some limitations; if the scene being combined has temporal variability and the presence of cloud it cannot give the correctly imputed values. V.Rulloni et al. [33] proposed a linear regression technique for imputing Landsat7 ETM+ SLC-off by using the temporary accurate image with lower resolution and using regression between image to be imputed and lower resolution image in the regression. The experimental result showed that this algorithm gave accurate imputed values in missing area. However, the limitation of this method is that it must use other images for the regression which may be a problem if those missing images are not accurate in lower resolution. C.Zhanga et al.[34] proposed a method for gaps-fill of SLC-off Landsat 7 ETM+ by using the Kriging method which is a geostatistical approach. For

imputing missing values with Kriging method, the following process will be done. First, model the spatial dependency of values in each position which is known as a variogram model. Calculate the relation of all points by finding sill and range from the variogram model, after that the variogram model are selected depends on their distribution. Next, find Kriging weight and use this weight to calculate the missing points. Kriging method depends essentially on assumed second-order or intrinsic stationary of random fields and corresponding covariance or variogram function models. The use of a wrong model will have an effect on the accuracy of the imputation process. From the limitation of those methods, in this study, the imputation algorithms for Landsat 7 ETM+ SLC-off was proposed by using two-step imputation processes. First, impute the missing pixels by using neural networks after that repeat imputing missing values by using the most similar area of the missing pixels to be a reference data for imputing missing values. This comparison is based on testing whether or not the two images have the same distribution.

5.2 Imputing incomplete SLC-off imagery based on neural networks

Typical Landsat7 ETM+ imagery is composed of eight bands. In each band, they have different characteristics, application details, and spectral range as shown in Table 5.1. Digital Number (DN) which is a gray scale value ranging between 0-255 for displaying color information appears in each band. The concerned points in Landsat 7 ETM+ SLC-off image are pixels which have missing values or pixels which have 0 numbers. An example of Digital Number in band 1 of Landsat7 ETM+ imagery which have missing values are shown in Fig. 5.1.

Table 5.1: Characteristics of Landsat 7 ETM+ and SLC sensors which are composed of 8 bands.

Band	Spectral Range	EM Region	Application details
1	0.45-0.52	Visible Blue	Coastal water mapping differentiation of vegetation from soil.
2	0.52-0.60	Visible Green	Assessment of vegetation rigion.
3	0.63-0.69	Visible Red	Chlorophyll absorption for vegetation differentiation.
4	0.76-0.90	Near Infrared	Biomass surveys and differentiation of water bodies.
5	1.55-1.75	Middle Infrared	Vegetation and soil moisture measurement differentiation between snow and cloud.
6	10.40-12.50	Thermal Infrared	Thermal mapping soils moisture studies and plant heat stress measurement.
7	2.08-2.35	Middle Infrared	Hydrothermal Mapping
8	0.52-0.90	Near Infrared	Large area mapping, urban change studies.

An example of Landsat 7 ETM+ imagery from eight bands with size 500×500 pixels is shown in Fig. 5.2.

34	38	36	40	31	36	38	36	38	40
31	34	34	40	40	38	40	36	29	46
38	36	34	36	40	38	34	34	34	38
65	55	44	40	38	31	34	36	36	36
0	0	0	0	0	34	29	31	34	34
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
27	34	42	46	40	34	34	0	0	0
31	31	40	42	51	55	40	40	44	51
36	31	31	44	63	61	40	40	44	55
36	34	38	59	78	63	34	34	40	42

Figure 5.1: An example of Digital Number in Band 1 of Landsat 7 ETM+ image of Bangkok area. The 0 numbers are position where missing values occur.

These eight bands were combined into a multi-spectral image format by using Multi-spectral software. In this dissertation, only band 1 - band 5 and band 7 were used for the experiments which have the same spectral range. An example of missing data in Landsat7 ETM+ imagery is shown in Fig. 5.1 which is an image having a striping of 22% missing values.

5.2.1 Notations and definitions

Let $\mathbf{P}$ be a matrix of size $n \times 3$ of Landsat7 ETM+ with SLC-off and

$$\mathbf{p}_i = (x_i \ y_i \ z_i) \in \mathbf{R}^3$$

be the i-th row of $\mathbf{P}$. (x_i, y_i) is the 2-dimensional coordinates of the location of i-th data and z_i is the digital number corresponding with that location. They can be considered as the input and the output as shown in equation. 5.1

$$z_i = f(x_i, y_i), \quad 1 \leq i \leq n. \tag{5.1}$$

The missing data in each pixel is denoted by $\mathbf{p}_i^m = [x_m \ y_m \ z_m]^T$ and is called the target vector. Thus, we are interested in predicting the value $z_m = f(x_m, y_m)$.

5.2.2 Neural networks for landsat7 ETM+ SLC-off imputation

Our approach is based on the assumption that the radiometric of image pixels close to each other are more similar than image pixels which are far apart. So, the process for imputing missing data is done as follows. First, calculate nearest neighbors of each missing pixels by using Euclidean distance between the missing pixel and k-nearest neighbors to

itself. The number k is calculated by the following process. At first, let $b = 4$ where b is a number of nearest neighbors. For each $i = 1, 2, \ldots, n_1$, where n_1 is number of observed data sets which does not have missing values. Let v_i^* be the mean of b nearest neighbors of $\mathbf{p}_i$ calculate from the bootstrap algorithms. Next, compute the error caused by using these b elements as mean. To determine b, the cross-validation method to the complete data is used.

$$CV(b) = \sum_{i=1}^{n} (z_i - v_i^*)^2, 4 \leq b \leq 16, 1 \leq i \leq n. \tag{5.2}$$

Finally, let k be the number which gives the smallest value of $CV(b)$.

$$k = argmin_{b=4}^{16} CV(b) \tag{5.3}$$

Use this k number to be the number of nearest neighbors of missing values. The k-nearest neighbors are calculated by using k minimum Euclidean distances from the following equation

$$\mathbf{p}_i^d = min_{d=1}^{k} (\Sigma_{j=1}^{n_1} (\mathbf{p}_i - \mathbf{p}_j)^2), \tag{5.4}$$

where $\mathbf{p}_i^d$ is nearest neighbors of p_i, d is number of nearest neighbors of $\mathbf{p}_i$.

ALGORITHM 9: Calculate k value

1. Let k be a number of nearest neighbors.
   Let n_1 be a number of non-missing data.
   Let $\mathbf{p}_i$ be non-missing data at i position.
   Let v_i^* be a mean value of k nearest neighbors.
2. **for** $b = 4$ to 16 **do**
3. **for** $i = 1$ to n_1 **do**
4. Find k nearest neighbors of $\mathbf{p}_i$ using Euclidean distance denote by $\mathbf{p}_i^1, \mathbf{p}_i^2, \mathbf{p}_i^3, ..., \mathbf{p}_i^b$
5. Find minimum digital value(z_i) of k- nearest neighbors of $\mathbf{p}_i$ by min_value=min($z_i^1, z_i^2, z_i^3, ..., z_i^k$)
6. Compute maximum digital value(z_i) of b-nearest neighbors of $\mathbf{p}_i$ by max_value=max($z_i^1, z_i^2, z_i^3, ..., z_i^b$)
7. Randomly select data in the range of min_value and max_value 10 times denoted by $z_{i*}^1, z_{i*}^2, z_{i*}^3, ..., z_{i*}^{10}$

8. Calculate mean value of this 10 data sets with
   $v_i^* = 1/10 \sum_{j=1}^{10} z_{i*}^j$
9. **end**
10. Calculate the cross-validation by summation of error between z_i versus v_i^* with
    $CV(b) = \sum_{i=1}^{n_1}(z_i - v_i^*)$

8. **end**
9. Select the smallest $CV(b)$ with
   $k = argmin_{b=4}^{16} CV(b)$.
11. Set k as a number of nearest neighbors for the next steps.

Use k as the number of nearest neighbors of pixel $\mathbf{p}_i$. If the missing data occurred at $\mathbf{p}_i^m$, find k nearest neighbors of $\mathbf{p}_i^m$ denoted with $\mathbf{p}_d^m$. d is the index of nearest neighbors. Also, with this $\mathbf{p}_d^m$ find k nearest neighbors of $\mathbf{p}_d^m$ denoted with $\mathbf{p}_{d,j}^m$. Use the following data to be a training pattern,

- $(x_d^m, y_d^m, z_{d,1}^m, z_{d,2}^m, ..., z_{d,k}^m)$ is input pattern where x_d^m, y_d^m is a position of nearest neighbor of missing pixel p_i^m. $z_{d,1}^m, z_{d,2}^m, ..., z_{d,k}^m$ are output attribute of the $k-$ nearest pixels of $\mathbf{p}_d^m$.
- z_d^m is a desired output.

Train only on the training set by setting the stopping criteria and the network parameters. In this study, feed-forward multilayer neural network with back propagation learning algorithms was used. The network consists of one input layer, two hidden layers, and one output layer. Set of inputs and desired output were fed into the neural network to learn the relationship of data. The process in hidden layer is to adjust weight connected to each node of input. The root mean square error is compared between desired output and its calculated output. If the error is not satisfied with the predefined values, it will propagate error back to the former layer. This will be done from the direction of the upper layer towards the input layer. This algorithm will adjust weight from initial weight until it gives the satisfied the mean square error. The mean square error is calculated from the following equation,

$$E = \frac{1}{2} \sum_p \sum_i (d_{ip} - o_{ip})^2, \tag{5.5}$$

where

- p is the pattern index.
- i is node index.
- d_{ip} is a desired output of node i pattern p.
- o_{ip} is a calculated output of node i pattern p.

In each node of hidden layer, a node k in layer h is described by the following pair of equations,

$$u_k^{(h)} = \sum_{j=1}^{m} w_{k,j}^{(h)} u_k^{(h-1)} \tag{5.6}$$

and

$$y_k^{(h)} = f(u_k^{(h)} + \theta_k^h), \tag{5.7}$$

- k is node index in layer h.
- h is the layer number.
- $y_k^{(0)}$ is the input node and $y_i^{(n)}$ is output node.
- θ_k^h is bias in h layer.
- $y_k^{(n)}$ is the final output which uses to impute the missing value.

In this neural network architecture, in each process of missing data imputation, will be calculated from the most nearest neighbor pixels. Each neuron performs a weighted summation of the inputs, and then passes this weighted summation of the inputs into a nonlinear activation function. Next, use sigmoid function to be activation function in each hidden layer by this equation,

$$sigmoid(x) = \frac{1}{1 + e^{-x}}. \tag{5.8}$$

Finally, the network output which is called the output layer is formed by weighted summation of the outputs of the neurons in the hidden layer. In this study, only one single output is used for the approximation of missing values. Use the weights in the previous step as the imputation step by using the following data,

- $(x_m, y_m, z_m^1, z_m^2, ..., z_m^k)$ is input pattern.
- z_m is desired output.

Use z_m to impute missing values $\mathbf{p}_m$,

ALGORITHM 10: Neural network training.

1. Let $\mathbf{p}_i$ be a vector for pixel i at position (x_i, y_i),
   $\mathbf{p}_i = [x_i\ y_i\ z_i]^T$ for pixel i, $1 \leq i \leq n$ where (x_i, y_i) gives the position
   of pixel i and z_i gives the intensity of pixel i.
   $\mathbf{p}_i^m$ denote output data of pixel i is missing.
2. Let $\mathbf{m}_i$ be an index vector to denote whether or not data at position (x_i, y_i) is missing.
3. Let m be a number of missing pixels and n be a number of pixels in an image.
4. Let k be a number of nearest neighbors of vector $\mathbf{p}_i^m$.
5. Let (x_i, y_i) be an input attribute of pixel i,
   z_i be an output attribute of pixel i.
6. **for** i=1 **to** n **do**
7. **if** a missing data $\mathbf{m}_i$ exists **then**
8. Let $\mathbf{K}$ be a set of nearest neighbors of vector $\mathbf{p}_i$ for pixel i
   at position (x_i, y_i) denoted by $(x_j, y_j) \in \mathbf{K}$
9. Find k nearest neighbors of vector $\mathbf{p}_i$ based on
   minimum Euclidean distance by
   $$\underset{(x_j, y_j) \in \mathbf{K}}{\operatorname{argmin}} \ (\Sigma_{j=1}^{n}(x_i - x_j)^2 + (y_i - y_j)^2)$$
10. Suppose k nearest neighbors of missing data $\mathbf{p}_i^m$
    are $\mathbf{p}_m^1, \mathbf{p}_m^2, ..., \mathbf{p}_m^k$, $\mathbf{p}_m^d \in \mathbb{R}^3$ and $1 \leq d \leq k$.
11. In each $\mathbf{p}_m^d$, find its k nearest neighbors denoted with $\mathbf{p}_m^{d,l}$, $1 \leq l \leq k$.
12. Use the following data set to be a training pattern:
    $(x_j^m, y_j^m, z_{j,1}^m, z_{j,2}^m, ..., z_{j,k}^m)$ is input pattern.
    z_j^m is target output pattern.
    where j is an index of nearest neighbors of a missing pixel, $1 \leq j \leq k$.
13. Train only on the training set by setting the stopping criteria
    and the network parameters.
14. Stop training as soon as the error is reaching the mean square error.
15. Use the weights in the previous step as the imputation step
    by using the following data,
    $(x_m, y_m, z_m^1, z_m^2, ..., z_m^k)$ is input pattern.
    z_m is desired output.

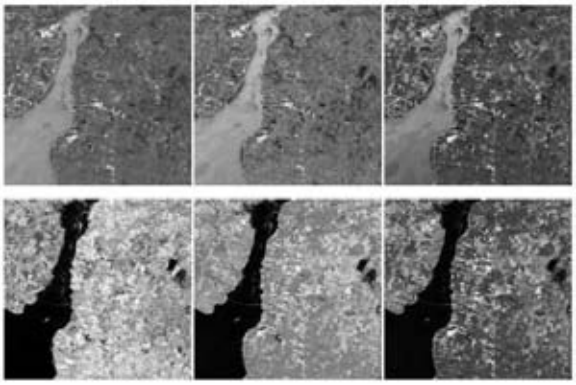


Figure 5.2: The example of Landsat 7 ETM+ imagery from six bands with size 500×500 pixels. In this research, we use only band 1- band 5 and band 7 for experiments.

16. Use z_m to impute a missing value $\mathbf{p}_i^m$.

17. **end**

18. **end**

5.2.3 Similarity measurement between two clusters

After preliminary imputing incomplete Landsat 7 ETM+ SLC-off data, the next process is to impute missing values depending on the similarity between two groups of pixels called clusters. The similarity measurement could be performed by measuring the similarity of clustering shape, clustering distribution, variance and density of two clusters.

5.2.3.1 Distribution based comparison between two clusters

This section explains the processes of how to measure the similarity between two clusters by adapting the Wald-Walfowitz Runs Test for Randomness. This algorithm is performed by comparing shape and distribution of two relevant clusters. The probability was checked that whether or not the candidate cluster which is used to impute missing values has similar distribution with target cluster. Let Ω_m^T ={ $\mathbf{tc}_1$,$\mathbf{tc}_2$,$\mathbf{tc}_3$,...,$\mathbf{tc}_{n_1}$ } be a target cluster, $\mathbf{tc}_i$ is a vector for pixel i in target cluster where some output attribute of $\mathbf{tc}_i$ contains missing value. Let ψ_i^R ={$\mathbf{rc}_1$,$\mathbf{rc}_2$,$\mathbf{rc}_3$,...,$\mathbf{rc}_{n2}$} be a reference cluster to be compared to check whether the chance that these two clusters came from the same distribution. $\mathbf{rc}_i$ is a vector for pixel i in a reference cluster. The test is done by using a run which is defined as a consecutive sequence of identical labels. The number of runs is used as a statistical test. The algorithm is performed as follows. First, compute the principle component analysis (PCA) to consider the distribution or variance of each data in Ω^T.

The variance of data in cluster depends on the axis of principle component in decreasing order. Second, compute principle component in reference cluster (Ψ^R) in the same manner as calculating principle component of target cluster. Next, the distribution similarity measurement between two clusters are calculated by applying the Wald-Walfowitz Runs Test for Randomness as follows.

1. Merge two clusters into one cluster denoted by matrix **A**.
2. Calculate the minimum spanning tree between two clusters in matrix **A**.
3. From the minimum spanning tree in step 2, calculate the statistical test by using a number of connection between two different groups in matrix **A** denoted with R. If there is the connected graph between two different groups then increase R by one. Otherwise, if the connection is in the same group of cluster, do nothing. Calculate R until all paths are already computed. Finally, increased R by one.
4. Calculate the statistical test value of R, as follows.

$$W = \frac{R - \mu}{\sigma} \tag{5.9}$$

$$\mu = \frac{2n_1n_2}{N} + 1 \tag{5.10}$$

$$\sigma = \frac{2n_1n_2(2n_1n_2 - N)}{N^2(N-1)} \tag{5.11}$$

5. Calculate p-value of W as same as calculating Z-statistics.
6. Compare the 95% confidence interval between p-value of W and α with the following hypothesis:
   (a) H_0: two clusters come from same distribution.
   (b) H_1: two clusters come from different distribution.

   With the above hypothesis, if p-value of W greater than α, then the null hypothesis H_0 are accepted. Otherwise reject the null hypothesis.
7. Add the reference cluster which is already accepted by the test statistics, and is from the same distribution with target cluster into the similarity clustering list (C_{list}).

