Night Vision and
Electronic Sensors Directorate

AMSRD-CER-NV-TR-246

Image Based Synthesis for Airborne Minefield Data

December 2005

Approved for public release; Distribution unlimited

CONQUEST OF DARKNESS

Fort Belvoir, Virginia 22060-5806

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
December 2005

3. REPORT TYPE AND DATES COVERED
Final Thesis - 2004

4. TITLE AND SUBTITLE
Image Based Synthesis for Airborne Minefield Data

5. FUNDING NUMBERS
DAAB07-01-D-G601/
W15P7T-05-R-B497

6. AUTHOR(S)
Thandava Krishna Edara

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Missouri-Rolla
Graduate School of Electrical Engineering

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Night Vision and Electronic Sensors Directorate
Countermine Division (Anh Trang)
10221 Burbeck Road
Fort Belvoir, VA 22060-5806

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AMSRD-CER-NV-TR-246

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approve for Public Release; Distribution Unlimited

12b. DISTRIBUTION CODE

A

13. ABSTRACT *(Maximum 200 words)*

Image-based synthesis is a process of generating new images from already existing images such that newly created images are indistinguishable when compared with the real-world images. Minefield synthesis, in this framework, refers to the process of generating different realistic minefield scenarios from previously collected images of the mine, clutter, and background. In this thesis, minefield synthesis using patch-based sampling of previously acquired airborne MWIR (mid wave infrared) imagery is explored. The main idea is to synthesize a new minefield by selecting appropriate small patches from existing images and stitching them together in a consistent manner to simulate realistic imagery data representing different minefield scenarios. The selected patches include those from different background types in conjunction with natural and man-made clutter and different mine types. The background imagery is modeled using a first order Markov process, assuming that the statistical characteristics of image-patch at particular location are dependent only on the characteristics of patches in the adjacent neighborhood. A patch-based Markov model for different backgrounds is estimated from the available images. The proposed model is capable of generating any desired terrain condition (homogenous and inhomogeneous) from available background/terrain data. In addition, different minefield layouts such as patterned and scattered can be generated using mine patches. Minefield imagery for single frame as well as those simulating a flight path over a terrain map can be synthesized. Minefield image data synthesized using this procedure should be valuable for airborne mine and minefield detection program t evaluate most mine detection as well as minefield detection algorithms

14. SUBJECT TERMS
Landmine Detection, Pixel-based image synthesis, patch-based image synthesis, Markov Random Fields (MRF) model, KL Transform (KLT)

15. NUMBER OF PAGES
115

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	None

IMAGE BASED SYNTHESIS FOR AIRBORNE MINEFIELD DATA

by

THANDAVA KRISHNA EDARA

A THESIS

Presented to the Faculty of the Graduate School of the

UNIVERSITY OF MISSOURI-ROLLA

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

2004

Approved by

______________________________ Sanjeev Agarwal, Advisor

______________________________ Randy H. Moss

______________________________ R. Joe Stanley

ABSTRACT

Image-based synthesis is a process of generating new images from already existing images such that newly created images are indistinguishable when compared with the real-world images. Minefield synthesis, in this framework, refers to the process of generating different realistic minefield scenarios from previously collected images of the mine, clutter, and background. In this thesis, minefield synthesis using patch-based sampling of previously acquired airborne MWIR (Mid Wave Infra Red) imagery is explored. The main idea is to synthesize a new minefield image by selecting appropriate small patches from the existing images and stitching them together in a consistent manner to simulate realistic imagery data representing different minefield scenarios. The selected patches include those from different background types in conjunction with natural and man-made clutter and different mine types. The background imagery is modeled using a first order Markov process, assuming that the statistical characteristics of image-patch at a particular location are dependent only on the characteristics of patches in the adjacent neighborhood. A patch-based Markov model for different backgrounds is estimated from the available images. The proposed model is capable of generating any desired terrain condition (homogenous and inhomogeneous) from available background/terrain data. In addition, different minefield layouts such as patterned and scattered can be generated using mine patches. Minefield imagery for a single frame as well as those simulating a flight path over a terrain map can be synthesized. Minefield image data synthesized using this procedure should be valuable for airborne mine and minefield detection programs to evaluate most mine detection as well as minefield detection algorithms.

ACKNOWLEDGMENTS

First, I would like to express sincere gratitude to my advisor, Dr. Sanjeev Agarwal, for his patience in guiding me through my graduate studies and the motivation he has given me in finding elusive grounding to research problems. Dr. Agarwal has been abundantly helpful, and has assisted me in numerous ways. Without his dedicated supervision, it would have been difficult for me to complete this thesis on time. I would like to thank my committee members, Dr. Randy Moss and Dr. Joe Stanley, for their valuable and timely suggestions during my thesis work.

I would like to thank Countermine Division of Night Vision and Electronic Sensors Directorate (NVESD) for providing the data and financial support for my research.

I would like to thank all my friends whose constant encouragement helped me to achieve my goals.

Finally, I would like to thank and dedicate this thesis to my parents, for their love, support and encouragement throughout my studies.

TABLE OF CONTENTS

Page

APPENDICES

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

Table Page

1. INTRODUCTION

Image/texture synthesis is one of the widely used concepts in the field of image processing, computer graphics, and virtual reality. Image synthesis can be defiends as a process of generating new images with or without using previous images. Representative applications of image synthesis include artificial texture generation [1], image repairing [2], photometric image rendering [3] and ultrasound imaging simulations [4]. Since the last decade, the image synthesis area has witnessed the development of a number of different algorithms. The main aim of all these algorithms is to synthesize real world like images based on mathematical model derived from available real-world images. The existing image/texture synthesis algorithms can be divided broadly into two groups: algorithms that use pixels as the basic building blocks, and the algorithms that use a specified block of image patch as the building blocks, in synthesizing new images. The rest of this section contains a brief review of the both pixel-based and patch-based image synthesis, motivation for minefield synthesis, review of the other mine/minefield synthesis efforts, and an overview of this thesis.

1.1. PIXEL-BASED IMAGE SYNTHESIS

Heeger et al. [5] developed a pyramid-based texture algorithm that approximately matches marginal histograms of filter responses. This algorithm is primarily based on psychological studies of the human visual system. According to this model, it is possible to capture all the relevant spatial information that characterizes a given image, by using the first order statistics of a set of filters. This kind of approach can be viewed as a parametric model of image synthesis because; the synthesis procedure is mainly dictated by parameters that are computed from the original image. However, this procedure works well only for homogeneous textures. Zhu et al. [6] introduced a mathematical model called FRAME, which incorporates filters and histograms into Markov Random Field (MRF) models. They made use of the minmax entropy principle in selecting feature statistics from the original image. Paget et al. [7] proposed texture synthesis using non-parametric multi scale Markov Random Fields (MRF) models. In this approach, the authors proved that it is possible to capture all the relevant characteristics of the image in a unique statistical model and that the texture can be synthesized using stochastic relaxation (SR) [8] (also called Monte Carlo Annealing). This method proved to be very efficient in synthesizing natural textures but was computationally expensive and time consuming. Wei et al. [9] proposed fast texture synthesis using tree-structured quantization. Using this method, the authors succeeded in accelerating the synthesis procedure using tree-structured vector quantization. This process is also based on pixel wise generation of the synthesized image while preserving the local similarities between the original and synthesized images.

1.2. PATCH-BASED IMAGE SYNTHESIS

All the techniques discussed above use a lattice structure composed of pixels as the building blocks in generating the new image. Though the quality of the images synthesized using these techniques is good, most of them are computationally expensive. To overcome this problem Xu et al. [10] developed a synthesis algorithm by pasting texture/image blocks (patches) of the input image onto the synthesized image. This algorithm is capable of synthesizing textures images extremely fast, however, at the expense of effectiveness. The main disadvantage of this approach is its inability to handle boundary conditions well. Liang et al. [11] proposed a Patch-Based Sampling approach with better edge handling capability. In this model, the new image is generated based on Markov Random field model. Minefield synthesis discussed in this thesis is based mainly on this approach.

1.3. MOTIVATION FOR MINEFIELD SYNTHESIS

Although the main applications of image/texture synthesis till now have been in the computer graphics and virtual reality, it is a very useful concept in creating natural looking environments for many computer applications where a lack of adequate data is a major concern. For some computer applications such as military aviation, having an adequate amount of data is a very important factor and the number of real-world images that are available at hand may not be sufficient to obtain reliable and accurate results.

Image synthesis is considered to be a very good choice under these circumstances as it generates real-world like images.

Infrared (IR) images are used in different warfare scenarios, and flight simulations. But obtaining such imagery for all possible environmental conditions and tactical scenarios is practically an impossible task. To overcome this problem, Weijie et al. [12] proposed a model for infrared image synthesis that effectively synthesizes infrared images of various objects under different ambient and viewing conditions. This in turn simplifies the need for an extensive set of data collection for all possible lightning conditions. Kamei et al. [13], proposed a model that synthesizes a new view of an image by pasting image strips that are extracted from pre-acquired source images. This method eliminates the need to photograph the same image at different viewing angles. Vince [14] addressed the need for a large set of realistic data in creating flight simulator applications. He also point out the importance of image synthesis in generating realistic data for these applications. Meredith et al. [15] studied the aptness of synthetic imagery for Aided-Target Recognition (AiTR) evaluation. In this study, authors compared the performance of AiTR for a different real and synthetic imagery and concluded that the image scenes synthesized based on visual metric such as noise, target-to-background contrast, and mean edge contrast, may provide realistic scenes for AiTR evaluation. For this thesis, the concept of image synthesis is used to generate imagery for the evaluation of various mine/minefield detection programs.

As part of airborne mine and minefield detection programs, various researchers have developed several mine/minefield detection algorithms; however, the true capabilities of these algorithms in various environmental conditions are still under investigation. To verify the detection capabilities of these algorithms and their future developments, one needs to collect an extensive set of imagery, which, in most cases, may prove to be a very expensive and time-consuming task. Swonger [16] in a memo pointed out the near impossibility of collecting an exhaustive set of minefield data for different environments, time of the day, terrain conditions and minefield layouts.

As part of this effort, a patch-based minefield synthesizer that constructs minefield scenarios using pre-acquired mine as well as the background images is explored. This approach uses only a limited set of background-only images, and images with different clutter and mine targets for all the background of interest (like road, grassland and desert). The biggest advantage of the proposed approach is the ability to synthesize any desired terrain condition (homogenous or inhomogeneous). This in turn simplifies the need for an extensive set of data collected for different terrain and minefield scenarios.

1.4. MINE AND MINEFIELD SYNTHESIS EFFORTS

The concept of image synthesis for the simulation of mine and minefields environments, has been an emerging topic over many years. Various researchers have proposed different models that assist in testing and evaluating various mine/minefield detection programs and other related efforts. As a part of these developments, Sjokvist et al. [17] developed a model that simulates the thermal contrast of solar radiated surfaces such as sand, containing buried mine-like objects. In this effort, the authors studied the thermal physics of buried mine-like objects in sand and its variations in noticeable thermal contrasts over the surface. The authors also proposed most suitable conditions to search for the mines, based on the simulation results. Liao et al. [18] developed a physics-based simulator to study the passive infrared signature of surface-laid anti-vehicle mines. Authors have succeeded in generating results that are in reasonable agreement with experimental imagery acquired by a mid-wave infrared (MWIR) camera with respect to the signature shape and intensity contrast.

All the above-mentioned utilities are primarily aimed at improving the mine detection capabilities under various circumstances, but these algorithms are not intended to evaluate minefield detection algorithms. On the other hand, Swonger [16] proposed a model that seeks to evaluates effectiveness of the mine and minefield detection algorithms over different backgrounds. He proposed to use an exhaustive collection of background data and use pre-acquired mine signatures to generate different minefield scenarios over the collected background data. Even though this approach eliminates the need to lay different minefield scenarios, it still requires huge background data collection effort. Veredian [19] had developed an application for minefield simulation and evaluation (genMF) as part of airborne mine and minefield detection program. However, this minefield simulation program heavily relies on the assumed distribution for RX

(Reed-Xiaoli) response of the background clutter and mine targets that may not be reliable. Further, minefield data simulated using this approach is not sufficient to evaluate many of the advanced minefield detection algorithms that are currently under development. The image-based minefield synthesis effort in this thesis is aimed towards generating synthesized minefield data which will be useful for the evaluation of both mine and minefield algorithms and provide realistic and believable imagery for human visualization and interpretation.

1.5. OVERVIEW OF THE THESIS

This thesis is divided into 6 sections and is organized as follows; Section 2 describes the minefield synthesis problem along with its basic architecture. Section 3 describes the fundamental elements of the patch-based modeling, Markov Random Field (MRF), quad tree representation of patch, and image database. Section 4 describes the selection of patches and single frame synthesis of minefields. An algorithm for minefield synthesis and related results are also included in this section. Section 5 describes synthesis of flight path and issues associated with it. Finally, Section 6 includes conclusion and potential future improvements.

2. MINEFIELD SYNTHESIS

2.1. PROBLEM STATEMENT

Minefield synthesis in the current context is a process of generating different realistic mine signatures and minefield scenarios. The principal aim of creating different minefield structures is to evaluate and validate various mine/minefield detection algorithms for different terrain and environmental conditions. To achieve this goal, synthesized minefields should serve the same purpose as the original imagery. In this thesis, minefield synthesis using patch-based sampling of previously acquired airborne MWIR imagery is explored. The reason for choosing the patch-based sampling technique [11] is that, unlike pixel-based synthesis algorithms, it uses patches of the input image as building blocks in synthesizing new images. The generation of minefields using this approach is faster than pixel-based synthesis procedures; also, the modeling problem is simpler when compared to modeling the texture or the image characteristics at the pixel level. The idea is to synthesize a new (minefield) image by selecting appropriate small patches from the existing images. These patches include those from different background types in conjunction with natural clutter and different mines types. The selected patches are then stitched together, to simulate realistic imagery data for different minefield scenarios. Given a set of sample image(s) $\{I_{IN}\}$, terrain information, and minefield layout, a minefield image $\{I_{OUT}\}$ is synthesized progressively, by placing one image-patch at a time until the entire new minefield is generated.

2.2. SYNTHESIS ARCHITECTURE

Figure 2.1 shows the pictorial representation of the proposed minefield synthesis architecture. As shown in the figure, the minefield synthesizer has three inputs and one output. The first input, image database provides the statistical model and sample patches for synthesis and largely dictates the quality of the synthesized image. The second input, terrain map, dictates the terrain structure of the synthesized image and the third input, mine layout, dictates the minefield structure of the synthesized image. The minefield synthesizer, which works on the patch-based sampling concept, synthesizes a new minefield based on the input parameters. The output part of the architecture holds the synthesized image from the synthesizer. This section gives a brief description of the input parameters and the minefield synthesizer. The role of the blocks shown in Figure 2.1, in minefield synthesis is also discussed briefly in this section. The detailed description of these parameters and their role are given later in the thesis.

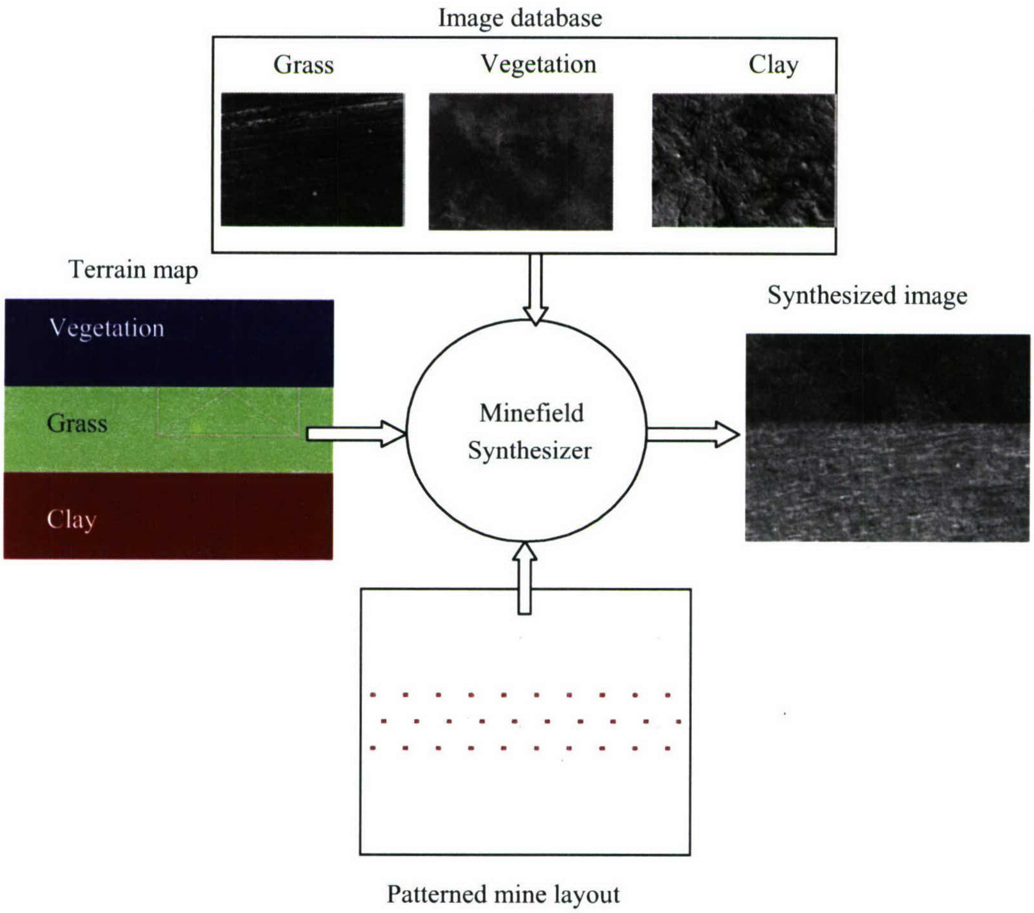


Figure 2.1 Schematic of Minefield Synthesis

2.2.1. Image Database. The image database contains sample patches from the pre-acquired image sets from different backgrounds. An image database with only three backgrounds is shown in the Figure 2.1 for the sake of simplicity. These image sets include background-only image patches, image patches with mines, and patches with manmade and natural clutter. A background-only image patches do not contain any mine and clutter signatures. Clutter patch is defined as a patch containing any significant anomaly other than mine. Storing samples images and corresponding patches requires lots of memory. Also, raw image format increases the computational time for patch selection. To overcome this problem, patches from each background are stored in a structured format. A full description of the structure used to describe the image database is discussed in Section 3.5.

2.2.2. Terrain Map. A terrain map contains all the related terrain information such as different backgrounds that are presented in the terrain map, terrain map resolution, terrain map size and terrain map geographic location. For easy visualization, each background of the terrain map in Figure 2.1 is represented with a different color viz. vegetation (top color), grass (middle color), and clay (bottom color). The box drawn on the terrain map represents the part of the terrain area selected for synthesis representing a single frame. However, it is also possible to synthesize multiple frames of the terrain map representing the image data over a flight path. The concept of flight path synthesis is discussed in Section 5. For the current purpose, the terrain map is assumed as an indexed image with each index representing different types of background in the image database. In future, it may be possible to use actual geographic maps of the desired areas to define the terrain map.

2.2.3. Minefield Layout. Minefield layout contains the information related to the mine type and relative mine location in the terrain map. In general, minefield layout exists in one of the two forms:

- Patterned structure
- Scattered/Random structure

For patterned structures, mines are placed as per regular pattern. On the other hand, for scattered/random structure, mines are placed in a random fashion. Figure 2.1 shows a fixed pattern mine layout. Patterned and scattered minefield structures are very popular in many real world scenarios and, hence the synthesis of these two mine structures on different terrain is studied in this thesis. Previously acquired image database and terrain map, are the essential inputs to the minefield synthesis algorithm. Synthesis of a new image is, however, possible without knowing the mine layout. Newly synthesized images without mines are termed as background-only images.

2.2.4. Minefield Synthesizer. This is the heart of the minefield synthesis architecture. Most of this thesis mainly deals with the algorithm used to implement minefield synthesizer given the allied information. Given an image patch in the neighborhood, a Markov model is utilized to select adjacent partially overlapping image patches (discussed in Section 3.4). These selected patches are stitched appropriately so that the edges of the patches are not visible, and do not result in undesirable artifacts (discussed in Section 4.2). The major steps involved in minefield synthesis are:

- **Modeling:** Modeling involves the estimation of stochastic parameters from the original input images. A detailed description of the patch-based modeling is given in Section 3.
- **Patch selection for synthesis:** This deals with the development of effective and efficient procedure for selecting image patch to synthesize new minefield. Detailed description of the selection process is presented in the Section 4.1.
- **Rendering:** Rendering deals with the issues of seamless stitching of neighboring patches and required mine or clutter patches so that no obvious visual or statistical artifacts are generated in the synthesized image. Blending of the image patches is described in detail in Section 4.2.

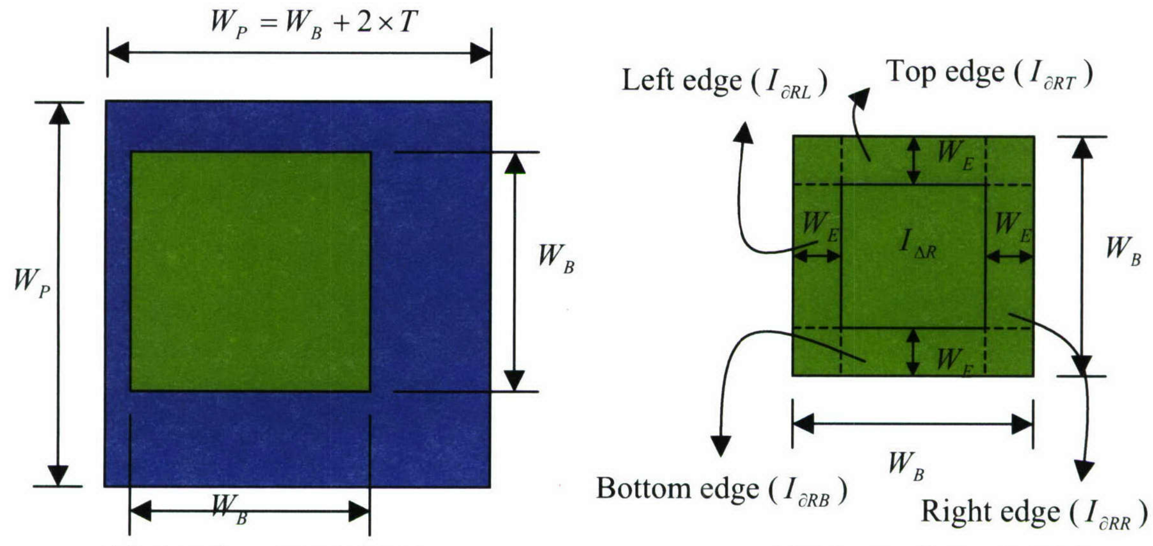
3. PATCH-BASED MODELING

To synthesize natural scenes using sample images it is necessary to capture all the visual characteristics of the original image in the form of a mathematical model. The mathematical model derived from the original images should be accurate enough to synthesize new images that are indistinguishable from real-world images. This thesis exploits the model based on a patch of the input image. This section includes fundament elements of patch-based modeling, quad tree representation of patches, the concept of Markov Random Fields, and image database. Clear understanding of the fundamental elements of the patch-based modeling such as a patch, feature based representation of patch, and their structure, is essential as they play a very important role in the modeling, selection and the subsequent synthesis procedure.

3.1. DEFINITION OF PATCH

A patch is a small portion of image area that is extracted from the original image. Each input images is divided into small overlapping patches. Image patches stored in the image database are slightly larger than the patches used in synthesis. The patch used in the synthesized image is called Synthesis Patch (SP), while larger patch stored in the image database is called a Database Patch (DP). The reason behind selecting a slightly larger patch size for DP is to ensure good correlation at the edges between neighboring patches in synthesized image. The synthesis patch of the specified size is extracted from the DP so that it has the best correlation with edge of the neighborhood patches. Detailed description of this idea is given in the Section 4.1.4. Figure 3.1 (a) shows a typical structure of Database Patch of size $W_P \times W_P$. Figure 3.1 (b) shows a typical structure of Synthesis Patch of size $W_B \times W_B$ extracted from the DP as shown in the Figure 3.1 (a). The parameters $I_{\Delta R}$, $I_{\partial RT}$, $I_{\partial RB}$, $I_{\partial RR}$ and $I_{\partial RL}$ shown in Figure 3.1 (b), represent the central, top edge, bottom edge, right edge and left edge of a synthesis patch respectively. For easy visualization, in Figure 3.1(a), both DP and SP are drawn in different color. Figure 3.2 (a) and 3.2 (b) shows an example image patches of sizes 60×60 and 40×40

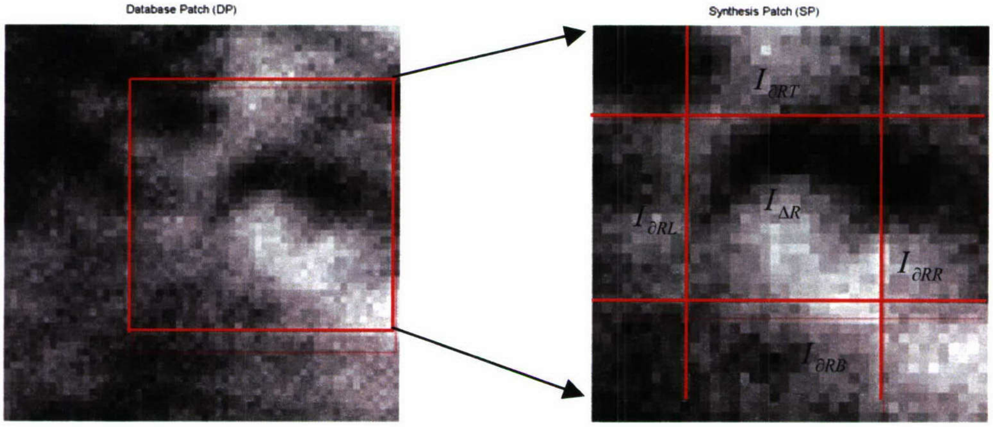
(in pixels), which correspond to database patch and synthesis patch respectively. The importance of various parameters shown in the Figure 3.1 (a) and Figure 3.1 (b) and their influence on the modeling and synthesis procedure is discussed in the following sub sections.


$W_P = W_B + 2 \times T$
W_P
W_B
W_B
Left edge ($I_{\partial RL}$)
Top edge ($I_{\partial RT}$)
W_E
W_E
$I_{\Delta R}$
W_E
W_E
W_B
W_B
Bottom edge ($I_{\partial RB}$)
Right edge ($I_{\partial RR}$)

(a) Database Patch (DP)

(b) Synthesis Patch (SP)

Figure 3.1 Patch-based Modeling Structures of Database Patch and Synthesis Patch



(a) Database Patch

(b) Synthesis Patch

Figure 3.2 Example Image for DP and Corresponding SP

3.1.1. Patch Size (W_B). The patch size W_B represents the size of the patch that is used in actual image synthesis. The size of the patch affects the appearance of the synthesized image. A smaller size indicates more number of patches for given set of sample images and allows for more matching possibilities between neighborhood patches at the expense of high computational time. This in turn implies weaker statistical constraints and less similarity between the input image and the synthesized counterpart. A bigger patch means higher similarity between the input image and the synthesized image and lesser computational time. For the sake of simplicity, only square patches are assumed. For this thesis, size of the synthesis patch is considered as 40 pixels as shown in Figure 3.2. (b).

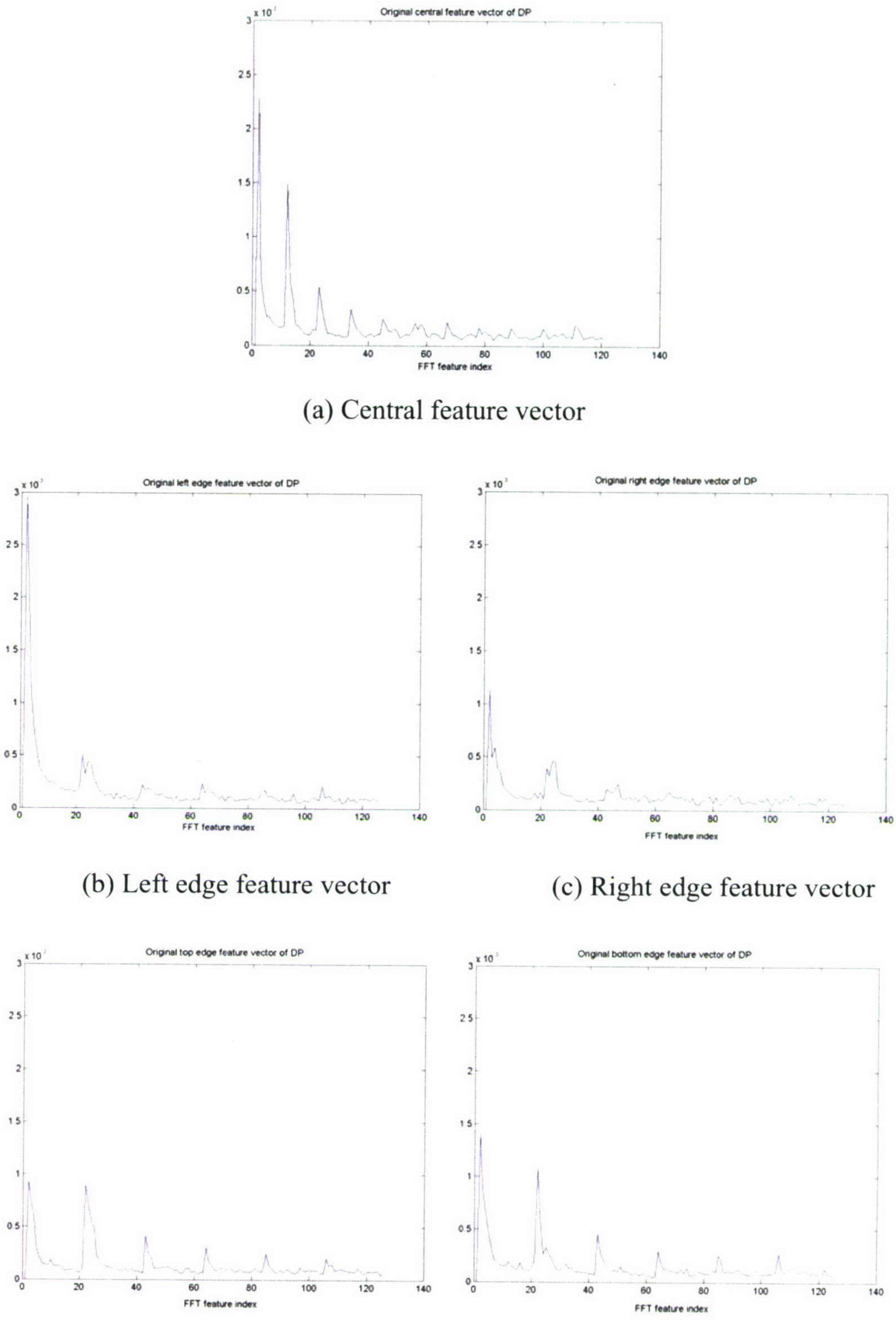
3.1.2. Edge Width (W_E). The width of the edge affects the blending between the adjacent patches of the synthesized image. Smaller width may results in poor blending. A bigger edge width results in high-quality blending between adjacent patches, but may results in possibly poor match with the original image characteristics. For this thesis, edge width of the synthesis patch is considered as 10 pixels.

3.1.3. Tolerance (T). The tolerance represents the extra image area included during the extraction of the patch from sample images. The actual size of the database patch extracted from the input image is $(W_B + 2\times T)\times(W_B + 2\times T)$. A bigger tolerance T means a better correlation can be established between neighborhood patches at the expense of increased computation and search. In this thesis, tolerance of 10 pixels is used in extracting database patch from the sample image.

3.2. FEATURE BASED REPRESENTATION OF PATCH

The feature-based representation is used to characterize a patch area under consideration. If the feature vector represents the central portion ($I_{\Delta R}$) of the synthesis patch in Figure 3.2, it is called central feature vector $F_{\Delta R}$. If it represents the edge portion of the synthesis patch shown in Figure 3.2. (b), it is called an edge feature vector. Edge features are designated as $F_{\partial RT}$, $F_{\partial RB}$, $F_{\partial RR}$, and $F_{\partial RL}$ for top edge ($I_{\partial RT}$), bottom edge ($I_{\partial RB}$), right edge ($I_{\partial RR}$), and left edge ($I_{\partial RL}$) respectively. The feature vector for a particular region of the patch is obtained by starting with two-dimensional *Fast Fourier Transform (FFT)* of the corresponding region. In order to eliminate the effect of the DC component (in nearest neighborhood calculations) on the feature vector, the DC value of the two-dimensional FFT vector is replaced with zero. The resultant two-dimensional *FFT* vector is stacked into a single-dimensional vector to simplify future calculations.

The feature vector derived from the synthesis patch may not give the exact representation of the characteristics of the database patch. Since the original database patch size W_P stored in the database is greater than the synthesis patch size W_B (Figure 3.1), for each database patch, there exist $(2\times T)^2$ ($T =$ Tolerance) valid synthesis patches of size $W_B \times W_B$. Calculating and storing the central and the four edge characteristics for each possible synthesis patch during the synthesis procedure needs a lot of computational time as well as memory. To overcome this problem, the central and the edge characteristics for all possible synthesis patches are averaged over the database patch. The central feature of a database patch is, thus, represented by a single feature vector, which is the average of the central features of all possible synthesis patches. In the same way, the edge characteristics for each edge of a database patch is represented by a single feature vector, which is an average of corresponding edge characteristics of all possible synthesis patches in a given database patch. Figure 3.3 (a) through (e) represent the average central, left edge, right edge, top edge, bottom edge feature vectors of the same database patch of size 60×60 respectively. The central and the four edge feature vectors of the database patch shown in Figure 3.3 are obtained by considering frequency terms from (0, 0) to (p/2 +1, q/2 +1) of the two-dimensional *FFT* of corresponding region of size $p\times q$. Thus, the patch area of size $p\times q$ results in a feature vector with (p/2 +1) × (q/2 +1) frequency samples. For the present case, the central regions of size 20×20 results in a feature vector with 121 frequency samples and the edge of size 40×10 results in a feature vector with 126 frequency samples. Figure 3.4 (a) through (e) represent the average central, left edge, right edge, top edge, bottom edge feature vectors of 500 randomly selected database patches (each of size 60×60) extracted from 60 input images (each of size 512×640). All these 60 images belong to the same background (Mine area –01).



(a) Central feature vector

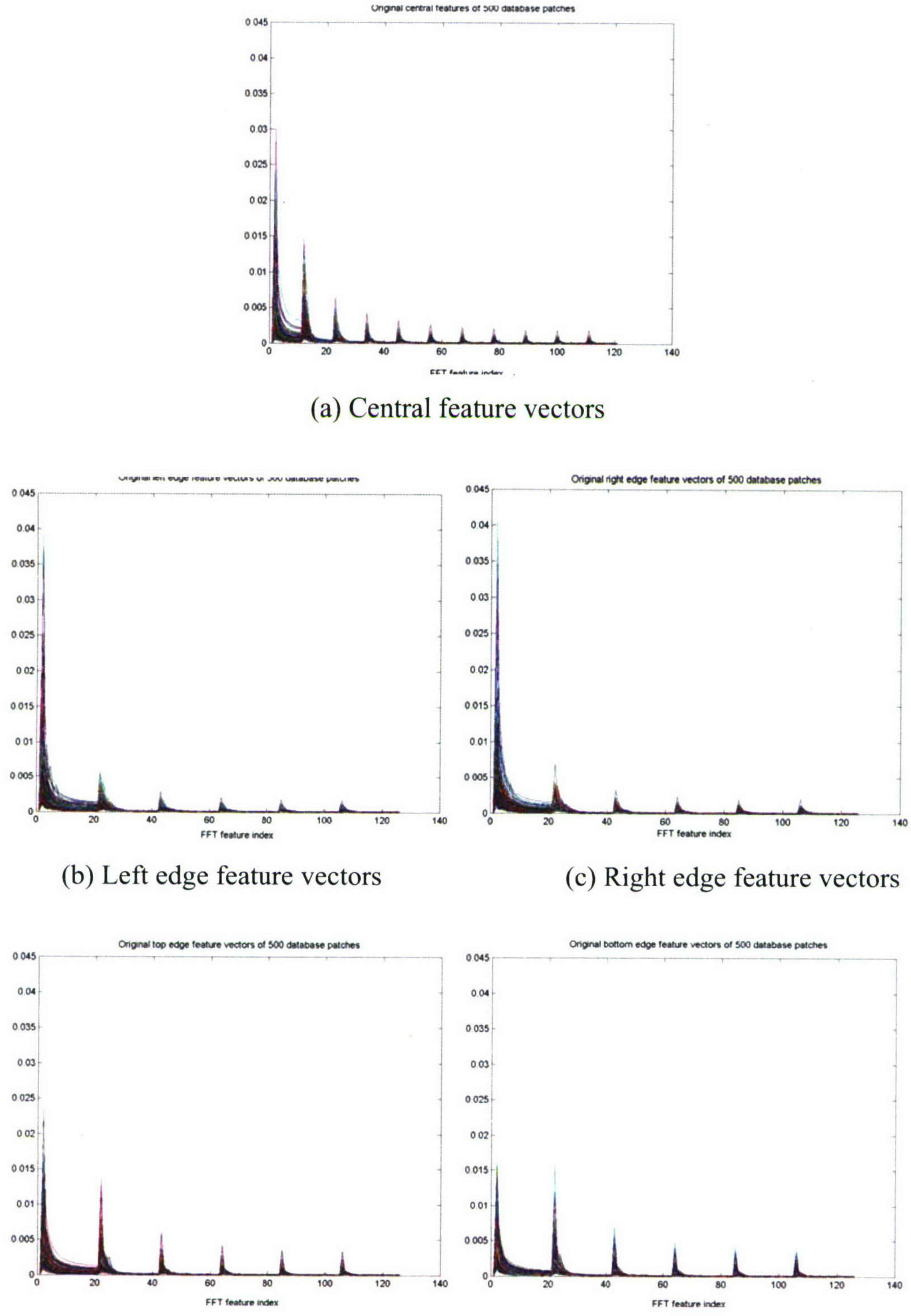
(b) Left edge feature vector

(c) Right edge feature vector

(d) Top edge feature vector

(e) Bottom edge feature vector

Figure 3.3 Original FFT Feature Vector for One Database Patch (Mine Area – 01)



(a) Central feature vectors

(b) Left edge feature vectors

(c) Right edge feature vectors

(d) Top edge feature vectors

(e) Bottom edge feature vectors

Figure 3.4 FFT Feature Vectors for a Set of 500 Randomly Selected Database Patches (Mine Area – Area 01)

The length of the feature vector obtained above, increases as the patch area under consideration increases. In addition, not all the features may be equally important in characterizing the patch area. In the other words, since the feature vector represents the distribution of energy of the patch area in frequency domain, most of the energy of the given feature vector is mainly concentrated in few features. Thus, it is sufficient to retain only those features with the most energy to get a compact representation of given image area. Moreover, reduction in number of features also reduces computational time and storage space. To accomplish this task a Karhunen-Loeve Transform (KLT) (also called Hotelling Transform or Principal Component Analysis (PCA)) is used.

3.3. KL TRANSFORM FOR FEATURE REPRESENTATION

The KLT is a linear transform, in which the basis functions are estimated from the statistics of the signal. KLT gives optimal representation in the sense that; it places as much energy as possible in fewest coefficients. This approach minimizes the mean square error between the real vector and its approximation. The minimization is done by transforming the base axis of the present vector or signal to the new dimensions in which contribution of one or more dimensions is small enough to neglect. The generation of the KLT matrix is explained below. Let

$$X_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{in} \end{bmatrix},$$

represents a feature vector of size $n \times 1$, where n represents the number of features in each feature vector. Given M number of such vectors (M >> n), the *covariance matrix* C_X is calculated as [20]:

$$C_X = \frac{1}{M}\sum_{k=1}^{M} X_k X_k^T - m_X m_X^T, \qquad (1)$$

$$m_X = \frac{1}{M}\sum_{k=1}^{M} X_k \qquad (2)$$

C_X and m_X are matrices of size $n \times n$ and $n \times 1$, respectively. The covariance matrix obtained using the eqn (1), is real and symmetric. Eigenvectors and corresponding eigenvalues of C_X are formed using the following condition:

$$C_X e_i = \lambda_i e_i, \qquad \text{for } i = 1, 2, 3\ldots \text{n} \tag{3}$$

where, e_i and λ_i represent eigenvectors and corresponding eigenvalues respectively. Now form a matrix A based on the eigenvectors of the covariance matrix C_X such that the first row of A corresponds to the highest eigenvalue and last row corresponds to the least eigenvalue. Eqn (3) can be formulated as:

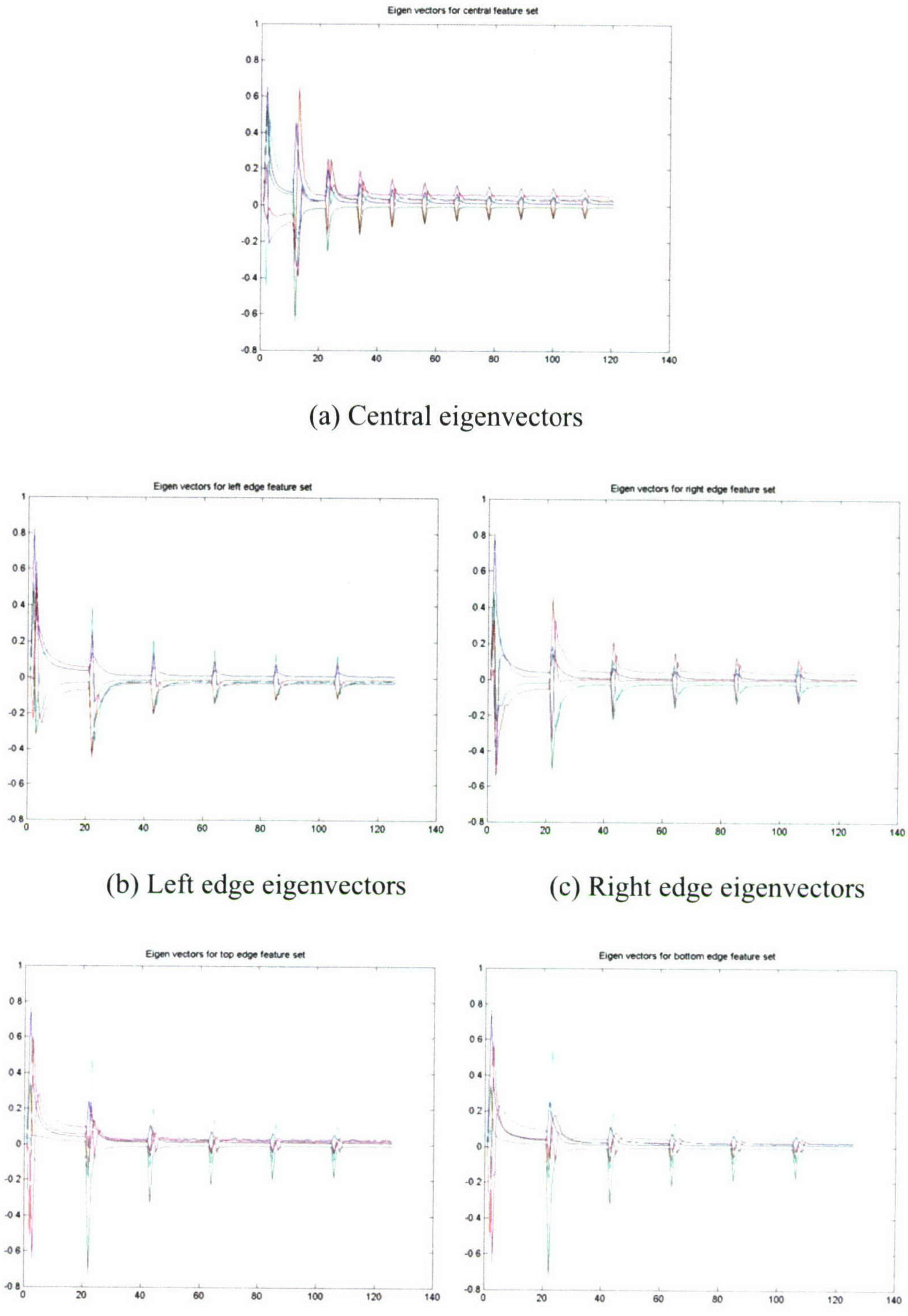
$$A C_X A^T = \Lambda \tag{4}$$

where, Λ is a diagonal matrix with eigenvalues in decreased order.

Now select a new matrix, A_P, of size $p \times n$ from A, such that p rows of A_P correspond to the p eigenvectors with p largest eigenvalues. For this particular case, the value of p is selected such that at least 96% of the total energy is retained and $p \geq 5$. The resultant KLT matrix A_P is used to reduce the number of features in the feature vector from n to p. Reduced feature vector of size $p \times 1$, which is used in all future calculations, is obtained by multiplying KLT matrix A_P of size $p \times n$ with the feature vector of size $n \times 1$. The reduced feature vector can be expressed as:

$$\hat{X}_i = A_p X_i \tag{5}$$

Figure 3.5 represents the five most prominent basis or eigenvectors derived from the set of central, left right, top, and bottom features shown in Figure 3.5. These five eigenvectors in each figure corresponds to the five largest eigenvalues. Figure 3.6 shows the central and edge feature vectors with reduced KLT features.



(a) Central eigenvectors

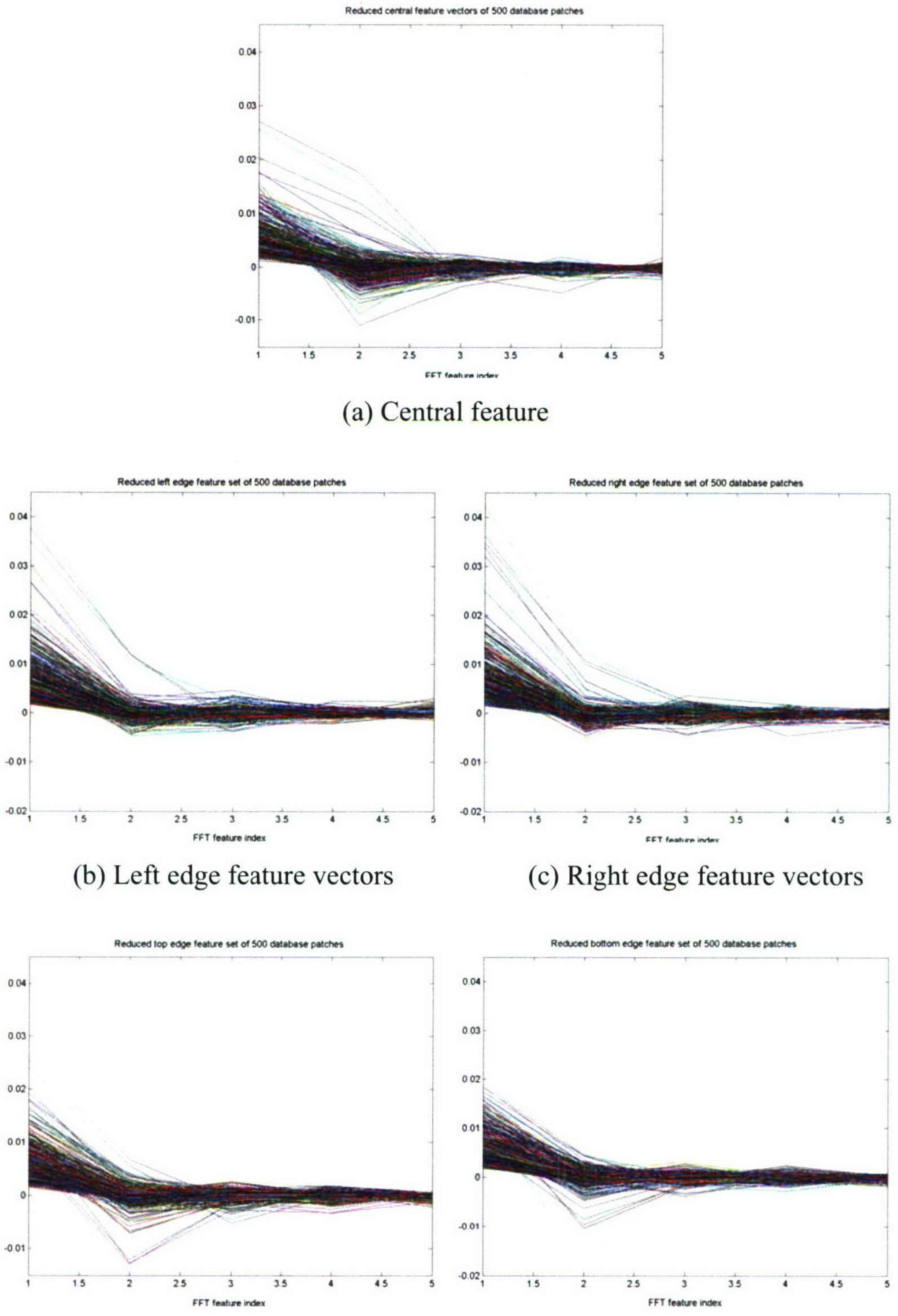
(b) Left edge eigenvectors

(c) Right edge eigenvectors

(d) Top edge eigenvectors

(e) Bottom edge eigenvectors

Figure 3.5 Eigenvectors Correspond to Five Largest Eigenvalues for FFT Features Shown in Figure 3.4



(a) Central feature

(b) Left edge feature vectors

(c) Right edge feature vectors

(d) Top edge feature vectors

(e) Bottom edge feature vectors

Figure 3.6 KLT Features for Database Patches Shown in Figure 3.4

All the features shown in Figure 3.6 obtained by using Eqn (5) and the original central and edge feature vectors shown in Figure 3.4, respectively. All new central and edge feature vectors, as shown in Figure 3.6, have only five features per feature vector. However, this may not always be the case.

3.4. QUAD TREE REPRESENTATION FOR PATCHES

The concept of KL transform discussed in previous section is aimed to decrease the memory requirements and the computation time of the synthesis algorithm. However, the KLT approach above proves ineffective as number of input images increases. Large number of input images is desired for high quality of synthesized image, since an increase in the number of input images gives more input patches, which in turn increases the probability of getting a suitable patch for a given criteria and eventually results in better synthesized image. At the same time, an increase in the number of patches results in more search time for the suitable patches that in turn results in more synthesis time. The high computational time can be overcome by employing better techniques in searching for the suitable patches, which is the focus of this section. The present approach uses a quad-tree based structure to overcome this problem.

Patch based synthesis algorithm synthesizes a new minefield by stitching one image patch at a time until the entire minefield is covered. In order to select appropriate patch for synthesis, two sets of image patches should be selected:

- Set of all patches whose center features are more likely to be adjacent to the neighborhood patch in the synthesized image.
- Set of all patches whose edge features are similar to the corresponding edge features of the neighborhood patches in the synthesized image.

To construct the first set, one needs to search all the available patches in the given database, which results in heavy computational cost and time. In this work, a quad tree representation of the patches is developed to overcome this problem. The quad-tree formation is based on the idea to divide all the available patches into different nodes based on the patch features. Thus the patches that are similar in their statistical characteristics are assigned to the same cluster. In the quad-tree implementation, each parent node has four children; hence, it is possible to divide the entire data set of patches into $k = 4^{level}$ clusters, where *level* represents the depth of the quad-tree (assuming base node is at the zero-level). A K-Mean clustering algorithm is used to form clusters and the corresponding codebooks.

3.4.1. K-Mean Clustering Algorithm. The main aim of this algorithm is to partition all the available feature vectors into k (k = 4level), clusters by minimizing the squared errors between the centroids [21]. At each level *l*, each of the KLT feature vector $\hat{X}_i$ is assigned to a cluster with cluster center c_j such that a squared error measure is minimized. The squared error at level *l* (ε_l^2) is calculated over the KLT feature vectors and is defined as:

$$\varepsilon_l^2 = \sum_{j=1}^{k}\sum_{i=1}^{M} \| \hat{X}_i - c_j^l \|^2 \tag{6}$$

where, $\hat{X}_i$ = i^{th} KLT feature vector,

$c_j^{\,l}$ = Cluster center (vector) for j^{th} cluster at level *l*,

M = Total number of feature vectors,

$\|.\|$ = Represents Euclidian distance between vectors $\hat{X}_i$ and $c_j^{\,l}$

The main steps involved in this algorithm are:

1. Randomly choose 4 centroids in the space represented by feature vectors.
   a) Assign all the feature vectors to these 4 centroids based on the nearest neighbor computation.
   b) Once all the vectors are assigned, recalculate the 4 centroids.
   c) Repeat (a) and (b) till all the centroids are stabilized and store the respective centroids.
2. Check the number of feature vectors in each cluster. If this number is greater than the minimum population, which is defined as minimum number of feature vectors per cluster, declare that cluster as parent and split into 4 different clusters as explained in step 1. If the number of features is less than the minimum population, declare that cluster as child and go to the next parent

cluster. This process continues until all the clusters become child nodes or the level reaches its maximum value.

Figure 3.7 represents a quad-tree with *level* = 2. Each level contains 4^{level} nodes. Each parent node contains a codebook with four centroids, and the child node contains all the patch identities belonging to that node. The numbers (1, 11, 123...etc) inside the node, shown in Figure 3.7, represent the cluster identification number at each node.

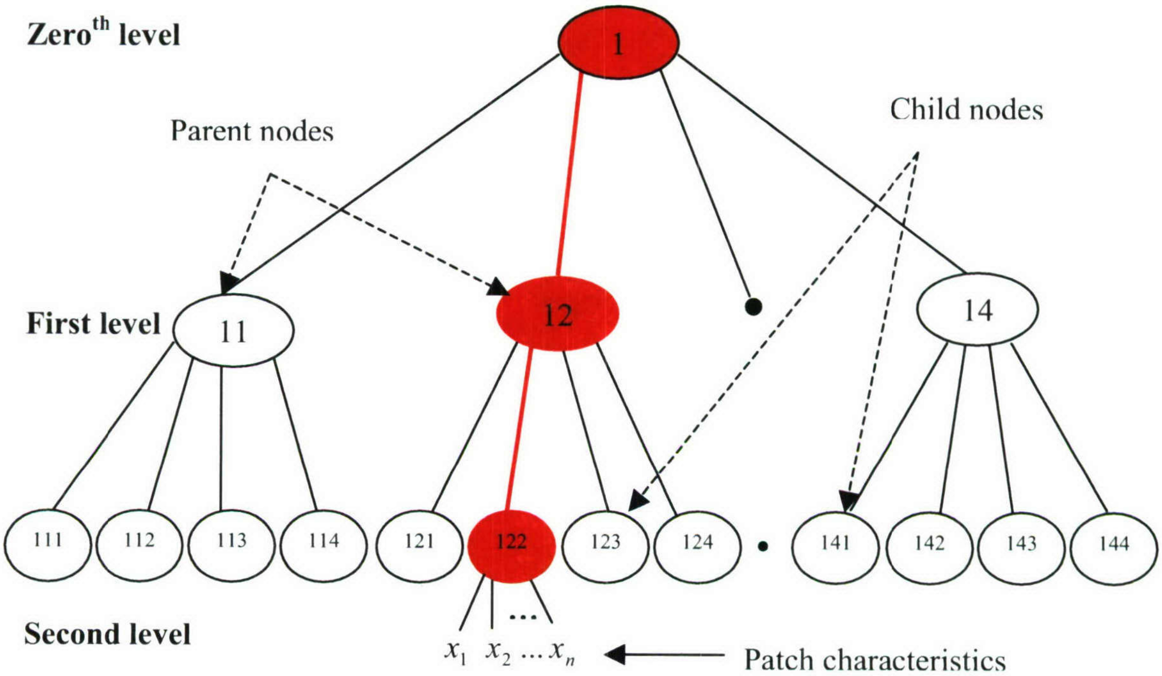


Figure 3.7 Pictorial Representation of the Quad-tree Structure with 2 Levels

The assignment of these identification numbers is as follows: Initially an ID of one is given to the node at the Zeroth level. From this level onwards, for each node, one digit (from 1 to 4) is appended from the right to the previous ID depending on the location of the current node with respect to its parent node. This process continues until either the child node or the last level is encountered. For example, consider a node at level *l*, the identification number that corresponds to this node contains (*l*+1) digits. The

$(l+1)^{th}$ digit represents the location of the current node at l^{th} level with respect to its parent at $(l-1)^{th}$ level.

3.4.2. Vector Quantization. Once the quad tree is formed and the codebooks are created, a Vector Quantization [22] technique is used to assign all the patches to the appropriate cluster based on the nearest neighborhood computation. Using this approach, each characteristic (centre, left edge, top edge etc) of a patch is assigned to the appropriate quad-tree's (central, left edge, top edge etc) child cluster whose center is nearest to the patch characteristic. The assignment process for each patch is summarized below,

1. Start from the Zeroth level of the quad-tree; find a cluster whose center is closest to the incoming patch characteristic using nearest neighborhood computation.
2. Verify if the above cluster is a parent or a child. If it is a parent, find the nearest child cluster under that cluster. If the cluster found is a child, stop the process and assign the child cluster ID to the patch characteristic.
3. Repeat step 2 till the process encounters a valid child cluster.
4. Repeat steps 1 to 3 for each database patch.

For this thesis, three quad-trees, each with six levels, are created for each available terrain. Out of these three, one tree is based on the central characteristics and the other two are based on the top and left edge characteristics. Bottom and right edge feature IDs are assigned based on top and left edge trees respectively.

3.4.3. Branch of a Quad-tree (B_Q^q). Branch of a quad-tree Q for node q represents the path in a quad-tree structure from parent node at zero level to the node q and is represented as B_Q^q. The red path shown in Figure 3.7 represents the branch of a quad-tree for the node whose ID is q =122, and $x_1\ x_2 \ldots x_n$ represent the patch IDs associated with that node. For any node q in quad-tree structure, branch of a quad-tree includes all the parent nodes, at different levels, in between the parent node at zero level to the present node q. Figure 3.7 shows a branch for node 122 with only one parent node with ID equal to 12 (at first level) in between the node 1 and node 122.

3.5. MARKOV RANDOM FIELDS (MRF) MODEL

So far the fundamental elements of patch based modeling and some of the optimization techniques, which are aimed to improve the performance of minefield synthesis, have been discussed. However, to retain the natural structure of synthesized image a proper mathematical model for the background is desired. This section presents the mathematical model that is considered in synthesizing the new minefield images. Figure 3.8 shows the patch based lattice structure for image. Each lattice point over the image lattice represents the valid site that needs to be considered for the synthesis. These lattice points also represent the center of the patches that are synthesized later. Lattice points are spaced by $(W_B - W_E)$ in both vertical and horizontal direction. In this thesis, mathematical model based on the Markov random field model is considered in synthesizing minefields.

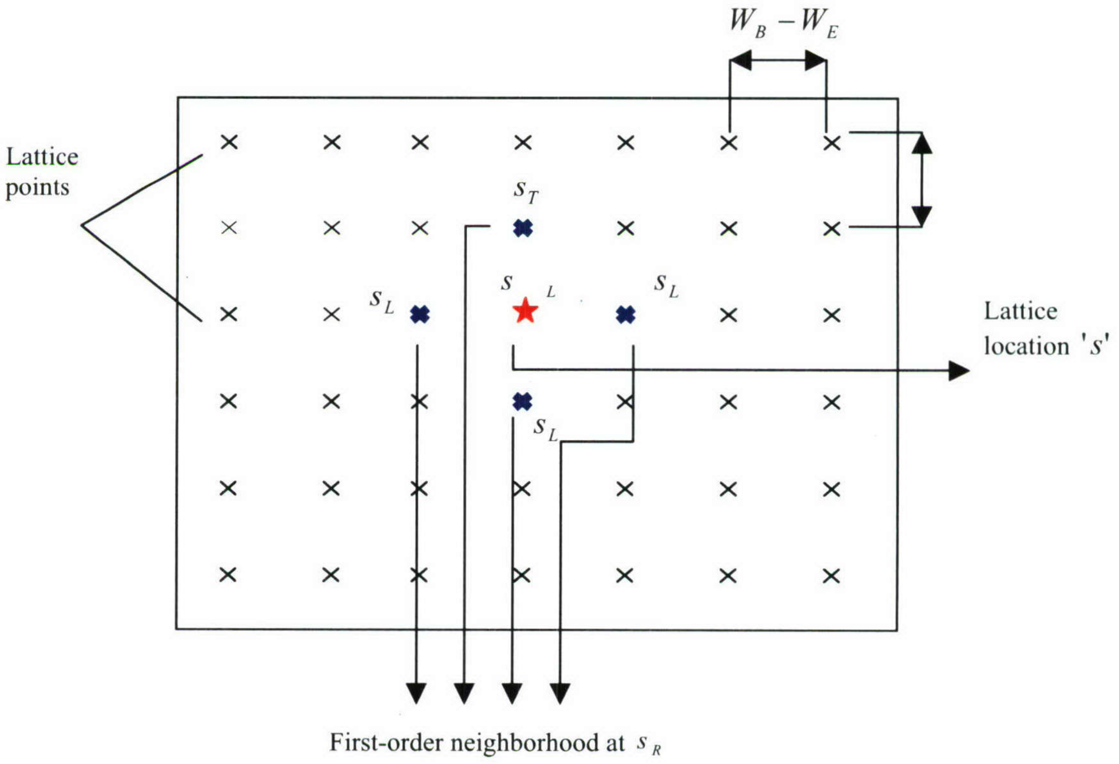


Figure 3.8 Patch-based Lattice Structure for Image-based Modeling and Synthesis

Markov process is a very important statistical model in image synthesis because the model is able to represent spatial continuity of natural images. The concept of Markov process can be explained in the following way: Let 's' be any site on the image and Λ_s be the set of all possible values of site 's'. Then under Markov random model, a variable x_s at site 's' on an image can take value $X_s \in \Lambda_s$. The probability of $x_s \equiv X_s$ depends only on the values at the neighborhood lattice point $\{x_r = X_r; r \in N_s^o\}$.Where x_r is the value of the patch in the neighboring locations to 's' [7]. Thus, under MRF model

$$p(x_s = X_s) = p(X_s \mid X_r, r \in N_s^o, s \in S) \tag{7}$$

where,

S = Image domain,

X_s = Value of the image patch at site 's', and

$N_s^o = o^{th}$ order neighborhood of location $s = (i, j)$

N_s^o at location *(i, j)* is given as:

$$N_s^o = \{r = (k, l) \in S : (k - i)^2 + (l - j)^2 \le o^2\} \tag{8}$$

For the current implementation, first-order Markov model is assumed. Figure 3.8 shows the image patch at any particular site $s = (i, j)$ and its four neighborhood patches, whose centers are represented with thick cross marks. Thus, only points at the sites s_T = *(i, j-1)*, s_R = *(i+1, j)*, s_B = *(i, j+1)*, and s_L = *(i-1, j)* are included in the neighborhood of the lattice point at site $s = (i, j)$. The left upper corner of the image lattice is assumed to be the origin. Hence, the characteristics of the selected patch at the location $s = (i, j)$ depend on the characteristics of the already synthesized patches (if any) at the locations s_T, s_R, s_B, and s_L.

Equation 7 gives the *local conditional probability density function* (LCPDF) with respect to the neighborhood N_s. To model a minefield as MRF, each patch at 's' is considered a site in the image domain *S*. The site value x_s is contained in the state space Λ_S that consists of all the available patches from the pre-acquired image data and their associated characteristics. Under the MRF model, select a realization of $X \in \Omega$ such that *X* has the highest probability in stochastic sense in Ω. Where Ω is called a *configuration space* consist of all realizations {*X*} for synthesized background and $X = \{x_s; \forall\ s \in S\}$ represents the one realization of the desired image. This means that the realization of X should produce the minimum 'cost function'. The cost function is defined as the visual perceptual difference between the training image and its synthesized counterpart. The configuration space for the realization of the image is defined by the set of patches at each site and set of all such possible realizations of the image X.

3.6. MODELING OF CENTRAL FEATURES

For the present work, the central characteristics of the patches are modeled as MRF. For each background, a model is generated based on the neighborhood characteristics as well as the central quad-tree. A neighborhood model is defined as a model that contains the first-order neighborhood information for each database patch. This neighborhood information is obtained from original imagery from which the patches are extracted. A central quad-tree, as described in Section 3.3, is a tree structure created based on the central characteristics of all the available patches, which belong to a unique background.

The model generated this way helps in finding the most probable patches that can be placed adjacent to the present patch. This model is represented in the form of a probability matrix for each level of central quad-tree. At each level, a probability matrix of size $k \times k$ ($k = 4^l$) is generated. The p^{th} row q^{th} column element of the probability matrix, at each level, represents the conditional probability of a patch from node *q* given a patch from p^{th} node is the neighborhood. The conditional probability for a patch from *q* at level *l* of the central quad-tree is calculated as the ratio of the number of patches, from node *q*, that have neighborhood patches from node *p* to the total number of patches from node *p*. The probability distribution at each level can be expressed as:

$$p^{l}_{pq} = \frac{t^{l}_{pq}}{T^{l}_{p}} \tag{9}$$

where,

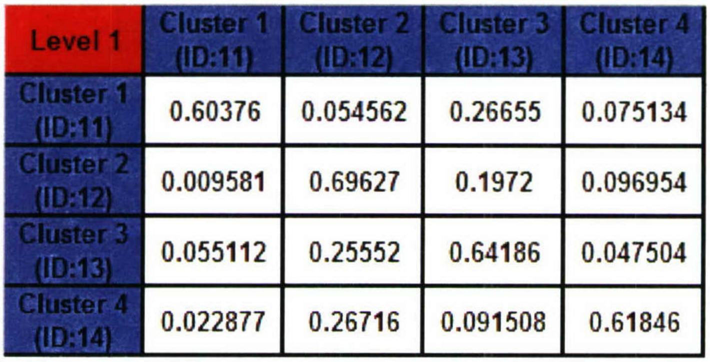
T^{l}_{p} = Total number of patches assigned to the node p at level l,

t^{l}_{pq} = Total number of patches from T^{l}_{p}, which have an appropriate neighbor patch from node q,

p^{l}_{pq} = Probability of the patch from node q at a location, given a patch from node p in the neighborhood

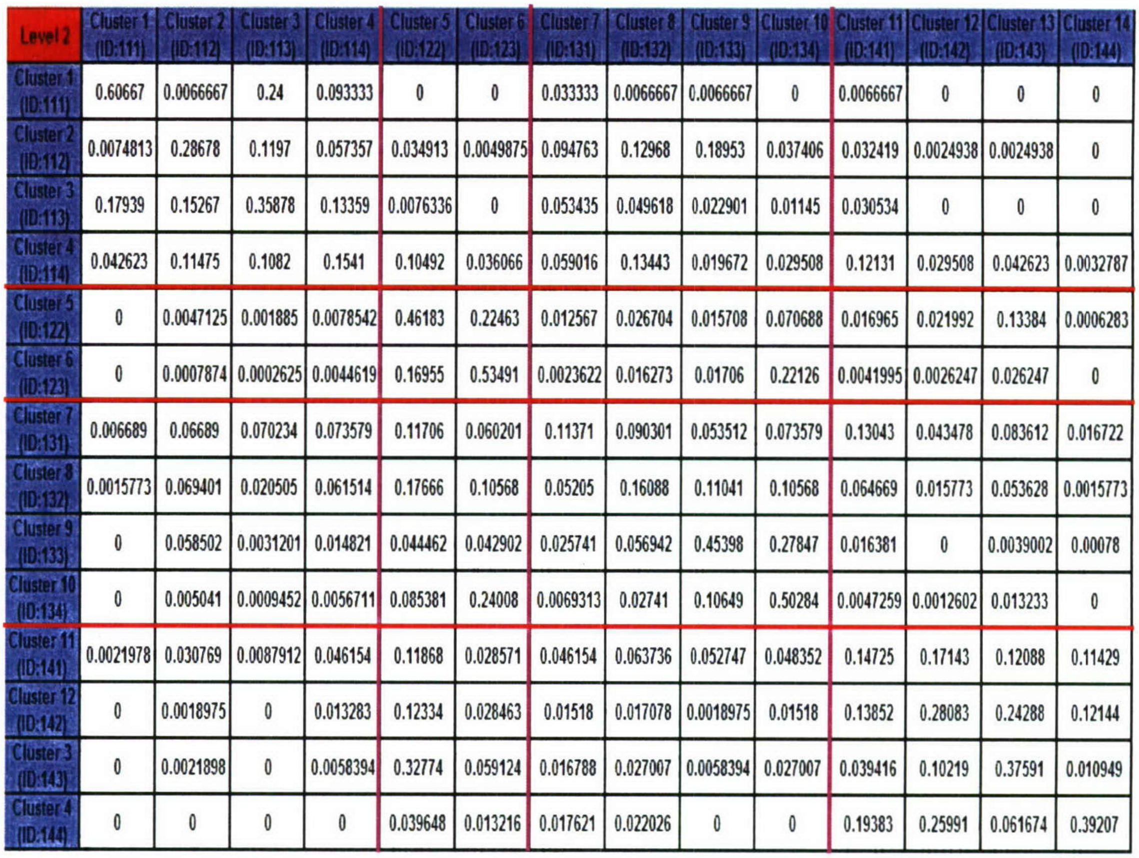
Tables 3.1 and 3.2 represent the example matrices for level 1 and level 2 of the MRF model. These example matrices are taken from the background model generated based on the Mine area (01) and represent model for central quad-tree structure. First row and column in each table represent the number of clusters and their identification numbers (ID) present in central quad-tree structure. Each row represents the selection probabilities of other clusters that can be selected as neighborhood to the cluster shown in the first column of that row. Thus, the sum of all probabilities at each row in the probability matrix is equal to one. From the first row of the Table 3.1 (row corresponding to the *Cluster 1*- ID: 11), we can say that the probability of *Cluster 1* (ID: 11) being a neighborhood cluster to the *Cluster 1* (ID: 11) is high (0.60376) as compared with the other clusters in that level.