ALGORITHM 11: Clustering Similarity Comparison

1. Let $I(x_i, y_i)$ be an image represented in the from of matrix of $r \times c$
   having some missing pixels.
   The position of any pixel is denoted by its coordinate (x_i, y_i).
2. Let $\mathbf{M}$ be a matrix of missing value, $\mathbf{m}_i$ denoted whether or not output
   attribute in pixel i is missing.
3. Let Ω_m^T be a target mask window of missing pixel $\mathbf{m}$.
4. Let $\mathbf{C}_{list}$ be a list of clusters that most similar to Ω_m^T.
5. Let Ψ_i^R be a sub-window in an image with the same size of Ω_m^T
   where i is index of each sub-window. $1 \leq i \leq w$, where w is number of sub-windows.
6. Compare the similarity between Ω_m^T and Ψ_i^R.
7. **For** $i = 1$ **to** w **do**
8. Merge these two clusters: Ω_m^T and Ψ_i^R into matrix $\mathbf{A}$
9. Create minimum spanning tree of each data in matrix $\mathbf{A}$.
10. Count number of runs (R), where R come from number of connected paths
    between two clusters.
11. **If** there is a connected path between two different clusters **then**
    $R = R + 1$.
12. **If** there is a connected path between two similar clusters **then**
    do nothing.
13. Calculate W by
    $W = (R - \mu)/\sigma$
    $\mu = ((2n_1n_2)/N) + 1,$
    $\sigma = 2n_1n_2(2n_1n_2 - N)/N^2(N - 1)$
14. Calculate p-value of W.
15. **If** p-value $> \alpha_{0.05}$ **then**
    the null hypothesis H_0 are accepted,
    otherwise reject null hypothesis.
16. Add this cluster into similarity list $\mathbf{C}_{list}$ by $\mathbf{C}_{list} = \mathbf{C}_{list} \cup \Psi_i^R$.
17. **end**
18. From the similarity list (C_{list}), find the most similarity cluster
    with Ω_m^T by using ALGORITHM 12.
19. Use Ψ_m^R which is the most similar to Ω_m^T to impute a missing area.
20. **end**

5.2.4 Angle based similarity measurement between two clusters

After the similarity clustering list $C_{list} = \{\Psi_1, \Psi_2, ..., \Psi_n\}$ is calculated, compute the most similar cluster between target cluster and similarity clustering list as follows

1. Compute angle (θ) of each points in target cluster by measuring each point with $\mathbf{tc}^T_{center}$ where $\mathbf{tc}^T_{center}$ is target cluster center calculated from

$$\mathbf{tc}^T_{center} = \frac{1}{n_1}\Sigma^{n_1}_{i=1}(\mathbf{tc}^T_i) \tag{5.12}$$

where n_1 is number of elements in target cluster and $\mathbf{tc}^T_i$ is element i in target cluster. Calculate angle of each point by

$$\theta_i = cos\theta^{-1}(\frac{\mathbf{tc}^T_i.\mathbf{tc}^T_{center}}{||\mathbf{tc}^T_i||.||\mathbf{tc}^T_{center}||}) \tag{5.13}$$

where θ_i is angle between $\mathbf{tc}^T_i$ and $\mathbf{tc}^T_{center}$.

2. Compute angle(β) of each points in reference cluster by measuring each point with $\mathbf{rc}^R_{center}$ where $\mathbf{rc}^R_{center}$ is reference cluster center. To compare the similarity of direction between two clusters, move the center of each reference cluster to the same position as follows,

$$\mathbf{rc}^R_{oldcenter} = \frac{1}{n2}\Sigma^{n2}_{i=1}(\mathbf{rc}^R_i) \tag{5.14}$$

where n_2 is number of elements in reference cluster and $\mathbf{rc}^R_i$ is element i in reference cluster.

$$\mathbf{rc}^R_{newcenter} = \mathbf{tc}^T_{center} \tag{5.15}$$

then the new position of each element in reference cluster is calculated by,

$$\mathbf{rc}^{R'}_i = \mathbf{rc}^R_i - \mathbf{rc}^R_{newcenter} \tag{5.16}$$

where $\mathbf{rc}^{R'}_i$ is the new position of reference cluster refers to $\mathbf{rc}^R_{newcenter}$. Next, calculate angle of each points with following equation

$$\beta_i = cos\beta^{-1}(\frac{\mathbf{rc}^{R'}_i.\mathbf{rc}^R_{newcenter}}{||\mathbf{rc}^{R'}_i||.||\mathbf{rc}^R_{newcenter}||}) \tag{5.17}$$

where β_i is angle between $\mathbf{rc}^{R'}_i$ and $\mathbf{rc}^R_{newcenter}$.

3. Compute the Euclidean distance between two clusters by using a two-dimensional graph as follows: Each point in two clusters is numbered, referring to the most similar direction from eq.(5.13) and eq.(5.17). Next, a two-dimensional graph is created for comparing the similarity between two clusters. A graph is drawn with two axes: x-axis represents vector position order by $cos\theta$ and $cos\beta$, y-axis represents Digital Number of each vector in this cluster. After that, calculate the Euclidean distance between two clusters using this two-dimensional graph.

4. Repeat steps 2 and 3 for every reference cluster in the similarity clustering lists. Select the most similar value compared to target cluster for imputing missing values.

5.2.5 Imputing missing values in target cluster

After the most similar cluster with target cluster is selected denoted with Ψ_m^R, this reference cluster is used to impute the missing values. In this process the missing values in target cluster are imputed with the following equation:

$$\mathbf{tc}_m^T = \mathbf{rc}_m^R - \frac{1}{2}(\mathbf{rc}_{m-1}^R - \mathbf{tc}_{m-1}^T) - \frac{1}{2}(\mathbf{rc}_{m+1}^R - \mathbf{tc}_{m+1}^T) \tag{5.18}$$

where

- $\mathbf{tc}_m^T$ is the missing vector in target cluster.
- $\mathbf{rc}_m^R$ is the reference cluster.
- $\mathbf{rc}_{m-1}^R$ is the element at the left of missing data in reference cluster.
- $\mathbf{tc}_{m-1}^T$ is the element at the left of missing data in target cluster.
- $\mathbf{rc}_{m+1}^R$ is the element at the right of missing data in reference cluster.
- $\mathbf{tc}_{m+1}^T$ is the element at the right of missing data in target cluster.

ALGORITHM 12: Two-dimension Similarity Measure.

1. Let $C_{list} = \Psi_1, \Psi_2, ..., \Psi_n$ be a list of reference clusters
   having the same distribution with target cluster.
2. Compute center of target cluster center $\mathbf{tc}_{center}^T$ by
   $\mathbf{tc}_{center}^T = \frac{1}{n1}\Sigma_{i=1}^{n1}(\mathbf{tc}_i^T)$

where $n1$ is number of elements in target cluster,

$\mathbf{tc}_i^T$ is vector i in the target cluster.

3. Calculate angle between each vector in the reference cluster and the target cluster center ($\mathbf{tc}_{center}^T$) by

   $\theta_i = cos\theta^{-1}(\frac{\mathbf{tc}_i^T.\mathbf{tc}_{center}^T}{||\mathbf{tc}_i^T||.||\mathbf{tc}_{center}^T||})$

   where θ_i is the angle between $\mathbf{tc}_i^T$ and $\mathbf{tc}_{center}^T$.

4. The angle of each vector is $\theta_1, \theta_2, ..., \theta_{n1}$

   where i is index of each vector in target cluster and $n1$ is number of elements in target cluster. After that sort this angle in increasing order.

5. Compute center of reference cluster center ($\mathbf{rc}_{center}^R$) by

   $\mathbf{rc}_{oldcenter}^R = \frac{1}{n2}\Sigma_{i=1}^{n2}(\mathbf{rc}_{center}^R)$

   where $n2$ is number of element in reference cluster,

   $\mathbf{rc}_i^R$ is vector i in reference cluster.

6. Transform the center of reference cluster into same position with target cluster center for calculating the angle by

   $\mathbf{rc}_{newcenter}^R = \mathbf{tc}_{center}^T$

   and move all element in reference cluster into new position refer to $\mathbf{rc}_{center}^R$

   $\mathbf{rc}_i^R = \mathbf{rc}_i^R - \mathbf{tc}_{center}^T$

   where $\mathbf{rc}_{center}^R$ is new position compare to $\mathbf{tc}_{center}^T$.

7. Calculate angle between each point in cluster and reference cluster center ($\mathbf{rc}_{center}^R$)

   $\beta_i = cos\beta^{-1}(\frac{\mathbf{rc}_i^R.\mathbf{tc}_{center}^T}{||\mathbf{rc}_{center}^R||.||\mathbf{tc}_{center}^T||})$

   where β_i is angle between $\mathbf{rc}_{center}^T$ and $\mathbf{tc}_{center}^T$.

8. The angle of each point is $\beta_1, \beta_2, ..., \beta_{n2}$

   where i is index of each points in reference cluster and $n2$ is number of elements in reference cluster. After that sort this angle in an increasing order.

9. Compute the Euclidean distance between two cluster in the two dimensional graph by step 10-12.

10. Each point in two clusters are numbered regarding to the most similar direction calculated from steps 2 to 8.

11. Create two dimension graph for comparing the similarity between two clusters where x-axis denote position of each vectors in cluster order by $cos\theta$ and $cos\beta$, y-axis denote the value of output attribute at this vector.

12. Calculate the Euclidean distance between two cluster in two dimensional graph.

13. Repeat this steps 2 to 10 for every reference cluster

in the similarity cluster lists.

14. Select the minimum Euclidean distance of the similarity cluster lists to impute the missing values.

CHAPTER VI

EXPERIMENTAL RESULTS

6.1 Introduction

This chapter shows the experimental results with the proposed algorithms and the competitive methods. The experimental results were divided into three parts:

1. Time-series data imputation
2. Image imputation
3. Landsat Scan Line Corrector(SLC)-off imagery imputation

6.2 Time-series data imputation

The following data, parameters, and comparison results were considered in the experiments.

1. Data used in the experiments are 1201 Mackey Glass chaotic time series data, 1071 sunspot dataset from A.D. 1915 to 2002 ,and 2000 daily gauge height data set at Ben Luang gauging station, Maetun stream, Ping river in Thailand. The characteristics of each data set are shown in Fig. 6.1 to Fig. 6.3.
2. The percentage of missing data was set to various levels as follows: 10%, 20%, 30%, 40%, 50%, 60%, and 70%, using MCAR(Missing Completely At Random) mechanism.
3. The following three comparison methods were used: Cubic interpolation, MI interpolation by using NORM Software, and Varies Window Similarity Measure (VWSM) algorithms. In the experiments, the performance of the proposed algorithms was tested by repeating the algorithms 30 times for each percentage of missing level.

6.2.1 Experimental Results

The results from the proposed algorithms are shown in Table 6.1. Table 6.1 shows the MSE of Mackey-Glass chaotic time series data by using four methods: Cubic interpolation,

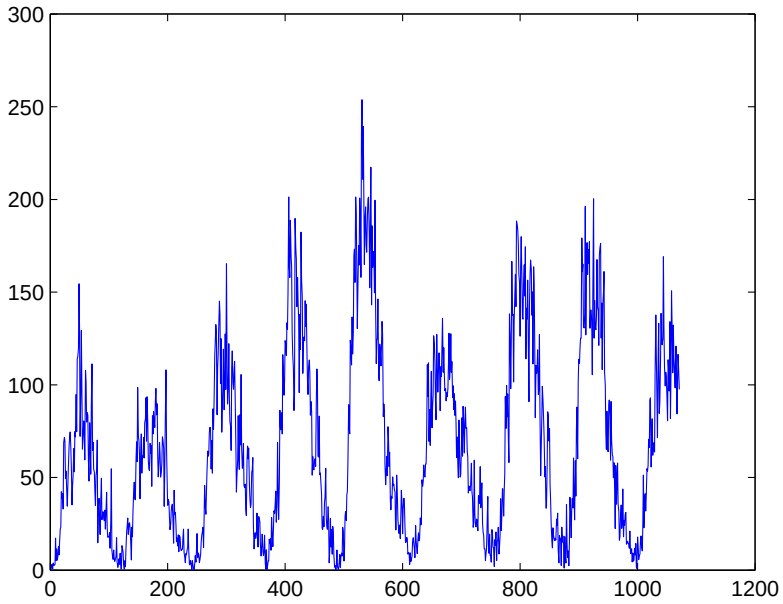


Figure 6.1: The time series data of Monthly Sunspot.

MI interpolation, VWSM, and the proposed algorithms which is RGGB algorithms. The experimental results showed that the proposed method gave the lowest MSE for every level of missing data. Table 6.1 shows the MSE of Gauge Heigth time series data by using four methods. The proposed algorithms gave the lowest MSE for every level of missing data. Furthermore, Table 6.1 shows the MSE of Sunspot time series data data by using four methods. The proposed algorithms gave the lowest MSE for every level of missing data.

Fig. 6.4 showed the actual and imputed data for Sunspot data set by using the proposed algorithms when the missing rate is 70%. For Mackey Glass data set, the given data with 70% missing rate is shown in Fig. 6.5. After applying the proposed imputing algorithm, the result is illustrated in Fig. 6.6. There are 500 data points in this tested set.

6.2.2 Performance Measure

To measure the performance of the proposed algorithms, the error of real value and predicted value were calculated by using mean square error(MSE) and mean absolute percentage error(MAPE) by equation(6.1) and equation(6.3) .

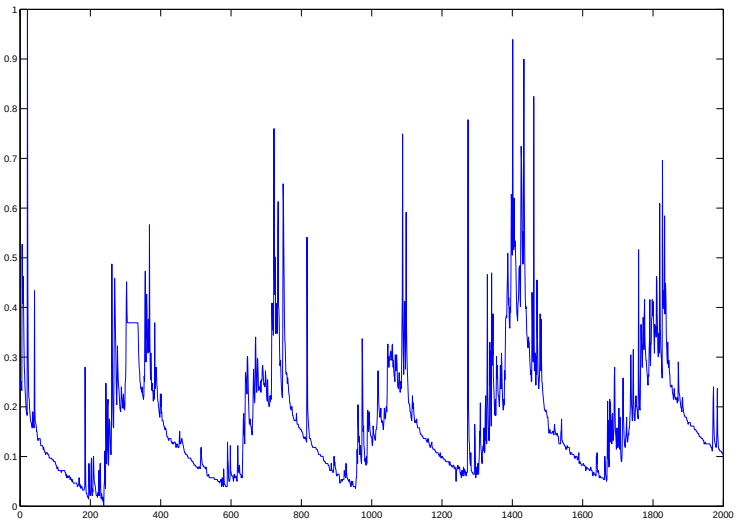


Figure 6.2: The time series data of Gauge Heigth.

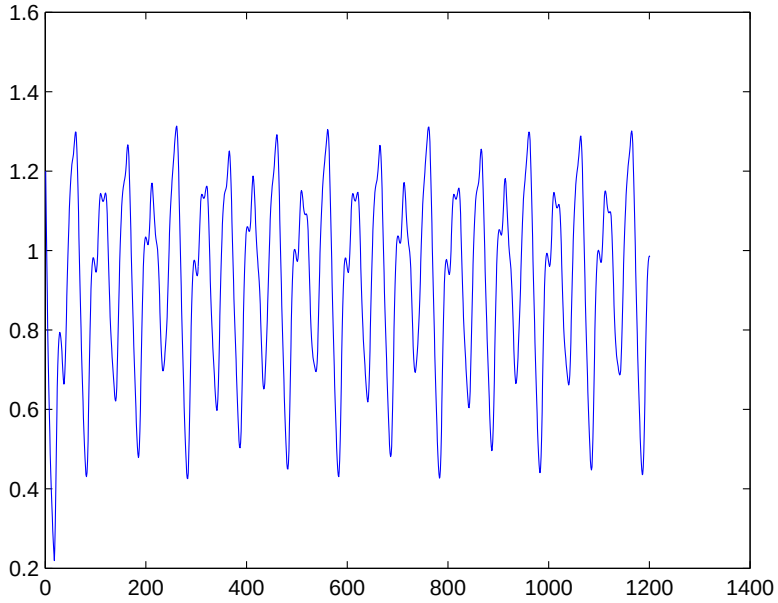


Figure 6.3: The time series data of Mackey-Glass.

$$MSE = (\sum_{1 \leq i \leq n} (\hat{T}_i - T_i))/nmiss \tag{6.1}$$

and all experiments are calculated by using eq.(6.2).

$$MSE = (\sum_{1 <= i <= n} (\hat{T}_i - T_i))/nmiss \times r \tag{6.2}$$

$\hat{T}_i$ is an imputed incomplete time series data by using Cubic interpolation, MI algorithms, VWSM algorithms and the proposed algorithms. T_i is complete time series data. $nmiss$ is the number of missing data. r is the number of experiments. Mean absolute percentage error(MAPE) was calculated from the following equation:

$$MAPE = \frac{100}{n} \times \sum_{i=1}^{n} |\frac{\hat{T}_i - T_i}{T_i}| \tag{6.3}$$

$\hat{T}_i$ is an imputed incomplete time series data. T_i is complete time series data. n is the number of missing data. After that, the accuracy of imputed image was calculated from equation(6.4),

$$Accuracy = 100 - MAPE\% \tag{6.4}$$

The comparison of MSE for each data set and each level of missing value are shown in Fig. 6.7 - Fig. 6.19.

6.2.3 Time complexity

The time complexity of the proposed algorithms is $O(n)$ where n is the number of data. The computation time in step 1 of **ALGORITHM 1** do not depend on the number of data. So, the time used in this step is constant $O(1)$. The time complexity in steps 2 to 12 of **ALGORITHM 1** is depended on the number of time step in the data set. However, the time complexity in step 8 of **ALGORITHM 1** is depended on the time complexity of **ALGORITHM 2** which is $O(n)$. Moreover, the time complexity in step 10 of **ALGORITHM 1** is depended on the time complexity of **ALGORITHM 3** which is $O(n)$. So, the time complexity in steps 2 to 12 of **ALGORITHM 1** is $O(n)$. Thus, the time complexity in **ALGORITHM 1** is

$$T_{alg} = O(1) + O(n) = O(n) \tag{6.5}$$

Table 6.1: The mean square error (MSE)$\times$ 10^{-6} of each time series data by using cubic spline interpolation, MI interpolation method, VWSM algorithm, and the proposed algorithms denoted by RGGB.

Missing rate	Cubic	MI	VWSM	**RGGB**
Mackey Glass chaotic time series				
10%	4.11	**0.8**	92.56	2.01
20%	8.59	4.69	14.18	**2.69**
30%	18.3	13.83	27.95	**3.72**
40%	53.05	50.91	58.71	**10.03**
50%	258.39	**140.59**	170.57	140.84
60%	685.43	589.01	555.65	**430.96**
70%	936.31	1356.20	903.80	**664.50**
Gauge Height time series data				
10%	2.94	4.51	2.34	**1.05**
20%	2.62	4.52	2.00	**1.68**
30%	3.31	5.01	2.18	**1.83**
40%	4.11	5.37	2.80	**2.02**
50%	4.92	5.96	3.64	**1.79**
60%	5.25	6.99	4.56	**2.61**
70%	7.47	7.68	6.81	**3.0**
Sunspot time series data				
10%	4.98	7.11	4.25	**2.88**
20%	5.59	7.55	4.73	**3.03**
30%	6.76	8.00	5.01	**3.56**
40%	7.67	8.23	5.28	**3.57**
50%	8.27	8.98	5.99	**4.94**
60%	9.58	9.67	7.05	**5.01**
70%	12.53	10.83	8.42	**6.02**

Table 6.2: The percentage of the accuracy in each time series data by using cubic spline interpolation, MI interpolation method, VWSM algorithm, and the proposed algorithms denoted with RGGB.

Missing rate	Cubic	MI	VWSM	**RGGB**
Mackey Glass chaotic time series				
10%	98.15	**99.64**	58.51	99.09
20%	96.05	97.84	93.48	**98.76**
30%	85.55	89.08	77.93	**97.06**
40%	86.48	87.03	85.04	**90.68**
50%	83.06	**90.54**	85.42	89.27
60%	80.44	81.79	82.25	**87.64**
70%	79.70	75.08	70.05	**85.43**
GH time series				
10%	74.88	61.47	80.00	**91.03**
20%	77.61	61.38	82.91	**85.64**
30%	71.72	57.20	81.37	**84.36**
40%	64.88	54.12	76.08	**82.74**
50%	57.96	49.08	68.90	**84.70**
60%	55.15	40.28	61.04	**77.70**
70%	36.18	34.39	41.82	**74.37**
Sunspot time series				
10%	86.75	81.08	88.69	**92.34**
20%	85.13	79.91	87.41	**91.94**
30%	82.02	78.72	86.67	**90.53**
40%	79.59	78.11	85.95	**90.50**
50%	78.00	76.11	84.06	**86.86**
60%	74.51	74.28	81.24	**86.67**
70%	66.67	71.19	77.60	**83.98**

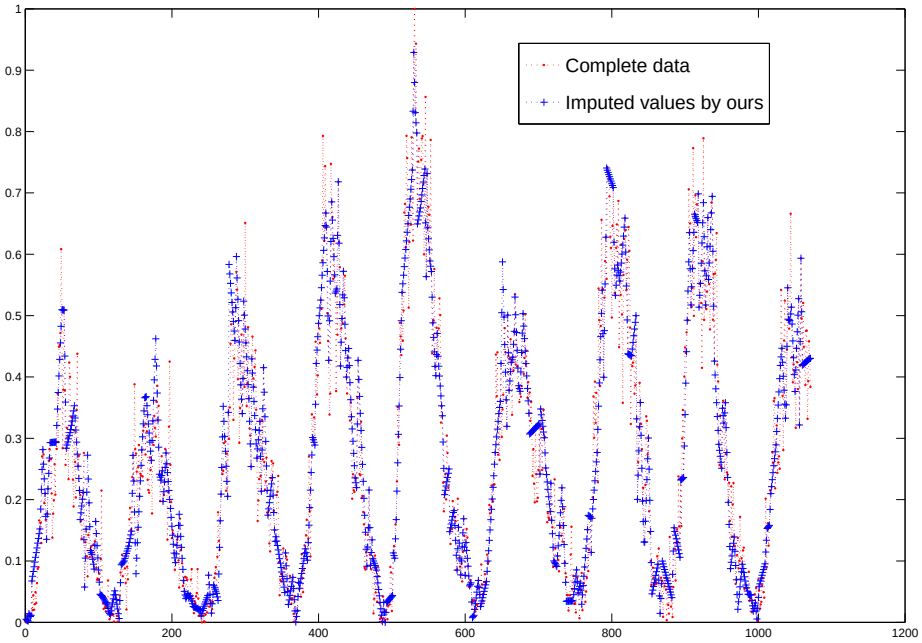


Figure 6.4: The actual data and imputed data by using the proposed algorithms in the sunspot data set with 70% of missing data.

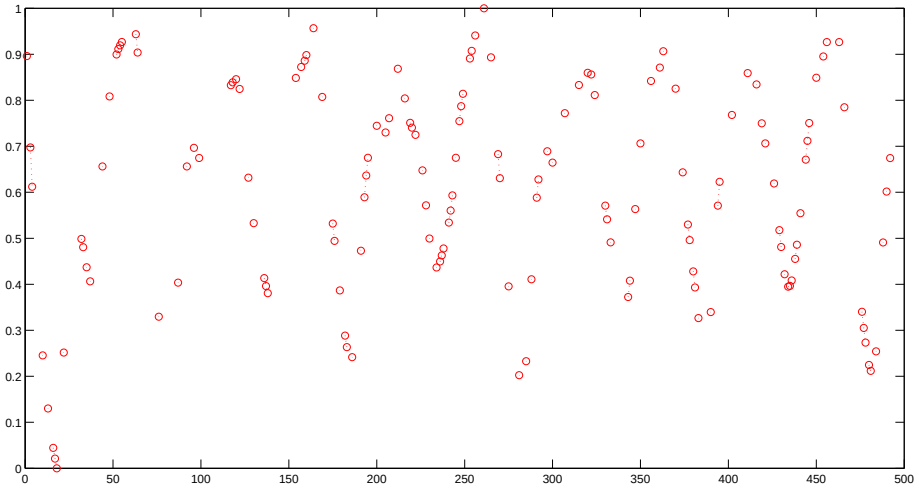


Figure 6.5: The first 500 data of Mackey Glass chaotic time series data set with 70% of missing data.

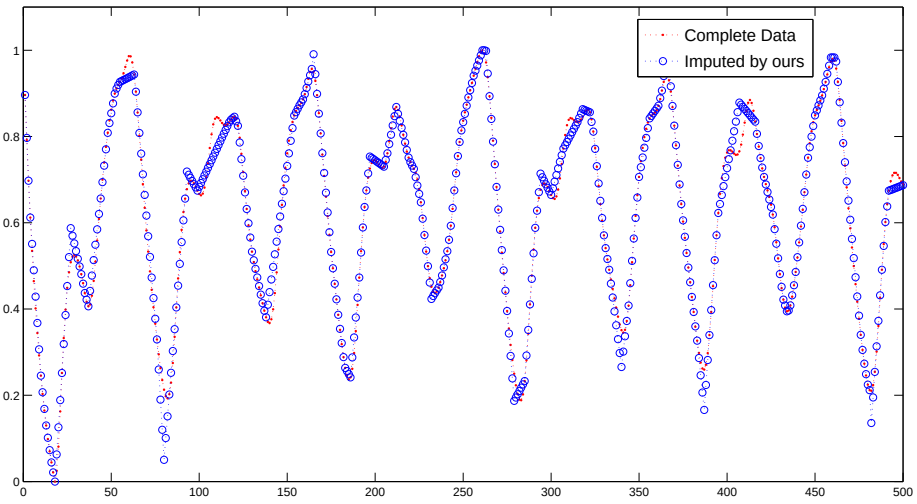


Figure 6.6: The actual data and imputed data by using the proposed algorithms in the first 500 data of Mackey Glass chaotic time series dataset with 70% of missing data.

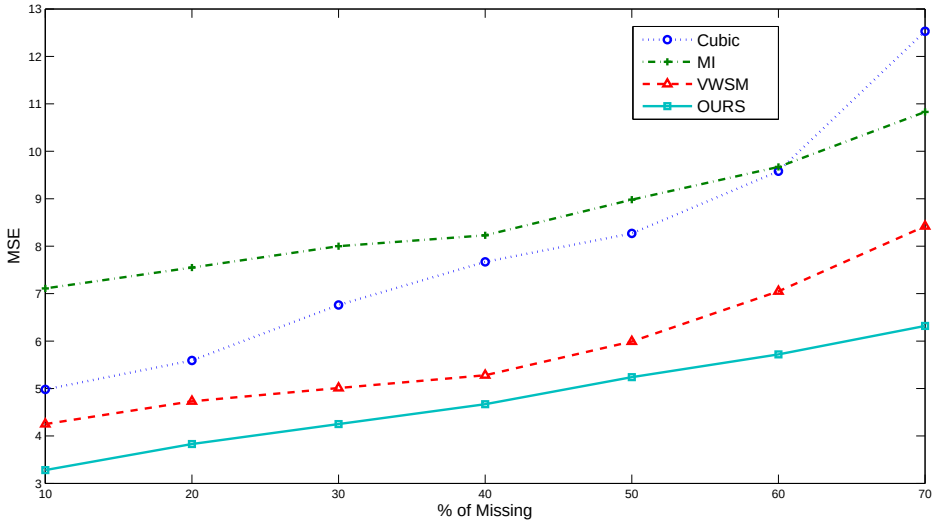


Figure 6.7: MSE comparison of each method with sunspot data set

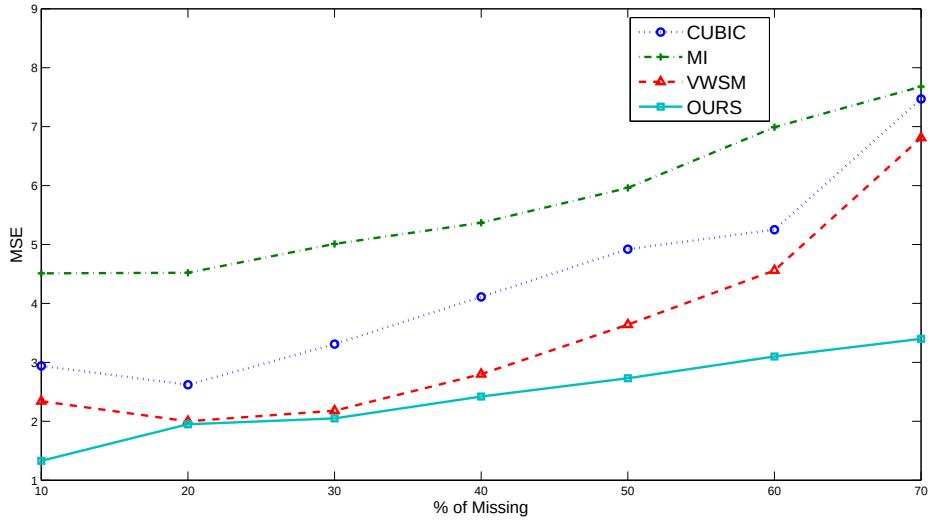


Figure 6.8: MSE comparison of each method with Gauge height time series data set.

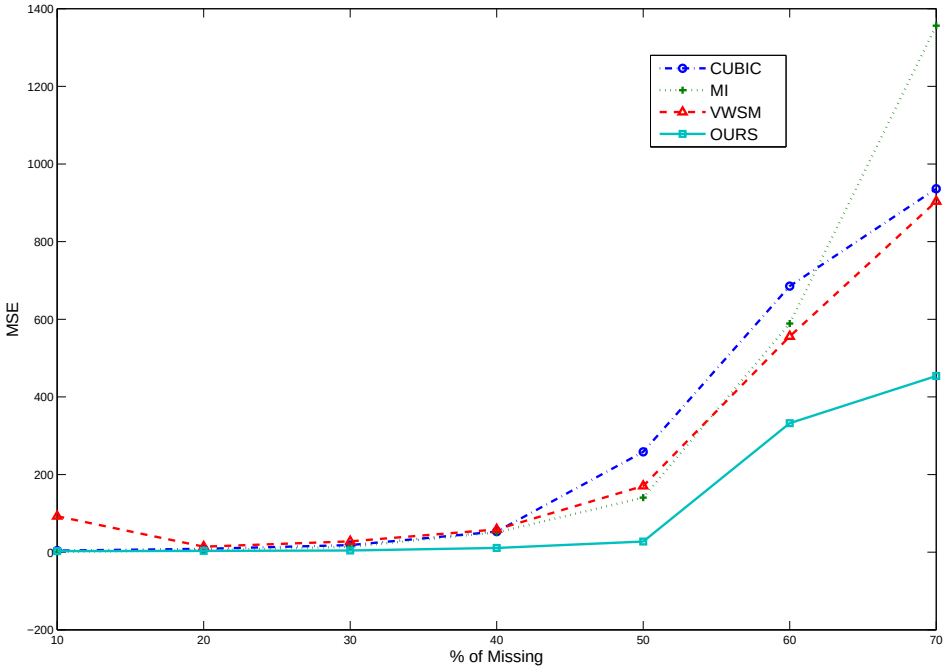


Figure 6.9: MSE comparison of each method with Mackey Glass chaotic time series data set.

Table 6.3: The running time in each time-series data by using cubic spline interpolation, MI interpolation method, VWSM algorithm, and the proposed algorithms denoted with RGGB.

Missing rate	Cubic	MI	VWSM	**RGGB**
Mackey Glass chaotic time series				
10%	**12**	15	600	20
20%	**17**	21	690	28
30%	**23**	28	780	35
40%	**27**	34	900	49
50%	**31**	40	1015	50
60%	**35**	45	1500	67
70%	**39**	51	1890	78
Gauge Height time series				
10%	**16**	18	670	25
20%	**21**	26	730	33
30%	**26**	34	790	38
40%	**34**	38	890	51
50%	**39**	45	985	60
60%	**48**	49	1400	67
70%	**54**	53	1780	85
Sunspot time series				
10%	**11**	14	570	15
20%	**14**	19	620	19
30%	**19**	24	780	27
40%	**25**	29	850	35
50%	**31**	34	900	40
60%	**39**	40	985	46
70%	**42**	46	1200	53

6.2.4 Discussions

The experimental result showed that, the proposed algorithms outperformed the other estimation methods in three test cases. The limitation of this algorithms is the occuring of fluctuating data patterns. If there are too many fluctuating patterns, then it is dificult to impute the missing data and acheive the high accuracy. One more problem is that if the consecutive sequence of missing data is long, then the acceptable accuracy of data may not be achieved because there may be some unpredictable mixture of positive and negative slopes. The other problem is that if there is a missing data having high values of left and right slopes with different signs, then it is rather hard to give the correct value of this data. These issues will be solved in the future.

6.2.5 Conclusions

In this dissertation, a new method based on the slopes of non-missing nearest neighbors of missing data and bootstrapping concept to impute those missing data was pro-

posed. The method was tested on three data sets: 1201 Mackey Glass chaotic time series data set, 1071 sun spot data set, and 2000 gauge height data set. The experimental results were compared with three competitors: Cubic interpolation, MI interpolation, and VWSM. The results showed that the by using the proposed algorithms, its outperformed those three methods in accuracy.

6.3 Image imputation

The following data, parameters, and comparison results were considered in the experiments. Tested data with different characteristics of damaged image were considered. In this dissertation, two characteristics of missing pixels were studied: randomly missing pixels and non-randomly missing pixels. In the randomly missing pixels, the missing pixels with different level of missing values were generated . The proposed algorithms based on the hybrid imputation with neural network and the similarity measurement in with different type of dataset were introduced. The data set used in the experiments composed of two groups. The first group of data set is standard image data sets which are Lena, Airfield, Airplane, Goldhills, Harbor, and Aerial. The second group of data set is the damaged images data sets.

6.3.1 Experimental Set-up

6.3.1.1 Selection of algorithms

Because the randomly missing pixels in an image are similar to the image which has noisy pixels. Thus, in this group of data set, the method which used the concept of noisy removal were used. The missing pixels as the noisy pixels were considered. According to the first group of data set, the following algorithms were considered to impute missing values.

1. Gaussian filter[16].
2. Soheil's Method[35].
3. Cellular Neural Network(CNN)[36].

In the second group of data set, the following algorithms were used to impute missing values in the damaged area.

1. Criminisi's algorithms[19].

Table 6.4: The comparison between the proposed method and the competitors.

Topics	the proposed method	Criminisri's algorithms	Huan's algorithms
1.Method for calculating order of missing pixels	Marching Method	Use isophote line. Use confidence and Data term	Marching Method
2.Window's size	Variable block of windows using global / local window	Fix size	Fix size
3.Direction of window	From Hough algorithms	-	-
4.Similarity measurement	Wald-Walfowitz test	SSD	SSD
5.Reference pixels	From the original given data	From the previously imputation	From the previously imputation
6.Color space	Gray scale/RGB	RGB	RGB

2. Huan's algorithms[18].

The reason for using these two algorithms for the comparison is that two algorithms are well-known method for imputing missing values in an image. These two algorithms give more accuracy compare to other method and the charateristics of them are similar to the proposed method in this dissertation. The comparison between the proposed method and the competitors are shown in Table 6.4.