Table 3.1 Probability matrix for first level (Mine Area – 01)



Level 1	Cluster 1 (ID:11)	Cluster 2 (ID:12)	Cluster 3 (ID:13)	Cluster 4 (ID:14)
Cluster 1 (ID:11)	0.60376	0.054562	0.26655	0.075134
Cluster 2 (ID:12)	0.009581	0.69627	0.1972	0.096954
Cluster 3 (ID:13)	0.055112	0.25552	0.64186	0.047504
Cluster 4 (ID:14)	0.022877	0.26716	0.091508	0.61846

Table 3.2 Probability matrix for second level (Mine Area – 01)



Level 2	Cluster 1 (ID:111)	Cluster 2 (ID:112)	Cluster 3 (ID:113)	Cluster 4 (ID:114)	Cluster 5 (ID:122)	Cluster 6 (ID:123)	Cluster 7 (ID:131)	Cluster 8 (ID:132)	Cluster 9 (ID:133)	Cluster 10 (ID:134)	Cluster 11 (ID:141)	Cluster 12 (ID:142)	Cluster 13 (ID:143)	Cluster 14 (ID:144)
Cluster 1 (ID:111)	0.60667	0.0066667	0.24	0.093333	0	0	0.033333	0.0066667	0.0066667	0	0.0066667	0	0	0
Cluster 2 (ID:112)	0.0074813	0.28678	0.1197	0.057357	0.034913	0.0049875	0.094763	0.12968	0.18953	0.037406	0.032419	0.0024938	0.0024938	0
Cluster 3 (ID:113)	0.17939	0.15267	0.35878	0.13359	0.0076336	0	0.053435	0.049618	0.022901	0.01145	0.030534	0	0	0
Cluster 4 (ID:114)	0.042623	0.11475	0.1082	0.1541	0.10492	0.036066	0.059016	0.13443	0.019672	0.029508	0.12131	0.029508	0.042623	0.0032787
Cluster 5 (ID:122)	0	0.0047125	0.001885	0.0078542	0.46183	0.22463	0.012567	0.026704	0.015708	0.070688	0.016965	0.021992	0.13384	0.0006283
Cluster 6 (ID:123)	0	0.0007874	0.0002625	0.0044619	0.16955	0.53491	0.0023622	0.016273	0.01706	0.22126	0.0041995	0.0026247	0.026247	0
Cluster 7 (ID:131)	0.006689	0.06689	0.070234	0.073579	0.11706	0.060201	0.11371	0.090301	0.053512	0.073579	0.13043	0.043478	0.083612	0.016722
Cluster 8 (ID:132)	0.0015773	0.069401	0.020505	0.061514	0.17666	0.10568	0.05205	0.16088	0.11041	0.10568	0.064669	0.015773	0.053628	0.0015773
Cluster 9 (ID:133)	0	0.058502	0.0031201	0.014821	0.044462	0.042902	0.025741	0.056942	0.45398	0.27847	0.016381	0	0.0039002	0.00078
Cluster 10 (ID:134)	0	0.005041	0.0009452	0.0056711	0.085381	0.24008	0.0069313	0.02741	0.10649	0.50284	0.0047259	0.0012602	0.013233	0
Cluster 11 (ID:141)	0.0021978	0.030769	0.0087912	0.046154	0.11868	0.028571	0.046154	0.063736	0.052747	0.048352	0.14725	0.17143	0.12088	0.11429
Cluster 12 (ID:142)	0	0.0018975	0	0.013283	0.12334	0.028463	0.01518	0.017078	0.0018975	0.01518	0.13852	0.28083	0.24288	0.12144
Cluster 3 (ID:143)	0	0.0021898	0	0.0058394	0.32774	0.059124	0.016788	0.027007	0.0058394	0.027007	0.039416	0.10219	0.37591	0.010949
Cluster 4 (ID:144)	0	0	0	0	0.039648	0.013216	0.017621	0.022026	0	0	0.19383	0.25991	0.061674	0.39207

3.7. IMAGE DATABASE

The image database deals with the information regarding patches that are extracted from different terrain such as Mine area, Rock 41, Rock 42, Rock 44, and Wash 5. These extracted patches, for each terrain, are again categorized into three different groups such as background-only patches, clutter patches and mine patches, based on the content of the image patch. For this thesis, clutter patches are not considered for synthesis. The detailed description of these groups of patches used in the image database is given below:

3.7.1. Background-only Patches.

A background-only patch is defined as an image patch without mines and clutter objects. To minimize the memory requirement in storing image patches, the raw images of the database patches are not stored. Instead each background patch is identified by the starting row and column position, file name, and the path of the input image from which the patch is extracted. Along with this information, each database patch is associated with the five identification (ID) numbers for the central, and the four edge characteristics that belong to the three corresponding quad-tree structures. For example, the central ID of a particular database patch gives the location of the child cluster to which the central characteristic of that patch is assigned in central quad-tree structure. These identification numbers helps in fast extraction of the appropriate patches later in the synthesis procedure. The detailed description on how these identification numbers are used in the selection process is explained in Section 4.1. In order to save search time during synthesis, the IDs of all the patches are pre-computed and are stored along with the other patch information. A vector quantization technique, explained in Section 3.3.2, is used to assign the identification number for all five characteristics of database patch. The identification numbers related to the center, the left edge and the top edge characteristics of database patch are directly derived from the corresponding quad tree structures. Identification numbers for the right edge and bottom edge characteristics are derived from the left edge and top edge quad tree structures respectively.

3.7.2. Mine Patches. This part of the image database contains the information regarding each individual mine patch. A mine patch is defined as an image patch with a mine at its center and is primarily identified by the type of the mine (such as MP_A, SM_A, LP_B, and LM_A, where MP stands for medium plastic, SM stands for small metallic, LP stands for large plastic and LM stands for large metallic mines) present in the middle of the mine patch. Like background-only patches, mine patches are also identified by starting row and column position, file name, and path of the image frame from which the patch is extracted. The structure of the mine patch database is different from that of background patch database explained in previous section. The mine patches do not have identification numbers corresponding to the central characteristics. The identification numbers for all four edges of a mine patch are derived from background edge quad-trees that are derived from the background-only patches whose background is same as mine. The identification numbers for the left edge and right edge of a mine patch are derived from the left edge quad-tree, whereas the identification numbers for the top edge and bottom edge characteristics are derived from the top edge quad tree structure.

4. PATCH SELECTION AND SYNTHESIS

The basic idea behind patch-based synthesis is that, at each step, a patch from the image database is selected and pasted onto the synthesized image until the synthesized image is fully covered. In order to synthesize minefields without any visible artifacts, one should select adjacent image patches intelligently. This selection is affected by a number of factors such as the statistical characteristics of the patches that are already pasted, the terrain information provided by the terrain map, and the statistical model of the background at that location. Minefield synthesis based on such selection is visually rich as compared with one synthesized by using random selection of patches. This section is divided into six sub sections. Section 4.1 deals with the selection criterion for background patches and estimation of correlation between the adjacent patches. Suitable blending of the adjacent patches is discussed in Section 4.2. Section 4.3 deals with the minefield synthesis procedure on a regular image lattice. Placement of mine signature over synthesized background-only image is discussed in Section 4.4. A complete step-by-step algorithm for the proposed minefield synthesizer is presented in Section 4.5. Results for synthesis of minefield frames are presented in Section 4.6.

4.1. SELECTION OF SYNTHESIS PATCH

The selection of a patch at any particular location is completely dictated by the terrain characteristics at that particular location and the characteristics of the already synthesized neighborhood patches. This section presents the systematic procedure of the patch selection.

4.1.1. Selection of Patches Based on Terrain Characteristics. The terrain characteristics determine the type of the background patches that needs to be considered for the synthesis. These terrains characteristics are obtained from the terrain map that is given as one of the inputs to synthesis algorithm. The terrain map holds the information regarding the terrain structure of the desired synthesized minefield area. The background information presented in the terrain map is stored in the form of an indexed image and each background is identified by a unique index value. Set of database patches (Ψ_T)

whose background matches the background information of the terrain map at the site $'s'$ is obtained in the following way:

$$\Psi_T = \{x \mid T(x) \equiv T_s, x \in \Psi\}, \tag{10}$$

where,

$T(x) =$ Identity of background for the patch $x \in \Psi$

$T_s =$ Identity of background at site $'s'$ on the terrain map,

$\Psi =$ Image database created using input images $\{I_{IN}\}$

Note that, $\Psi_T \subseteq \Psi$

4.1.2. Selection of Patches Based on Central Characteristics. The central characteristics or central features of the synthesized neighborhood patches are important parameters for the selection process as it dictates the overall gray level consistency of the synthesized image. Given the central characteristic for the neighborhood patches, the synthesis algorithm picks the proper patches with appropriate central characteristics. The first-order Markov model for the background under consideration gives the most probable patches to be placed adjacent to the synthesized neighborhood patches. On the other hand, the central quad-tree, explained in Section 3.4, gives all the appropriate patches based on the desired central characteristics of the patch. Selection of the patches based on the central characteristics is summarized below:

1. Given the identity of the central characteristic for neighborhood patches; obtain all possible parent nodes. As explained in Section 3.3.2, all parent nodes to the child node in different levels can be obtained by inspecting the ID of the child node. These nodes are termed as *target central nodes* at different levels of the central quad-tree structure. For example, the child node ID 1231223 will have parent nodes with following IDs 12, 123, 1231, 12312, 123122 at each level (starting form level 1 to 5) of the 6-level central quad-tree structure. Note that there may be more than one neighborhood patches at the give site on the image lattice.

2. Initially consider the target central node at the second level ($l = 2$), and select a node at level l to which patch at present location should belong, based on MRF model probabilities at level l. The terrain model, created in 3.4.1, gives the appropriate neighborhood probabilities at each level. This step gives all possible patches, at level l, which can be placed adjacent to the present neighborhood patch along with the relative probabilities. If the numbers of patches obtained in this step are less than the required minimum number (for this work this required minimum number is set to 5) then the present process stops here and returns all the selected patches to the next level of selection process, else continue with step 3.
3. In this step, node that is selected in step 2 become parent nodes at level l+1 and the same process, as explained in step 2, continues till a child node is encountered or the number of selected patches is below a specified minimum. If the number of patches obtained in this step is less than the required minimum number then the process picks the patches from the node present one level above to the final selected node. Finally, the patches whose central IDs matches with the selected child node IDs are collected and are sent to the next level of the selection process.

This is a hierarchical process such that at each target central node a set of patches will be selected and the size of the set decreases as the process advances to the last node of the target central nodes. Finally the patches with high neighborhood probabilities are selected statistically and are passed to the next level of the selection process. The set of patches (Ψ_{CT}) selected based on the central characteristics of neighborhood patch can be expressed mathematically in the following way.

$$\Psi_{CT} = \{x \mid p(q_s) \overset{\Delta}{=} p(q_s \mid q_r ; \forall r \in N_s^o), x \in \Psi_T , x \in B_{CF}^{q_s} \} \qquad (11)$$

where,

q_s = ID of the node for central quad-tree selected as per conditional probability distribution $p(q_s) \overset{\Delta}{=} p(q_s \mid q_r ; \forall r \in N_s^o)$

q_r = Central ID for neighborhood patch at 'r'

$B_{CF}^{q_s}$ = Branch of the central quad-tree for the q_s node

Note that, $\Psi_{CT} \subseteq \Psi_T$

4.1.3. Selection of Patches based on Edge Characteristics. The edge characteristics of the synthesized patches are primarily used to minimize any anomalies that might result while blending the boundary zones. The edge characteristics of the synthesized neighborhood patches are used to select patches whose edge characteristics match the previously synthesized neighborhoods. Given the edge characteristics of the synthesized neighborhood patches, the selection process searches the previously formed set (Ψ_{CT}) for the patches with edge characteristics similar to the neighborhood. Selection of the patches based on the edge characteristics is summarized below:

1. Given the edge identity of the node to which the edge characteristic of neighborhood patches belong; obtain all possible parent nodes to that child node and represent them as *target edge nodes.* Note that there may be more than one neighborhood patches at the give site on the image lattice.
2. Consider the first target edge node, which is at level *l* and find all patches below this. Finally, find the common patches between the patches selected in this step and patches from Ψ_{CT}. If the number of common patches is less than the required minimum number, stop search and send the resultant common patches to the next level of the synthesis algorithm.
3. If the number of patches is greater than the minimum required number then select the next parental node from the target edge node list and repeat step 2.

Like the previous case, the present selection process here is also hierarchical such that at each target edge node, a set of patches from Ψ_{CT} will be selected and the size of this set decreases as the process advances to the child node of the target central node. The set of patches (Ψ_{ECT}) selected based on the edge characteristics of neighborhood patch can be expressed mathematically in the following way.

$$\Psi_{ECT} = \{x \mid x \in \Psi_{CT}, x \in \cap\{B_{EF}^{q_r}, \forall r \in N_s^o\}\} \tag{12}$$

where,

q_r = Edge ID for neighborhood patch at 'r'

$B_{EF}^{q_r}$ = Branch of a edge quad-tree for a node with ID of q_r

Note that, $\Psi_{ECT} \subseteq \Psi_{CT}$

4.1.4. Selection of a Patch Based on Correlation. It may be noted that the patches selected based on central and edge characteristics are actually database patches. Within the database patch, each possible synthesis patch may have large variation in the mean gray value and may not possess very good correlation with its neighbors for all possible synthesis patches. A randomly selected synthesis patch may results in artifacts on the synthesized image. To overcome this problem, one more level is added to the synthesis process. The main purpose of this level is to select a synthesis patch that has best possible correlation with its neighborhood patches. The search for highly correlated synthesis patch for synthesis is summarized below:

1. Select a database patch from Ψ_{ECT}. Extract the edge region of the neighborhood patch (patches) from the synthesized image and slide over a selected database patch by storing all the correlation coefficient values between database patch edge region and neighborhood patch edge region, and their corresponding synthesis patches. Find a patch from the above stored list of synthesis patches that has the maximum correlation coefficient. If this value is greater than or equal to the predefined threshold, stop the search and return the corresponding synthesis patch as the final patch for synthesis.
2. If the maximum value is less than the threshold selected in step 1, pick the next database patch from Ψ_{ECT} and repeat step 1. This process continues until a synthesis patch whose edge region has a correlation coefficient greater than the predefined threshold with the neighborhood edge region is found. However, there exists a case where all correlation coefficient may not exceed

the threshold, in which case, synthesis patch with the highest correlation coefficient will be selected.

The correlation coefficient (r_{ij}) between the edge region of i^{th} synthesis patch extracted from j^{th} DP and the neighborhood edge region extracted from the synthesized image is calculated using the following formula:

$$r_{ij} = \frac{(A_{ij} - m_{A_{ij}})^T (B - m_B)}{\sqrt{(A_{ij} - m_{A_{ij}})^2 \times (B - m_B)^2}} \tag{13}$$

where,

A_{ij} = i^{th} synthesis patch edge region extracted from j^{th} database patch,

$m_{A_{ij}}$ = Mean of A_{ij},

B = Neighborhood edge region extracted from the synthesized image,

m_B = Mean value of B.

Figure 4.1 shows the hierarchical representation of the selection process explained in Section 4.1. As shown in the figure, each stage of the selection hierarchy is associated with a specific criterion such that, at each stage number of database patches, starting from the set of all input patches (ψ), are reduced based on the stage criterion and pass the resultant patches to the next stage for further processing. Final set of database patches (ψ_{ECT}) obtained after the last stage of the selection process contains patches that satisfy all the conditions such as the terrain characteristics, the central characteristics, the statistical model of the background, and the edge characteristics with its neighborhood synthesized patches. Finally, a synthesis patch is selected from ψ_{ECT} based on the best correlation with the synthesized neighborhood.

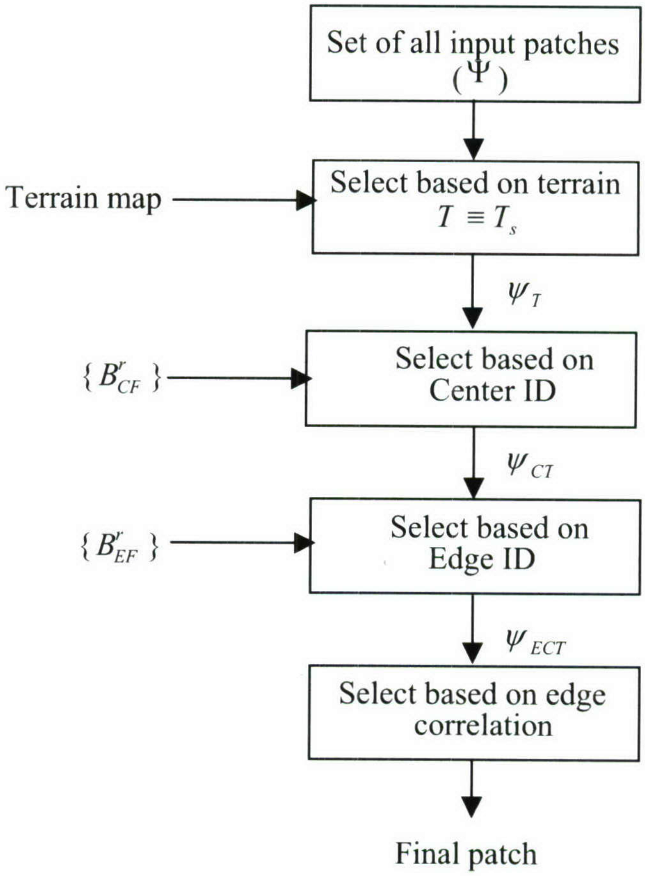


Figure 4.1 Flow Diagram of the Selection Process

4.2. BLENDING OF BACKGROUND-ONLY PATCHES

Blending is a process mainly aimed to reduce the discontinuities in intensity and color between the images being blended. This step provides a smooth transition between adjacent image patches. This operation is performed after the final patch that satisfies the previous selection criterions. A simple feathering algorithm [11] is used to blend the two images.

This algorithm is based on the concept of weighted average. In this framework weight matrix is formed based on the Euclidean distance. A weight is assigned to each pixel in the patch. This weight is proportional to the distance of the pixel to the boundary of the patch. Figure 4.2 shows the computation of the distance metric (*d*). As shown in the diagram the nearest distance to the edge (d_1) is chosen amongst the four highlighted distances (d_1, d_2, d_3, d_4). The purpose of assigning this weight distribution is to reduce the intensity contribution of the pixels that are close to the boundaries. The following equation is used to blend the given two images after the weight matrix is constructed:

$$I_0(x,y) = \frac{\sum_k d_k(x,y) I_k(x,y)}{\sum_k d_k(x,y)} \tag{14}$$

where $I_k(x,y)$ is the gray value of the pixel at (x, y) for the k^{th} image (since two images are used in blending $k \in \{1,2\}$), $I_0(x,y)$ is the gray value of the resultant image.

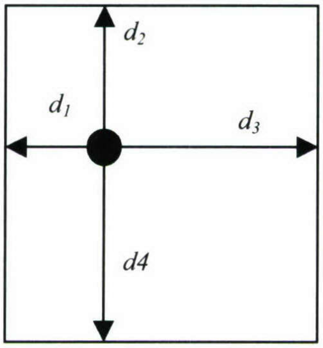


Figure 4.2 Calculation of Weight Value for Feathering Algorithm

4.3. SYNTHESIS OF BACKGROUND-ONLY FRAME

This section deals with synthesis of the background-only images. The background structure of the newly synthesized image completely depends on the terrain map. As explained in Section 3.4, terrain map that needs to be synthesized is divided into lattice points. Each lattice point over the image lattice represents the valid site that needs to be considered for the synthesis. Lattice points are spaced by $(W_B - W_E)$ in both vertical and horizontal direction. Here, W_B represents size of the synthesis patch (SP) and W_E represents the overlap between the neighborhood synthesis patches. The synthesis algorithm synthesizes each lattice point from left to right and top to bottom based on the terrain information at these lattice points. The entire background synthesis procedure can be divided into four different scenarios. The summary of each of these scenarios is discussed in the following sub-sections.

4.3.1. Placing the First Image Patch. Selection of the first patch depends only on the terrain information at the first lattice point. In this case, the algorithm randomly picks an image patch corresponding to the background information and places it on the upper left most corner of the image to be synthesized as shown in Figure 4.3.

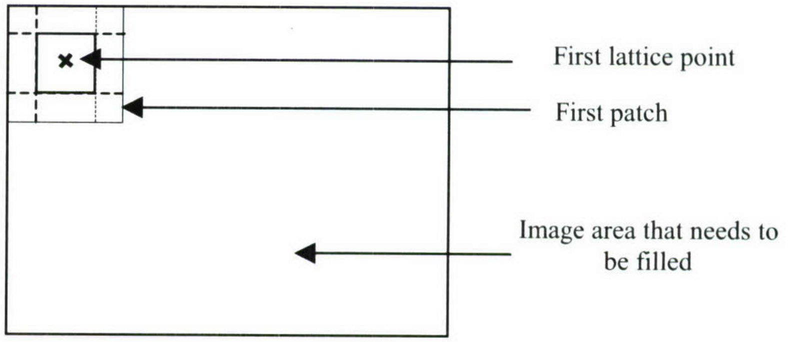


Figure 4.3 Placement of the First Patch on Image Lattice

At this stage, set of patches (Ψ_{ECT}) are selected based on the terrain information at present lattice point only.

$$\Psi_{ECT} = \Psi_T = \{x \mid T(x) \equiv T_s, x \in \Psi\} \qquad (15)$$

4.3.2. Placing the First Row of Patches. At this stage, selection of the next patch depends on the terrain information of the lattice point at the present location (T_s) as well as the image patch immediately to the left (x_L) (shaded with different color) of that location as indicated in Figure 4.4.

The set of patches (Ψ_{ECT}) are selected such that the central characteristics of the left patch are in confirmation with the MRF model of the background. Also, the left edge characteristics of the selected patches match with the right edge characteristics of the left patch (x_L). This can be represented mathematically in the following way.

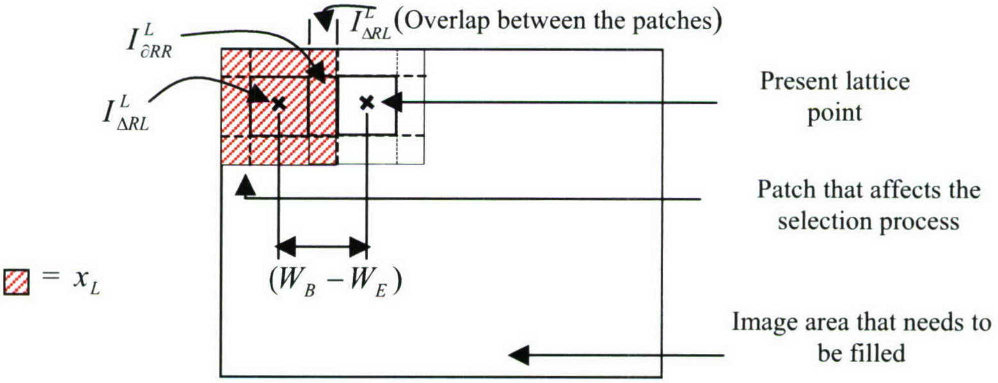
$$\left.\begin{aligned} &\Psi_T = \{x \mid T(x) \equiv T_s, x \in \Psi\}, \\ &\Psi_{CT} = \{x_1 \mid p(q_s) = p(q_s \mid q_{CL}), x_1 \in \Psi_T, x_1 \in B_{CF}^{q_s}\}, \\ &\Psi_{ECT} = \{x_2 \mid x_2 \in \Psi_{CT}, x_2 \in B_{LEF}^{q_{LE}}\}, \end{aligned}\right\} \tag{16}$$

where,

q_{CL} = Central ID of the left neighborhood patch x_L.

$B_{CF}^{q_s}$ = Branch of a central quad-tree structure for node ID q_s

$B_{LEF}^{q_{LE}}$ = Branch of a left edge quad-tree structure for patch x_2 with ID $= q_{LE}$.



$I_{\Delta RL}^{L}$ (Overlap between the patches)
$I_{\partial RR}^{L}$
$I_{\Delta RL}^{L}$
Present lattice point
Patch that affects the selection process
▨ = x_L
$(W_B - W_E)$
Image area that needs to be filled

Figure 4.4 Placement of the Patch on the First Row of the Image Lattice

4.3.3. Placing the First Column of Patches. At this stage, selection of the next patch depends on the terrain information of the lattice point at the present location (T_s) as well as on the patch that was pasted on top of the present location (x_T). Figure 4.5 gives the pictorial representation of this idea.

The set of patches (Ψ_{ECT}) are selected such that the central characteristics of the top patch are in confirmation with the MRF model of the background. Also, the top edge characteristics of the selected patches match with the bottom edge characteristics of the top patch (x_T). This can be represented mathematically in the following way.

$$\left.\begin{aligned} &\Psi_T = \{x \mid T(x) \equiv T_s, x \in \Psi\}, \\ &\Psi_{CT} = \{x_1 \mid p(q_s) = p(q_s \mid q_{CT}), x_1 \in \Psi_T, x_1 \in B_{CF}^{q_s}\}, \\ &\Psi_{ECT} = \{x_2 \mid x_2 \in \Psi_{CT}, x_2 \in B_{TEF}^{q_{TE}}\}, \end{aligned}\right\} \tag{17}$$

where,

q_{CT} = Central ID of the top neighborhood patch x_T

$B_{LEF}^{q_{TE}}$ = Branch of a top edge quad-tree structure for patch x_2 with ID $= q_{TE}$.

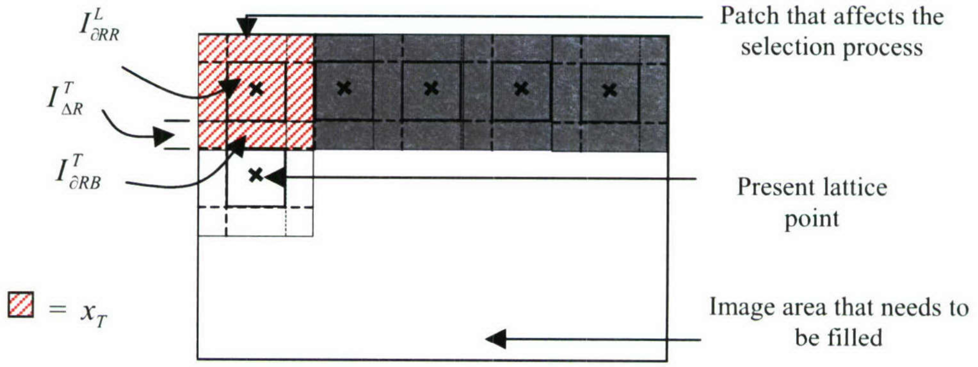


Figure 4.5 Placement of the Patch on the First Column of the Image Lattice

4.3.4. Placing the Remaining Patches. At this stage, selection of the next suitable patch depends on the terrain information of the lattice point at that location (T_s), and the left (x_L) and top (x_T) patches of the current location. Figure 4.6 gives pictorial representation of this idea.

Unlike the previous two cases, the selection of the patch for the present location depends on the two synthesized patches. In this case, the selection of patches (Ψ_{ECT}) are made such that the central characteristics of the selected patches and the central characteristics of the already synthesized patches located at the top and to the left of the present location are jointly in conformation with the respective MRF model for the given background . Along with this, the left and top edge characteristics of the selected patches also match the right edge characteristics of the synthesized left patch and bottom edge characteristics of the synthesized top patch in the neighborhood of the present location respectively. This can be represented mathematically in the following way:

$$\left.\begin{aligned}
&\Psi_T = \{x \mid T(x) \equiv T_s, x \in \Psi\},\\
&\Psi_{CT} = \{x_1 \mid p(q_s) = p(q_s \mid q_{CL}, q_{CT}), x_1 \in \Psi_T, x_1 \in B_{CF}^{q_s}\}\\
&\Psi_{ECT} = \{x_2 \mid x_2 \in \Psi_{CT}, x_2 \in \{B_{TEF}^{q_{CT}} \cap B_{LEF}^{q_{CL}}\}\},
\end{aligned}\right\} \tag{18}$$

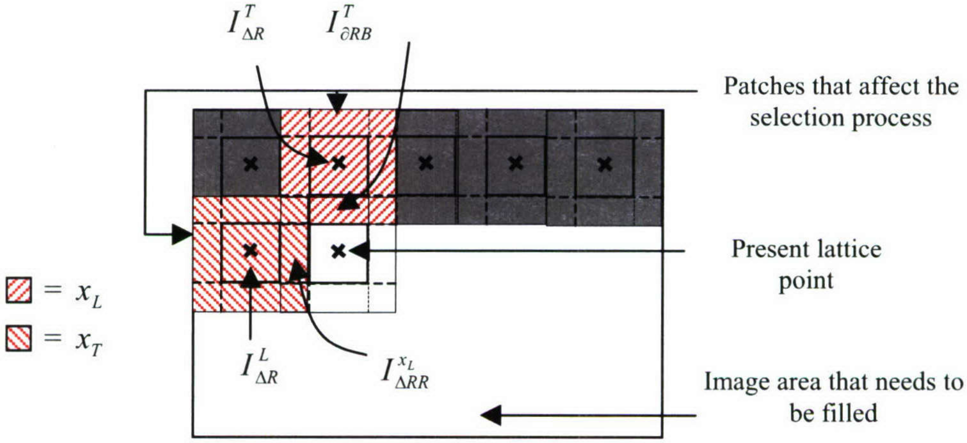


Figure 4.6 Placement of the Patch on the Rest of the Image Lattice

4.4. MINE PLACEMENT

Until now, the discussion was limited to the various issues concerned with the synthesis of the background. The next part of the synthesis procedure deals with the synthesis of clutter and mines. The proposed minefield synthesizer can place the image chips for various mine targets and/or clutter on already synthesized background-only image to synthesize different minefield scenarios. For the sake of simplicity, only mine patches are considered in synthesizing minefields. In this section, the synthesis of patterned/random mine layouts on already synthesized background is discussed.

Minefield synthesis is slightly different from that of background synthesis in the sense that the central characteristics of background-only synthesized image are not used in selecting the mine patches. The selection of a mine at a particular location is completely dictated by the identity of the mine type that needs to be placed and the edge characteristics of the equivalent background patch, which is centered at that location. It is also possible to use terrain information of the background, if desired, in selecting mine patches. To lay a mine patch at location $m = (i, j)$ on the synthesized background image, extract a background patch, centered at (i, j), of size equal to that of the mine patch. Select the mine patches from image database based on the mine type that needs to be placed. From this set, select a patch whose edge characteristics match the extracted background patch's edge characteristics. Finally background and selected mine patch are blended to obtain desired mine structure. In this case, a slightly different blending approach is employed when compared with the one described in Section 4.2.

The blending algorithm is based on the method of weighted averages as in the case of background-only patches. The difference between the blending of mine patch and the blending of background-only patch lies only in the computation of the distance metric. To blend a mine patch with its background, one needs to create two weight matrices with different distance metrics. The computation of distance for blending of mine patch is discussed below.

The location of the mine is assumed at the center of the mine patch of size $W_B \times W_B$ and occupies $y \times y$ pixels. Normally y takes values ranging from 10 to 20 pixels. The creation of the mine weight matrix in this part of synthesis is as that of the creation of the weight matrix explained in Section 4.2 with a small exception. In the former case, the weight value given to each pixel decreases as it move away from the center of the patch. For the present case, a constant weight is give to the all pixels that are present at the middle of the mine patch and this value decreases gradually as one move away from the center region. The construction of the background weight matrix is opposite to that of the mine weight matrix. In this case, a constant weight value of zero is given to all the pixels that are present in the middle of the background patch and this value increases gradually as one move away from the center region. The reason for choosing this kind of weight assignment is that in implanting mines over a synthesized background, mine location of the mine patch is an important factor. Thus, in creating weight matrices, more weight is given to the center pixels of the mine patch and less weight is given to the center pixels of the background area. Figure 4.7 gives the basic structure of the weight matrix. The shaded area in the middle, in Figure 4.7, is assigned a value of $(W_B - y)/2$ for the mine weight matrix and a value of zero for the background weight matrix.