6.3.1.2 Data set descriptions

All data sets used in the expeiments have the difference data density, number of data sets and characteristics of data set. Details of each data set are described as follows.

1. Standard image data sets. Six images were used: Lena, Airfield, Airplane, Goldhills, Harbor, and Aerial. The size of each image is 512×512 pixels with 8 bits standard gray scale values.

2. Damaged standard image data sets. Six damaged images were used: Lena image, the two circles image, the window image[18], the bunji jump image[19], Mural image and Monkey giant image.

6.3.1.3 Percentage of missing data

The percentage of missing data in randomly missing image was set to various levels as follows: 10%, 20%, 30%, 40%, 50%, 60%, 70% using Missing Completely At Random(MCAR) mechanism. Each percentage of missing data in each image was generated

5 times for checking the accuracy of imputed image.

6.3.1.4 Performance of algorithms

The performance evaluation of the proposed algorithms was compared with the other algorithms by using Mean Absolute Percentage Error(MAPE) and Peak Signal-to-Noise Ratio(PSNR). The MAPE compute from equation(6.6).

$$MAPE = \frac{100}{n} \times \sum_{i=1}^{n} |\frac{\hat{\mathbf{p}}_i - \mathbf{p}_i}{\mathbf{p}_i}| \tag{6.6}$$

where $\hat{\mathbf{p}}_i$ is an imputed incomplete multi-dimensional data, $\mathbf{p}_i$ is complete multi-dimensional data, n is the number of missing data. After that the accuracy of imputed image is calculated from equation(6.7),

$$Accuracy = 100 - MAPE \tag{6.7}$$

To evaluate the performance of the proposed imputation algorithms, PSNR (Peak Signal-to-Noise Ratio) were used as an indicator for measuring. The Peak Signal-to-Noise Ratio, PSNR, is a ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. PSNR is usually expressed in terms of the logarithmic decibel scale. A higher PSNR would normally indicate that the reconstruction is of higher quality. The performance of the method is evaluated in terms of both visual quality and the PSNR value of the restored images. The Peak Signal-to-Noise Ratio (PSNR) is defined by

$$PSNR = 10log_{10}\frac{255^2}{\frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(\hat{\mathbf{p}}_i - \mathbf{p}_i)^2} \tag{6.8}$$

where m and n are the width and height of the image, $\mathbf{p}_i$ and $\hat{\mathbf{p}}_i$ are the intensity values of the original image and of the restored image respectively. Typical values for the PSNR are between 30 and 50 dB, where higher is better.

6.3.2 Experimental results

The experimental results by using the proposed algorithms are presented in this section.

6.3.2.1 Randomly missing image reconstruction with Standard image data sets

The first group of data sets are standard image data sets: Lena, Airfield, Airplane, Goldhills, Harbor, and Aerial. Fig. 6.10 - Fig. 6.13 shows image imputation results by using the proposed method and competitive method with standard image data sets. The original images are shown in Fig. 6.10(a)-Fig. 6.13(a). The damaged images with 50% of missing values are shown in Fig. 6.10(b)-Fig. 6.13(b). The imputation of damaged images with the proposed algorithms are shown in Fig. 6.10(c)-Fig. 6.13(c). The reconstructed images by using the CNN algorithms are shown in Fig. 6.10(d)-Fig. 6.13(d). The reconstructed images by using the Gaussians filter algorithms are shown in Fig. 6.10(e)-Fig. 6.13(e). The reconstructed images by using the Soheil's algorithms are shown in Fig. 6.10(f)-Fig. 6.13(f). The values of PSNR in each image is shown in Table 6.5. From the experimental results, the proposed method outperformed the competitive methods. The reason for this accuracy comes from the fact that restoring the randomly damaged area, the localized data which set-up by the target mask window was used. The assumption for using localized data is the more data which nearest to missing pixels will have the same characteristics to missing pixels. So, the most nearest observed values to the missing pixel was used to impute. In the proposed method neural network was used to train the pixels in the observed area for becoming the pixels for imputing missing area. Moreover after imputing by neural network, the similarity measurement will use again for giving the highest accuracy to the imputed pixels. The experimental results showed that the proposed method significantly outperformed the Soheil's method, CNN, and Gaussian as shown in the Table 6.5.

6.3.2.2 Imputation as the object removal

In case of removing an object in an image, the experimental results are shown as follows.

The comparison of the imputed View image between the competitive algorithms and the proposed algorithm are shown in Fig. 6.20. The original image, the missing image, the imputed image with proposed algorithms, the imputed image with Criminisri algorithms, the imputed image wite Huan algorithms are shown in Fig. 6.20(a) - Fig. 6.20(e).
To remove the object and to restore the structure of blackground in this image which is a couple of human, the proposed algorithms can be used. The difficulty of imputing the damaged area in this picture is that area has a horizontal line. If use the traditional method in that area, the structure of this horizontal line can not detect. However, by

Figure 6.10: The interpolation of missing data in Lena image. (a) Original image (b) Damaged image with 50% of missing values (c) Reconstructed image with our proposed algorithms (d) Reconstructed image using the CNN. (e) Reconstructed image using the Guassian filter (f) Reconstructed image using Soheil's algorithms.

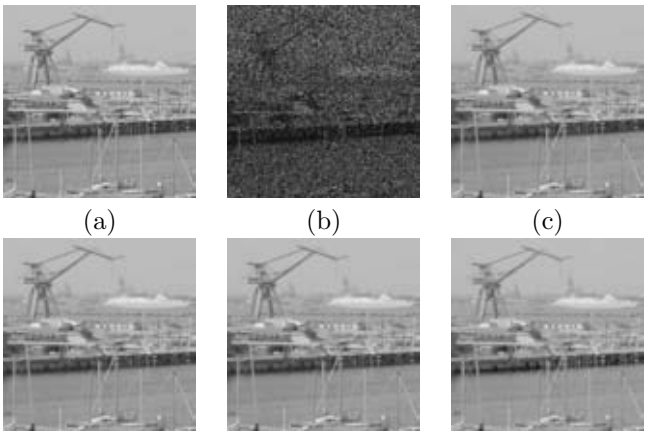
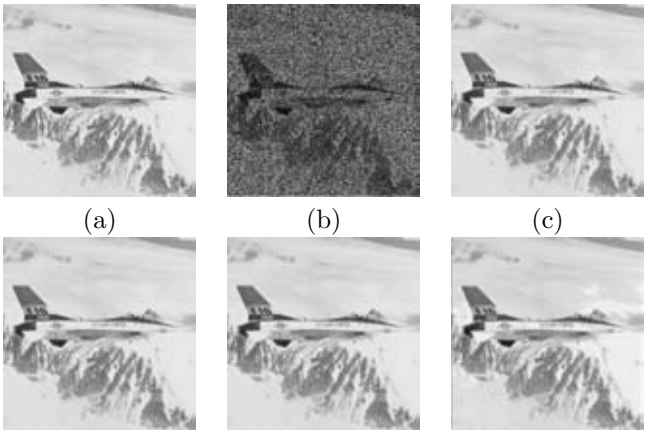


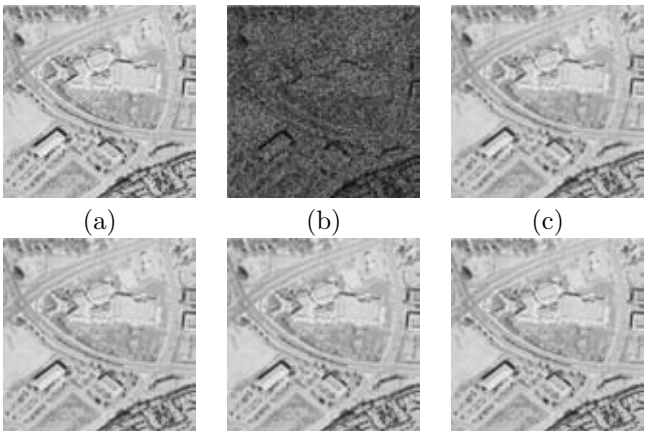
Figure 6.11: The interpolation of missing data in Harbor image. (a) Original image (b) Damaged image with 50% of missing values (c) Reconstructed image with our proposed algorithms (d) Reconstructed image using the CNN. (e) Reconstructed image using the Guassian filter (f) Reconstructed image using Soheil's algorithms.



(a) (b) (c)

(d) (e) (f)

Figure 6.12: The interpolation of missing data in airplane image. (a) Original image (b) Damaged image with 50% of missing values (c) Reconstructed image with our proposed algorithms (d) Reconstructed image using the CNN. (e) Reconstructed image using the Guassian filter (f) Reconstructed image using Soheil's algorithms.



(a) (b) (c)

(d) (e) (f)

Figure 6.13: The interpolation of missing data in Aerial image. (a) Original image (b) Damaged image with 50% of missing values (c) Reconstructed image with our proposed algorithms (d) Reconstructed image using the CNN. (e) Reconstructed image using the Guassian filter (f) Reconstructed image using Soheil's algorithms.

Table 6.5: The average PSNR values of the reconstructed images.

Dataset/Missing Rate	10%	20%	30%	40%	50%	60%	70%
Lena							
Proposed Method	**37.99**	**34.8**	**32.41**	**31.22**	28.13	25.87	**24.19**
Soheil's Method	34.13	32.24	29.37	27.43	25.24	24.21	20.81
CNN	35.89	33.60	31.55	29.75	**28.54**	**26.16**	23.14
Gaussian filter	33.03	30.62	29.61	28.19	25.70	22.81	21.57
Airfield							
Proposed Method	**32.00**	**30.24**	**28.68**	26.87	25.34	**23.84**	**21.71**
Soheil's Method	29.91	28.67	27.44	26.22	**25.78**	22.88	19.68
CNN	29.21	27.4	25.33	24.56	24.96	23.12	19.46
Gaussian filter	29.78	28.76	28.2	26.47	24.78	22.65	20.67
Airplane							
Proposed Method	**34.27**	**32.61**	**31.57**	**30.94**	**28.78**	**25.9**	**23.9**
Soheil's Method	32.44	31.3	28.91	27.82	24.93	23.33	20.51
CNN	31.69	30.55	27.88	27.37	26.87	24.58	21.95
Gaussian filter	31.51	30.21	29.05	28.28	25.55	23.66	21.67
Goldhills							
Proposed Method	**37.16**	**34.64**	**33.26**	**31.4**	**29.74**	**26.91**	**26.43**
Soheil's Method	34.5	32.56	30.72	29.59	26.51	24.74	24.57
CNN	33.57	31.33	29.56	28.57	27.74	25.51	25.26
Gaussian filter	30.01	28.8	26.79	26.6	24.59	22.66	22.09
Harbor							
Proposed Method	**30.005**	**29.285**	**28.53**	**27.53**	**26.405**	24.655	22.65
Soheil's Method	29.655	28.23	26.84	25.94	25.015	**24.7**	**23.89**
CNN	26.74	26.26	24.985	24.55	23.825	22.505	22.96
Gaussian filter	25.345	26.21	24.64	25.62	22.305	23.15	23.07
Aerial							
Proposed Method	**35.59**	**34.31**	**32.58**	**31.62**	**30.35**	**26.81**	**24.02**
Soheil's Method	32.115	32.09	26.935	27.875	26.455	25.145	22.97
CNN	30.11	28.74	27.465	26.84	25.605	24.405	21.535
Gaussian filter	29.87	28.33	27.255	25.5	25.455	23.215	21.27

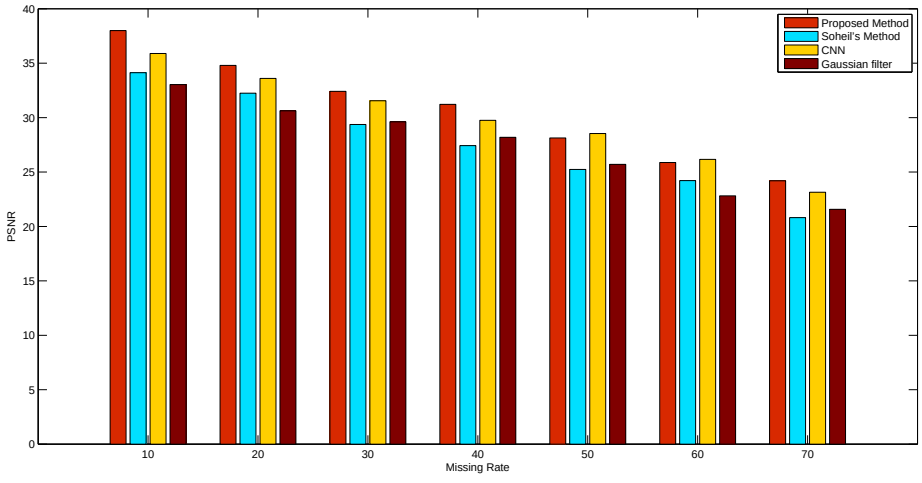


Figure 6.14: The PSNR comparison of each method with Lena data set.

Table 6.6: The accuracy of reconstructed images.

Dataset/Missing Rate	10%	20%	30%	40%	50%	60%	70%
Lena							
Proposed Method	**90.95**	**90.19**	**89.62**	**89.34**	88.60	88.06	**87.66**
Soheil's Method	90.03	89.58	88.90	88.43	87.91	87.67	86.86
CNN	90.45	89.90	89.42	88.99	**88.70**	**88.13**	87.41
Gaussian filter	89.77	89.19	88.95	88.62	88.02	87.33	87.04
Airfield							
Proposed Method	**89.52**	**89.10**	**88.73**	**88.30**	87.94	**87.58**	**87.07**
Soheil's Method	89.03	88.73	88.44	88.15	**88.04**	87.35	86.59
CNN	88.86	88.43	87.93	87.75	87.85	87.41	86.54
Gaussian filter	88.99	88.75	88.62	88.21	87.80	87.30	86.82
Airplane							
Proposed Method	**90.06**	**89.67**	**89.42**	**89.27**	**88.76**	**88.07**	**87.59**
Soheil's Method	89.63	89.36	88.79	88.53	87.84	87.46	86.79
CNN	89.45	89.18	88.54	88.42	88.30	87.76	87.13
Gaussian filter	89.41	89.10	88.82	88.64	87.99	87.54	87.06
Goldhills							
Proposed Method	**90.75**	**90.15**	**89.82**	**89.38**	**88.98**	**88.31**	**88.20**
Soheil's Method	90.12	89.66	89.22	88.95	88.22	87.79	87.75
CNN	89.90	89.36	88.94	88.71	88.51	87.98	87.92
Gaussian filter	89.05	88.76	88.28	88.24	87.76	87.30	87.16
Harbor							
Proposed Method	**89.05**	**88.88**	**88.70**	**88.46**	**88.19**	87.77	87.30
Soheil's Method	88.96	88.62	88.29	88.08	87.86	**87.78**	**87.59**
CNN	88.27	88.16	87.85	87.75	87.58	87.26	87.37
Gaussian filter	87.94	88.14	87.77	88.00	87.21	87.41	87.40
Aerial							
Proposed Method	**90.38**	**90.07**	**89.66**	**89.43**	**89.13**	**88.29**	**87.62**
Soheil's Method	89.55	89.54	88.32	88.54	88.20	87.89	87.37
CNN	89.07	88.75	88.44	88.29	88.00	87.71	87.03
Gaussian filter	89.02	88.65	88.39	87.97	87.96	87.43	86.97

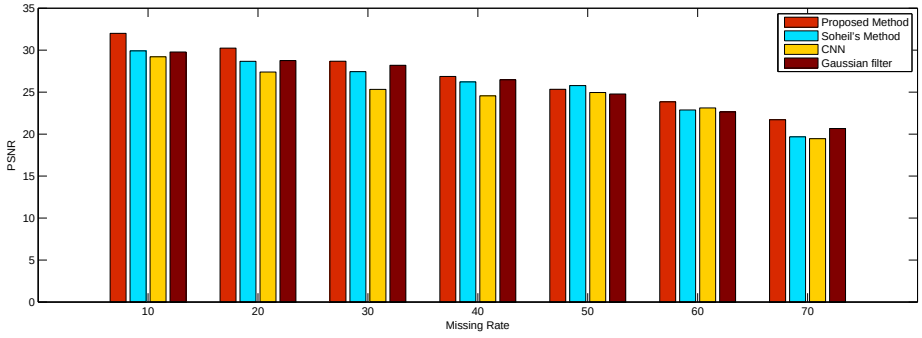


Figure 6.15: The PSNR comparison of each method with airfield data set.

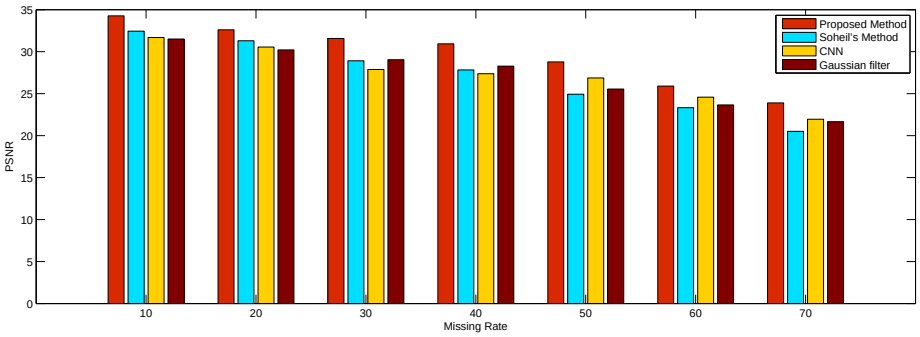


Figure 6.16: The PSNR comparison of each method with airplane data set.

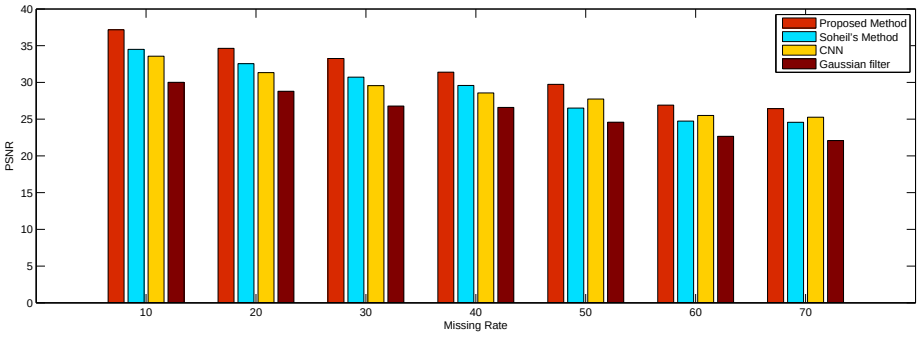


Figure 6.17: The PSNR comparison of each method with goldhills data set.

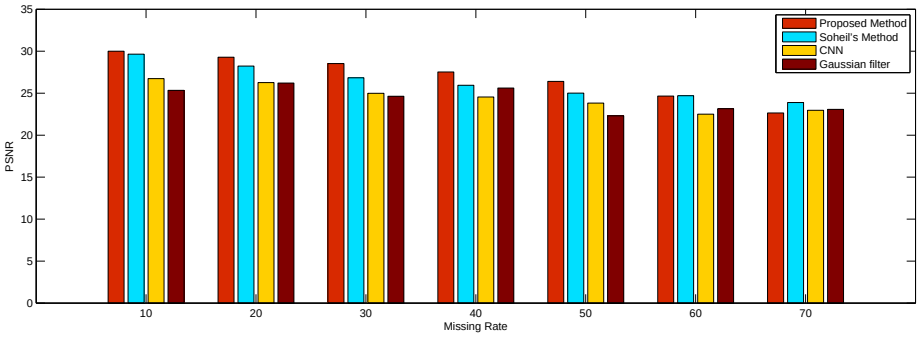


Figure 6.18: The PSNR comparison of each method with harbor data set.

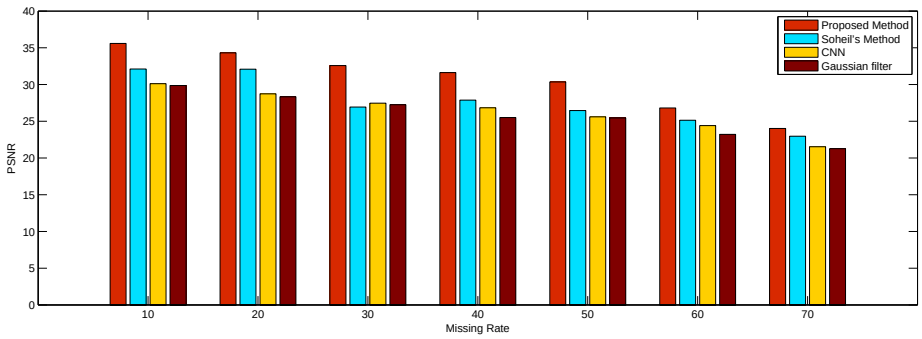
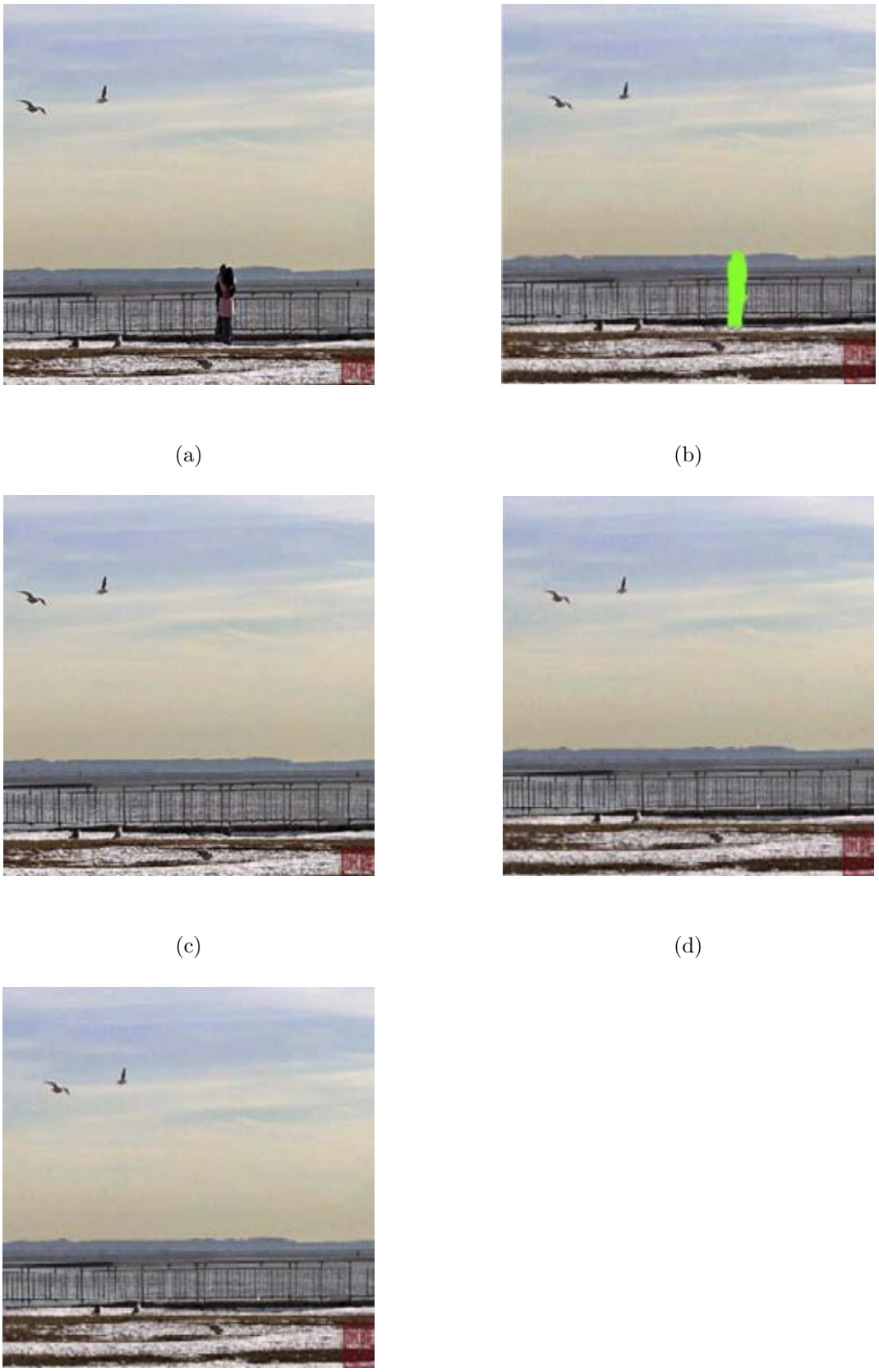


Figure 6.19: The PSNR comparison of each method with aerial data set.

using the proposed algorithm, the structure of overall image can be preserved. By using the global imputation algorithm, the mask window template can be set for comparing between the damaged area and the other area. The experimental results showed that the proposed algorithm can remove an object in a desired area and can be restored the structure of the blackground. Moreover, the proposed algorithms can give a better visual quality than the traditional method as shown in Fig. 6.20(c) - Fig. 6.20(e).

The comparison of imputed Bunji jump image between the competitive algorithms and the proposed algorithms are shown in Fig. 6.21. The original image, the missing image, the imputed image with proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms are shown in Fig. 6.21(a) - Fig. 6.21(e). In this image which is a man who is playing a bunji jump, to remove an object and to restore the structure of blackground, the proposed algorithms can be used by using the local imputation and global imputation algorithms depending on the characteristics of the damaged area. For example to remove a man's legs, the global imputation algorithms can be used because mean square error in this area and the observed area is less than a pre-defined threshold. So, the patch that most similar to the missing area can be used. However, in some area such as the middle of body can not be used with global imputation because while using the global algorithm in that area they do not have a similar area which less than the predefind threshold. Accordingly, in this area the local imputation must used. From the experimental results, the proposed algorithm can remove an object in an desired area and can detect the structure of the blackground. Moreover, the proposed algorithms can give a better visual quality than the traditional method as shown in Fig. 6.21(c) - Fig. 6.21(e).



(a) (b)

(c) (d)

(e)

Figure 6.20: The comparison of imputed View image between the competitive algorithms and proposed algorithms. (a) The original image (b) The missing image (c) Imputed image with proposed algorithms (d) Imputed image with Criminisri algorithms (e) Imputed image with Huan algorithms

(a)

(b)

(c)

(d)

(e)

Figure 6.21: The comparison of imputed Bunji jump image between the competitive algorithms and proposed algorithms. (a) The original image (b) The missing image (c) Imputed image with proposed algorithms (d) Imputed image with Criminisri algorithms (e) Imputed image with Huan algorithms

6.3.2.3 Imputation as the image restoration

The comparison of the imputed two-circles image between the competitive algorithms and tphe roposed algorithms are shown in Fig. 6.22. The missing image, the imputed image with proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms are shown in Fig. 6.22(a) - Fig. 6.22(d). The experimental results showed that using global imputation algorithms which using both two sides of the observed area of the missing area for imputing, the line of circle can be restored. In this image, the order of missing pixels to be imputed were calculated. The direction of border is used.

The comparison of the imputed Windows image between the competitive algorithms and the proposed algorithms are shown in Fig. 6.23. In this image, the direction of image was considered. The global imputation method can be used by considering the direction of image for giving the high accuracy after imputing the damaged area. However, not only use global imputation method but also use local imputation method as well. The local imputation method was used for imputing some position that global imputation can not be applied due to the mean square error is greater than the predefine threshold. The missing image, the imputed image with the proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms are shown in Fig. 6.23(a) - Fig. 6.23(d). The difficulty of imputation in this image is that how tp restore the shape inside each missing area. If the conventional method was used, then the pattern inside missing area can not be captured as shown in Fig. 6.23. In the other hand, by using the proposed method which uses two sides of area between the missing area for comparing the similarity between the observed area and the missing area, this technique can restore the complex structure inside an image.

The comparison of imputed Brick image between the competitive algorithms and proposed algorithms are shown in Fig. 6.24. The original image, the missing image, the imputed image with the proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms were shown in Fig. 6.24(a) - Fig. 6.24(d). In this image the direction of image was considered for giving the high accuracy of imputed image.

The comparison of imputed Lena image between the competitive algorithms and the proposed algorithm are shown in Fig. 6.25 to Fig. 6.28. In this image, four missing patterns were considered and tested. The reason for using this four patterns is that there

are different shape and different missing position. These four patterns were generated for testing the performance of algorithms to restore the shape of original image. In the original image with the missing image of pattern 1, the imputed image with the proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms are shown in Fig. 6.25(a) - Fig. 6.25(e). In the original image with the missing image of pattern 2, the imputed image with the proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms are shown in Fig. 6.26(a) - Fig. 6.26(e). In the original image with the missing image of pattern 3, the imputed image with the proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms are shown in Fig. 6.27(a) - Fig. 6.27(e). In the original image with the missing image of pattern 4, the imputed image with the proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms are shown in Fig. 6.28(a) - Fig. 6.28(e). The experimental results showed that when impute the missing area in an image, for restoring the shape of original image, the order of missing pixels to be imputed is considered. The experimental results showed that the proposed algorithm can give a high accuracy compare to other competitive method. The proposed algorithms can restore the surface structure as shown in Fig. 6.25 to Fig. 6.28.

The comparison of the imputed Mural image between the competitive algorithms and proposed algorithms are shown in Fig. 6.29 to Fig. 6.32. This image is an example of the complex image which has many texture inside their image. In this image, four missing patterns were considered and tested. The missing image with different shape and different missing position were generated. The experimental result showed that the proposed algorithm can give a high accuracy compare to other competitive method. The proposed algorithms can restore the surface structure as shown in Fig. 6.29 to Fig. 6.32. Details of restoration image are shown as follows. In the original image with the missing image of pattern 1, the imputed image with the proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms are shown in Fig. 6.29(a) - Fig. 6.29(e). In the original image with the missing image of pattern 2, the imputed image with the proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms are shown in Fig. 6.30(a) - Fig. 6.30(e). In the original image with the missing image of pattern 3, the imputed image with the proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms are shown in Fig. 6.31(a) - Fig. 6.31(e). In the original image with the missing image

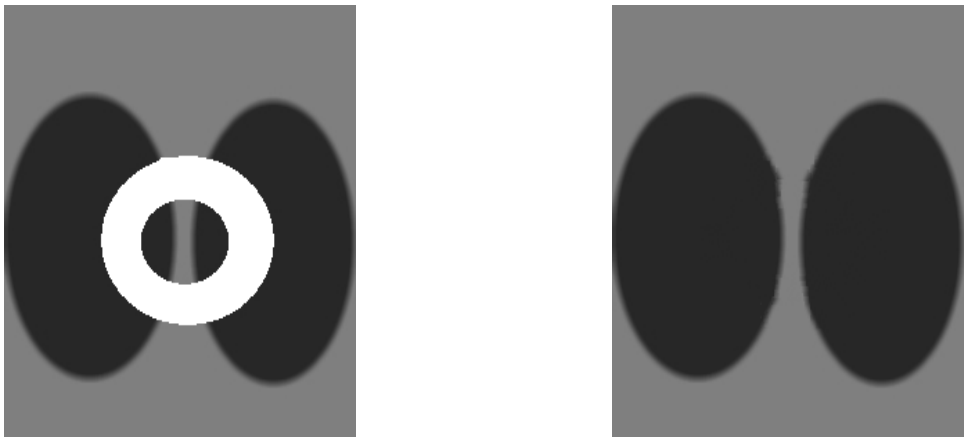
of pattern 4, the imputed image with the proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms are shown in Fig. 6.32(a) - Fig. 6.32(e). In this image, the order of missing pixels were considered before impute the missing area. This process can preserved the complex structure in an image.

The comparison of the imputed Giant image between the competitive algorithms and proposed algorithms are shown in Fig. 6.33 to Fig. 6.36. In this image, four missing patterns were considered and tested. This image is another example of complex image which has missing values. The experimental results shown that to impute the missing area in an image, for preserving the shape of original image, the order of pixels to be imputed is importance. The experimental result showed that the proposed algorithm can give a high accuracy compare to other competitive method. The proposed algorithms can restore the surface structure as shown in Fig. 6.33 to Fig. 6.36. In the original image with the missing image of pattern 1, the imputed image with the proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms are shown in Fig. 6.33(a) - Fig. 6.33(e). In the original image with the missing image of pattern 2, the imputed image with the proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms are shown in Fig. 6.34(a) - Fig. 6.34(e). In the original image with the missing image of pattern 3, the imputed image with the proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms are shown in Fig. 6.35(a) - Fig. 6.35(e). In the original image with the missing image of pattern 4, the imputed image with the proposed algorithms, the imputed image with Criminisri algorithms, the imputed image with Huan algorithms are shown in Fig. 6.36(a) - Fig. 6.36(e).

The PSNR in each image pattern is shown in Table 6.7. The accuracy of the restored image in each pattern is shown in Table 6.10. The experimental results showed that the proposed algorithms give a high PSNR in serveral case of missing patterns and serveral images.

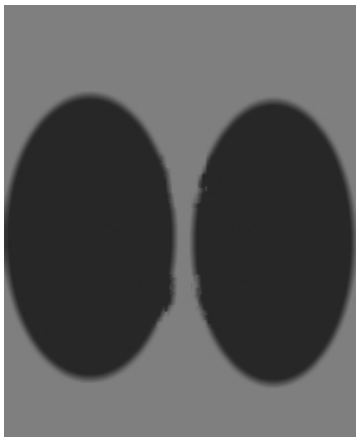
6.3.3 Time complexity

The time complexity of hybrid imputation algorithms composed of three parts depending on the shape of missing area in each image to be imputed. The proposed algorithms have two strategies that are: the first strategy used neural network and similarity measurement. The second strategy used the similarity measurement which is global imputation and local imputation.