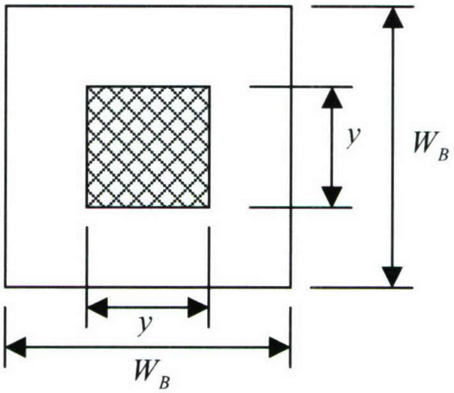


Figure 4.7 Structure of Weight Matrix for Blending of Mines

The following equation is used to blend the background and mine image patches:

$$I_0(x,y) = \frac{d_M(x,y)I_M(x,y) + d_B(x,y)I_B(x,y)}{d_M(x,y) + d_B(x,y)} \tag{19}$$

where

$I_M(x,y)$ = Gray value of the pixel at (x, y) of the mine patch,

$I_B(x,y)$ = Gray value of the pixel at (x, y) of the background patch,

$d_M(x,y)$ = Weight value of the of the mine weight matrix at (x, y),

$d_B(x,y)$ = Weight value of the of the background weight matrix at (x, y),

$I_0(x,y)$ = Gray value of the pixel at (x, y) of the resultant image.

4.5. ALGORITHM FOR SYNTHESIS OF MINEFIELD FRAME

4.5.1. Synthesis of Background

1. Select a set of patches ψ_T from the image database ψ based on the terrain information.
2. Select a set of patches ψ_{CT} from the chosen set of patches ψ_T based on the MRF model for the background and the neighborhood patches already synthesized. If ψ_{CT} is empty, set $\Psi_{CT} = \Psi_T$.
3. Select a set of patches ψ_{ECT} from ψ_{CT} such that the edge characteristics of each patch in the selected set are similar to the previously synthesized neighborhood. If ψ_{ECT} is empty, set $\psi_{ECT} = \psi_{CT}$.
4. Find a synthesis patch $x \in \Psi_{ECT}$ that has high correlation with the edge of the already synthesized background image.
5. Perform blending operation at the boundary zones between the neighborhood patches and the patch selected in step 4.
6. Repeat steps 1 to 5 for each lattice location on the desired image until the output image is fully covered.

4.5.2. Synthesis of Mines

7. Obtain the mine locations and mine types from the minefield layout over the terrain map.

8. Choose a mine patch, from the database, which matches the edge characteristics of the equivalent background patch.
9. Perform the blending of the mine patches at the boundary zones.
10. Repeat 8 and 9 until all the mines are placed.

4.6. RESULTS

In this section, some of the synthesized minefields frames using the proposed patch-based synthesis algorithm are presented. All the images shown in this section are of size 512×640 and represent 1.1-inch ground resolution. Original images from five different backgrounds identified as Mine area, Wash 5 area, Rock 41 area, Rock 42 Area, and Rock 44 Area, are used in synthesizing different minefield scenarios. These input images are collected in two time frames viz daytime (07:30 – 10:30) and nighttime (02:30 – 05:30). Daytime imagery includes Mine area, Wash 5 area, Rock 42 area, and Rock 44 area backgrounds and nighttime imagery includes Mine area, Wash 5 area, Rock 41 area, and Rock 44 area backgrounds. Note that, for the both time frames, mine signatures are available for only mine area background. These same mine signatures, from a specific time frame, are used to simulate various minefield scenarios on the other backgrounds from that time frame. Two kinds of image databases are created for each of the available time frame. In each image database, sixty images from each of the available backgrounds are used. In creating image database, patches are extracted with the following specifications (all dimensions shown in this section are in pixels):

- Size of database patch 60×60,
- Size of synthesis patch 40×40,
- Edge width of the synthesis patch 10,
- Allowed tolerance 10.

Two kinds of minefield structures such as patterned and scattered are used synthesizing minefield scenarios. For easy visual inspection, some of the mines over the synthesized images are represented with the arrows marks. In synthesizing patterned mine layouts different mine types such as LM_A, MP_A, SM_A, and LP_B are used. For scattered mine layout, only single mine type (MP_A) is considered to improve the visibility of mine signatures.

4.6.1. Daytime Imagery. Figure 4.8 through 4.11 represent original daytime images taken from the different backgrounds of Mine area, Wash 5 area, Rock 42 area, and Rock 44 area, respectively. Figure 4.8 shows the daytime image from mine area with mine signatures. Figure 4.16 through Figure 4.19 represent the different synthesized background-only daytime images with a single background. Figure 4.24 through Figure 4.26 represent different synthesized background-only daytime images with two backgrounds. The synthesized image shown in Figure 4.25 has very distinct interface between the two backgrounds, where as in images shown in Figure 4.24 and Figure 4.26 the interface is faint. This distinct interface in Figure 4.25 is because of large variations in gray values of the original imagery corresponding to these backgrounds. However, in other two images (4.24 and 4.26) the interface and the gray level variations from one background to other background are natural. Figures 4.30 to 4.32 represent the synthesized daytime images with patterned minefield structure and Figures form 4.36 to 4.39 represent the synthesized daytime images with scattered minefield structure.

4.6.2. Nighttime Imagery. Figure 4.12 through 4.15 represent original nighttime images taken from the different backgrounds such as Mine area, Wash 5 area, Rock 41 area, and Rock 44 area, respectively. Figure 4.12 shows the nighttime image from mine area with mine signatures. Figure 4.20 through Figure 4.23 represent the different synthesized background-only nighttime images with a single background. The physical and the statistical characteristics of these synthesized images are very similar to the corresponding original images. Figure 4.27 through Figure 4.29 represent different synthesized background-only nighttime images with two backgrounds. Unlike previous synthesized interfaces, synthesized images shown in Figure 4.27 through 4.29 have the natural interface between the two backgrounds. Figures 4.33 to 4.35 represent the synthesized nighttime images with patterned minefield structure and Figures form 4.40 to 4.42 represent the synthesized nighttime images with scattered minefield structure

In synthesized images with more than one background, the patches, which are presented at the interface, have the spatial features similar to that of the other background. Because of this, adjacent backgrounds are coupled gradually. All the synthesized images with minefield are having the mine signatures that are blended naturally with the surrounding synthesized background. The mine signatures placed over the background-only images that are synthesized from the nighttime sample images are prominent from the surrounding background and even small mines such as SM_A can be identified easily. However, the mine signatures over the background-only images that are synthesized from the daytime sample images are not visible in the backgrounds such as Wash 5, Rock 42, and Rock 44 because of large gray level variations between the mine patch and the surrounding background. In all the synthesized images, the variations of gray levels are natural and are uniform throughout the synthesized image. There are no discontinuities (patchy effect) at the patch boundaries. The average time taken for synthesizing all the frames (of size 512×640) shown above is about 650 sec per image on a 1.3 GHz Pentium 4 processor with 512 MB RAM computer.

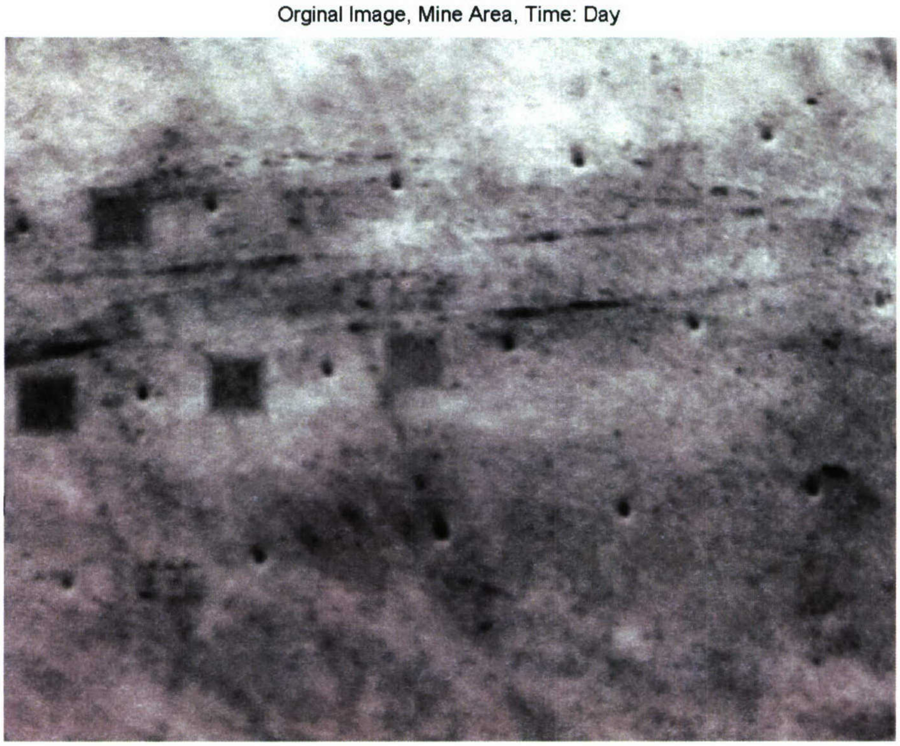


Figure 4.8 Original Mine Area Image – Daytime

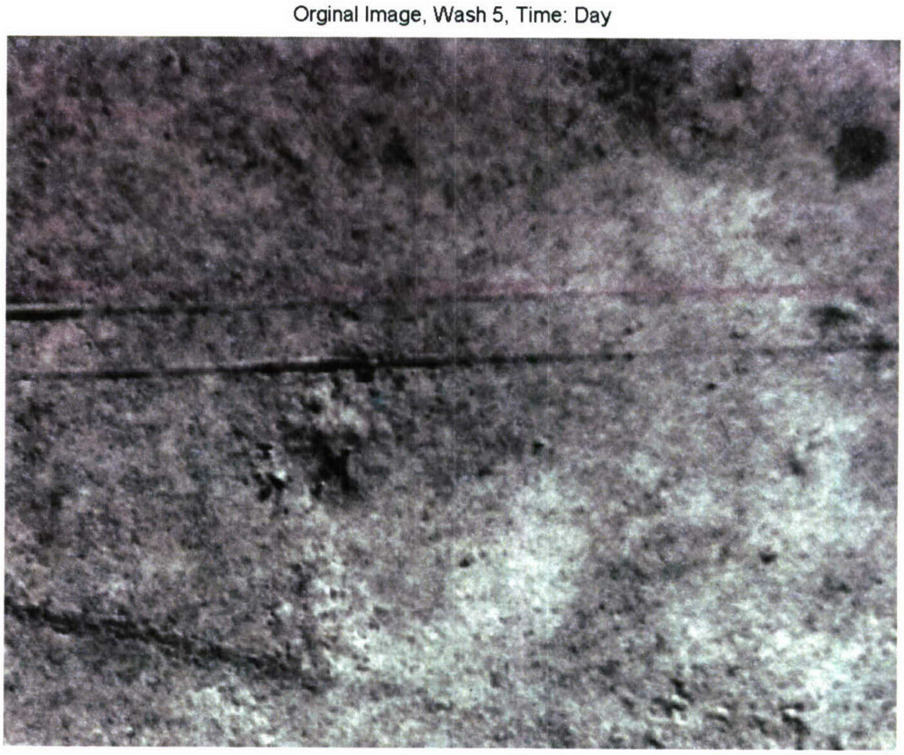


Figure 4.9 Original Wash 5 Area Image – Daytime

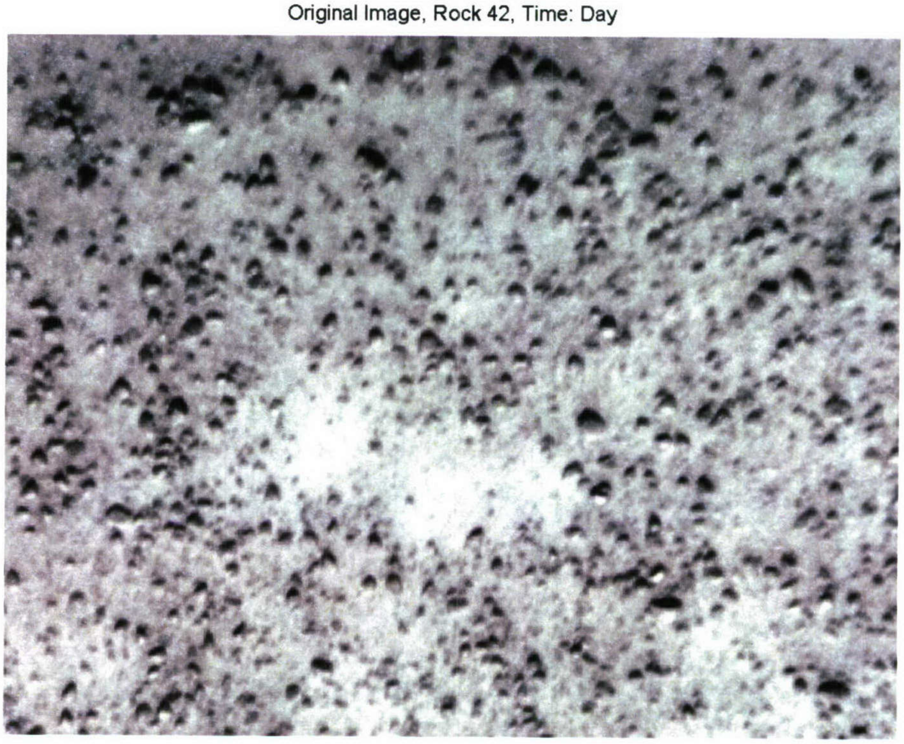


Figure 4.10 Original Rock 42 Area Image – Daytime

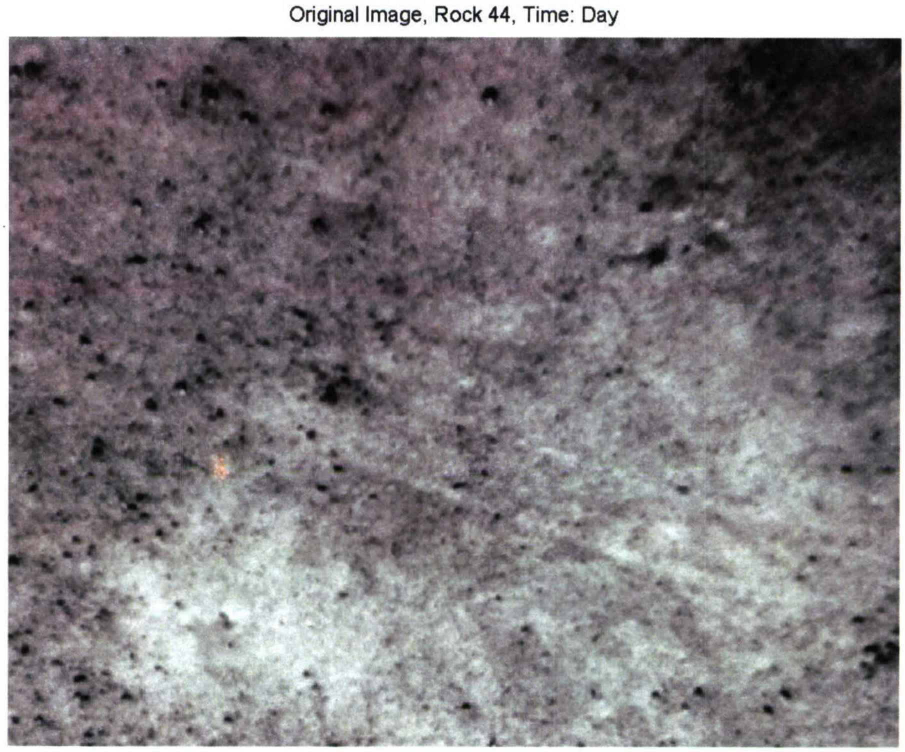

Original Image, Rock 44, Time: Day

Figure 4.11 Original Rock 44 Area Image – Daytime

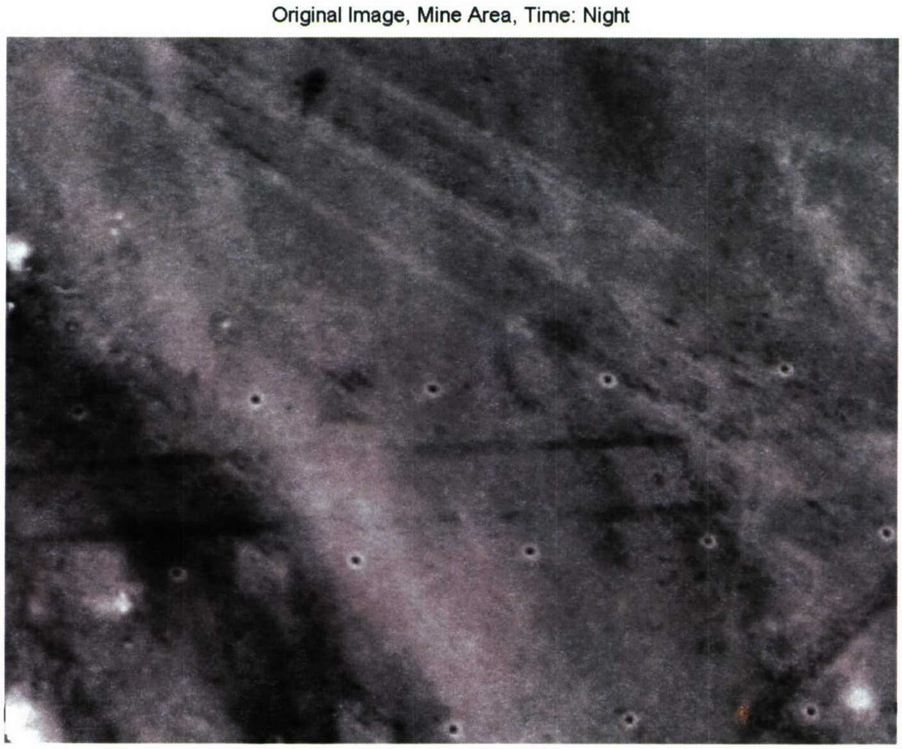
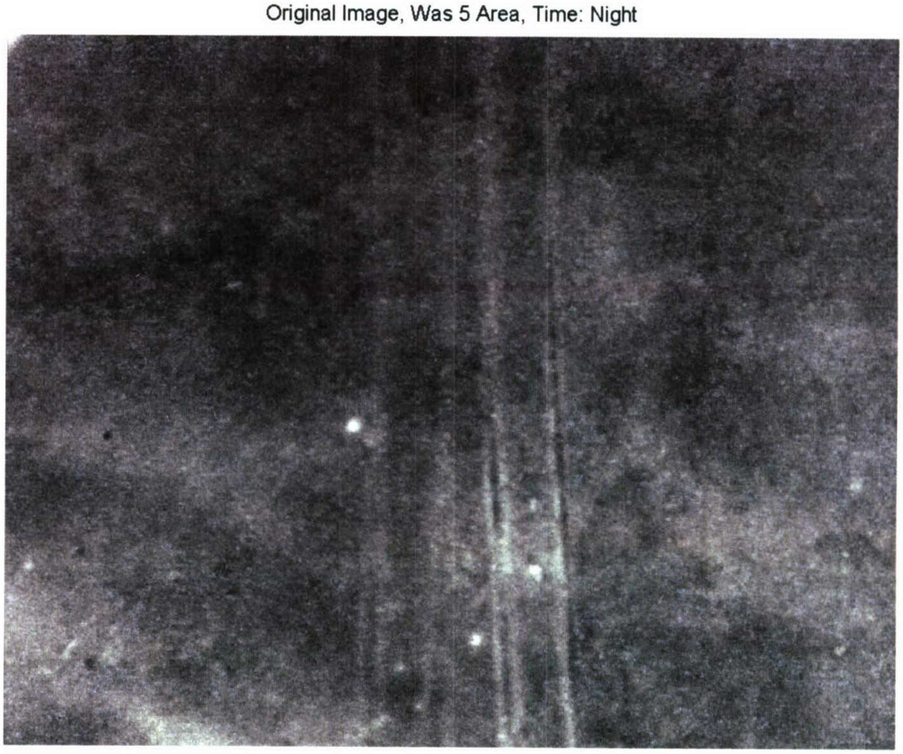
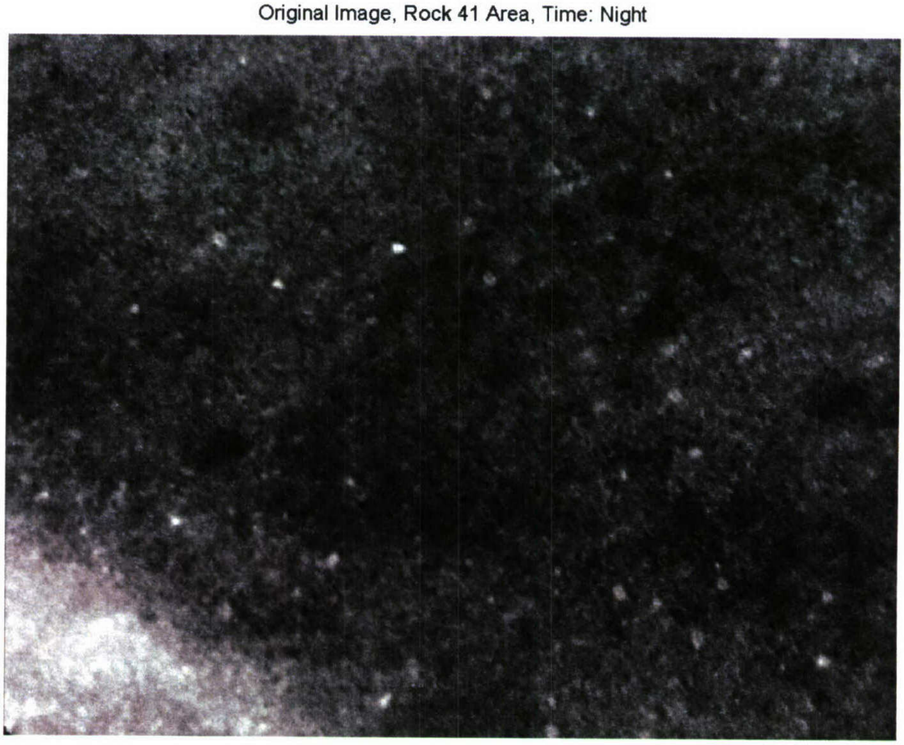

Original Image, Mine Area, Time: Night

Figure 4.12 Original Mine Area Image – Nighttime



Original Image, Was 5 Area, Time: Night

Figure 4.13 Original Wash 5 Area Image – Nighttime



Original Image, Rock 41 Area, Time: Night

Figure 4.14 Original Rock 41 Area Image – Nighttime

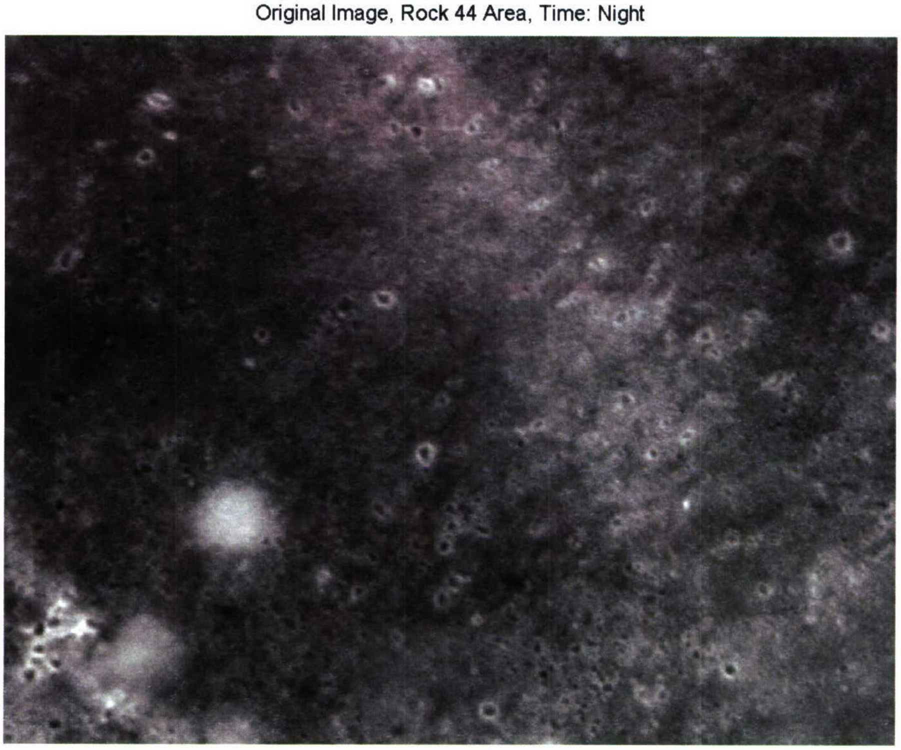


Figure 4.15 Original Rock 44 Area Image – Nighttime

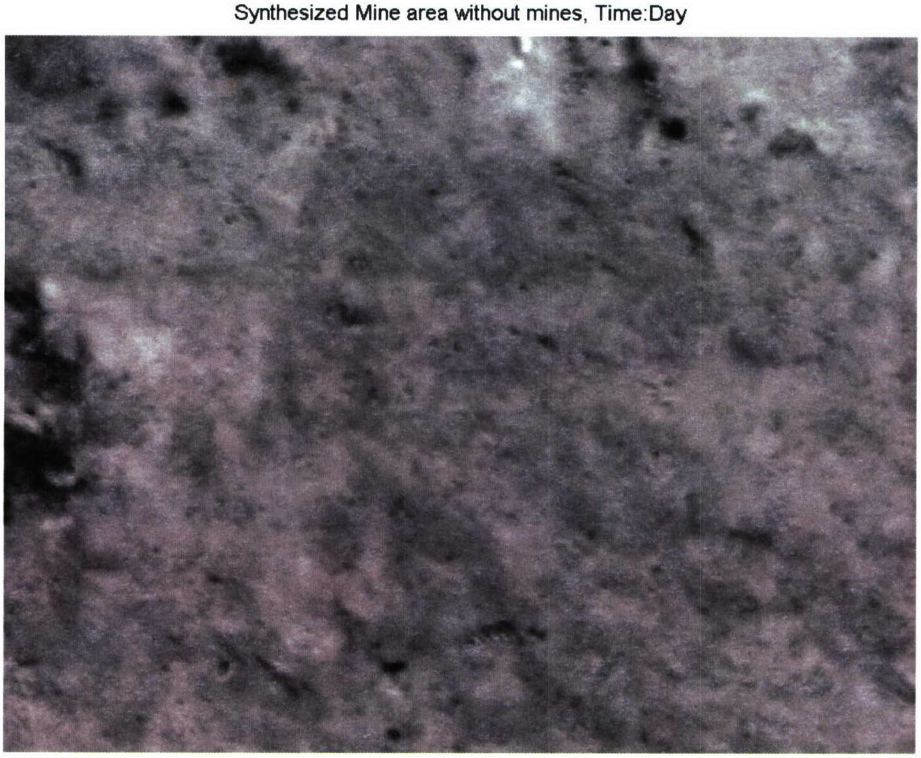


Figure 4.16 Synthesized Mine Area Without Mines- Daytime

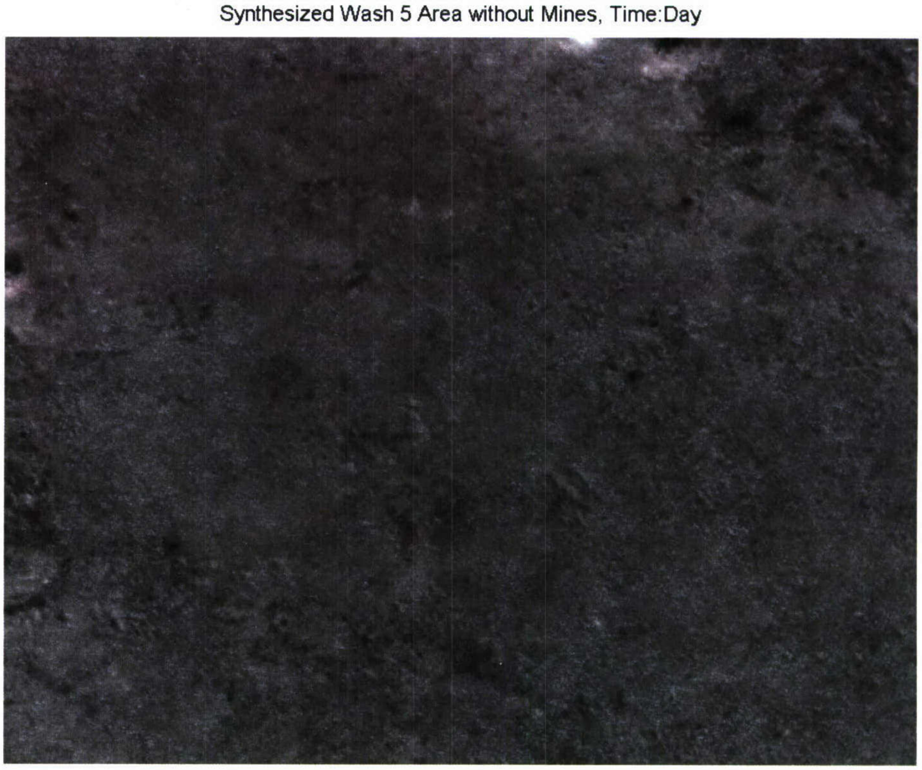

Synthesized Wash 5 Area without Mines, Time:Day

Figure 4.17 Synthesized Wash 5 Area Without Mines- Daytime

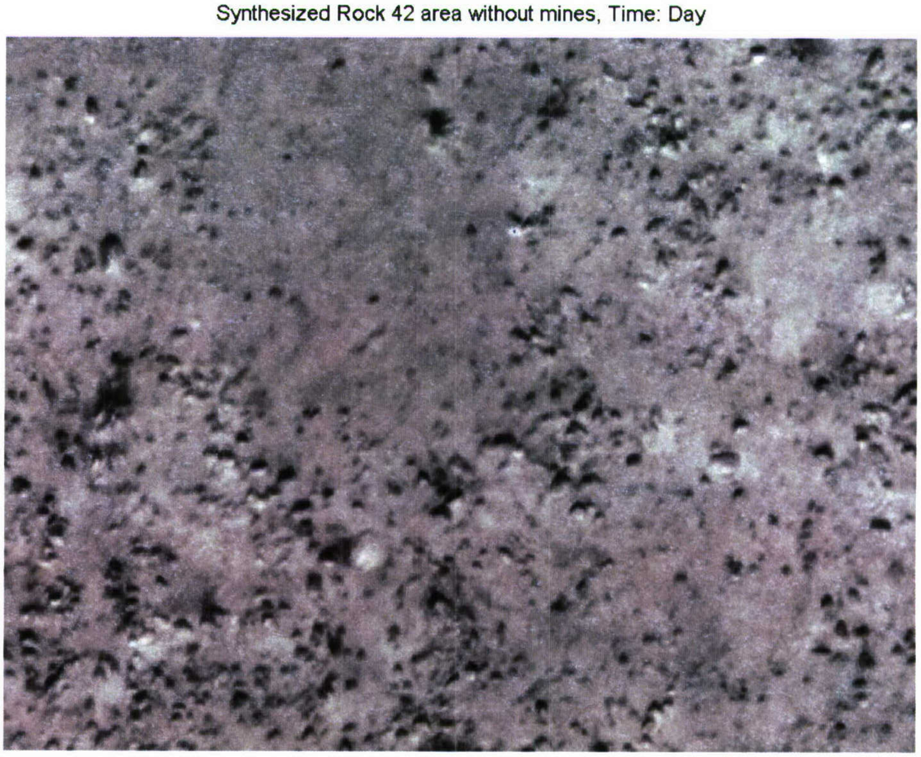

Synthesized Rock 42 area without mines, Time: Day

Figure 4.18 Synthesized Rock 42 Area Without Mines- Daytime

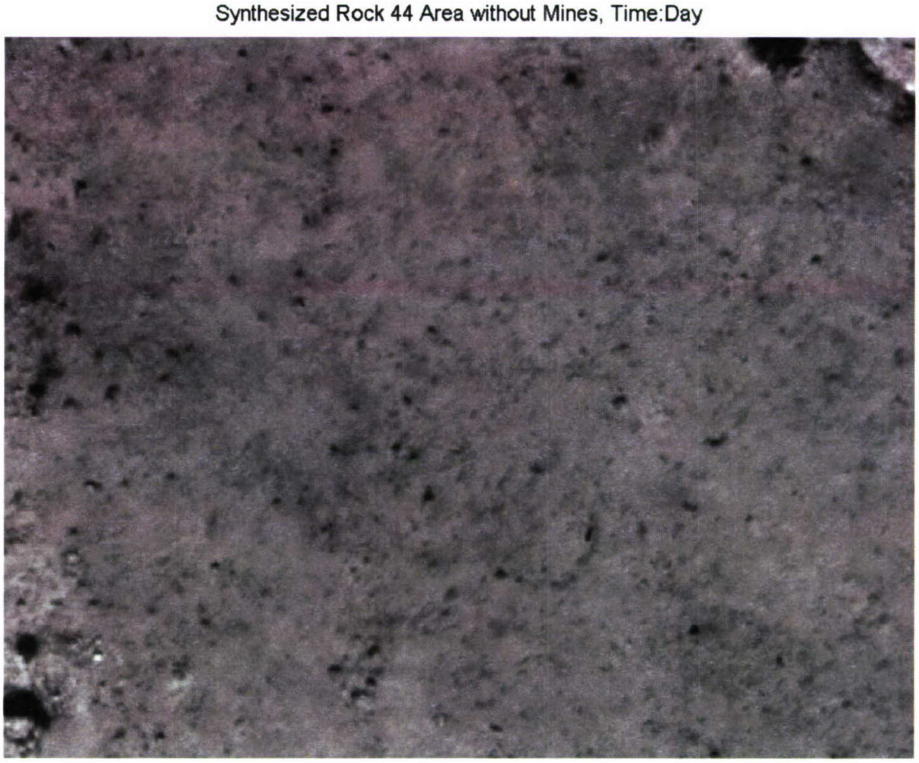


Figure 4.19 Synthesized Rock 44 Area Without Mines- Daytime

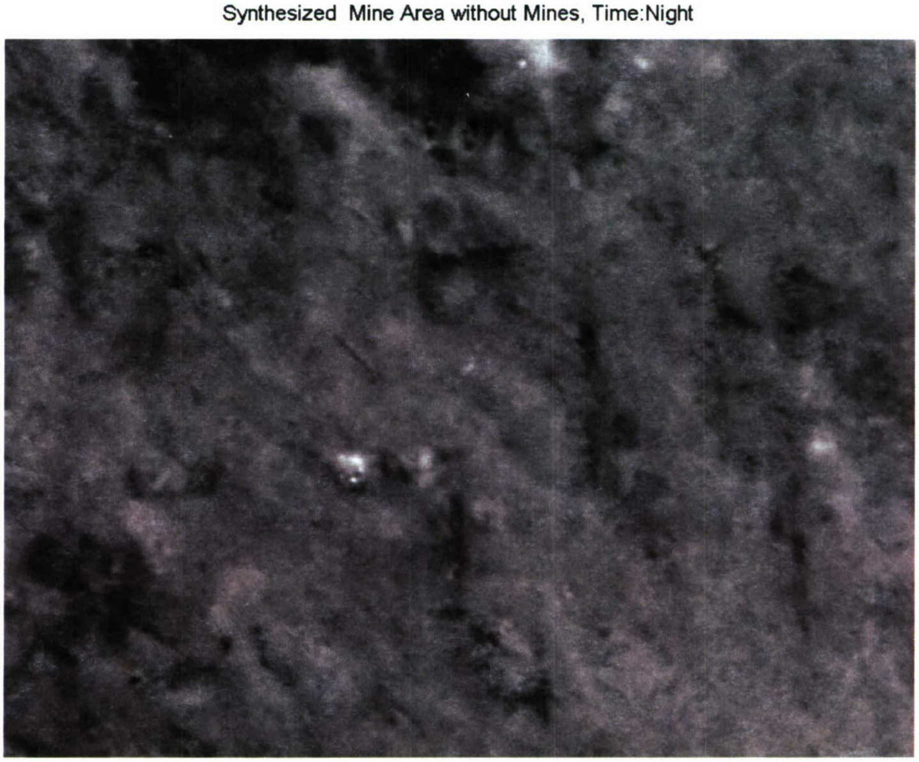


Figure 4.20 Synthesized Mine Area Without Mines- Nighttime

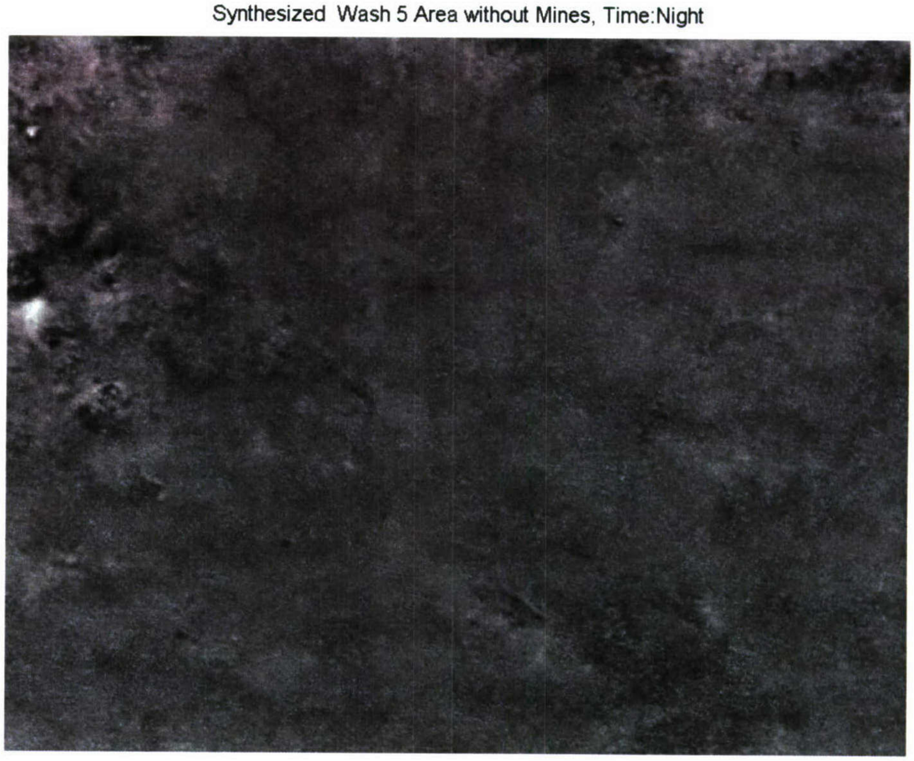


Figure 4.21 Synthesized Wash 5 Area Without Mines- Nighttime

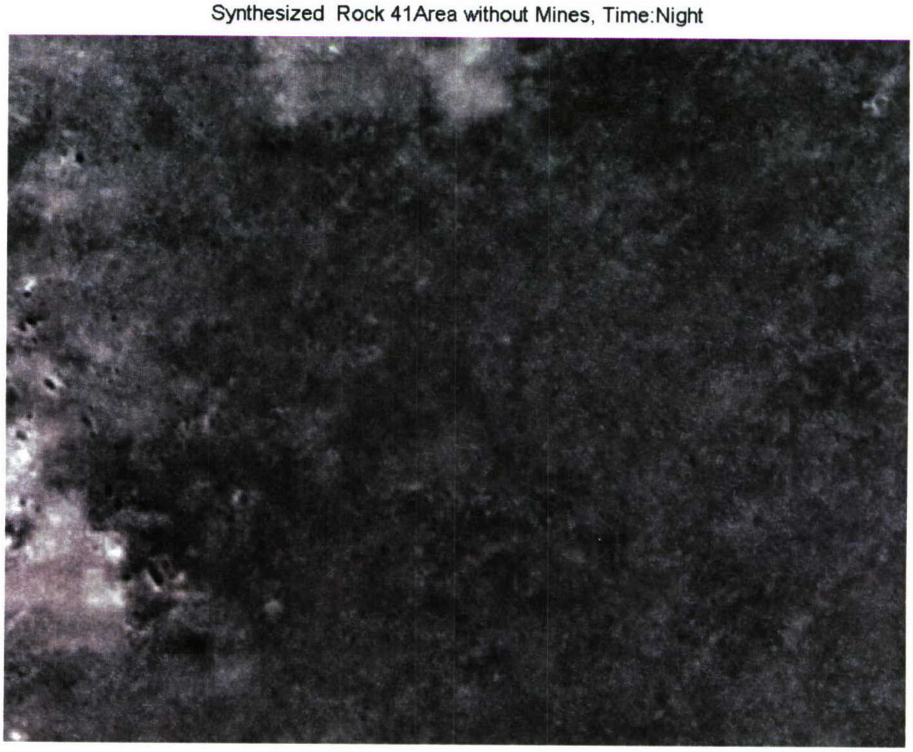


Figure 4.22 Synthesized Rock 41 Area Without Mines- Nighttime

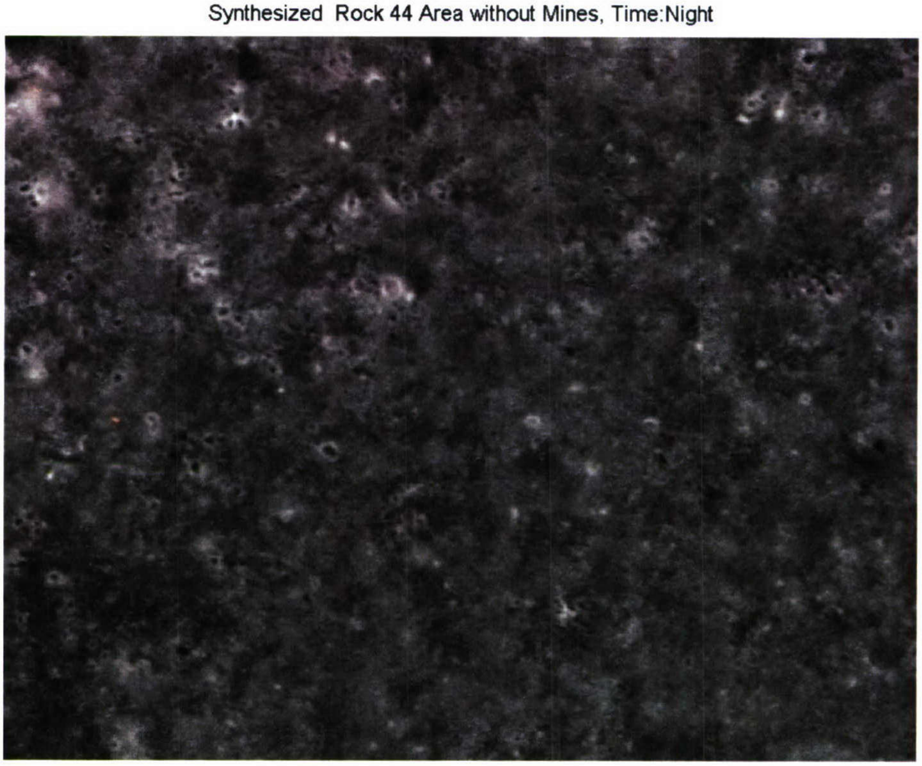


Figure 4.23 Synthesized Rock 44 Area Without Mines- Nighttime

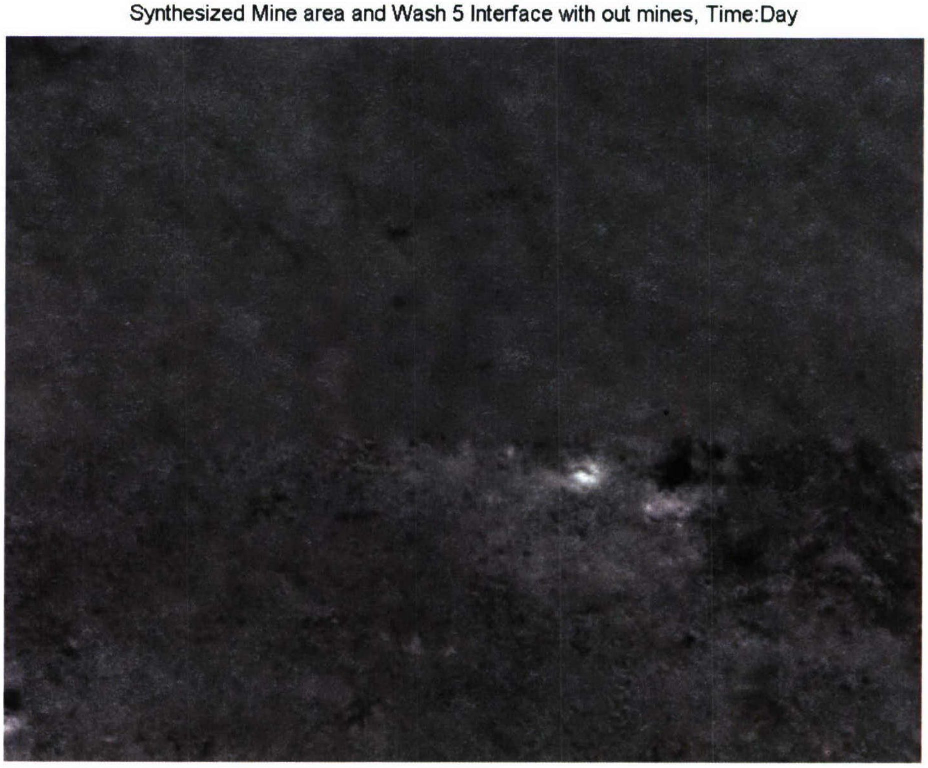
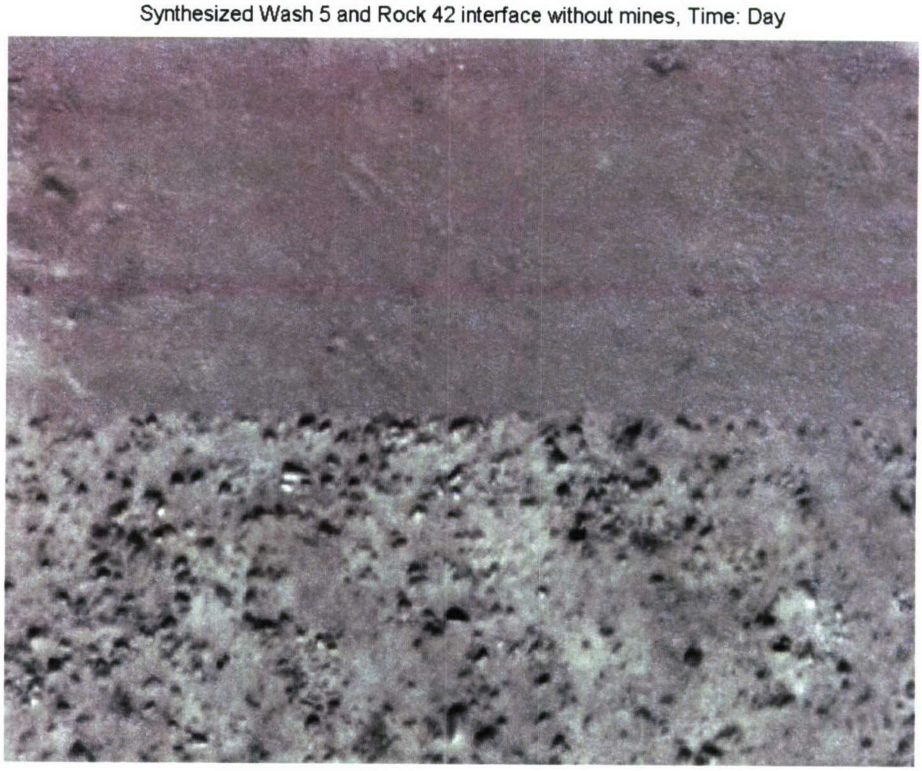
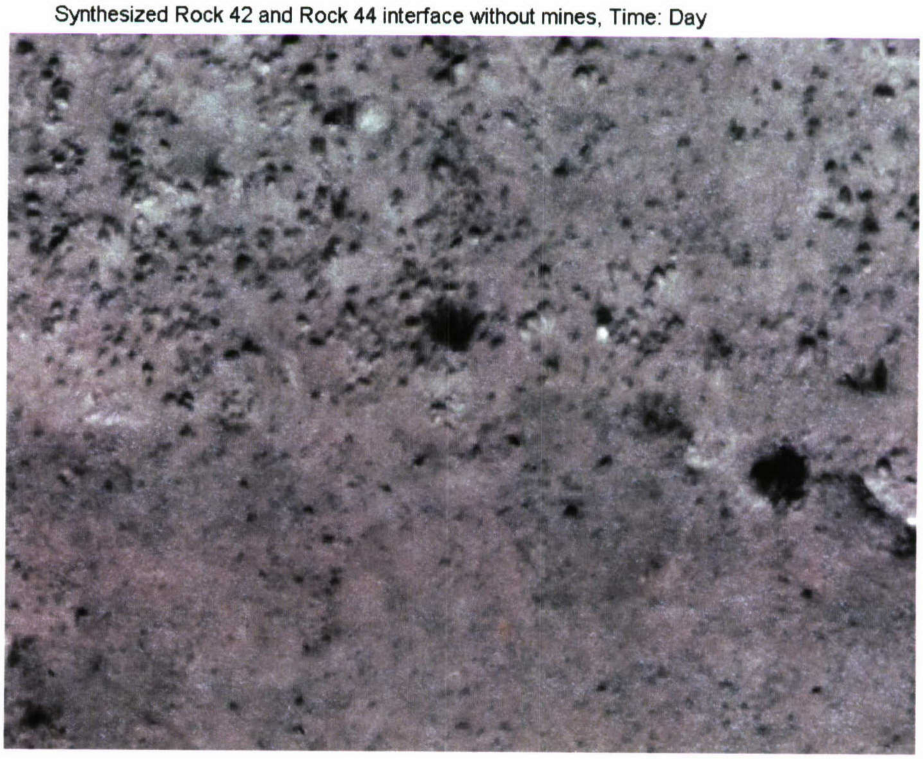

Synthesized Mine area and Wash 5 Interface with out mines, Time:Day

Figure 4.24 Synthesized Mine and Rock 44 Interface Without Mines- Daytime



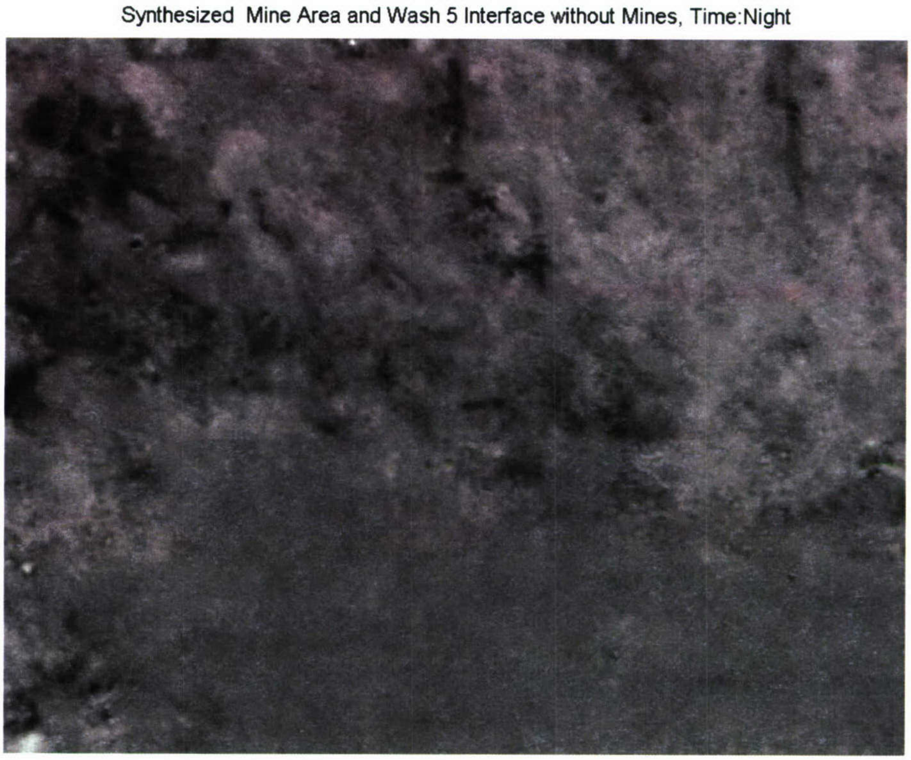
Synthesized Wash 5 and Rock 42 interface without mines, Time: Day

Figure 4.25 Synthesized Wash 5 and Rock 42 Interface Without Mines- Daytime



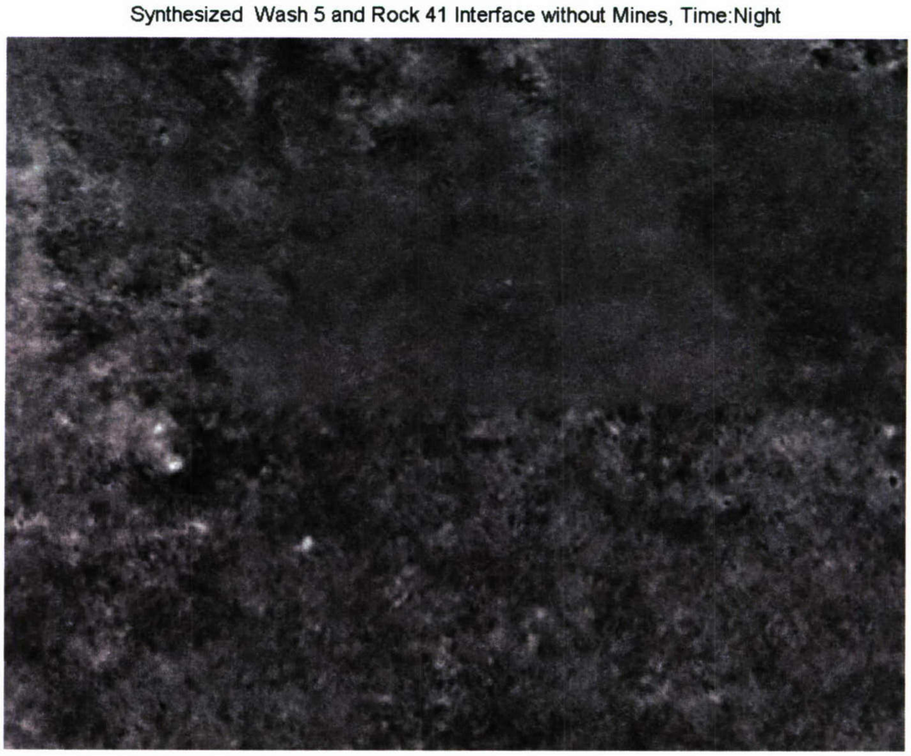
Synthesized Rock 42 and Rock 44 interface without mines, Time: Day

Figure 4.26 Synthesized Rock 42 and Rock 44 Interface Without Mines- Daytime



Synthesized Mine Area and Wash 5 Interface without Mines, Time:Night

Figure 4.27 Synthesized Mine and Wash 5 Interface Without Mines- Nighttime



Synthesized Wash 5 and Rock 41 Interface without Mines, Time:Night

Figure 4.28 Synthesized Wash 5 and Rock 41 Interface Without Mines- Nighttime

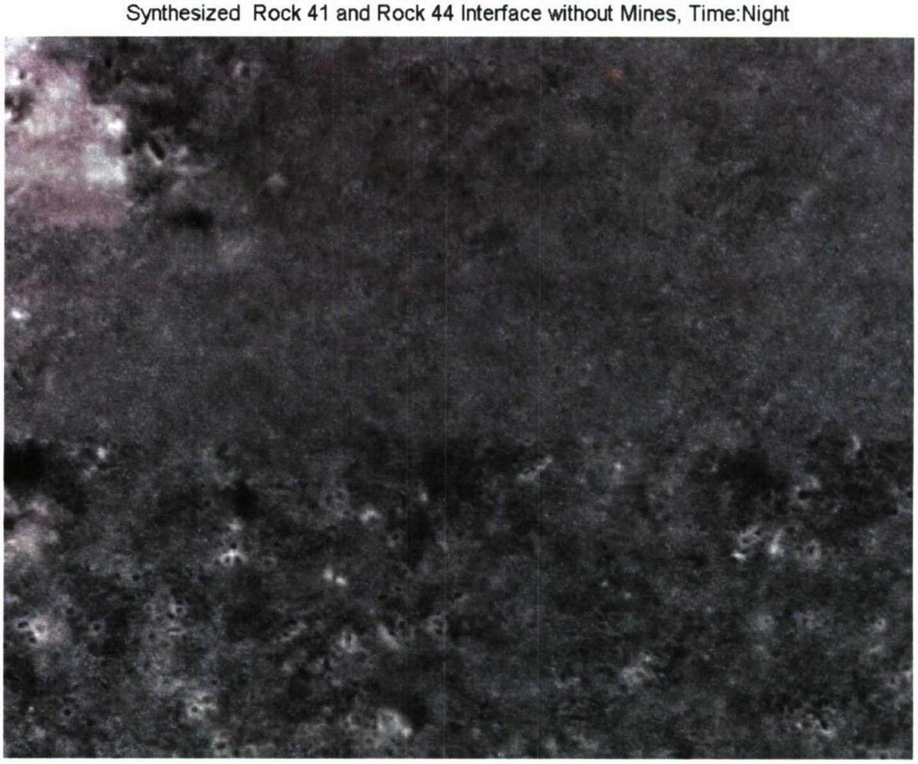

Synthesized Rock 41 and Rock 44 Interface without Mines, Time:Night

Figure 4.29 Synthesized Rock 41 and Rock 44 Interface Without Mines- Nighttime

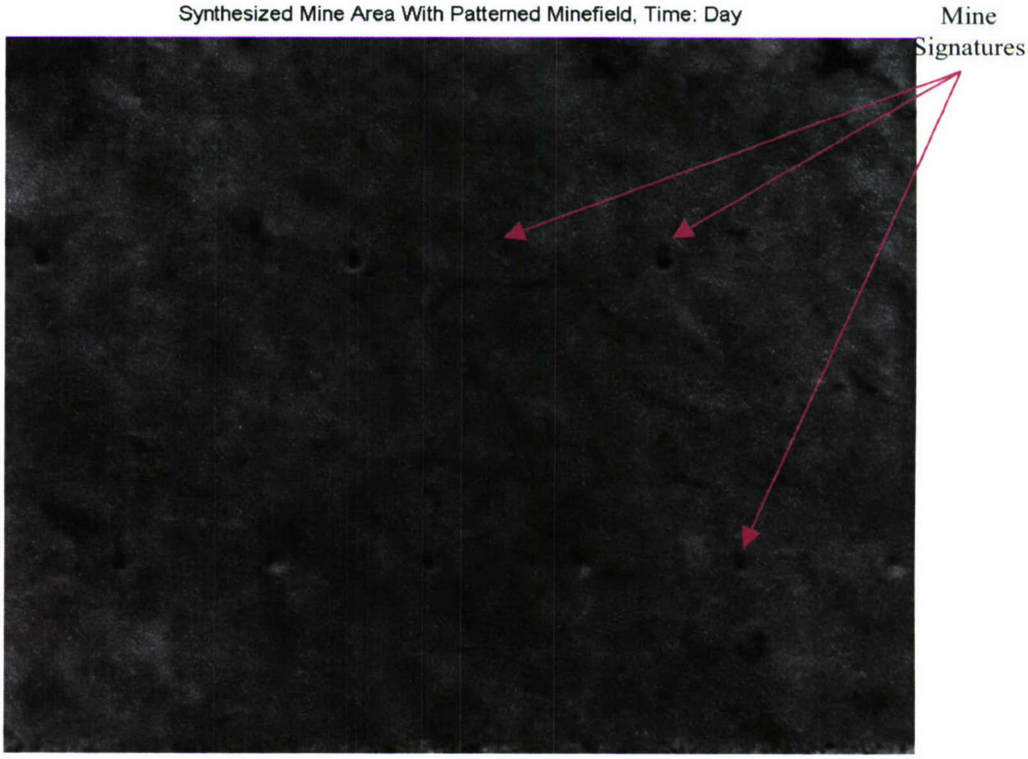


Figure 4.30 Synthesized Mine Area With Patterned Mines- Daytime

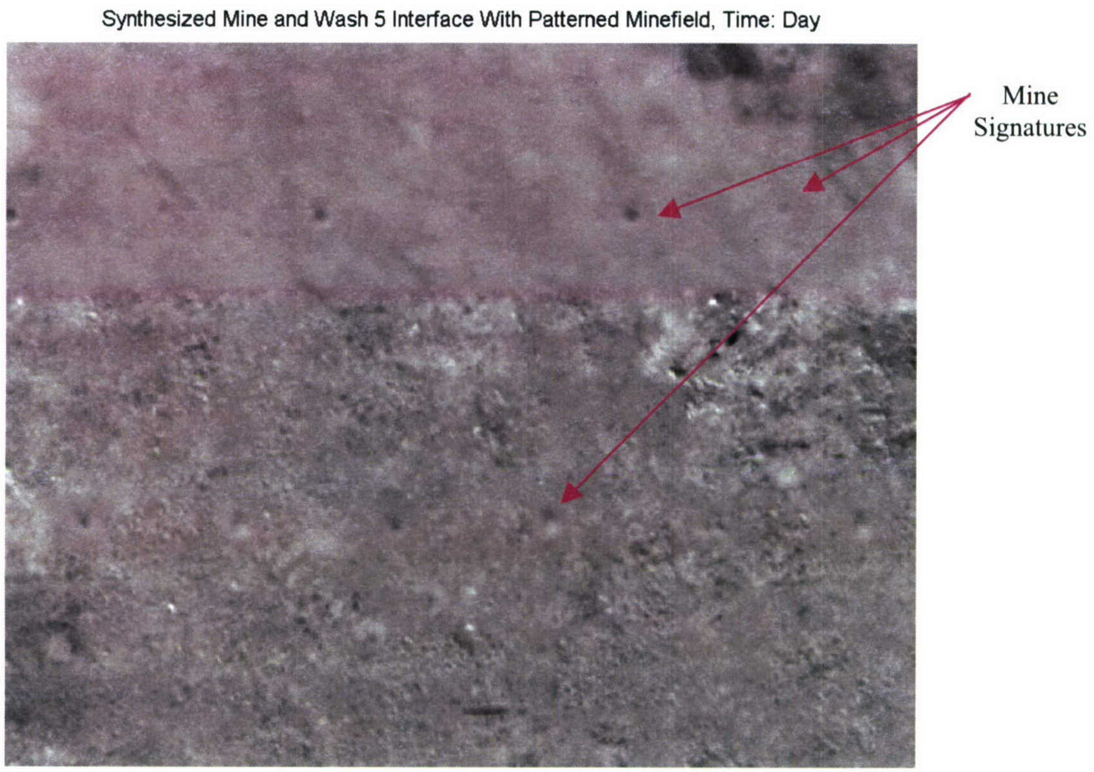

Synthesized Mine and Wash 5 Interface With Patterned Minefield, Time: Day
Mine
Signatures

Figure 4.31 Synthesized Mine and Wash 5 Interface With Patterned Mines- Daytime

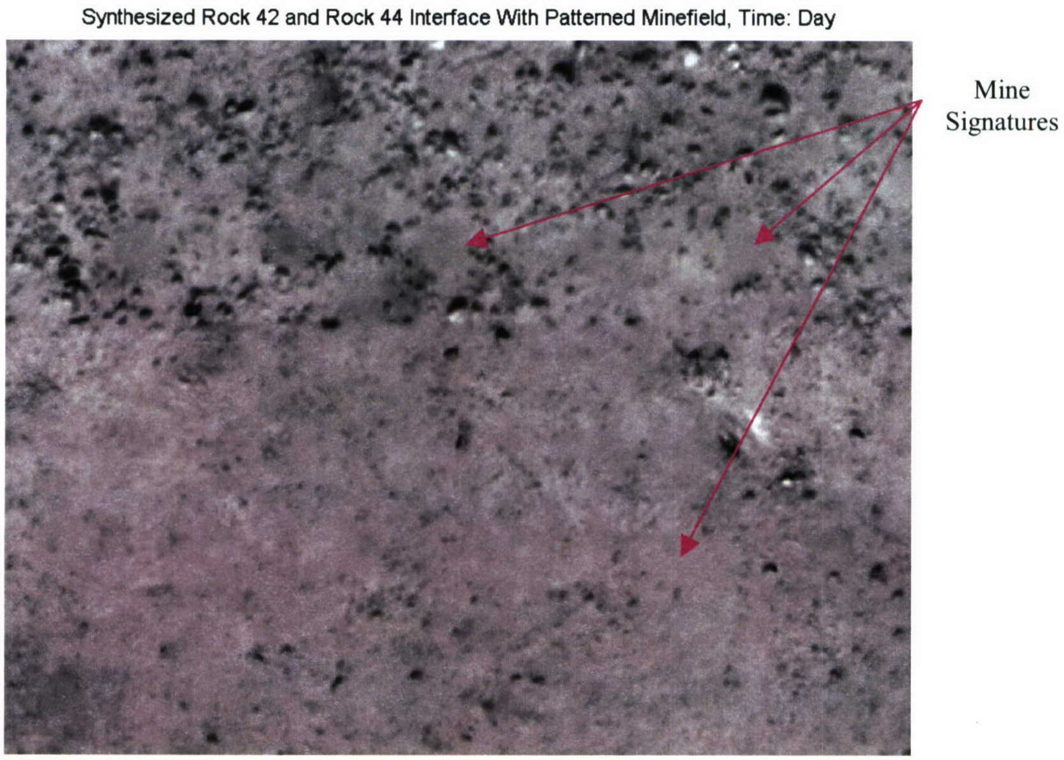

Synthesized Rock 42 and Rock 44 Interface With Patterned Minefield, Time: Day
Mine
Signatures

Figure 4.32 Synthesized Rock 42 and Rock 44 Interface With Patterned Mines- Daytime

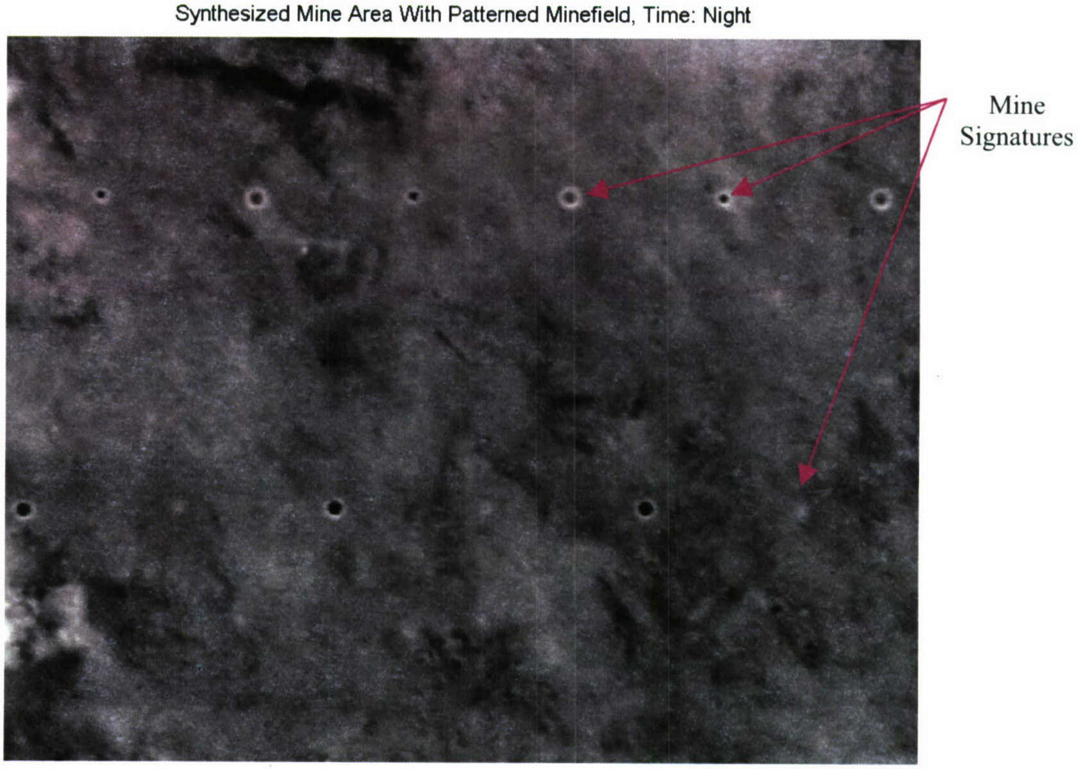


Figure 4.33 Synthesized Mine Area With Patterned Mines- Nighttime

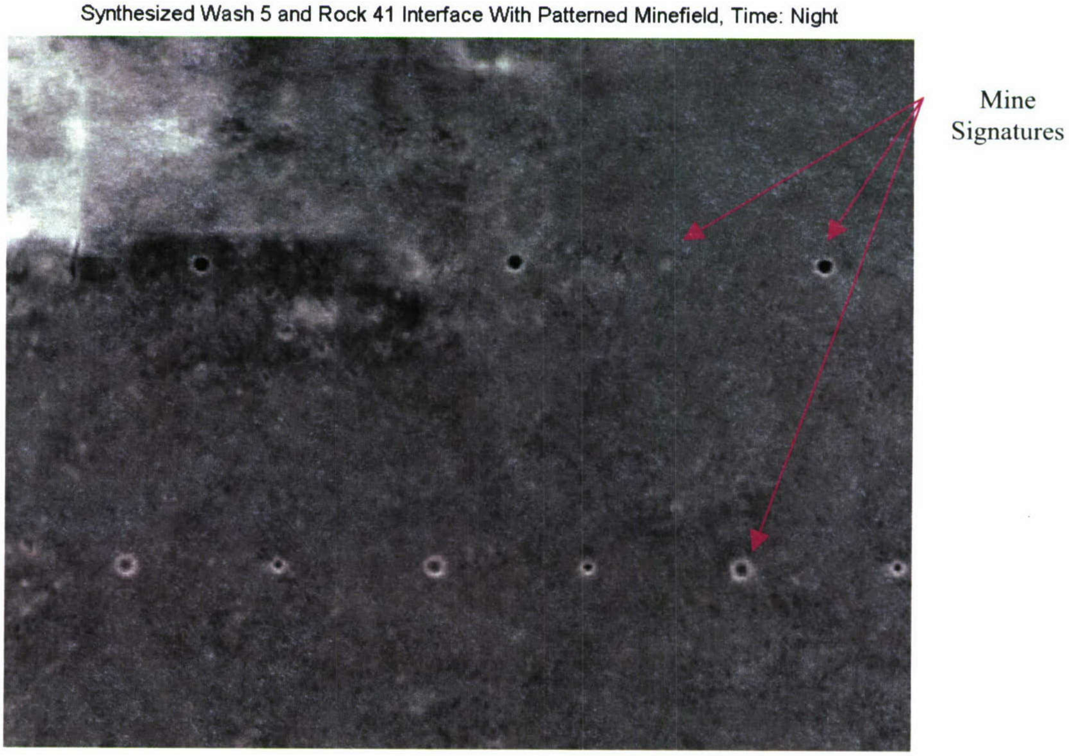

Synthesized Wash 5 and Rock 41 Interface With Patterned Minefield, Time: Night
Mine Signatures