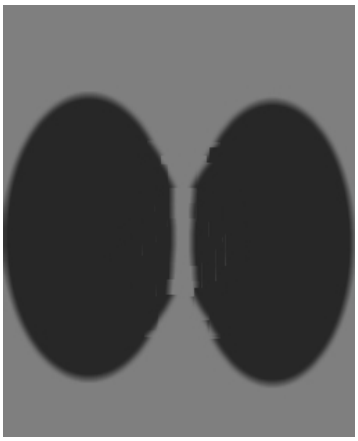


(a)

(b)

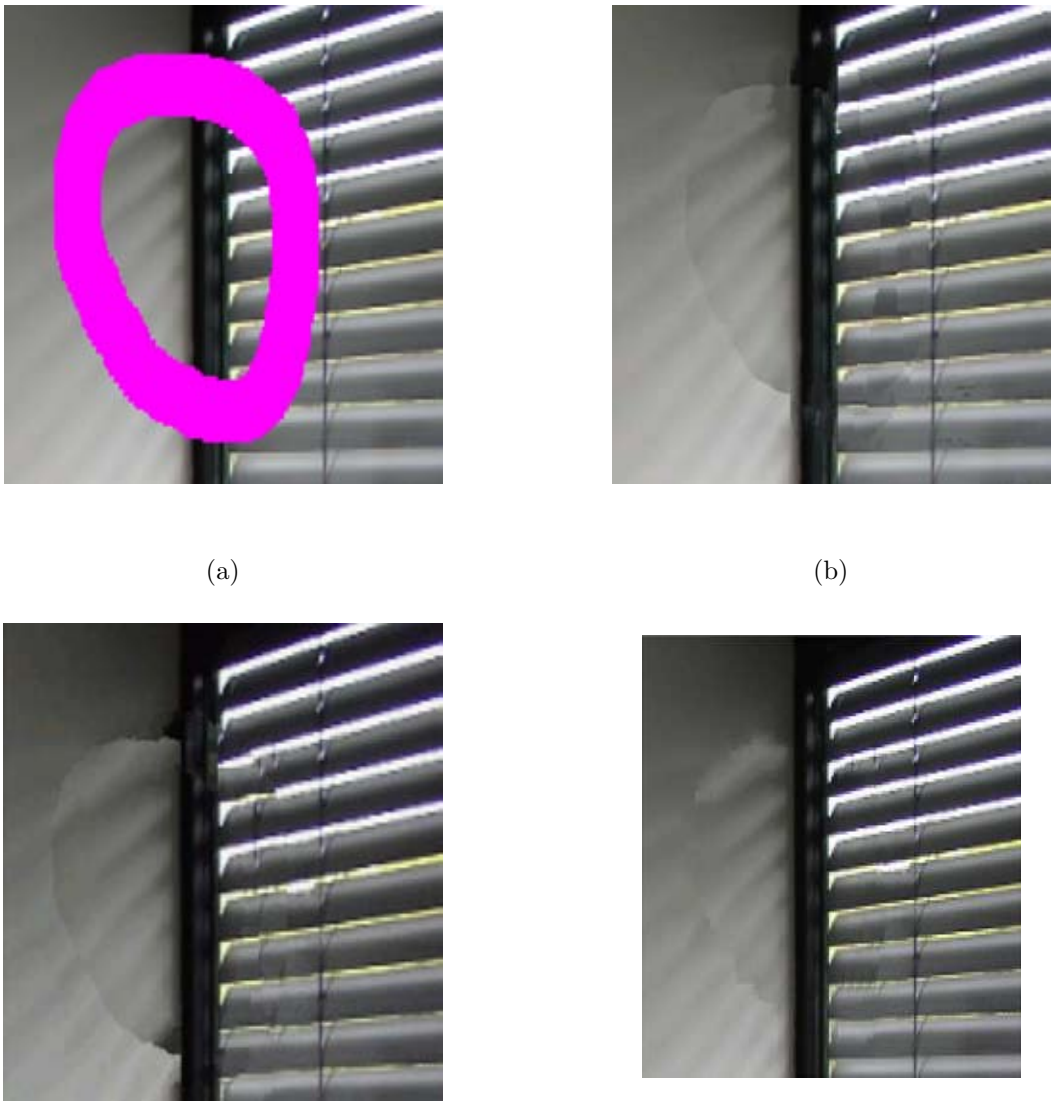


(c)



(d)

Figure 6.22: The comparison of imputed Two circles image between the competitive algorithms and proposed algorithms. (a) The missing image (b) Imputed image with proposed algorithms (c) Imputed image with Criminisri algorithms (d) Imputed image with Huan algorithms



(a) (b)

(c) (d)

Figure 6.23: The comparison of imputed Window image between the competitive algorithms and proposed algorithms. (a) The missing image (b) Imputed image with proposed algorithms (c) Imputed image with Criminisri algorithms (d) Imputed image with Huan algorithms

(a)

(b)

(c)

(d)

Figure 6.24: The comparison of imputed Brick image between the competitive algorithms and proposed algorithms. (a) The missing image (b) Imputed image with proposed algorithms (c) Imputed image with Criminisri algorithms (d) Imputed image with Huan algorithms

(a)

(b)

(c)

(d)

(e)

Figure 6.25: The comparison of imputed Lena image between the competitive algorithms and proposed algorithms. (a) The original image (b) The missing image(pattern 1) (c) Imputed image with proposed algorithms (d) Imputed image with Criminisri algorithms (e) Imputed image with Huan algorithms

(a)

(b)

(c)

(d)

(e)

Figure 6.26: The comparison of imputed Lena image between the competitive algorithms and proposed algorithms. (a) The original image (b) The missing image(pattern 2) (c) Imputed image with proposed algorithms (d) Imputed image with Criminisri algorithms (e) Imputed image with Huan algorithms

(a)

(b)

(c)

(d)

(e)

Figure 6.27: The comparison of imputed Lena image between the competitive algorithms and proposed algorithms. (a) The original image (b) The missing image(pattern 3) (c) Imputed image with proposed algorithms (d) Imputed image with Criminisri algorithms (e) Imputed image with Huan algorithms

(a)

(b)

(c)

(d)

(e)

Figure 6.28: The comparison of imputed Lena image between the competitive algorithms and proposed algorithms. (a) The original image (b) The missing image(pattern 4) (c) Imputed image with proposed algorithms (d) Imputed image with Criminisri algorithms (e) Imputed image with Huan algorithms

(a)

(b)

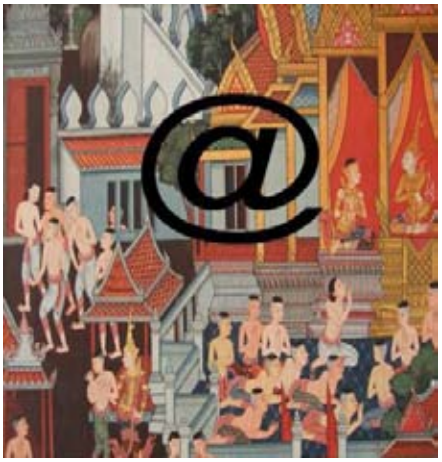
(c)

(d)

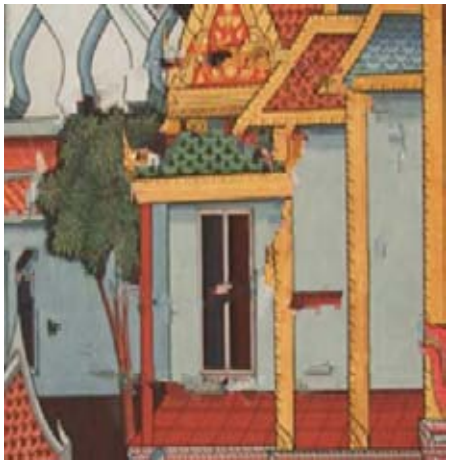
(e)

Figure 6.29: The comparison of imputed Mural image between the competitive algorithms and proposed algorithms. (a) The original image (b) The missing image (pattern 1) (c) Imputed image with proposed algorithms (d) Imputed image with Criminisri algorithms (e) Imputed image with Huan algorithms

(a)



(b)



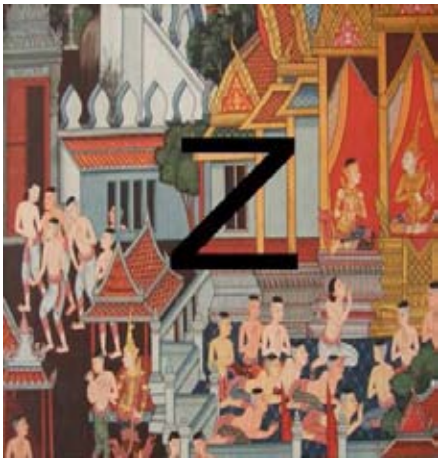
(c)

(d)

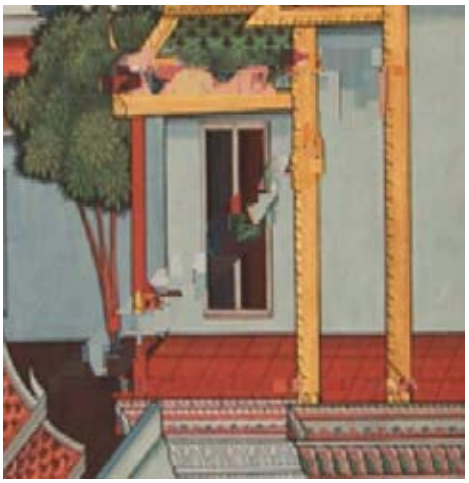
(e)

Figure 6.30: The comparison of imputed Mural image between the competitive algorithms and proposed algorithms. (a) The original image (b) The missing image (pattern 2) (c) Imputed image with proposed algorithms (d) Imputed image with Criminisri algorithms (e) Imputed image with Huan algorithms

(a)



(b)



(c)

(d)

(e)

Figure 6.31: The comparison of imputed Mural image between the competitive algorithms and proposed algorithms. (a) The original image (b) The missing image (pattern 3) (c) Imputed image with proposed algorithms (d) Imputed image with Criminisri algorithms (e) Imputed image with Huan algorithms

(a)

(b)

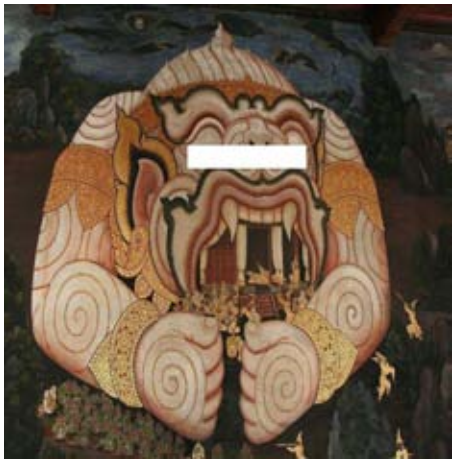
(c)

(d)

(e)

Figure 6.32: The comparison of imputed Mural image between the competitive algorithms and proposed algorithms. (a) The original image (b) The missing image (pattern 4) (c) Imputed image with proposed algorithms (d) Imputed image with Criminisri algorithms (e) Imputed image with Huan algorithms

(a)



(b)

(c)

(d)

(e)

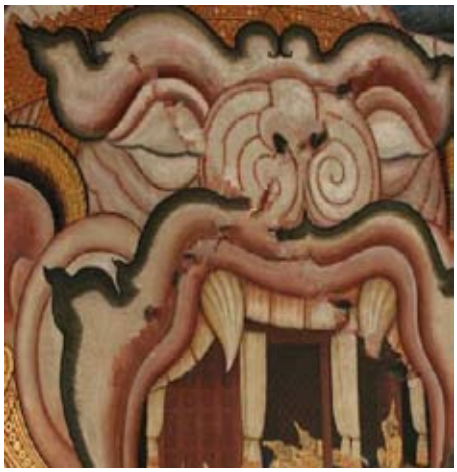
Figure 6.33: The comparison of imputed Giant image between the competitive algorithms and proposed algorithms. (a) The original image (b) The missing image(pattern 1) (c) Imputed image with proposed algorithms (d) Imputed image with Criminisri algorithms (e) Imputed image with Huan algorithms

(a)

(b)

(c)

(d)

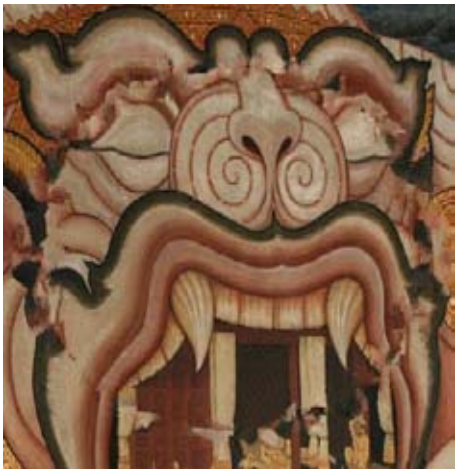


(e)

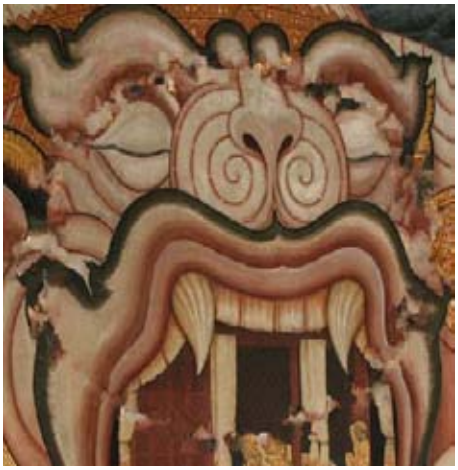
Figure 6.34: The comparison of imputed Giant image between the competitive algorithms and proposed algorithms. (a) The original image (b) The missing image(pattern 2) (c) Imputed image with proposed algorithms (d) Imputed image with Criminisri algorithms (e) Imputed image with Huan algorithms

(a)

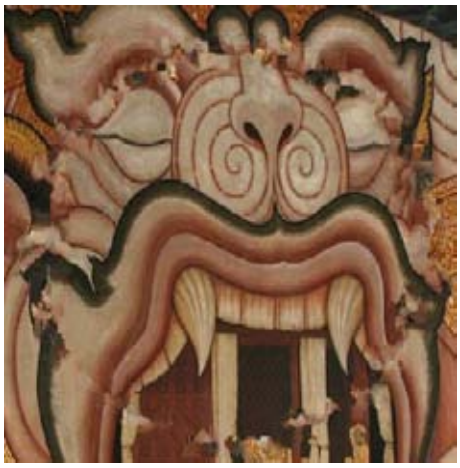
(b)



(c)



(d)

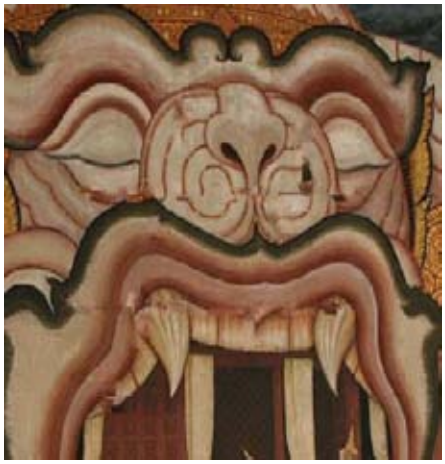


(e)

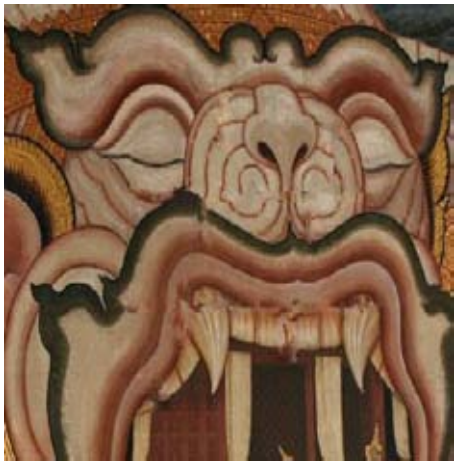
Figure 6.35: The comparison of imputed Giant image between the competitive algorithms and proposed algorithms. (a) The original image (b) The missing image(pattern 3) (c) Imputed image with proposed algorithms (d) Imputed image with Criminisri algorithms (e) Imputed image with Huan algorithms

(a)

(b)



(c)



(d)

(e)

Figure 6.36: The comparison of imputed Giant image between the competitive algorithms and proposed algorithms. (a) The original image (b) The missing image(pattern 4) (c) Imputed image with proposed algorithms (d) Imputed image with Criminisri algorithms (e) Imputed image with Huan algorithms

Table 6.7: The PSNR in each image pattern.

Dataset/Method	Type	Proposed algorithms	Criminisri	Huan
Lena	Image restoration			
- pattern 1		**41.75**	40.13	38.55
- pattern 2		**53.48**	49.28	49.65
- pattern 3		**32.74**	30.74	30.27
- pattern 4		**31.46**	26.76	27.06
Mural	Image restoration			
- pattern 1		**30.47**	29.05	28.51
- pattern 2		**30.32**	27.89	27.97
- pattern 3		**31.23**	29.64	29.13
- pattern 4		**27.61**	25.09	24.97
Giant	Image restoration			
- pattern 1		**28.93**	28.34	25.29
- pattern 2		**34.41**	32.98	30.17
- pattern 3		**29.75**	27.57	26.34
- pattern 4		**37.53**	34.98	34.13
Two circles	Image restoration	**37.47**	33.97	32.43
Brick	Image restoration	**30.17**	29.96	29.34
Windows	Image restoration	N/A	N/A	N/A
View	Object removal	N/A	N/A	N/A
Bunji-jump	Object removal	N/A	N/A	N/A

Table 6.8: The accuracy of restored images.