Figure 4.34 Synthesized Wash 5 and Rock 41 Interface With Patterned Mines- Nighttime

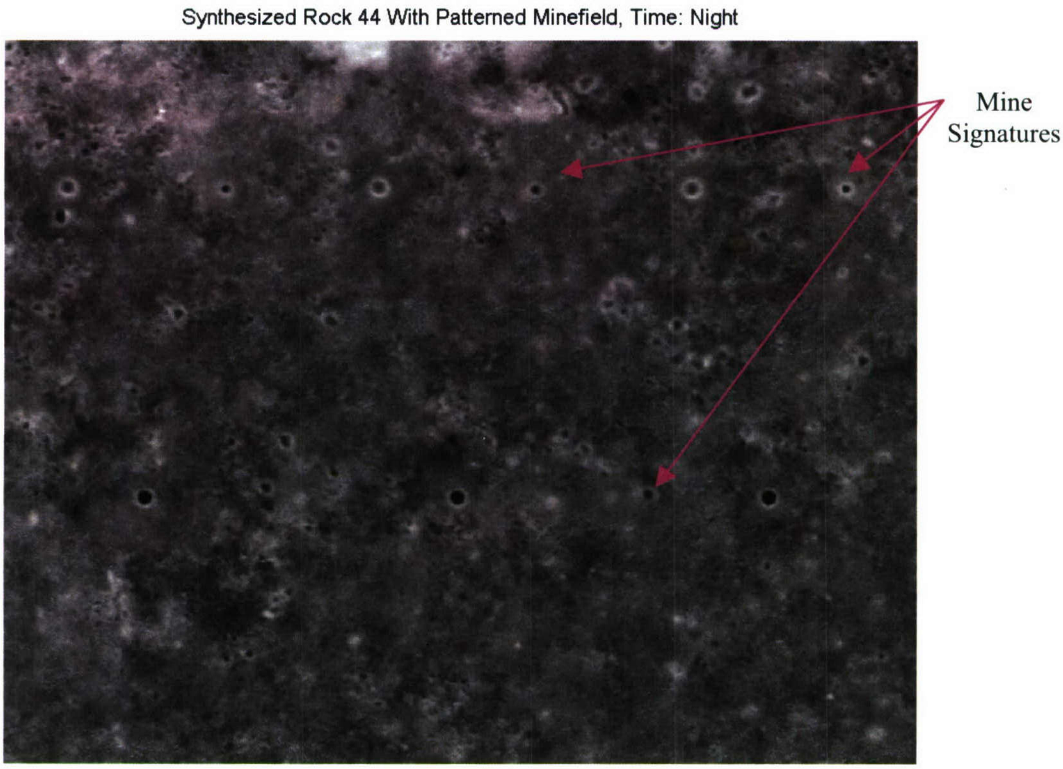


Figure 4.35 Synthesized Rock 44 Area With Patterned Mines- Nighttime

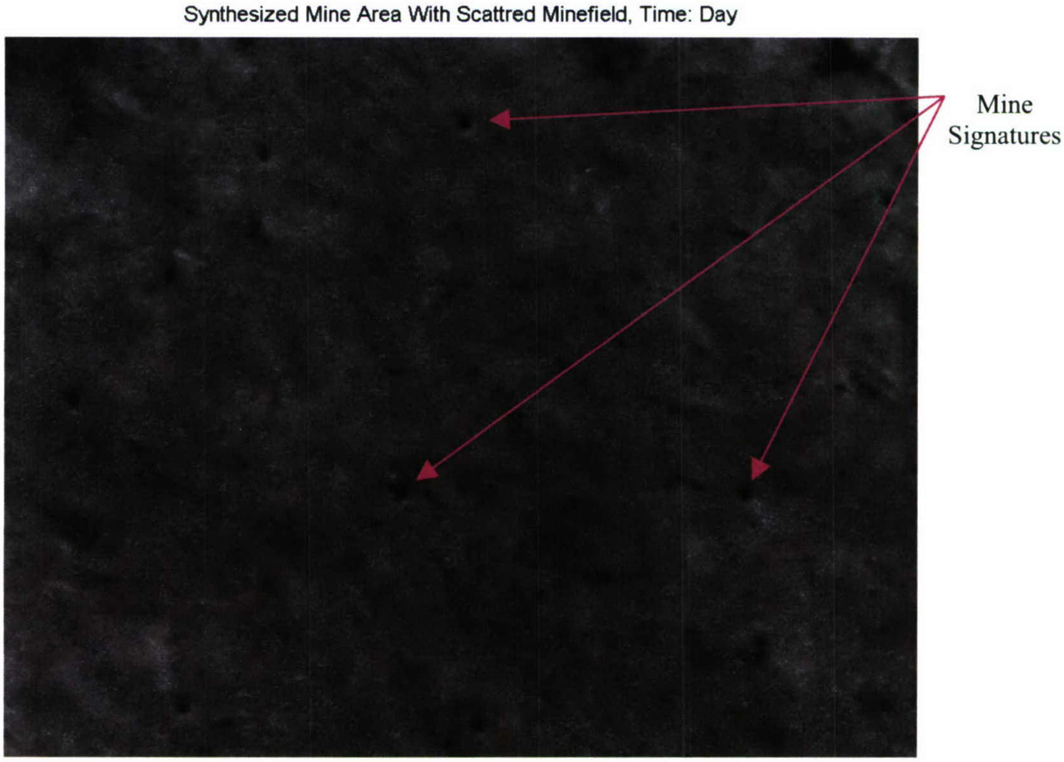


Figure 4.36 Synthesized Mine Area With Scattered Mines- Daytime

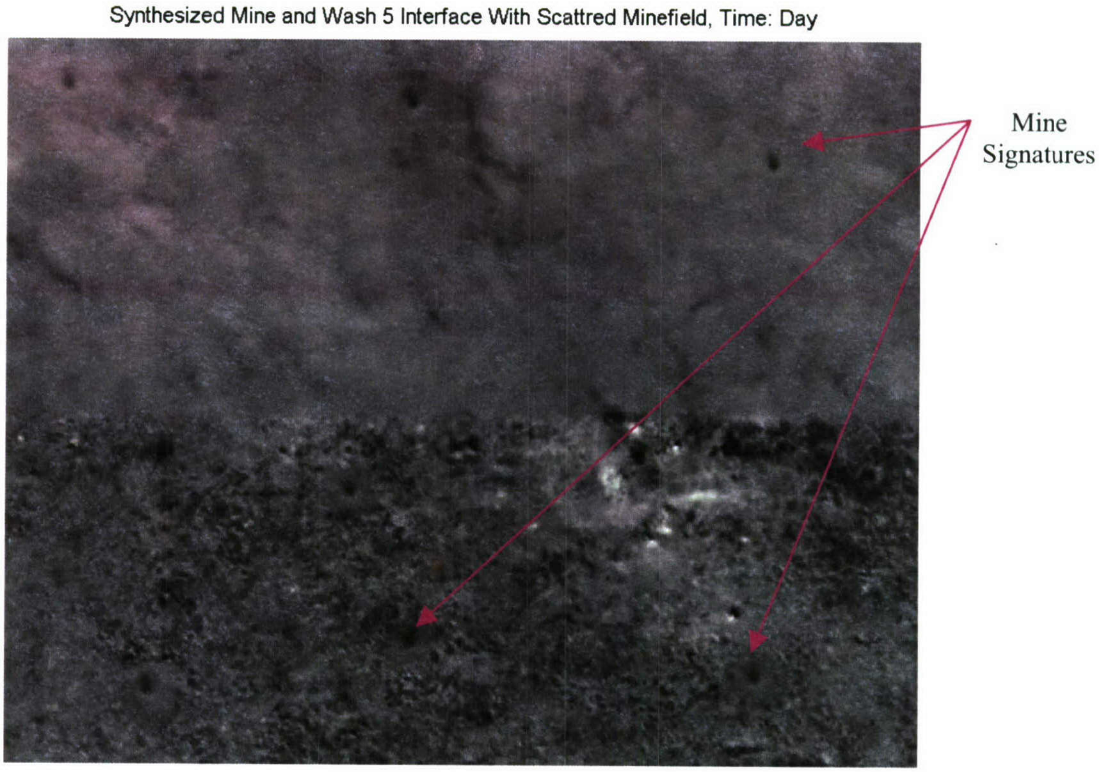

Synthesized Mine and Wash 5 Interface With Scattred Minefield, Time: Day
Mine Signatures

Figure 4.37 Synthesized Mine and Wash 5 Interface With Scattered Mines- Daytime

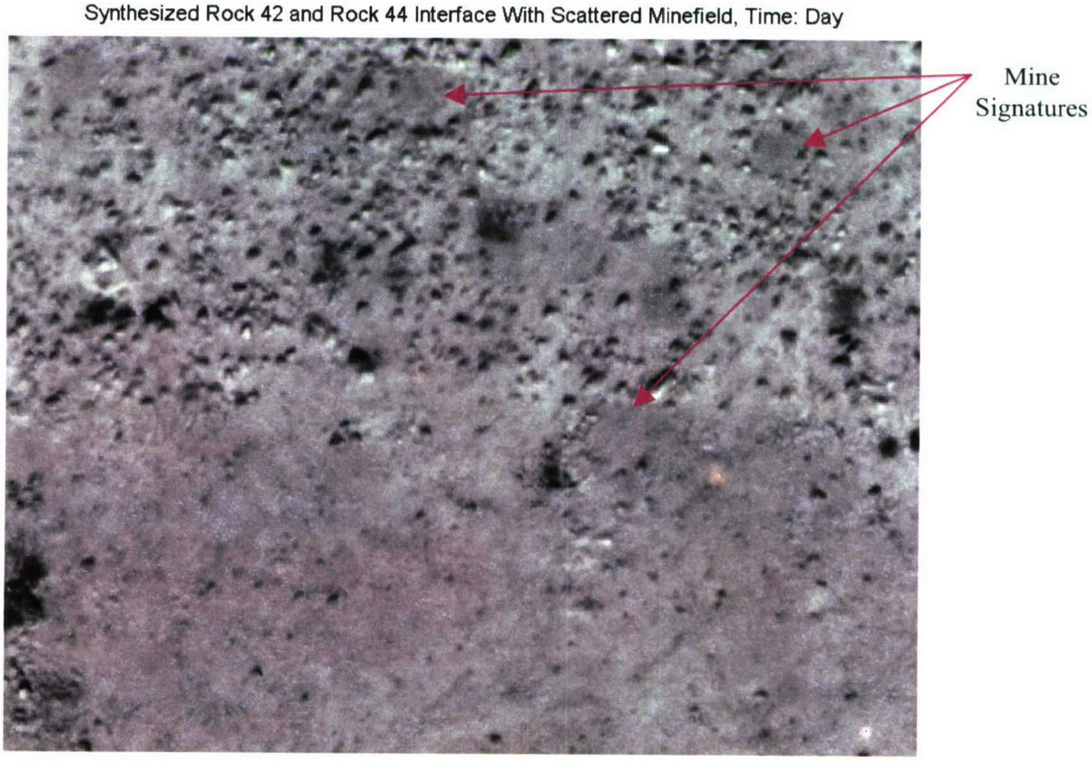

Synthesized Rock 42 and Rock 44 Interface With Scattered Minefield, Time: Day
Mine Signatures

Figure 4.38 Synthesized Rock 42 and Rock 44 Interface With Scattered Mines- Daytime

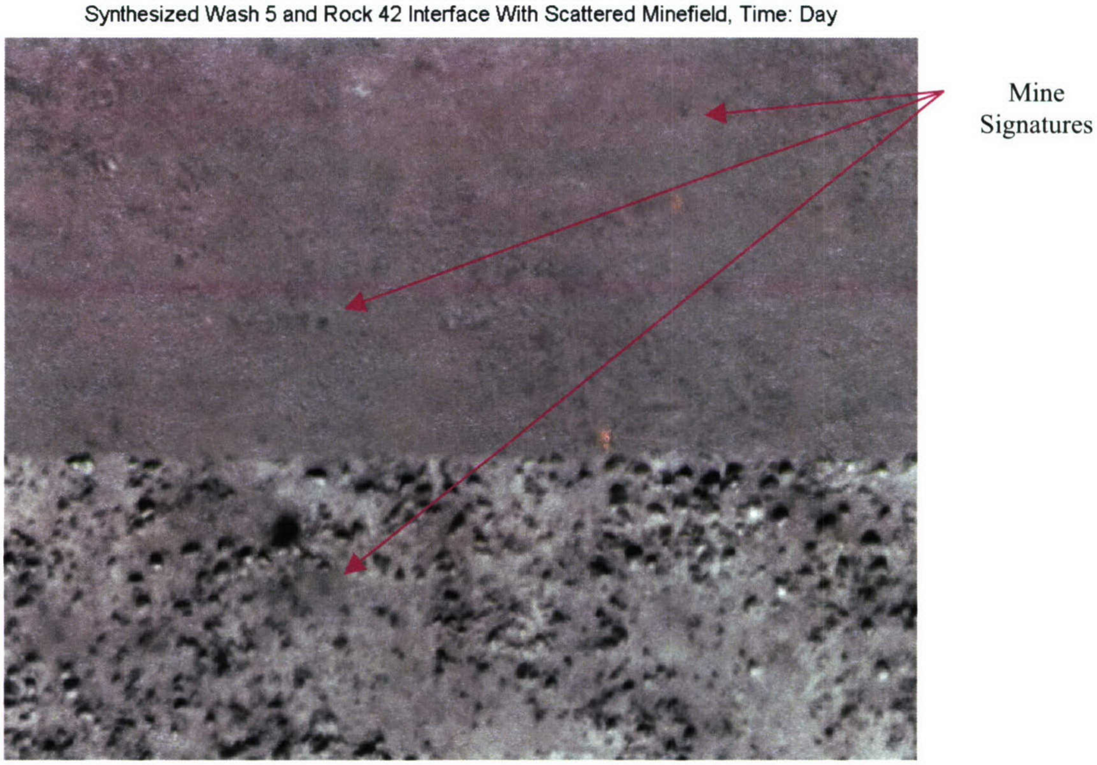

Synthesized Wash 5 and Rock 42 Interface With Scattered Minefield, Time: Day
Mine
Signatures

Figure 4.39 Synthesized Wash 5 and Rock 42 Interface With Scattered Mines- Daytime

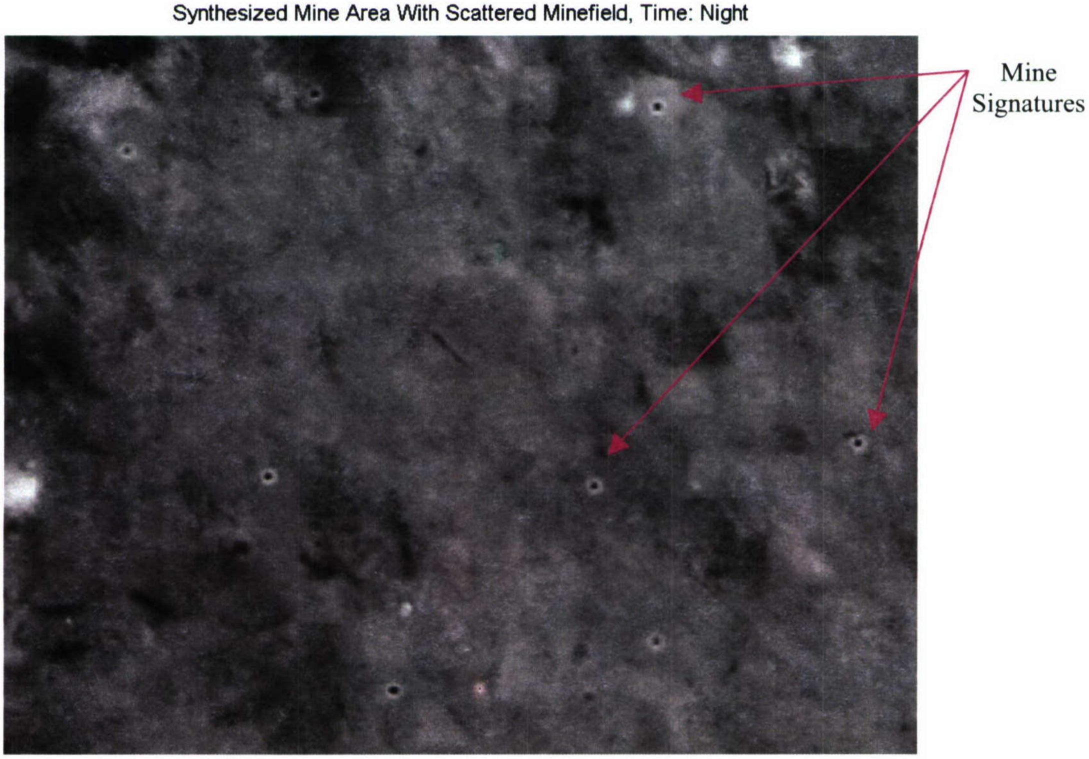


Figure 4.40 Synthesized Mine Area With Scattered Mines- Nighttime

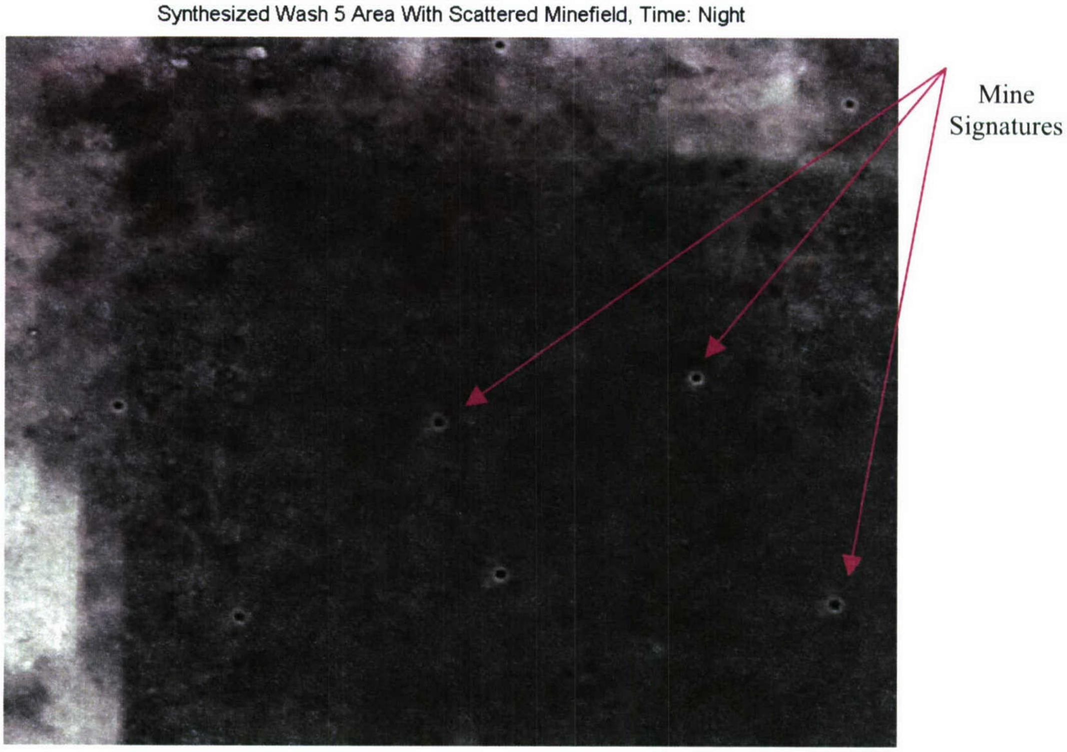

Synthesized Wash 5 Area With Scattered Minefield, Time: Night
Mine
Signatures

Figure 4.41 Synthesized Wash 5 Area With Scattered Mines- Nighttime

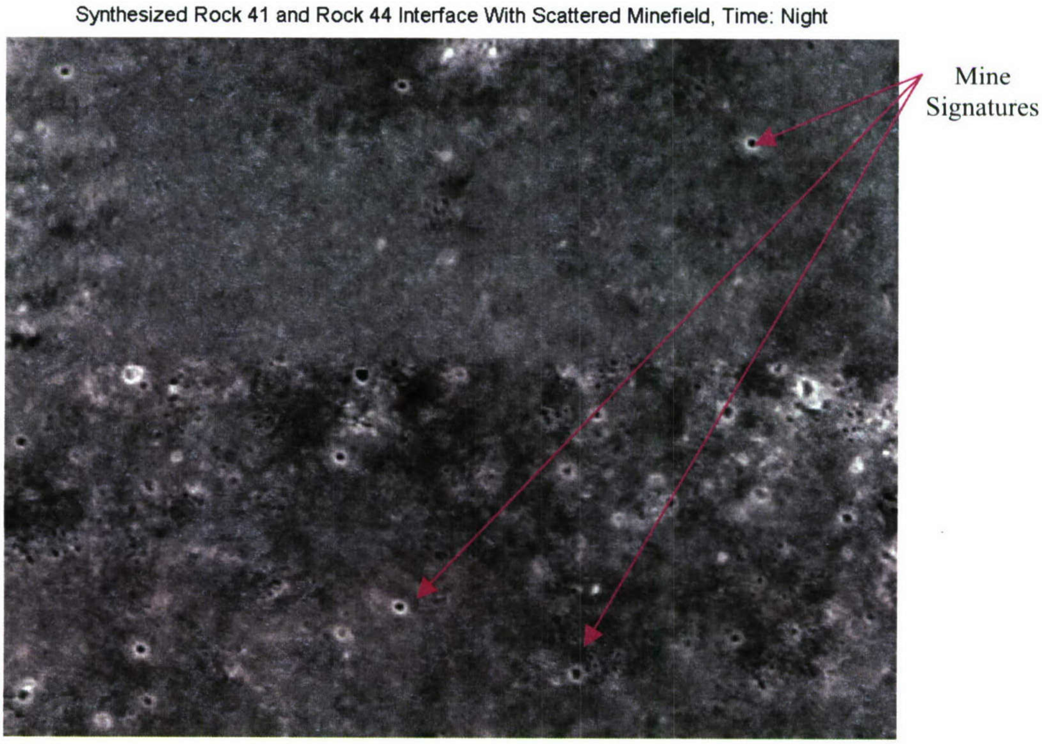

Synthesized Rock 41 and Rock 44 Interface With Scattered Minefield, Time: Night
Mine
Signatures

Figure 4.42 Synthesized Rock 41 and Rock 44 Interface With Scattered Mines-Nighttime

5. SYNTHESIS OF FLIGHT PATH

5.1. INTRODUCTION

So far the synthesis of single frame at specified location on the terrain map has been discussed. But it is possible to extend the same concept to synthesize multiple frames, which can be viewed as a flight path, over the terrain map. Imagery generated in this way should be similar to one obtained by airborne sensor and would be valuable to evaluate minefield detection algorithms. The flight path synthesis is primarily divided into two parts:

- *Define flight path*: The location of the sequence of frames for the flight path over terrain map is defined. Effects of the various components such as flight speed, on board camera frame rate, and the altitude of the flight are discussed here.
- *Synthesize imagery for flight path*: The imagery data is synthesized corresponding to the frame sequence of terrain data identified in previous step.

5.2. DEFINING FLIGHT PATH

This part of the flight path synthesis extracts the terrain information corresponding to the frames that would be imaged by an aircraft during specified flight path over the terrain map. Given the beginning and the ending location of the flight, all the frames on the terrain map are extracted. For the sake of simplicity, the flight path is assumed to be a straight line between the starting $c_0 = (x_0, y_0)$ and the end point $c_e = (x_e, y_e)$. Based on the starting and the end locations, the angle is calculated which is used in updating the camera step for each frame.

Center location for each frame, $l_i = (x_i, y_i)$, is calculated based on flight speed (v in inches/sec), flight angle (θ), and camera frame rate (N in Hz). Flight angle, flight speed, and frame rate are used to update the camera position for each frame until it reaches the end point. This idea is shown in the Figure 5.1, where Δx and Δy gives the updation of the step in x-direction and y-direction respectively. Angle θ gives the direction of the flight path over terrain map. Next frame center is calculated using the eqn (21).

$$\theta = \tan^{-1}\left(\frac{y_e - y_0}{x_e - x_0}\right) \tag{20}$$

$$c_{i+1} = c_i + (\Delta x, \Delta y) \tag{21}$$

where

$$\Delta x = \Delta c \times \sin\theta \text{, and } \Delta y = \Delta c \times \cos\theta \tag{22}$$

and Δc represents the distance moved by the sensor between two consecutive frames and is give as:

$$\Delta c = v / N \tag{23}$$

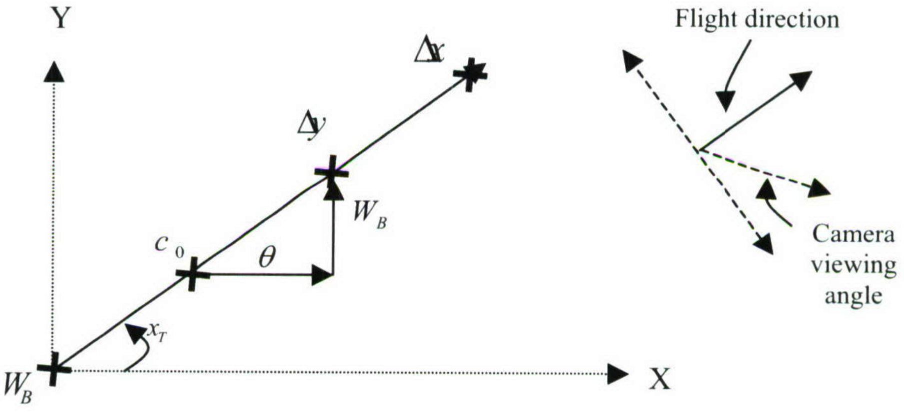


Figure 5.1. Calculation of Frame Center on Flight Path

In all the above calculations, camera-viewing angle is considered to be constant (no stair-step moment of the camera) and perpendicular to the flight direction. Also the camera axis assumed to be aligned with the flight direction. The resultant frame centers are then used to extract the terrain map for each frame. For the implementation convenience, each frame is initially considered as a rectangular box with specified frame parameters such as frame height and frame width, centered at the calculated frame center. This rectangle is then rotated by an angle of $\phi = \pi/2 - \theta$ in the clockwise direction to obtain the final frame. This process is called *pre-synthesis rotation*. Figure 5.2 gives the pictorial representation of this process. In the figure, all the centers along the direction of the flight path are represented with symbols c_0 c_1 $\ldots c_e$, where c_0 represents the starting center and c_e represents the ending center. The dotted boxes FBR_0 FBR_1 $\ldots FBR_e$ represent the frames before rotation centered at c_0 c_1 $\ldots c_e$ respectively. The boxes with thick boundaries, denoted by FAR_0 $FAR_1 \ldots FAR_e$, represent the frames after rotation about the respective centers in the direction of flight. The angle θ gives the direction of flight with respect to the X-axis.

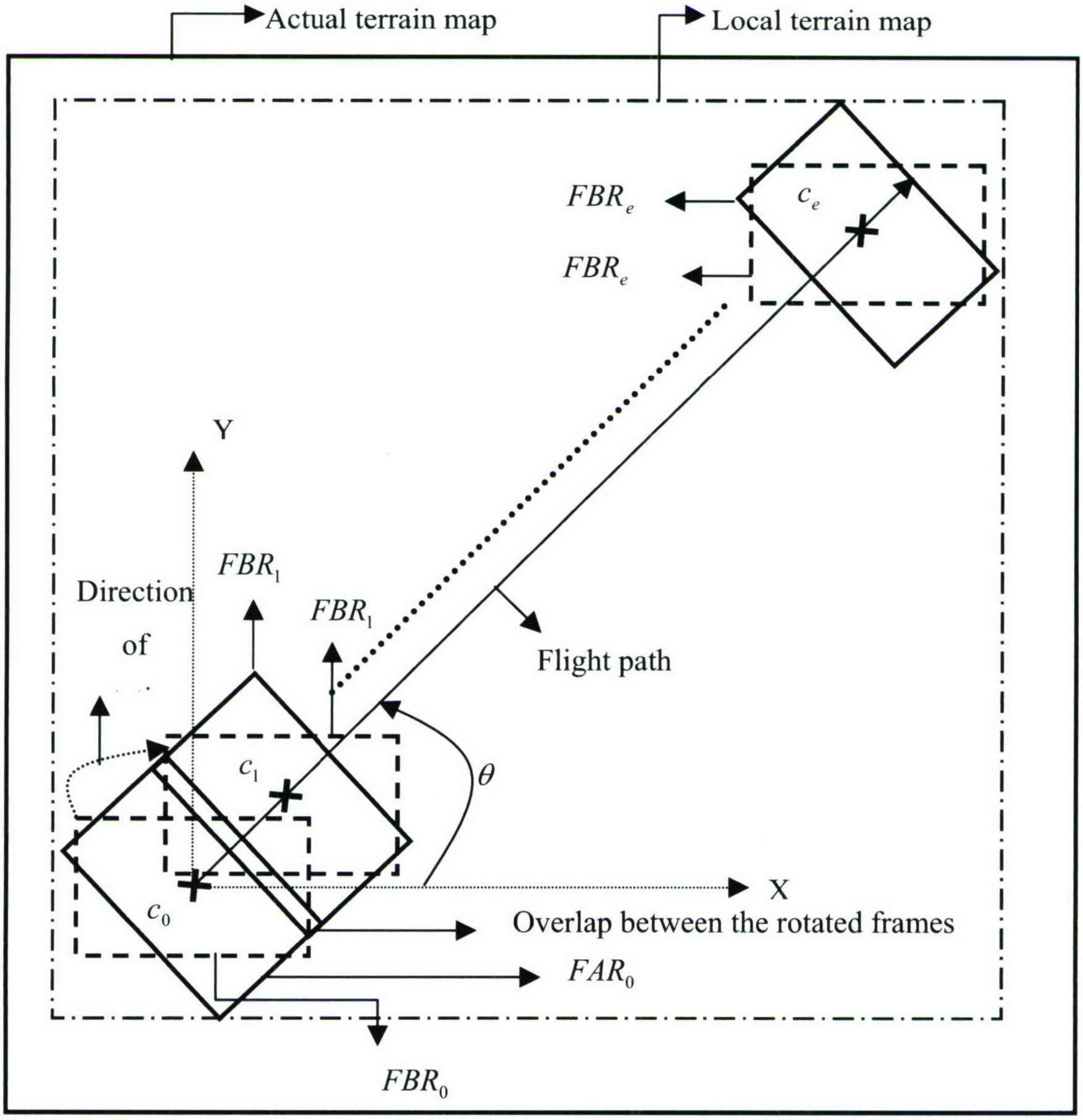


Figure 5.2. The Extraction of Frames for Flight Path Synthesis Over a Terrain Map

Given the direction of the flight and all the four coordinates of the FBR_i, the new co-ordinates of rotated frame (FAR) are calculated as:

$$[\overline{p}_i, \overline{q}_i] = [p_i, q_i]\begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \tag{24}$$

where,

$\phi = \dfrac{\pi}{2} - \theta$,

$[\overline{p}_i, \overline{q}_i]$ = i^{th} corner of FAR,

$[p_i, q_i]$= i^{th} corner of FBR,

Four corners of FBR (assuming upper leftmost corner of local terrain map is at center) centered at $c_i = [x_i, y_i]$ are calculated as:

$$\left.\begin{array}{l} \text{Left upper corner } [p_1, q_1] = [x_i - h/2,\ y_i - w/2], \\ \text{Right upper corner } [p_2, q_2] = [x_i - h/2,\ y_i + w/2], \\ \text{Right bottom corner } [p_3, q_3] = [x_i + h/2,\ y_i + w/2], \\ \text{Left bottom corner } [p_4, q_4] = [x_i + h/2,\ y_i - w/2], \end{array}\right\} \tag{25}$$

where, h = Frame height, and w = Frame width.

The local terrain map, as shown in Figure 5.2, represents the mine terrain map that encompasses all the flight frames of the terrain map. The resultant local terrain map along with the mine information is then passed to the next stage for synthesis. For easy representation of the flight frames over local and actual terrain maps, two kinds of co-ordinates systems, namely, local co-ordinates and global co-ordinates are employed. In the global co-ordinate system points such as the frame center and all the four corners refer to the actual co-ordinates of the original terrain map. The local co-ordinate system stands for the system that is compatible with local terrain map that has been selected for synthesis. Another reason to maintain different co-ordinate systems is that, in most cases, the resolution of the terrain map may not be equal to that of resolution of input imagery that is stored in the image database. In such cases, an appropriate scaling factor is needed while extracting the frames from the local terrain map for error-free synthesis. The global co-ordinates are represented in the units of terrain map resolution; where as the local co-ordinates are represented in the units of input imagery resolution. Synthesis of flight path now boils down to the synthesis of each individual frame alone. However, there are some other issues, which need to be addressed such as preserving the natural overlap between successive frames and accomplish synthesis efficiently. The solution for these problems is explained in Section 5.2.

5.3. RENDER THE FLIGHT PATH

This section explains the synthesis of the frames sequence defined in the previous section. Figure 5.3 shows the local terrain map that is extracted from the desired flight path of Figure 5.2. Having the local terrain map and all the related information about the flight frames, the rest of the process is to synthesize one frame at a time. From Figure 5.3, it is evident that only part of a local terrain map (area covered by the rotated frames) needs to be synthesized. However, synthesizing the rotated frame is difficult and is not compatible with the single frame synthesis explained in the Section 4.3. To get around this issue, a rectangular box that covers the desired frame is synthesized instead of rotated frame itself. The frame that covers FAR is referred as a *Bound Frame* (*BF*). All the bound frames, shown with thick red boxes, are denoted as BF_0, BF_1,... BF_e in Figure 5.3. In order to save synthesis time, the synthesis procedure is only confined to the area covered by bound frames and measures are taken not to synthesize the common area between frames more than once. For this, the local terrain map is divided into synthesis points (similar to one shown in Figure 3.6). Each synthesis point represents the center of the patch on the synthesized image and is spaced by $W_B - W_E$ with its adjacent synthesis points in both horizontal and vertical direction. The reason for dividing the local terrain map into synthesis points is to confine synthesis procedure to only synthesis points that are covered by bound frames.

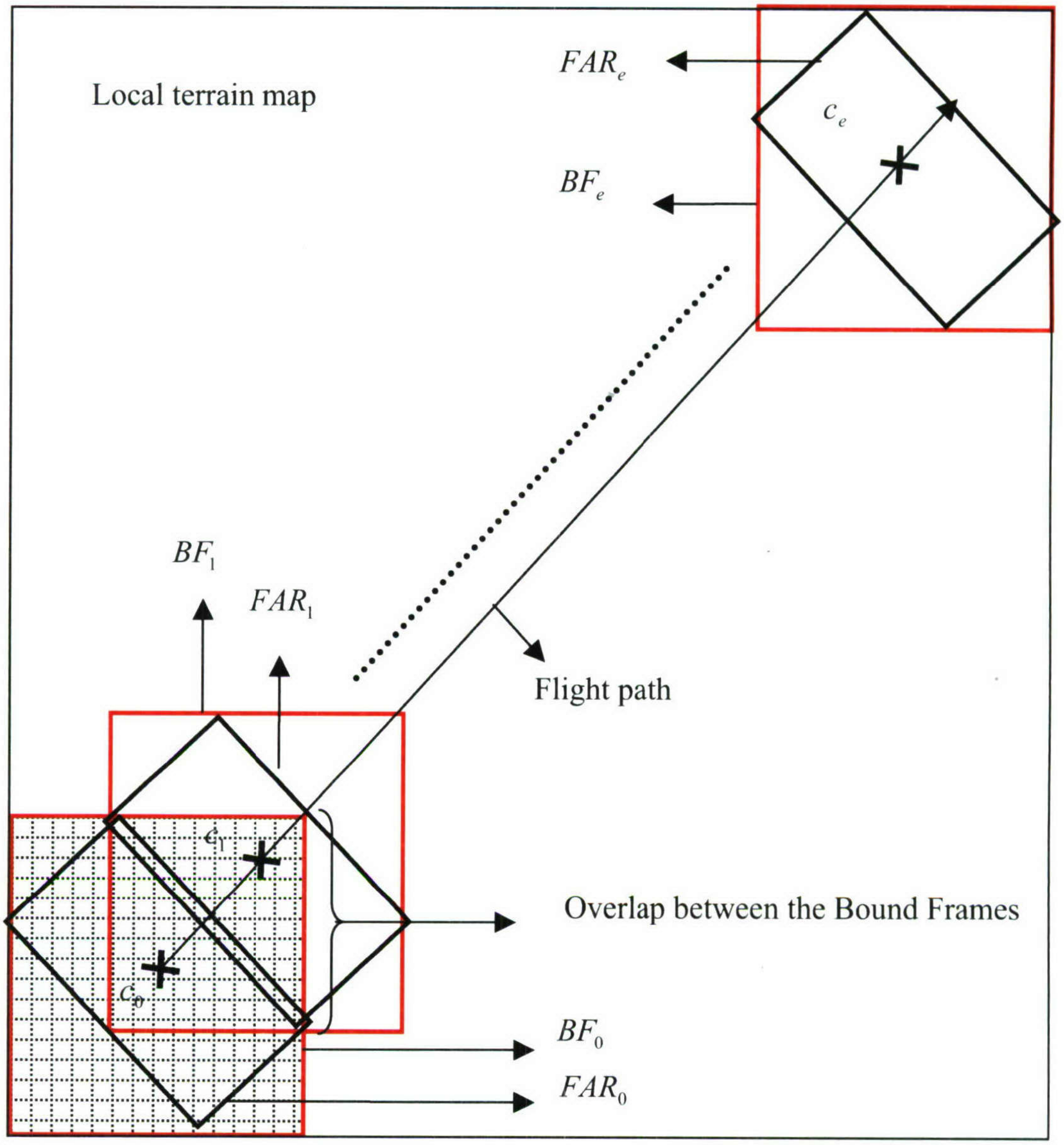


Figure 5.3. Synthesis of Flight Frames

The terrain portion for the first frame is obtained by taking all the synthesis points covered by BF_0 shown in Figure 5.3. The extracted portion along with the mine information (if available) is sent to the synthesis algorithm as an input. Synthesis of the first bound frame is nothing but the synthesis of a single frame as explained in Section 4. All the sample points and mines that are synthesized under this frame are recorded to ensure that these sample points and mine patches are not synthesized in later frames. The synthesis procedure is the same for all the bound frames present in the flight path. But the major difference lies in the extraction of the terrain map and mine locations. This difference is clearly observable from the second frame onwards, where there is an overlap between successive frames. The terrain map for the second frame is obtained by taking all the sample points and mine locations that are included in the second bound frame and excluding the sample points and mine patches that are common to both BF_1 and BF_2. The unfilled part of BF_2 in Figure 5.3 represents the effective terrain map that needs to be synthesized for the second bound frame. The resulting terrain map along with the mine information is then passed to the synthesis algorithm. In addition to this, the common synthesized area between the frames BF_1 and BF_2 is also supplied to minefield synthesizer as an input to maintain the natural flow in image representation between the successive frames. The common synthesized area from BF_1 is utilized in synthesizing the rest of the terrain map represented by BF_2. The same procedure continues till the minefield synthesizer synthesizes the last bound frame BF_e in the flight path.

All the bound frames synthesized in this part are sent to the next stage for further processing called *post-synthesis rotation*. The main aim of the post-synthesis rotation is to extract the actual synthesized flight frames FAR_0, FAR_1 ... FAR_e from the synthesized bound frames. At this stage, each synthesized bound frame is rotated by an angle of $\phi = \pi/2 - \theta$ in anti-clockwise direction to obtain the final synthesized flight frame. Nearest neighbor interpolation method is used in rotating the synthesized bound frames. Figures 5.4 shows the synthesized bound frame (of size 790×820) and the red box represents the actual flight frame that needs to be extracted. Figure 5.5 shows the extracted synthesized frame (of size 512×640) from Figure 5.4 after post-synthesis rotation. In Figure 5.4, X, Y represent the axes of the bound frame and X1, Y1 represent axes of the final flight frame along the direction of the flight. Angle ϕ represents the direction of the rotation of the bound frame.

Bound Frame

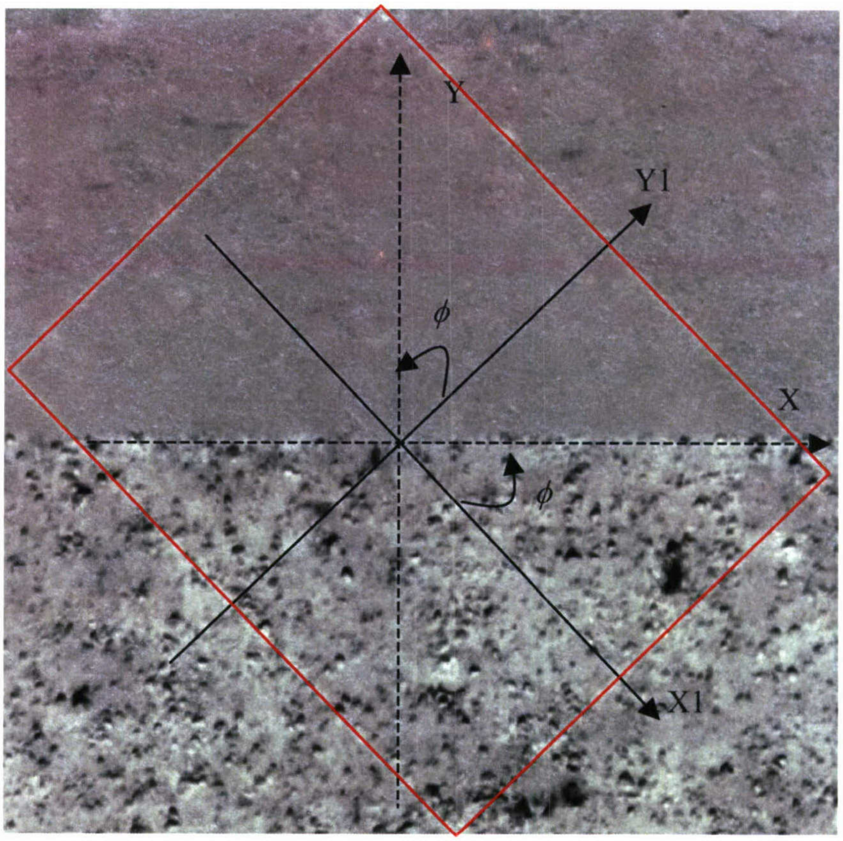


Figure 5.4. Synthesized Bound Frame

Final Synthesized Flight Frame

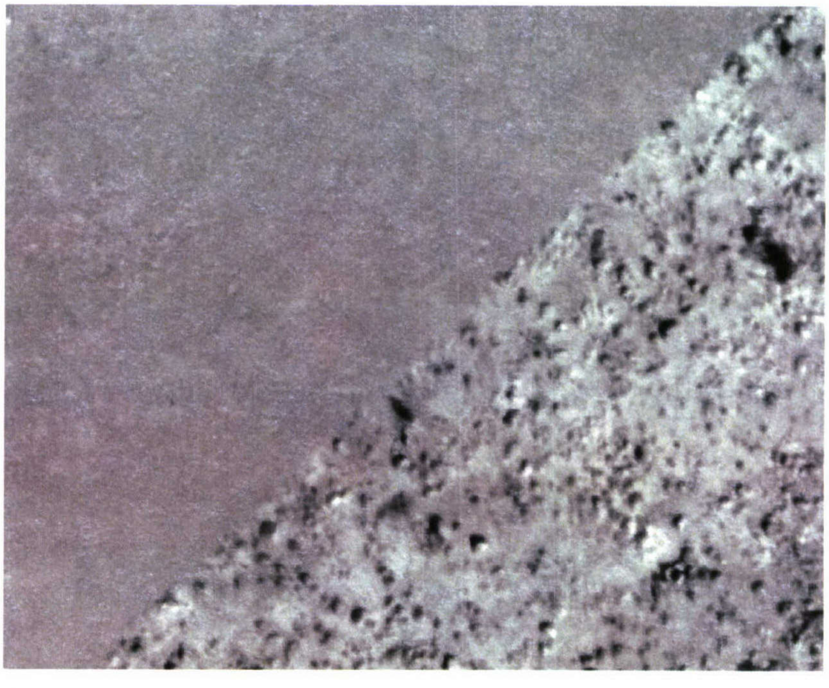


Figure 5.5. Synthesized Flight Frame After Post-Synthesis Rotation

5.4. RESULTS

In this section, some of the synthesized frames of flight path over the terrain map are presented. All the synthesized images shown in this section are of size 512×640, 1.1-inch ground resolution, and represent the images after the post synthesis rotation.

Figure 5.6 shows the direction (South-East to North-West) of the flight path over the terrain map with scattered mine layout. In this figure, flight path is covering the three different kinds of the backgrounds such as Mine Area, Wash 5 and Rock 41. Images shown in Figure 5.7 represent the four consecutive synthesized frames of the flight path defined in Figure 5.6. In Figure 5.7, frame 1 enclosed three different backgrounds where as the other three frames enclosed only two backgrounds and all the frames have mine signatures. Figure 5.8 represents the flight direction (South to North) over the scattered minefield structure. In this case, frames 2 and 4 have only one background and frame 1 and frame 3 has two different backgrounds. Figure 5.9 represent the synthesized frames of the flight path defined in Figure 5.8. In Figure 5.9, mine signatures are available on the all frame except frame 1. Figures 5.10 show the direction (South-West to North-East) of the flight path over the terrain map. In this case, all the frames have two backgrounds and all the frames expect frame 1 have mine signatures on them. Figure 5.11 represent the synthesized frames of the flight path defined in Figure 5.10. Figure 5.12 represent the direction (South-West to North-East and West to East) of flight and corresponding flight frames over the terrain map with the scattered minefield structure. Figure 5.13 represent respective synthesized flight frames defined in Figure 5.12. All the synthesized flight frames shown in Figure 5.13 have two backgrounds and the mine signatures on them. In all the above synthesized flight frames, overlap between the successive frames is preserved. Gray level variation from one background to other background and one frame to the other frame are natural.

All the flight frames shown above are extracted from the same terrain map and are synthesized from the input imagery, which are collected during the nighttime. Center locations of the flight frames in the above four flight paths are calculated based on the flight speed at 130 inches/sec and the camera frame rate at 6 frames/sec. In calculating center frames fixed camera viewing angle is considered. Flight frames shown in figures 5.6 and 5.10 are synthesized in 1900 sec per four frames, where as flight frames shown in

Figure 5.8 and 5.12 are synthesized in 1700 sec. The reason for less synthesis time for the flight frames shown in figures 5.8 and 5.12 is that there exist more overlap between the successive frames than the flight frames shown in figures 5.6 and 5.10.

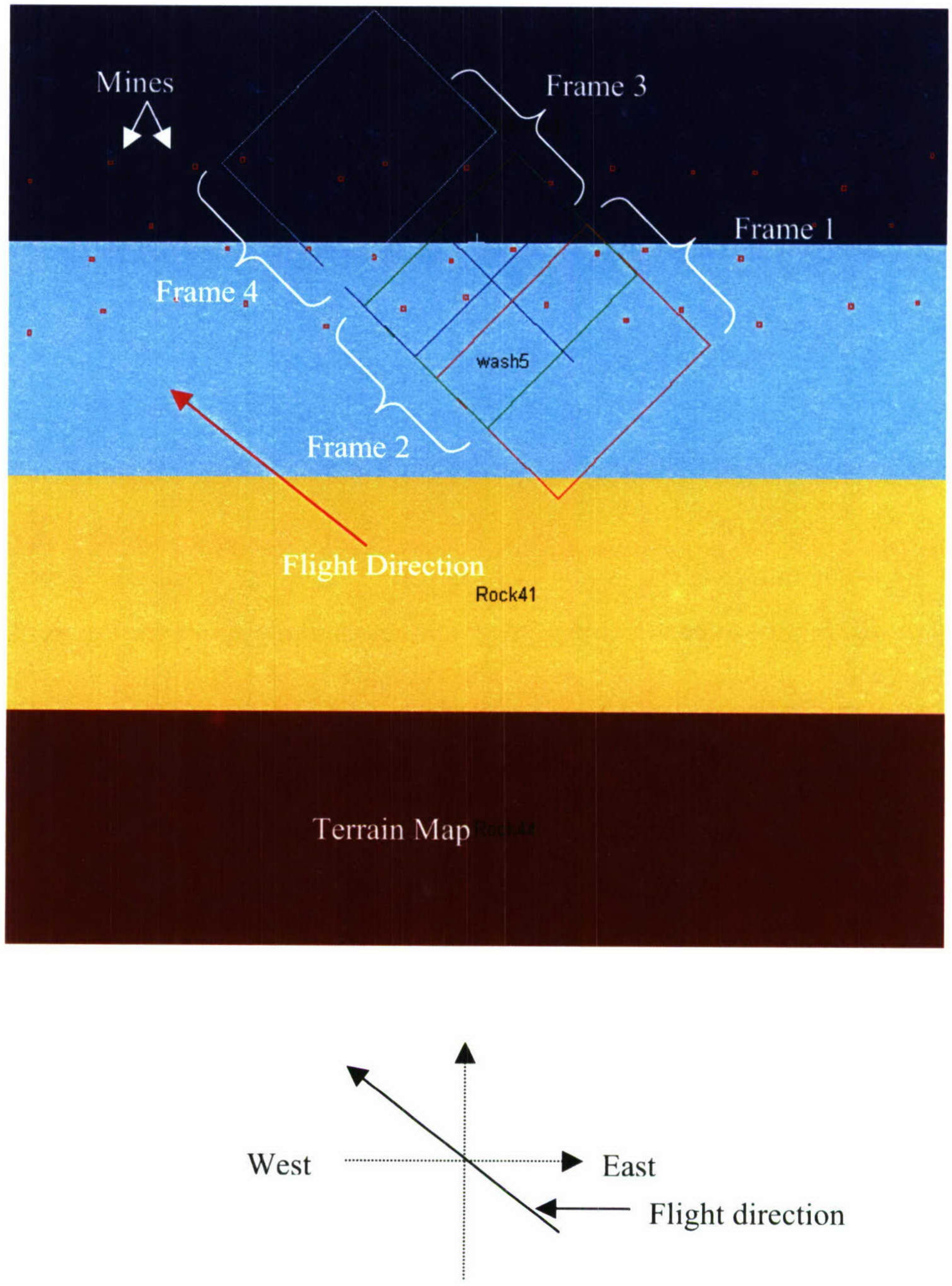
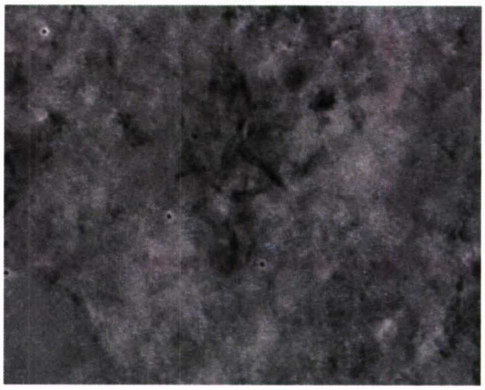
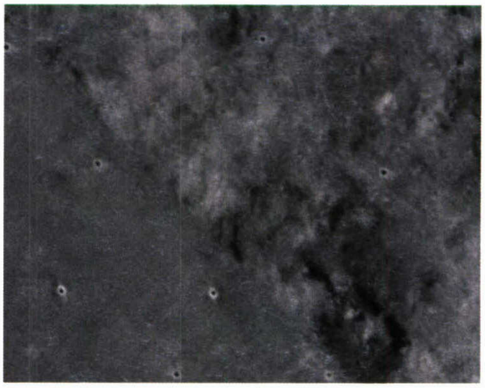


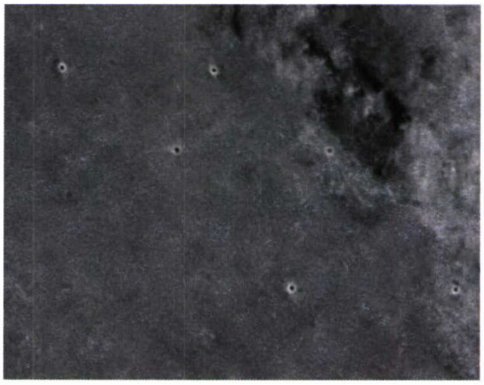
Figure 5.6 Flight Direction (South-East to North-West) Over Terrain Map



Frame 4: Mine area and Wash 5 area interface with scattered mines



Frame 3: Mine area and Wash 5 area interface with scattered mines



Frame 2: Mine area and Wash 5 area interface with scattered mines

Frame 1: Wash 5 area and Rock 41 area interface

Figure 5.7 Synthesized Flight Frames for South-East to North-West Flight

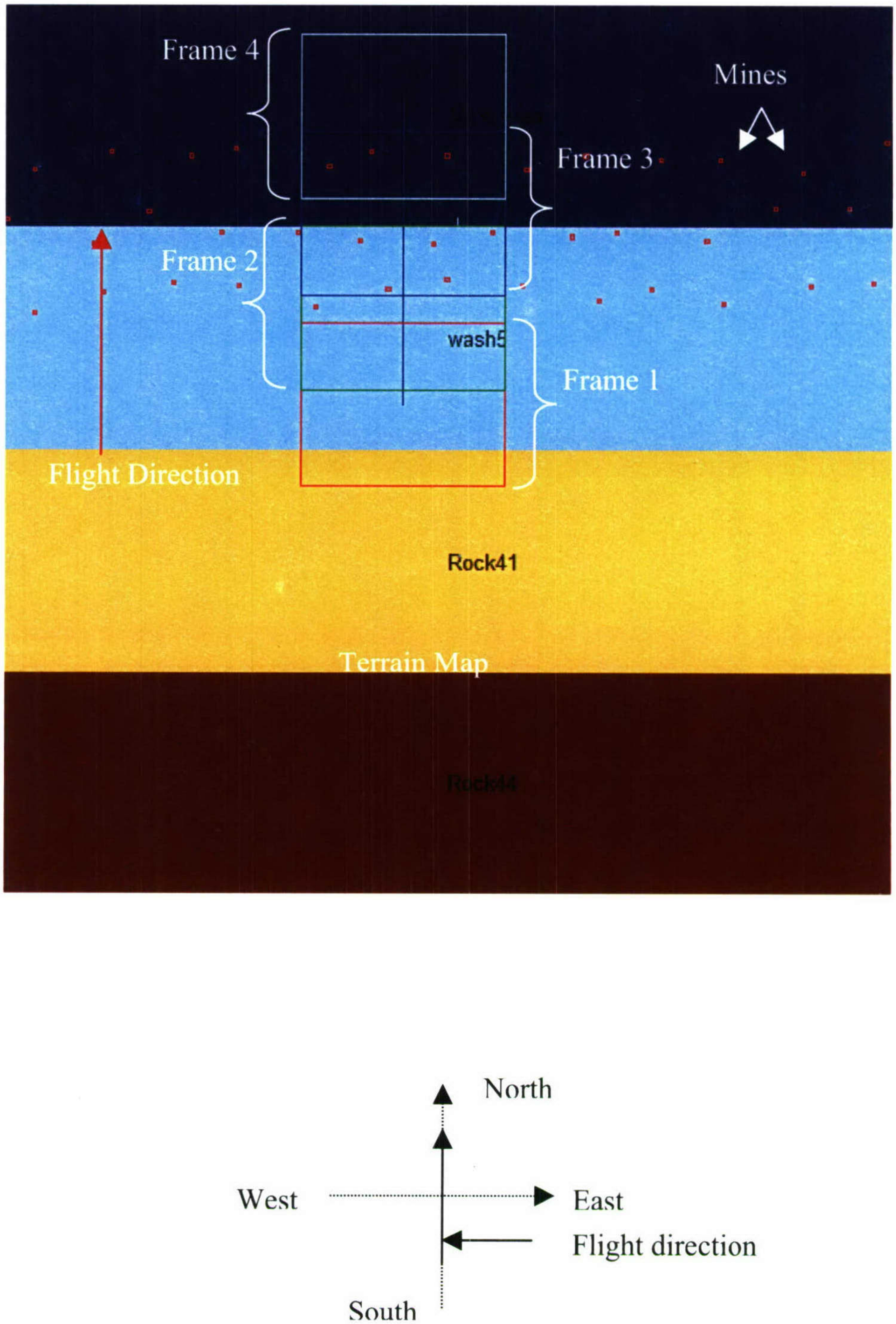
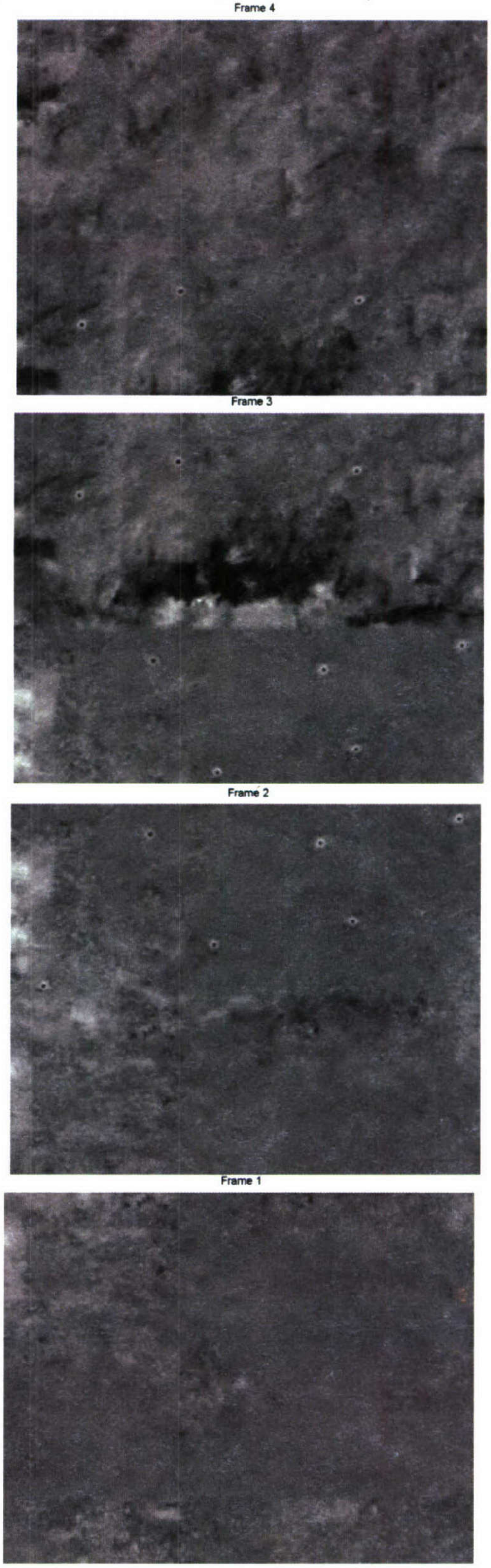


Figure 5.8 Flight Direction (South to North) Over Terrain Map



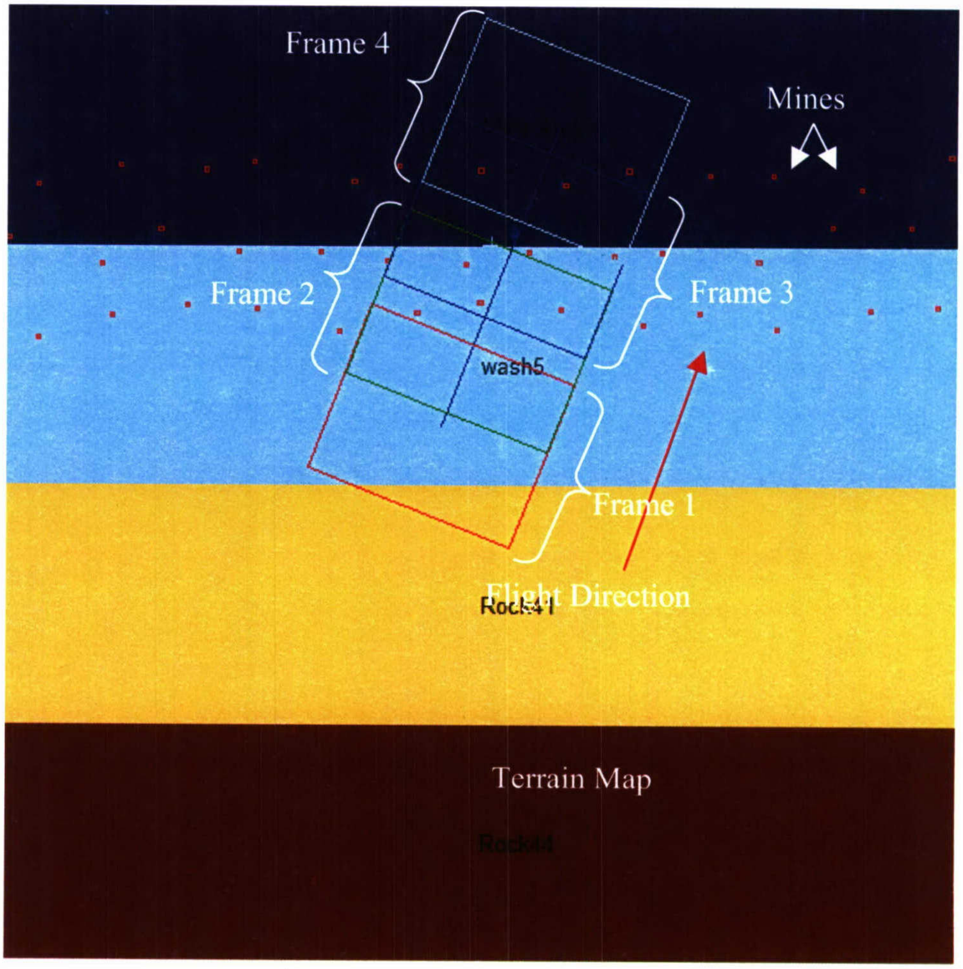
Frame 4: Mine area with scattered mines

Frame 3: Mine area and Wash 5 area interface with scattered mines

Frame 2: Wash 5 area with scattered mines

Frame 1: Wash 5 area and Rock 41 area interface without mines

Figure 5.9 Synthesized flight frames for South to North Flight



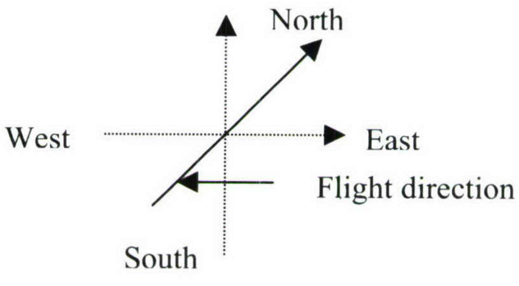
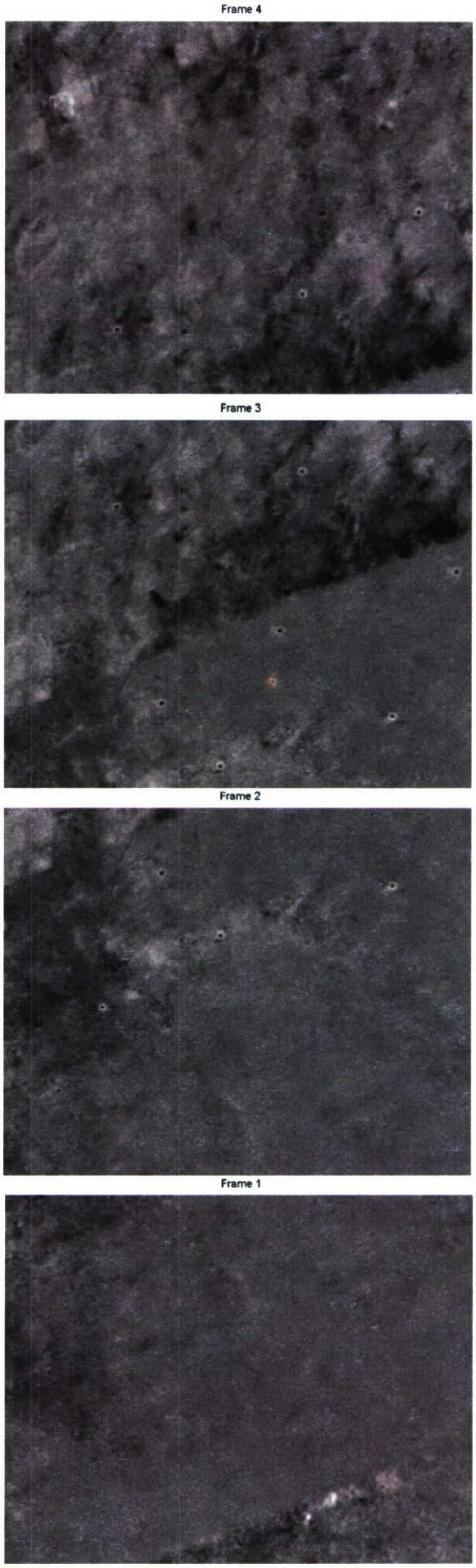