Dataset/Method	Type	Proposed algorithms	Criminisri	Huan
Lena	Image restoration			
- pattern 1		**92.62**	89.54	88.49
- pattern 2		**98.69**	96.14	95.90
- pattern 3		**98.45**	96.67	94.90
- pattern 4		**95.75**	90.25	89.68
Mural	Image restoration			
- pattern 1		**95.26**	93.72	92.14
- pattern 2		**95.87**	91.45	92.30
- pattern 3		**96.84**	95.43	94.26
- pattern 4		**93.27**	90.12	90.08
Giant	Image restoration			
- pattern 1		**93.14**	91.12	90.24
- pattern 2		**97.62**	96.54	95.18
- pattern 3		**94.21**	90.19	89.54
- pattern 4		**98.64**	95.21	94.85
Two circles	Image restoration	**96.37**	95.12	94.58
Brick	Image restoration	**92.11**	89.25	87.63
Windows	Image restoration	N/A	N/A	N/A
View	Object removal	N/A	N/A	N/A
Bunji-jump	Object removal	N/A	N/A	N/A

In the first strategy: the time complexity for the variable's definition in lines 1-5 of ALGORITHMS 4 is $O(1)$ and the time for training the observed and unobserved data in lines 6-17 of ALGORITHMS 4 is $O(n)$. The process to compare the similarity of two windows is $O(mn)$ where m is number of rows and n is number of columns. If m equals n time complexity is $O(n^2)$. Thus, in this strategy the time complexity is

$$T_{alg} = O(1) + O(n) + O(mn) = O(n^2) \tag{6.9}$$

In the second strategy, time complexity can be described as follows,

- time for definding variable in line 1 of ALGORITHM 5 is $O(1)$.
- time for calling marching method in line 2 of ALGORITHM 5 is $O(n)$.
- time for calculating azimute angle of sub image in line 3 of ALGORITHM 5 is $O(n)$.
- time for definding variable in lines 4-7 of ALGORITHM 5 is $O(1)$.
- time for calculating the mark template window in lines 8-10 of ALGORITHM 5 is $O(n)$.
- time for comparing the similarity between two sub-image in lines 11-15 of ALGORITHM 5 depending on calling ALGORITHM 7 which is $O(n^2)$.
- time for imputing missing value by local imputation algorithms in line 17 of ALGORITHM 5 depending on calling ALGORITHM 6 which is $O(n^2)$.

So, in this strategy time complexity is

$$T_{alg} = O(1) + O(n) + O(n) + O(1) + O(n) + O(n^2) + O(n^2) = O(n^2) \tag{6.10}$$

The actual running time tested with three types of missing data: randomly missing data, image restoration, object removal are shown in Table 6.9 and Table 6.10.

6.3.4 Disscussions

The experimental results showed that, in the first group of data set, the proposed algorithms can impute the missing pixels and can reconstruct the damaged image. The

Table 6.9: The actual running time($\times 10^3$second) of reconstructed images.

Dataset/Missing Rate	10%	20%	30%	40%	50%	60%	70%
Lena							
Proposed Method	8.60	17.50	26.8	39.7	47.5	54.9	65.1
Soheil's Method	2.8	4.9	7.3	9.2	11.7	14.5	16.9
CNN	3.3	5.3	7.5	10.5	12.3	14.8	17.9
Gaussian filter	2.9	5.1	6.9	10.2	11.9	14.4	17.3
Airfield							
Proposed Method	9.1	16.8	29.0	35.9	45.8	58.4	69.0
Soheil's Method	4.5	6.3	8.4	10.2	11.9	15.3	18.4
CNN	4.2	6.1	7.9	10.9	12.7	15.1	18.3
Gaussian filter	3.5	5.9	7.2	10.5	12.4	14.8	17.8
Airplane							
Proposed Method	7.1	19.3	26.5	7.04	45.0	56.8	67.2
Soheil's Method	3.7	5.9	8.7	10.2	12.0	13.9	17.1
CNN	3.2	5.4	7.6	10.4	12.1	14.7	17.7
Gaussian filter	2.8	5.3	7.1	10.1	11.8	14.3	17.2
Goldhills							
Proposed Method	10.3	18.9	28.4	37.9	47.0	55.7	66.2
Soheil's Method	3.7	6.4	8.9	11.9	12.6	16.1	17.9
CNN	3.9	6.1	8.3	11.4	13.1	15.7	18.9
Gaussian filter	3.4	5.9	7.8	10.9	12.8	15.3	18.5
Harbor							
Proposed Method	11.7	19.5	26.9	38.6	49.0	56.1	67.3
Soheil's Method	3.1	5.8	8.4	10.7	12.8	16.9	19.6
CNN	3.8	6.5	8.7	11.3	13.3	15.3	18.6
Gaussian filter	3.5	6.1	8.2	11.0	13.1	14.9	18.2
Aerial							
Proposed Method	9.8	18.7	27.4	36.5	45.4	54.9	63.2
Soheil's Method	3.7	4.9	8.3	11.2	12.5	14.9	18.2
CNN	3.5	5.7	7.9	10.8	12.7	15.0	18.3
Gaussian filter	2.9	5.5	7.4	10.2	12.1	14.7	17.8

Table 6.10: The actual running time($\times 10^3$second) of reconstructed images.

Dataset/Method	Type	Proposed algorithms	Criminisri	Huan
Lena	Image restoration			
- pattern 1		0.87	1.76	1.08
- pattern 2		1.97	2.98	2.12
- pattern 3		3.35	5.15	4.97
- pattern 4		3.19	4.13	3.93
Mural	Image restoration			
- pattern 1		1.25	2.09	2.76
- pattern 2		2.48	3.90	3.12
- pattern 3		4.90	3.98	3.19
- pattern 4		4.98	4.18	3.08
Monkey Giant	Image restoration			
- pattern 1		1.54	1.09	0.98
- pattern 2		2.76	3.10	3.13
- pattern 3		4.89	5.90	5.14
- pattern 4		5.75	4.09	4.98
Two circles	Image restoration	6.87	8.31	7.57
Brick	Image restoration	0.19	1.43	0.98
Windows	Image restoration	3.89	4.91	4.03
View	Object removal	0.65	2.75	2.09
Bunji-jump	Object removal	7.29	9.28	8.07

performance of reconstructing image gives a high value of PSNR as shown in Table 6.7. An apparent problem on the proposed method as well as any method is the edge effect problem. If a missing pixels locates in the edge of regions of available data, it may give a wrong imputed value. This problem may cause a low PSNR and it will be solved in the future. From the last group of data set, the experiment results showed that by using the hybrid imputation between neural network and the similarity comparison can give the high accuracy of the reconstructed image. Although the proposed algorithms can give a satisfied imputed image, some problem was occured which affects to the performance of the restoration image that is the size of window of missing pixels. If there are gaps that are large, then the accuracy could not be reached. Moreover, the method for selecting the nearest neighbors is an other problem which will be concerned in the future.

6.3.5 Conclusions

In this section, the imputation technique for incomplete multi-dimensional data in output attributes which were simulated from an unknown function of the input attributes is focused. The proposed ideas were based on the hybrid method of the semi-supervised learning and the comparison between two clusters for imputing incomplete multi-dimensional data. The experimental results showed that the proposed method give high accuracy of imputed data in several of data set.

6.4 Imputing incomplete SLC-off imagery based on neural network and similarity comparison

6.4.1 Experimental set-up

To test the performance of the proposed algorithms compared to the traditional imputation algorithms for the SLC-off imagery, the following experiments were setup.

6.4.1.1 Selection of algorithms

Three comparing methods to compare the accuracy with the proposed method were used. Each method can be used for imputing Landsat 7 ETM+ with SLC-off imagery. The following algorithms were used in the experiments.

1. LLHM algorithms.
2. Linear regression algorithms.
3. Kriging algorithms.

6.4.1.2 Case Study

In the experiments, the Bangkok imagery was used. Two similar images acquired before and after the malfunction of Landsat 7 ETM+ imagery sensor(SLC-off) occurred were collected. Two images which have the same position and resolution were used for the experiments. The radiometric values from Landsat 7 ETM+ imagery was used for testing the accuracy of the proposed algorithms. Each acquired data set was composed of missing values. Each scene of Landsat 7 ETM+ imagery was composed of 8 band separately. In this research, only band 1- band 5 and band 7 were used for testing and imputing missing values in each band separately. The used data set was Bangkok area from Landsat 7 ETM+ with SLC-off. The old images which do not have missing values was acquired at path 129, row 50 and the acquired date was 02/11/2000. The cloud cover of this image was 0.00%. The missing Landsat 7 ETM+ imagery caused by SLC-off was acquired at Path 129 Row 50 which is the same position of the missing image. The cloud cover of this image was 0.00 %. These two images were downloaded for experiments on 27/10/2010. In each band, each image 500×500 pixels was used.

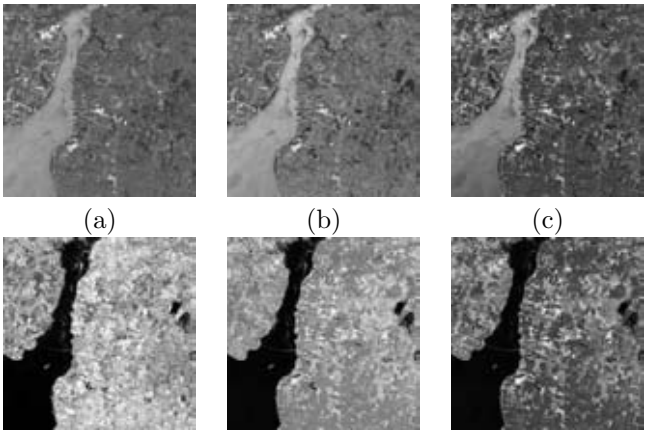


Figure 6.37: the complete image of Bangkok at each band. (a) Band 1 (b) Band 2 (c) Band 3 (d) Band 4 (e) Band 5 (f) Band 7.

6.4.1.3 Performance of algorithms

The performance evaluation of the proposed algorithms compared to other algorithms used the root mean squared error(RMSE) of the observed data and predicted data for accuracy measurement, calculated by the following equation,

$$RMSE = \sqrt{1/n\Sigma_{i=1}^{n}(\hat{\mathbf{p}}_i - \mathbf{p}_i)^2} \tag{6.11}$$

$\hat{\mathbf{p}}_i$ is imputed pixel. $\mathbf{p}_i$ is original pixel. The closer the values of RMSE to zero, the more precise the imputation is.

6.4.2 Experimental results

The complete image of Bangkok in each band (Band1-5 , Band 7) is shown in Fig. 6.37. The missing image of Bangkok in each band are shown in Fig. 6.38. The image which shows interpolation results using the proposed algorithms, Kriging, regression, LLHM are shown in Fig. 6.39, 6.40 , 6.41 and 6.42 respectively. Fig. 6.43 shows the interpolated image using proposed algorithms, Kriging, regression, LLHM with multispectral image in ordering. Statistics summary of complete image in six bands of Landsat 7 ETM+ SLC-off are shown in Table. 6.11. The RMSE, mean, standard deviation of imputed image by the proposed algorithms, Kriging algorithms, LLHM algorithms, and linear regression are shown in Table. 6.12.

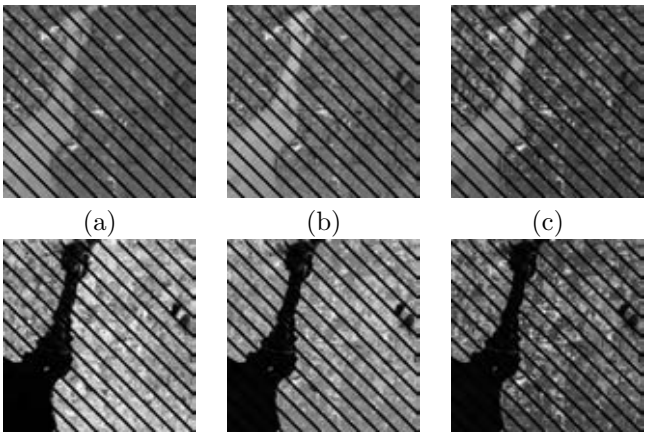


Figure 6.38: The missing image of Bangkok at each band. (a) Band 1 (b) Band 2 (c) Band 3 (d) Band 4 (e) Band 5 (f) Band 7.

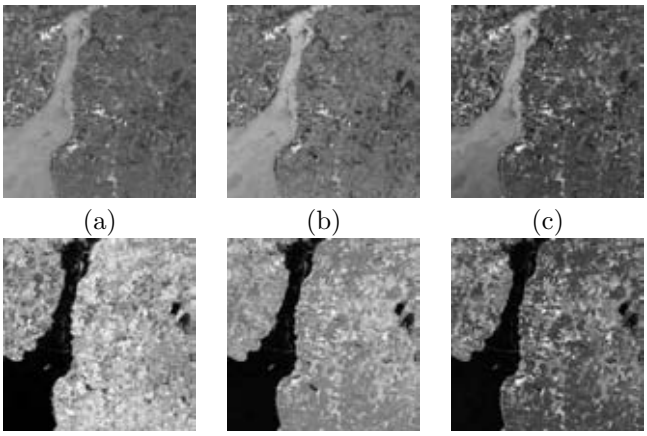


Figure 6.39: An imputed image by the proposed algorithms. (a) Band 1 (b) Band 2 (c) Band 3 (d) Band 4 (e) Band 5 (e) Band 7.

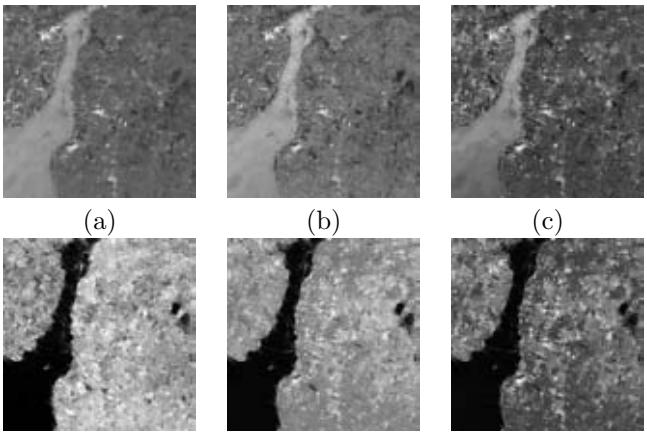
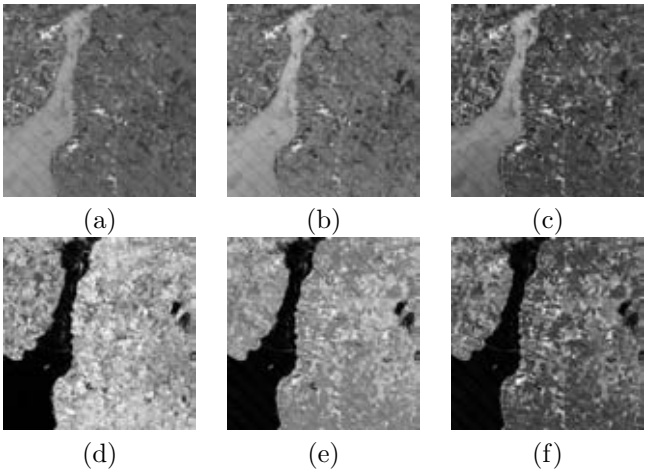


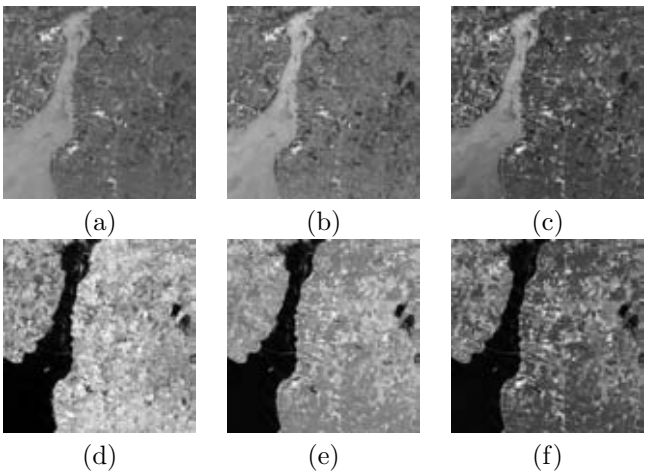
Figure 6.40: An imputed image by LLHM algorithms. (a) Band 1 (b) Band 2 (c) Band 3 (d) Band 4 (e) Band 5 (f) Band 7.



(a) (b) (c)

(d) (e) (f)

Figure 6.41: An imputed image by regression algorithms. (a) Band 1 (b) Band 2 (c) Band 3 (d) Band 4 (e) Band 5 (f) Band 7.



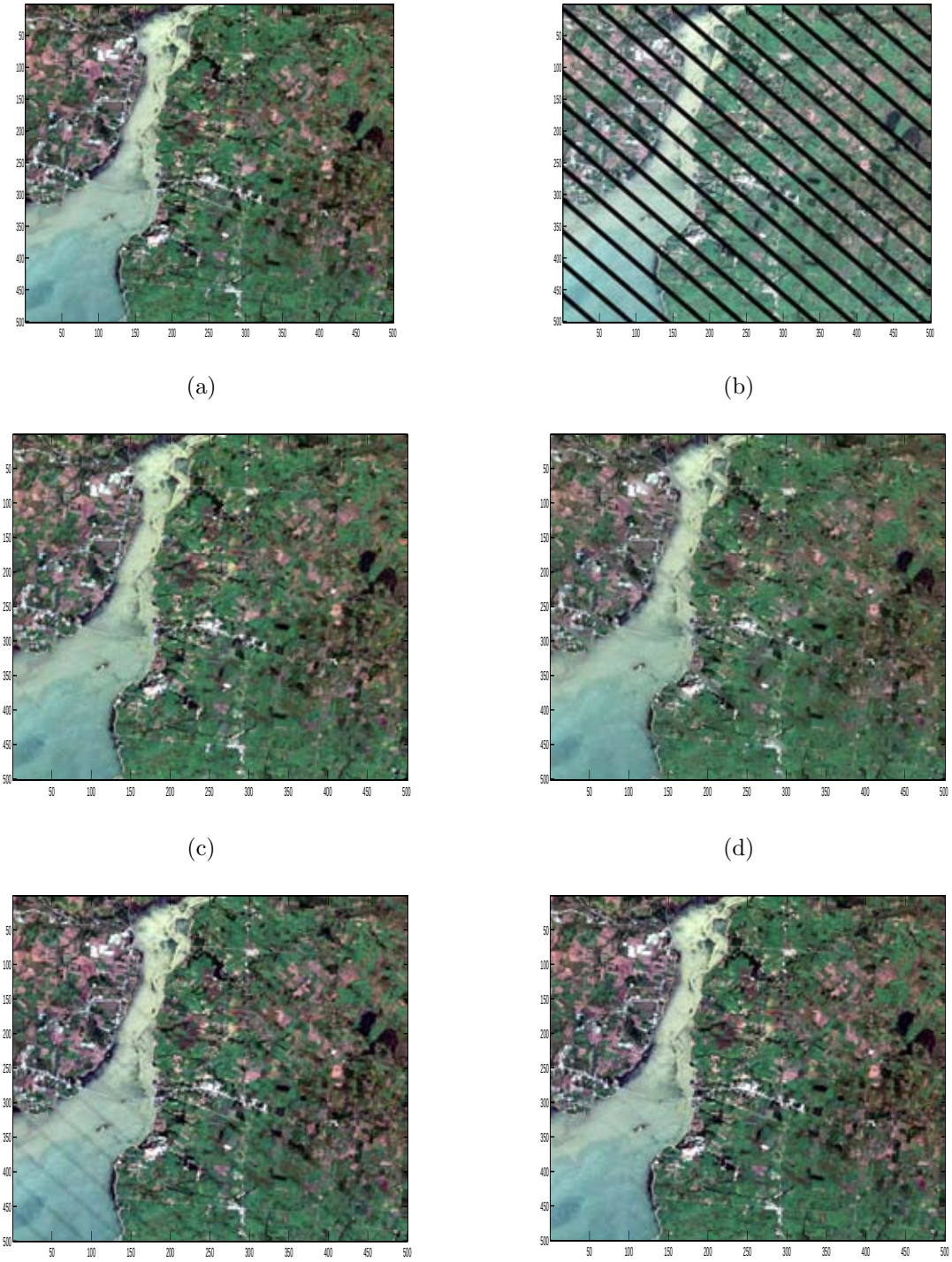
(a) (b) (c)

(d) (e) (f)

Figure 6.42: An imputed image by Kriging algorithms. (a) Band 1 (b) Band 2 (c) Band 3 (d) Band 4 (e) Band 5 (f) Band 7.

Table 6.11: Statistical summary of complete image in six bands in Landsat7 ETM+ SLC-off.

Statistics / Band	**1**	**2**	**3**	**4**	**5**	**7**
n	250000	250000	250000	250000	250000	250000
m	22%	22%	22%	22%	22%	22%
min	0	0	0	0	0	0
max	252	252	252	253	253	253
MEAN	64.4221	84.8780	70.1746	127.6604	109.2084	78.2593
SD	44.5657	45.3880	47.9974	62.5675	57.5259	49.3226



(a) (b)

(c) (d)

(e) (f)

Figure 6.43: The comparison of imputed image of Bangkok in each band (Band1-5 , Band 7) taken by Landsat 7 ETM+ imagery. (a)The original image with multispectral image(bands 1-5,7). (b)The missing image at Bangkok imagery of Landsat7 ETM+(bands 1-5,7). (c)The interpolation image using the proposed algorithms(bands 1-5,7). (d)The interpolation image using the LLHM algorithms(bands 1-5,7). (e)The interpolation image using regression algorithms(bands 1-5,7). (f)The interpolation image using the Kriging algorithms(bands 1-5,7).

Table 6.12: The RMSE values and related statistical values between complete image and imputed image of six bands in Landsat7 ETM+ SLC-off.

Method / Band	**1**	**2**	**3**	**4**	**5**	**7**
Imputed image by the proposed algorithms						
K	5	6	5	5	5	5
RMSE	**15.9376**	**15.2098**	**15.4173**	**16.6698**	**18.1420**	**16.5431**
MEAN	61.3192	80.6962	67.1872	124.4742	105.0827	74.4152
SD	42.8291	44.3468	46.5256	61.5823	56.3772	48.0734
Imputed image by Kriging						
K	8	8	8	8	8	8
RMSE	17.5632	17.1711	18.1278	21.5048	21.2381	17.9765
MEAN	64.9851	84.8708	69.9956	128.5881	109.4758	78.4432
SD	40.9710	41.7075	44.4708	60.2221	54.7356	46.6375
Imputed image by LLHM						
K	-	-	-	-	-	-
RMSE	32.2411	32.0687	32.2033	36.0538	37.2715	34.0550
MEAN	67.3294	86.6297	72.7107	129.2284	111.2040	80.4138
SD	48.1883	48.5700	51.5764	65.2025	61.3163	49.3227
Imputed image by regression						
K	-	-	-	-	-	-
RMSE	28.7146	28.5625	28.8892	33.1079	34.0624	29.2554
MEAN	67.1697	86.4065	69.9956	72.2829	129.6626	80.3717
SD	46.6554	46.9543	50.2592	64.0791	59.6959	52.2447

6.4.3 Disscussions

From the experimental results, by using a two-step processes for imputing missing values with neural network and similarity measurement between two groups of window in an image, the accurate imputed pixels were obtained more than compared to using only neural network to impute. The proposed algorithm in each missing pixels imputed by using the most k nearest neighbors of missing pixels in the imputation process. From the experiments, number of k varies in each band depending on its characteristics. In this study, numbers of k are $5, 6, 5, 5, 5$ and 5 for bands $1, 2, 3, 4, 5$ and 7, respectively. The imputed image showed that the algorithms give a smoother image than the imputed image by competitive algorithms. The RMSE of complete images and imputed image of six bands in Landsat 7 ETM+ SLC-off which were imputed by the proposed algorithms were 15.9376, 152098, 15.4173, 16.6698, 18.1420, and 16.5431, respectively. These RMSE values are lower than the RMSE of competitive algorithms.

Although the proposed algorithms gave a satisfied imputed image, some problems concerning the size of window of missing pixels were found. If there are gaps that are too large, the accuracy could not be obtained. Moreover, the method for selecting the nearest

neighbors is an other problem which will be concerned in the future.

6.4.4 Conclusions

In this part, a 2-step imputation process for imputing missing values was proposed by using neural network for tentative imputing missing values and the similarity measurement for finding the most similar cluster with target cluster. The algorithms details were described for imputing missing values in landsat 7 ETM+ with SLC-off by using neural networks using k-elements of nearest neighbors to be input of neural networks. Next, the details of algorithms for comparing the similarity between two clusters by adopting Wald-Wolfowitz test were introduced. The experimental results showed that the proposed algorithms gave the best accuracy compared with the competitive methods.

CHAPTER VII

CONCLUSIONS

7.1 Dissertation Summary

This research focused on the imputation technique for incomplete multi-dimensional data in output attributes which were simulated from an unknown function of the input attributes. The experiments were separated into two data sets: time series data set and image data set. For time-series data set, the imputation of incomplete data set were based on regional-gradient guided bootstrapping algorithms. Since the considered data were in a sequence of times, the whole data sequence can be partitioned into three consecutive groups. The first group is the sequence of data to the left of missing data sequence. The second group is the missing data sequence lying next to the first sequence. The last group is the data sequence to the right of the second sequence. In case of image data set, the proposed ideas were based on the hybrid method of the semi-supervised learning and the comparison between two clusters for imputing incomplete multi-dimensional data. This technique has two strategies depending on the characteristics(e.g., the shape) the damaged area inside a picture. If the image has a large missing pixels, then the similarity measurement with global and local strategy is used. If the image has missing pixels considered as noisy image, then the neural network and similarity measurement were used. The experimental results showed that the proposed method gives high accuracy of imputed data in most of data sets.

7.2 Further Improvement and Extension

In case of imputing time-series data set, the following factors can effect the accuracy of the imputed values in this domain:

1. **The length of missing data.** If the length of time series data is longer than 20 time steps, the accuracy could not be reach because of in that missing area may have an fluctuated time series. So, in the proposed algorithms cannot capture this fluctuated patterns.

2. **The similarity between two clusters of time series.** To compare the most similar cluster between two clusters, if the algorithm which is used to check the

similarity between these two clusters cannot capture all of the similarity between two clusters, it may result to the decreasing accuracy.

3. **Time complexity.** In the experiments, some problems occurred during the imputation processes. The time complexity of large size data set consume a lot of time in the imputation process especially in the bootraping process. In fact, the proposed algorithms can give the most accuracy of imputed data. On the other hand, the time to compute the confidence interval in a boostraping process has taken a lot of time.

4. **The occuring of extream values.** The accuracy of imputed time series data may be degraded if there are some occurring of extreme values in some clusters. The calculated values from bootstrapping will give a wrong predicted value and effects the similarity comparison process.

In case of the damaged image's restoration, the following factors should be carefully focused in the future:

1. **Selection of nearest neighbors of missing data.** The number of nearest neighbors of missing pixels affects the accuracy of the imputation. The carful selection of this number should be focused.

2. **Window size of target mask template windows.** The window size of target mask windows is another factor which affects to the accuracy of the imputation process. The large size of window will give the global statistics of missing areas. On the other hand, the small size of windows will not capture the structure of data sets. So, the selection of suitable window size should be paid attention on it.

3. **Time complextity.** In the experiments, some problems occurred while imputing missing values that is the time complexity of large size image data set. The occurring of this problem was caused by the process to check the similarity in each sliding window between the missing area and non-missing area. Repeating check the data set from all of sliding windows in an image is necessary. So, from this process, it can consume a lot of time.

Although the proposed algorithms gave a satisfied imputed image, some problems occurred as described above. Accordingly, these problems will be solve in the future. The author strongly believe that, with the proposed algorithm, many interesting, or the solution for imputing the missing data could be seen in this dissertation.

References

[1] Jerez, J.M., Ignacio, M., Alba, E., Ribelles, N., Miguel, M., and Franco, L. "Missing data imputation using statistical and machine learning methods in a real breast cancer problem". *Artif.Intell.Med* 50, 2, (October 2010): 105-115.

[2] Chiewchanwattana, S., Lursinsap, C., and Chu, C.H. "Imputing incomplete time series data based on varied-window similarity measure of data sequence". *Pattern Recognition Letters* 28,1, (February 2007): 1091-1113.

[3] Prasomphan, S., Lursinsap, C., and Chiewchanwattana, S. "Imputing time series data by regional-gradient-guided bootstrapping algorithm". In *International Symposium in Computer and Information Technology*, (2009): 269-275.

[4] Rubin, D.B. "Multiple imputation *after 18+ year". Journal of the American statistical Association* 91, 434, (January 1996): 473-489.

[5] Wasito, I. and Mirkin, B. "Nearest neighbors approach in the least-squares data imputation algorithms". *Information Sciences*, 169, 1, (January 2005): 1-25.

[6] Dempster, A.P., Laird, N.M. and Rubin, D.B. "Maximum likelihood from incomplete data via the em algorithms". *J.T.Statist.SOC* 39, (January 1977): 1-38.

[7] Rubin, D.B. "Multiple imputations for non Responses in Surveys". *John Wiley& Sons.,* New York, (1987). Third Edition.

[8] Batista, G. E. A. P. A., and Monard, M.C."An analysis of four missing data treatment methods for supervised learning". *Applied Artificial Intelligence* 17, (January 2003): 519-533.

[9] Wu, X., and Barbara, D. "Modeling and imputation of large incomplete multi-dimensional data sets". *Lecture Notes in Computer science DataWarehousing and Knowledge Discovery*. Springer 2454, (January 2002):365-374.

[10] Peng, H., and Zhu, S. "Handling of incomplete data sets using ICA and SOM in data mining". *Neural Comput. Appl*. 16, 2, (February 2007): 167-172.

[11] Piela, P. "Introduction to self-organizing maps modeling for imputation technique and technology". *Research in Ocial Statistics*, 98, 2, (2002): 5-19.

[12] Hathaway, R.J., and Bezdek, J.C."Fuzzy c-means clustering of incomplete data" *IEEE Transactions on Systems, Man, and Cybernetics PART B: Cybernetics*, 31, 5, (October 2001): 735-744.

[13] Ogawa, T., and Haseyama, M. "Missing intensity interpolation using a kernel pca-based pocs algorithm and its applications". *IEEE Transactions on Image Processing* 20, 2, (February 2011): 417-432 .

[14] Hu, X., Cao, D., and Qiu, S. "Image restoration based on eigensubspace for image structure". In *IEEE International Symposium on Multimedia*, (December 2006): 781-782.

[15] Grover, S., Gupta, S., and Sarj, A.A.K. "A unifined approach for digital image inpainting using bounded search space". *International Journal on Graphics, Vision and Image Processing* 5, 6, (June 2005): 17-24.

[16] Ji, H., Shen, Z.,and Xu, Y. "Wavelet frame based image restoration with missing /damaged pixels". *East Asia Journal on Applied Mathematics* 1, 2, (January 2011): 108-131.

[17] Telea, A. "An image inpainting technique based on the fast marching method". *International Journal on Graphics, Vision and Image Processing* 9, 1, (September 1979).

[18] Huan, X., Murali, B., and Ali, A.L. "Image restoration based on the fast marching method and block based sampling". *Computer Vision and Image Understanding* 114, 8, (September 2010): 847-856.

[19] Criminisi, A., Prez, P., and Toyama, K. "Region filling and object removal by exemplar-based image inpainting". *IEEE Transactions on Image Processing*, 13, 9, (September 2004).

[20] Bertalmio, M., Sapiro, G., Caselles, V., and Balleste, C. "Image inpainting". In *Proceedings of SIGGRAPH*, (July 2000).

[21] Torres, G.J., Basnet, R.B., Sung, A.H., Mukkamala, S., and Ribeiro, B.M. "A similarity measure for clustering and its applications". *International Journal of Electrical and Electronics Engineering* 3, (March 2009): 164-170.

[22] Bae, E. "Clustering similarity comparison using density profiles". *Lecture Notes in Computer science*. AI2006. Springer 99, (2006): 342-351.

[23] Dong, Y., Sun, Z., and Jia, H. "A cosine similarity-based negative selection algorithms for time series novelty detection". *Mechanical System and Signal Processing* 20, (March 2005): 1461- 1472.

[24] Leauhatong, T., Hamamoto, K., Atsuta, K., and Kondo, S. "A new similarity measure for content-based image retrieval using the multidimensional generalization of the wald-wofowitz runs test". In *IEEE International Symposium in Computer and Information Technology*, (November 2008): 81-86.

[25] Zoubir, A.M., and Iskandes, D.R. "Bootstrap methods and application". *IEEE Signal Processing Magazine* 10, (July 2007): 1-10.

[26] Malladi, R., and Sethian, J. A. "Fast methods for shape extraction in medical and biomedical imaging". *Geometric Methods in Biomedical Image Analysis*, (2002): 1-13.

[27] Duda, R.O., and Hart, P.E. "Use of the hough transformation to detect lines and curves in pictures". *Artificial Intelligence Center*, (April 1971).

[28] Rabbani,T., and Heuvel, F. "Efficient hough transform for automatic detection of cylinders in point clouds". In *Proceedings of the 11th Annual Conference of the Advanced School for Computing and Imaging (ASCI '05)*, (June 2005).

[29] Sethian, J.A. "Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science". Cambridge University Press, Cambridge, (1999).

[30] Sidler, R. "Kriging and Conditional Geostatistical Simulation Based on Scale-Invariant Covariance Models". John Wiley&Sons, New York, (2003).

[31] Fernandez, G., and Calderon, A. "Spatial regression analysis vs kriging method for spatial estimation". *Int. Adv. Econ. Res* 10 (July 2008): 1-10.

[32] Scaramuzza, P., Micijevic, E., and Chander, G. "SLC gap-filled products phase one methodology". *ArXiv e-prints*, (2004).

[33] Rulloni, V., Bustos, O., and Flesia,A.G. "Large gaps imputation in remote sensed imagery of the environment". *ArXiv e-prints*, (June 2010).

[34] Zhanga, C., Lia, W., and Travis, D."Gaps-fill of slc-off landsat 7 etm+satellite image using a geostatistical approach". *International Journal of Remote Sensing* 28, 22, (January 2007): 5103- 5122.

[35] Feizi, S., Zahedpou, S. and Soltanolkotabi, M. "Salt and pepper noise removal for image signals". In *IEEE Transactions on Signal Processing*, (2008).

[36] Su, T.J., Huang, M.Y., Hou, C.L., and Lin, Y.J. "Cellular neural networks for gray image noise cancellation based on a hybrid linear matrix inequality and particle swarm optimization approach". *Neural Process Lett* 32, (September 2010): 147-165.

Biography

Name: Mr. Sathit Prasomphan.

Date of Birth: 21^{th} May, 1979.

Educations:

- Ph.D. Candidate in Computer Science , Department of Mathematics and Computer Science, Chulalongkorn University, Bangkok, Thailand.
- Visiting student, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Japan, 2010.
- M.Sc. Program in Software Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand, 2005.
- B.Sc. Program in Computer Science, Faculty of Science, Khon Kaen University, Khon Kaen, Thailand, 2002.

Publication papers:

- Prasomphan, S., Lursinsap, C., Chiewchanwattana, S.: Imputing Time Series Data by Regional-Gradient-Guided Bootstrapping Algorithm, In *The 2009 International Symposium on Communications and Information Technologies (ISCIT 2009)* , vol. 2009.
- Prasomphan, S., Lursinsap, C., Chiewchanwattana, S.: Two-Phase Imputation with Regional-Gradient-Guided Bootstrapping Algorithm and Dynamics Time Warping for Incomplete Time Series Data. In *The 6th International Conference on Intelligent Computing(ICIC 2010)* , pp.615–622

Scholarship: Thailand Government's Scholarship .