Figure 5.10 Flight Direction (South-West to North-East) Over Terrain Map



Frame 4: Mine area and Wash 5 area interface with scattered mines

Frame 3: Mine area and Wash 5 area interface with scattered mines

Frame 2: Mine area and Wash 5 area interface with scattered mines

Frame 1: Wash 5 area and Rock 41 area interface with scattered mines

Figure 5.11 Synthesized Flight Frames for South-West to North-East Flight

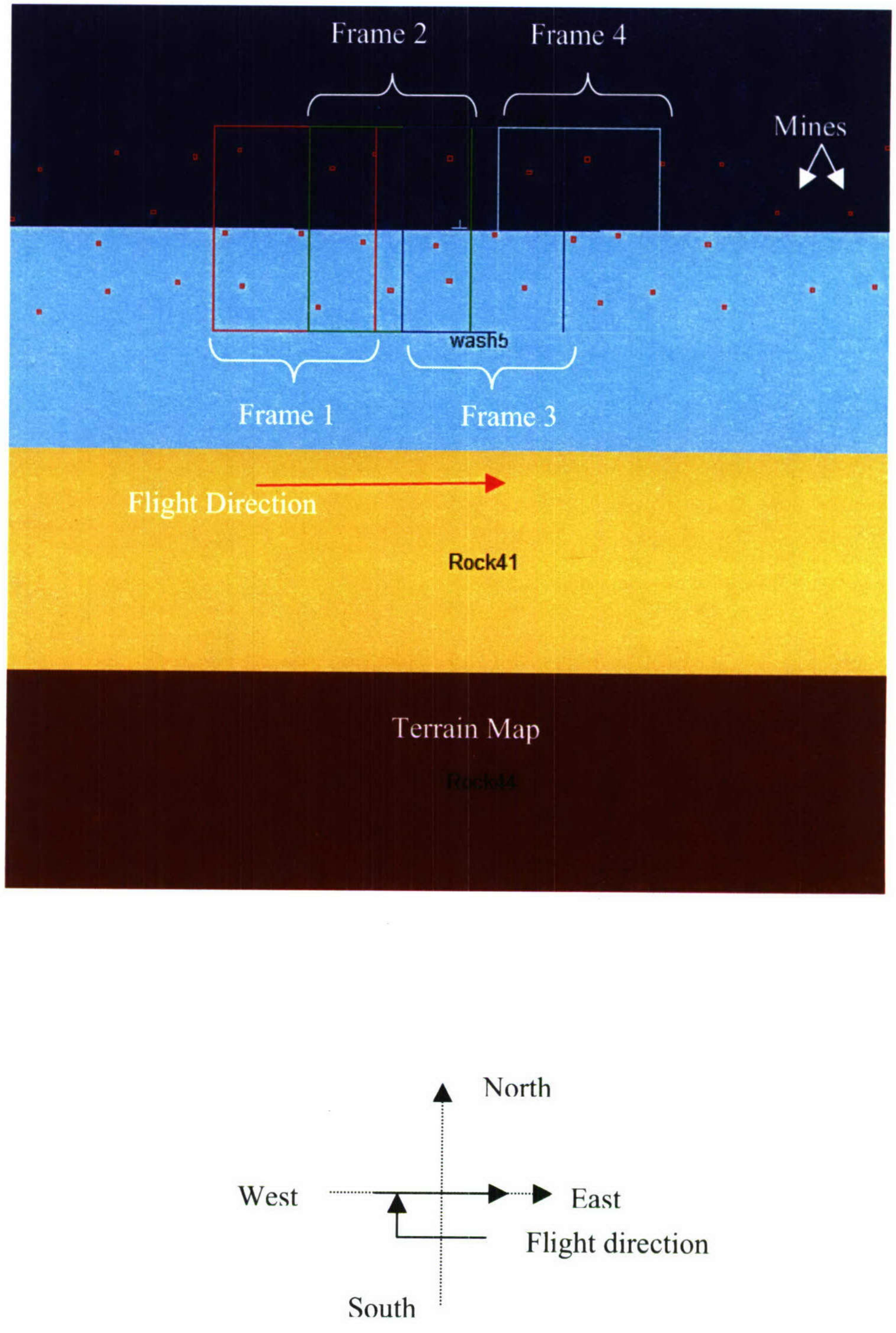
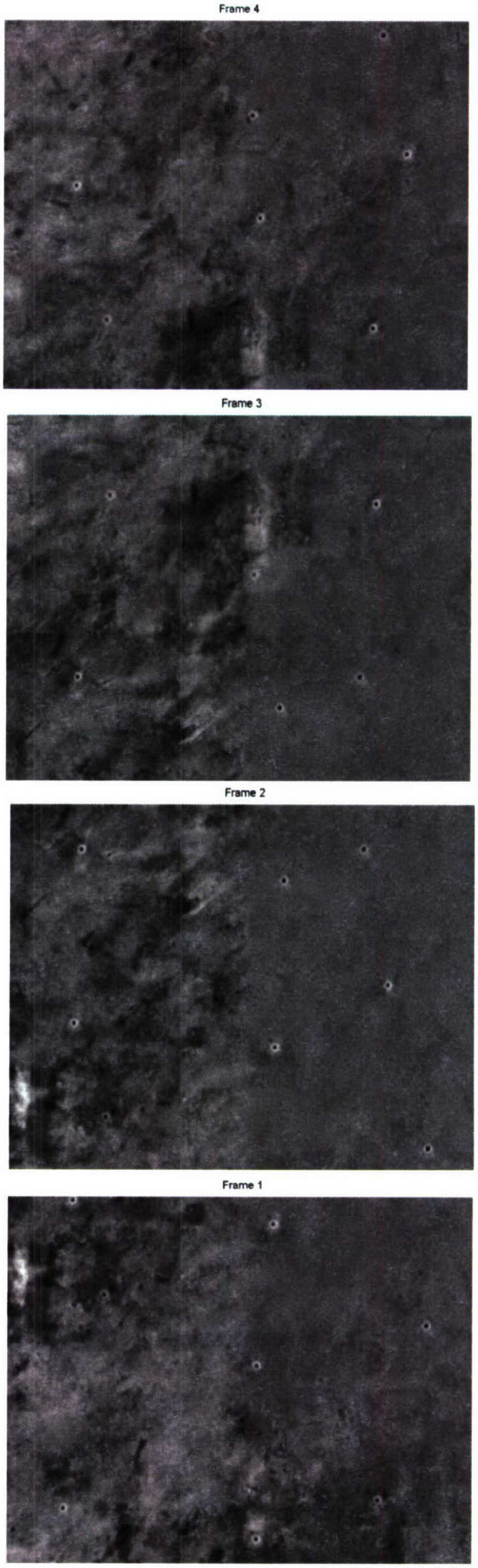


Figure 5.12 Flight Direction (West to East) Over Terrain Map



Frame 4: Mine area and Wash 5 area interface with scattered mines

Frame 3: Mine area and Wash 5 area interface with scattered mines

Frame 2: Mine area and Wash 5 area interface with scattered mines

Frame 1: Mine area and Wash 5 area interface with scattered mines

Figure 5.13 Synthesized Flight Frames for West-East Flight

6. CONCLUSIONS AND FUTURE WORK

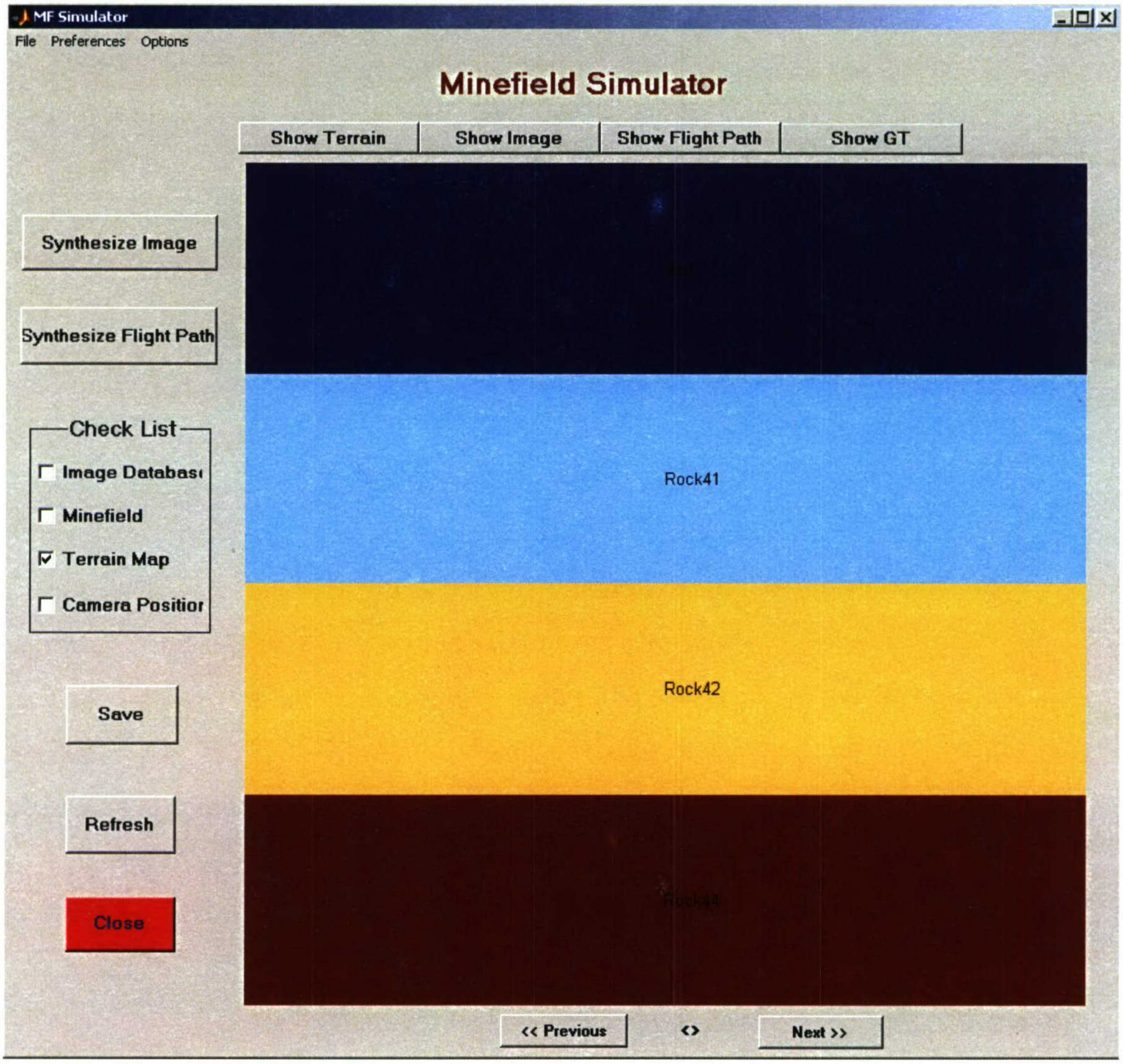
This thesis illustrated synthesis of various minefield scenarios using patch-based sampling technique and also showed how minefield imagery data could be generated for validation of various mine/minefield detection algorithms. This work also described the concept and working details of the single frame and multi frame (flight path) synthesis of the terrain map. Along with this, synthesis of mine-only frames over different backgrounds is also explored to evaluate various minefield layout strategies for the given type of background information. The proposed synthesis procedure works well for the wide variety of the input imagery and is capable to generate imagery with any desired terrain structure. In this thesis, concepts such as Quad-Tree structure, Karhunen-Loeve Transform, and Markov random fields are also explored to improve the performance of the synthesis algorithm and succeeded in getting better performance. Most of the imagery generated in this work is very near to real world imagery and can be used for many practical purposes.

Quality of synthesized images is one of the most important aspects of image based minefield synthesis and can be improved by using better modeling techniques in selecting neighboring image patches. Apart from modeling there is one more factor that directly affects the quality of the synthesized image i.e. number of input images. This is because an increase in the number of input images gives us more input patches to spare, which in turn increases the probability of getting a suitable patch for a given criteria that eventually results in a better image quality. In the present case, only a few number of images (less than 100 images of size 512×640) are considered for the synthesis. To synthesize larger terrain maps one needs to have a lot of imagery to avoid the repetition of the patches in synthesized image. However, an increase in the number of input images may result in large space constraints. Increase in number of input images also adversely affects the speed of the synthesis algorithm. These problems can be overcome by using better storage and data management techniques. Present approach uses the quad-tree approach to speed-up the search for best neighborhood and is effective for relatively small number of input images. If the number of input images increases, searching through quad-tree itself takes a sufficiently large time that may not be desirable in real world operations. In synthesizing flight path straight path between the starting and end points is considered. However, in practice, flight can take any path between the starting and the end points and should be synthesized using the geometric properties of the flight path. Also current synthesis is developed for push-broom mode and can be expanded for stair-step mode of data collection.

APPENDIX A
A GRAPHICAL USER INTERFACE FOR MINEFIELD SYNTHESIS

A. INTRODUCTION

The main aim of this graphical user interface is to provide a flexible and integrated environment to synthesize various minefield images using minefield synthesis algorithm proposed in the present thesis. One of the main motivations behind the development of this utility is to understand the true capabilities of the proposed synthesis algorithm in generating real-world like imagery for different kinds of input imagery. Moreover, the results generated by the minefield synthesizer are also used for the evaluation of various mine/minefield detection algorithms as part of the airborne mine and minefield detection programs.



Check List
☐ Image Databas
☐ Minefield
☑ Terrain Map
☐ Camera Positior

Figure A.1. Main Interface Window for MF Simulator

Figure A.1 shows screen shot of the main user interface window for the minefield simulator. The present version of the minefield simulator, called as MFSimulator, is equipped with all the features discussed in the above thesis such as, options to load different terrain and minefield structures, and option to place minefield layout in different locations on the terrain map. The utility is capable to synthesize a single frame data as well as list of frames along a flight path. The application is developed in MATLAB-6.5 using the GUIDE design tool and has been tested only under Windows-2000 environment. However, it may be possible to run the same application under other operating systems with some simple modifications. A complete summary of the various features of this application and user interface can be found in the MF Simulator user manual [23]. Some of the main utilities integrated in the user interface are listed below:

1. Import - Image Database. The image database contains the information regarding patches that are extracted from different terrain and associated quad tree information to be used in minefield synthesis. An *Image Database* panel [23] that shows the information such as, number of input images as well as patches associated with each background, number of available mine signatures per each background, resolution of the input images, and size of both database patch (DP) and synthesis patch (SP), will be displayed at the end of loading process.

2. Import - Pre-defined Terrain Map. A terrain map contains all the relevant terrain information in the form of an indexed image. A *Terrain Map Data* panel that displays all the allied information to the terrain map such as different backgrounds that are presented in the terrain map, terrain map resolution, terrain map size and terrain map geographic location is displayed. Using this panel it is also possible to compare the name of the background and associated index value in terrain map.

3. Import - Pre-defined Minefield Layout. The minefield layout contains the information related to the mine type and relative mine location in the terrain map.

4. Relocate Minefield on the Terrain Map. This feature is used to relocate the loaded minefield structure on the terrain.

5. Synthesis Parameters. This feature allows to specify certain parameters such as image frame width, image frame height, flight speed, frame rate, and flight altitude, which dictate the course of the minefield synthesis procedure.

6. Camera Position. This feature picks a location on the terrain map that acts as the center of the area for which the minefield image is going to be synthesized. The *Set Camera Position* then extracts the terrain map based on image size and resolution and makes this information available (including mine information) to the minefield synthesis algorithm as an input.

7. Define Flight Path. This feature is used to define flight path over the terrain map. Once the starting and end points of the flight path are specified, *Define Flight Path* extracts all the possible frames in between the starting and the end points, assuming a straight line path. Overlap and resolution of the extracted flight frames will be dictated by the flight speed, frame rate, and altitude of the flight, respectively.

8. Synthesize Image. This feature synthesizes a single frame terrain map selected previously. The synthesis procedure can be started without loading the minefield structure.

9. Synthesize Flight Path. This feature initiates the minefield synthesis procedure based on the earlier defined flight path.

10. Save. This feature saves an individual image frame or set of frames for flight path in *itv* format. It also saves corresponding ground truth file in *csv* format.

APPENDIX B
CLUTTER REMOVAL

B.1. INRODUCTION

The main purpose of this utility is to remove any fiducial markings that are present in the original airborne MWIR imaginary. Any markings or objects other than mines are considered as a fiducials. Removal of the fiducials from the original imagery may improve the visual richness of the synthesized image and also help in better evaluation of mine detection algorithms. For human in loop evaluation of mine and minefield detections, removal of fiducials from the original imagery removes artificially added cues, which may bias the evaluation.

The basic idea behind the clutter removal is to extract the clutter patch and replace that with appropriate background-only patch. A clutter patch is defined as an image patch with a clutter object at its center and a background patch is defined as an image patch without mines and clutter objects. In order to preserve the originality of the image, background patch is selected from the same image. Initially a set of background patches are extracted from an image, from which the clutter needs to be removed. A clutter that needs to be removed is extracted by using ground truth of the input image and is replaced with the same size background patch. Ground truth of an input image contains both mine and clutter information and their location on original image. Background patch that is used to replace the node patch is selected based on the high correlation, as explained in Section 4.1.4, between the clutter patch and all the available background patches. To obtain a better correlation, a bigger size for background patches than the clutter patch size is assumed.

B.2 PATCH SELECTION

Selection of an appropriate background patch, which is used to replace the node patch, is summarized below:

1. Extract all background patches from the input image from which the fiducials needs to be removed.
2. Based on the ground truth of the input image obtain the first fiducial patch from the input image and extract the boundary region of the extracted patch. The extracted boundary patch is then slide over each background patch by storing all the correlation coefficient between the boundary regions of the

clutter patch and background patch. Find a portion of the background patch, whose size is equal to the size of the clutter patch, based on the high correlation coefficient. If this value is greater than or equal to the some predefined threshold (which is 0.5 for this work), stop the search at return the corresponding portion of the background patch.

3. If the maximum correlation coefficient found in step 1 is less than threshold value, pick next background patch and repeat step 1. This process continues until a portion of background patch whose boundary region has a correlation coefficient grater than the predefined threshold with the clutter edge region is found. However, there exists a case where all correlation coefficient may not exceed threshold, in such cases, patch with the highest correlation coefficient will be selected as final patch.
4. Finally blending operation is done between the clutter patch and the background patch to provide a smooth transition between freshly pasted background patch and its surroundings.

B.3. BLENDING OF SELECTED PATCH

Figure B.1.1 shows the structure of the clutter patch. The shaded region of the clutter patch is main area of interest in selecting background patch.

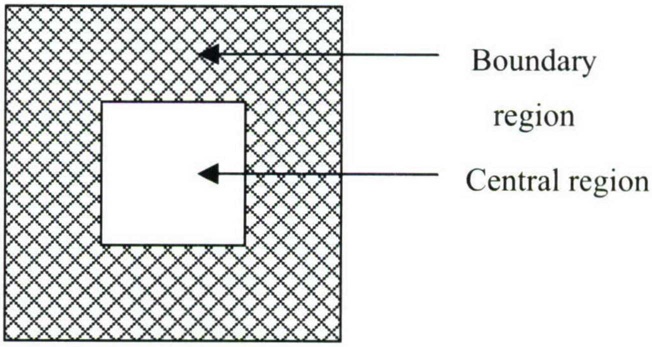


Figure B.1. Clutter patch structure

The blending operation performed in this part is same as that of mines blending explained in Section 4.2. However, for this case, mine weight matrix (created in Section 4.2) will be termed as background weight matrix and background weight matrix is termed as clutter weight matrix. In synthesizing mines, mine location of the mine patch is an important area. Thus, more weight is given to center pixels of the mine patch and less weight (zero weight) is given to the center part pixels of the background area in creating weight matrices. On contrary, in removing clutter, the clutter portion of the clutter patch is of least important. Hence, in creating weight matrices, more weight is given to center pixels of the background patch and zero weight is given to the center pixels of the clutter patch. The following equation is used to blend the background and clutter patches:

$$I_0(x,y) = \frac{d_B(x,y)I_B(x,y) + d_C(x,y)I_C(x,y)}{d_B(x,y) + d_C(x,y)} \qquad \text{(B.1)}$$

where

$I_B(x,y)$ = Gray value of the pixel at (x, y) of the background patch,

$I_C(x,y)$ = Gray value of the pixel at (x, y) of the clutter patch,

$d_C(x,y)$ = Weight value of the of the clutter weight matrix at (x, y),

$d_B(x,y)$ = Weight value of the of the background weight matrix at (x, y),

$I_0(x,y)$ = Gray value of the pixel at (x, y) of the resultant image.

Figure B.2 (a) and (b) represent the same image frame (of size 512×640**)** before and after removing fiducials. Original image frame has four fiducial in it. These fiducials are represented with cross marks in Figure B.2 (a). In Figure B.2 (a), these fiducials have been replaced with background patches extracted from the same image. The modified figure shows no artifacts and appears natural as compared with original image.

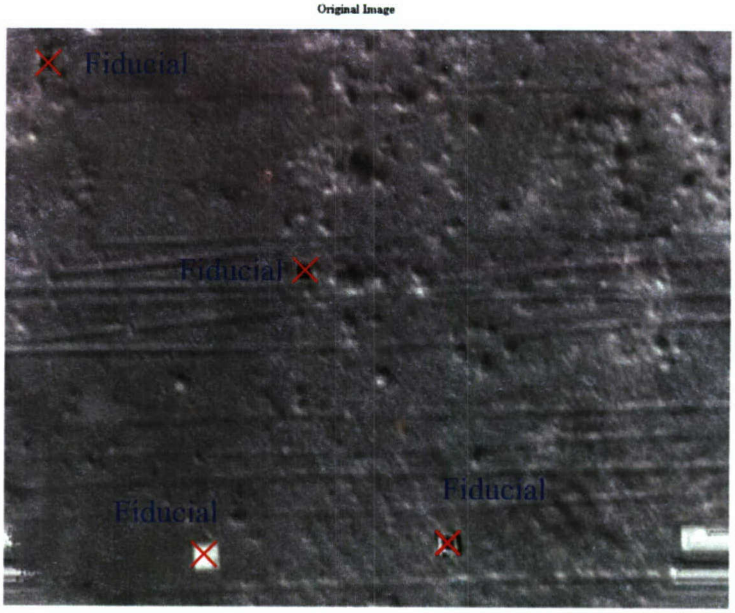


Figure B.2 Original input image with fiducials.

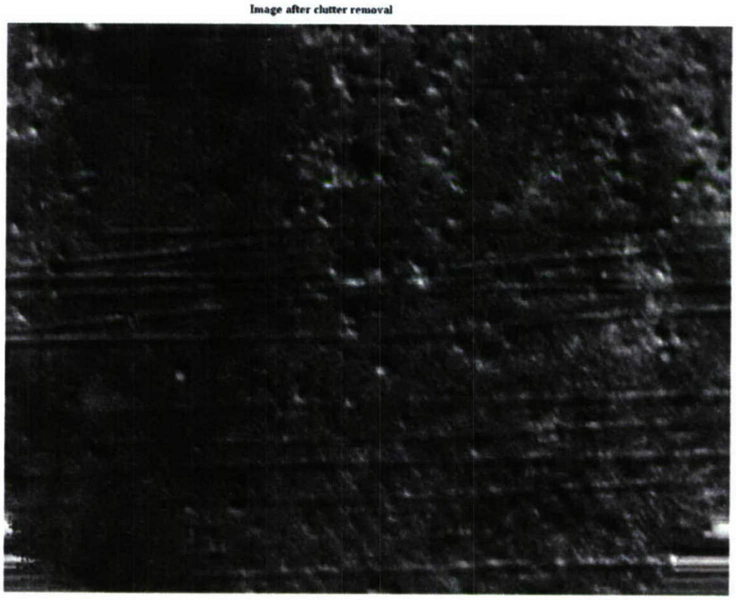


Figure B.3 Image in Figure B.2 after removal of fiducials

APPENDIX C
FUNCTION REFERENCE

C.1. Function Name: MinefieldSynthesisForKLT

Description

The main function that synthesizes the new minefield frame by using the patch-based sampling algorithm.

Usage

[finalSynthImg, bkgHistory] = MinefieldSynthesisForKLT (bkgData, mineData, dataParam, synthParam, terrainMap, minefieldLayout, dataDir)

Inputs

bkgData - background patch database

mineData - mine patch database

dataParam - database parameters such as database patch size, tolerance, synthesis patch size, and synthesis patch edge width.

synthParam - synthesis parameters such as image height and width.

terrainMap - terrain structure of the synthesized image

minefieldLayout - minefield structure of the synthesized image

dataDir - the directory where the original images are stored

Outputs

finalSynthImg – synthesized minefield frame

bkgHistory – background history such as patch number and terrain identification at each sample point of the synthesized image.

C.2. Function Name: GetPatch

Description

Reads the patch with specified size, from the input image.

Usage

[outPatch, meanVal] = Getpatch (patchInfo, dataDir, W)

Inputs

patchInfo - contains the information such as the patch number, starting row and column number on the input image, name of the original image, and the path name

dataDir - the directory where the original images are stored

W – size of the outpatch

Outputs

outPatch – Extracted patch from the input image

meanVal – mean of the input image from which current patch is extracted.

C.3. Function Name: selectCenterPatches

Description

Selects the set of database patches whose central characteristics are in confirmation with the MRF model of the background.

Usage

patches = selectCenterPatches(Tree, myTree, TModel, branchID, usedPatch, MinPatchNum)

Inputs

Tree - central quad-tree structure for specific background

myTree - contains the central identification number for all the available patches for a given background

TModel – Background model

branchID – central identification number of the neighborhood patch/patches

usedPatches – already synthesized patch information

MinPatchNum – minimum number of database patches needs to be selected (For present case this value is 50)

Outputs

patches – selected database patches (ψ_{CT})

C.4. Function Name: selectEdgePatches

Description

Selects the set of database patches whose edge characteristics match with the edge characteristics of the synthesized neighborhood patches.

Usage

patches = selectEdgePatches(Trees, myTrees, branchList, usePatch, MinPatchNum)

Inputs

Tree - edge quad-tree structure (left, top)

myTree - contains the edge identification numbers for all the available patches for a given background

branchList – edge identification and corresponding branch identification numbers of the neighborhood patch/patches

usedPatches – already synthesized patch information

MinPatchNum – minimum number of database patches needs to be selected (For present case this value is 5)

Outputs

patches – selected database patches (ψ_{ECT}).

C.5. Function Name: getPatchFromList

Description

Selects a synthesis patch from ψ_{ECT} that has the best possible correlation with its neighborhood patches.

Usage

[patchData, imgName, idx, row, col, val, meanVal, done] = getPatchFromList (finalPatchSet, dataParam, dataDir, synthPatch, mask, Tol)

Inputs

finalPatchSet – set of the patches selected from the function selectEdgePatches (ψ_{ECT})

dataParam - database parameters such as database patch size, tolerance, synthesis patch size, and synthesis patch edge width.

synthPatch – synthesized neighborhood patch /patches

mask – edge region of the synthesized neighborhood

Tol – tolerance allowed in extracting the database patch

Outputs

patchData – selected synthesis patch.

imgName – name of the original image from which the synthesis patch is extracted

idx – number associated with the database patch

row – starting row number of the synthesis patch in database patch

col - starting column number of the synthesis patch in database patch

val – correlation coefficient between edge region of the selected synthesis patch and neighborhood edge region

meanVal - mean of the input image from which current synthesis patch is extracted

done – correlation flag, this value is equal to one if the correlation coefficient associated with the selected synthesis patch is grater than or equal to the specified threshold; otherwise this value equals to zero.

C.6. Function Name: PlaceMines

Description

Places the mine patches belongs to various mine types on already synthesized background-only image

Usage

[finalSynthImg] = PlaceMines (mineData, minefieldLayout, synthImg, Wb, We, dataDir)

Inputs

mineData - mine patch database

minefieldLayout - minefield structure of the synthesized image

synthImg – synthesized background-only image

Wb – synthesis patch size

We – edge with of the synthesis patch

dataDir - the directory where the original images are stored

Outputs

finalSynthImg – synthesized minefield frame

C.7. Function Name: MinesBlending

Description

Blends the selected mine patch with the already synthesized background-only image

Usage

blendPatch = MinesBlending (mPatch, mWeight, bPatch, bWeight)

Inputs

mPatch – mine patch

mWeight – weight matrix for the mine patch

bPatch - equivalent background patch

bWeight – weight matrix for the background patch

Outputs

blendPatch – blended mine patch

C.8. Function Name: defineFlightPath

Description

Extracts the terrain information and other related information corresponding to the flight frames

Usage

[sLoc, eLoc, ang, globalLoc, localLoc, locacTerr, loclMineLoc] = defineFlightPath (terrMap, terrMapParam, globalMineLayout , Param)

Inputs

terrainMap – global or actual terrain map

terrainMapParam – terrain map parameters such as size and resolution

globalMineLayout - minefield structure presented over the global terrain map

Param – flight path synthesis parameters such as flight frame height and width, flight speed, and camera shutter speed.

Outputs

sLoc – starting location of flight

eLoc – ending location of flight

ang – direction of the flight over terrain map

globalLoc – flight frame locations on the global terrain map

localLoc – flight frame locations on the local terrain map locacTerr – local terrain map that encompasses all the flight frames

loclMineLoc – relative mine locations on the local terrain map

C.9. Function Name: simFlightPath

Description

Synthesizes and saves the extracted flight frames

Usage

out = simFlightPath (frameInfo, Param, fileName, pathname)

Inputs

frameInfo – information related to the flight frames, which is obtained from the defineFlightPath

Param – flight path synthesis parameters such as flight frame height and width, flight speed, and camera shutter speed.

filename – file name that is used to save the synthesized flight frames

pathname – name of the directory where synthesized flight frames need to be saved

Outputs

out – synthesized flight frames.

BIBLIOGRAPHY

1. L-Y. Wei, and M. Levoy, "Texture synthesis over a arbitrary manifold surfaces", Computer Graphics Proceedings, SIGGRAPHA 2001.

2. J. Jia, and C-K. Tang, "Image repairing: robust image synthesis by adaptive ND tensor voting", Proceedings of the IEEE, Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, pp I-643 - I-650, June 2003.

3. Y. Mukaigawa, S. Mihashi, and T. Shakunaga, "Photometric image-based rendering for virtual lighting image synthesis", Proceedings of IWAR, 2nd IEEE and ACM International Workshop on Augmented Reality, pp 115 -124, October 1999.

4. M. Song, R. M. Haralick, F.H. Sheehan, "Ultrasound imaging simulation and echocardiographic image synthesis", Proceedings of the IEEE, International Conference on Image Processing, Vol.3, pp 420 - 423, September 2000.

5. D. J. Heeger and J. R. Bergen, "Pyramid-Based Texture Analysis/Synthesis", In R Cook, editor, Conference Proceedings of the SIGGRAPH, Annual conference series, ACM SIGGRAPH, Addison Wesley, pp 229-238, August1995.

6. S. Zhu, Y. Wu, and B. Mumford, "Filter Random Fields and Maximum Entropy (FRAME) - Towards a Unified Theory for Texture Image Modeling", International Journal of Computer Vision, Vol. 27, No.2, pp 107-126, 1998.

7. R. Paget and I. Longstaff, "Texture Synthesis Via a Noncausal Nonparametric Multi-scale Markov Random Field", IEEE Transactions on Image Processing, Vol. 7, No.6, pp 925-931, June 1998.

8. S. Germane and D. Germane, “Stochastic Relaxation, Gibbs Distribution and the Bayesian restoration Images”, IEEE Transactions on Pattern Analysis, Machine Intelligence, Vol. 6, pp 721-741, 1984.

9. Li-Yi. Wei and M. Levoy, “Fast texture Synthesis using Tree-structured Quantization”, Computer Graphics Proceedings, pp 497-488, SIGGRAPH 2000.

10. Y. Xu, B. Guo, S-C. Zhu, and H-Y. Shum, “Asymptotically Admissible Texture Synthesis”, Technical Report MSR-TR-2001, Microsoft Research, May 2001.

11. L. Liang, C. Liu, Y. Xu, B. Guo, and H-Y. Shum, “Real-Time Texture Synthesis by Patch-Based Sampling”. ACM Transactions on Graphics, Vol. 20, No.3, pp 127-150, July 2001.

12. Y. Weijie, T. Hongming, P. Qunsheng, “An infrared image synthesis model”, Sixth Pacific Conference on Computer Graphics and Applications, pp 208-209, October 1998.

13. K. Kamei, M. Maruyama, K. Seo, “Scene synthesis by assembling striped areas of source images”, Proceedings of the IEEE, International Conference on Image Processing, Vol 2, pp 482-485, October 1997.

14. J. A. Vince, “Real-time synthetic images for flight simulation”, IEEE Colloquium on Realistic 3-D Image Synthesis , pp 4/1 -4/5, January 1990.

15. D. M. Meredith, C. P. Walters, C. W. Hoover Jr, “Suitability of Synthetic Imagery for ATR Evaluation”. Applied Imagery Pattern Recognition Workshop, pp 51-56, October 2001.

16. R. Swonger, “STRAWMAN Strategy for Creating a Substantial Future LAMD Minefield Database”, Technical memo, NVESD, Dec 2001.

17. S. Sjokvist, M. Georgson, S. Ringberg, D. Loyd, “Simulation of thermal contrast on solar radiated sand surfaces containing buried minelike objects”, 1998. Second International Conference on the Detection of Abandoned Land Mines (IEE Conf. Publication. No. 458), pp 115-119, October 1998.

18. W-J. Liao, I.K. Sendur, B. A. Baertlein and Joel T. Johnson, “Radiometric Signatures of Surface-Laid Land Mines”. Submitted to IEEE transactions on Geoscience and Remote Sensing.

19. Veredian Inc, “GenMF user manual”, 2000.

20. R. C. Gozalez, and R. E. Woods, “Digital Image Processing”, Pearson Education, Inc., 1992.

21. J. Makhoul, S. Roucos, H. Gish, “Vector Quantization in Speech Coding”, Proceedings of the IEEE, Vol. 73, No. 11, pp 1551-1588, November 1985

22. Y. Linde, A. Buzo, R.M. Gary, “An Algorithm for Vector Quantizer Design”, IEEE Transactions on Communications, Vol. 29, pp. 84-95, 1980

23. T. K. Edara, S. Agarwal, “MF Simulator: A Graphical User Interface for Minefield Synthesis – User’s Manual”, University of Missouri Rolla, 2004.

VITA

Thandava Krishna Edara was born on August 21, 1977, in Ongole, India. In May 2000, he received his B.Tech in Electronics and Communication Engineering from Jawaharlal Nehru Technological University, Hyderabad, India. He joined the Master's program in Electrical Engineering at the University of Missouri-Rolla in spring 2002. While pursuing his graduate degree, he was supported by the department of Electrical Engineering as a Graduate Research Assistant. His research interests include image processing, communications and signal processing. He received his Masters of Science Degree in Electrical Engineering from the University of Missouri-Rolla in May 2004.

Distribution List

8 Director
Night Vision And Electronic Sensors Directorate
10221 Burbeck Road
Attn: AMSRD-CER-NV-CMD
Fort Belvoir, VA 22060-5806

2 Defense Technical Information Center
8725 John J. Kingman Highway
Suite 0944
Fort Belvoir, VA 22060-6218