THESIS

# GENERATION OF TERRAIN TEXTURES USING NEURAL NETWORKS

Submitted by

Santiago Alvarez

Department of Computer Science

In partial fulfillment of the requirements

for the degree of Master of Science

Colorado State University

Fort Collins, Colorado

Fall, 1995

COLORADO STATE UNIVERSITY

September 30, 1995

WE HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER OUR SUPERVISION BY SANTIAGO ALVAREZ ENTITLED GENERATION OF TERRAIN TEXTURES USING NEURAL NETWORKS BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE.

Committee on Graduate Work

Adviser

Department Head

ABSTRACT OF THESIS

GENERATION OF TERRAIN TEXTURES USING NEURAL NETWORKS

Realistic visualization of terrain can be achieved by combining color and topographic information. Three-dimensional terrain models obtained from Digital Elevation Models (DEMs) can be rendered by mapping aerial photographs on top of them. However, there may be terrain models for which a texture (photograph) is not available. In those cases, it is useful to have some technique which generates artificial textures using terrain information of similar regions.

This project explores the use of neural networks, Multi-Layer Perceptrons (MLPs) in particular, to generate artificial terrain textures from DEMs. This type of neural network has been referred as a universal function approximator. In this case, the network approximates the mapping between elevation samples and pixel colors.

Three main issues were address in this project: first, the quality of the textures produce by the MLP; second, the effect of different input representations on the texture quality, in particular, the effect of including slope information and a neighborhood of elevation samples in the input vector; and third, the accuracy of the trained MLP in rendering unseen terrain with similar characteristics as those used during training.

Santiago Alvarez
Department of Computer Science
Colorado State University
Fort Collins, Colorado 80523
Fall, 1995

ACKNOWLEDGEMENTS

I wish to thank my advisor, Dr. Mike Goss, for his guidance and teachings throughout the completion of this degree. I am also grateful to Dr. Chuck Anderson who served in my committee and contributed to this work with numerous discussions and reviews. Thanks also go to Dr. Denis Dean for serving in my committee, providing me with useful information, and reading and commenting on the initial manuscript.

I wish to thank Dr. Mario Mejía-Navarro for all the material he shared with me and the time he devoted to help me. I am grateful to Alicia C. Lizarraga for providing me with important references. I have also benefited from other friends and colleagues who contributed to this work in many different ways.

I express here my gratitude to my parents and sister whose encouragement has always been essential to my work. *Sin su ayuda no hubiera alcanzado esta y muchas otras metas.*

DEDICATION

*A todos los que en veintiseis años han hecho de mi lo que soy. Inclusive a aquellos que fueron un obstaculo.*

CONTENTS

LIST OF FIGURES

LIST OF TABLES

# Chapter 1

# INTRODUCTION

Terrain textures are essential for rendering high-quality terrain images. Realistic visualization can be obtained by combining color and topographic information (Cohen and Gotsman, 1994; Miller, 1986; Taylor and Barrett, 1994). Terrain color can be obtained from aerial or satellite images. These photographs usually cover a small portion of terrain from a vertical view angle. Topographic information can be obtained from Digital Elevation Models (DEMs). A DEM contains a number of elevation samples that can be extracted from stereoscopic terrain photographs.

The rendering of terrain models using computer graphics has multiple applications. It is used by the military for flight simulation, simulation of the displays of some electro-optical weapons, and mission planning. This technology is also used in the civilian market for urban and rural planning, the generation of animations, and many other applications.

The most common method to render realistic images of actual terrain is to obtain a 3D polygonal model of the terrain from a DEM (Fowler, 1979; McCullagh, 1982; Tarvydas, 1984; Scarlatos, 1990). This polygonalization reduces the complexity of the terrain representation allowing faster access and processing. Once the terrain model has been created, the terrain texture is mapped onto it, and projected according to the desired view. The final realism of the rendered image depends on the resolution and quality of the 3D model generated, and the realism of the terrain texture mapped onto the model.

However, terrain textures are not always available. In those cases where a terrain texture cannot be obtained, it would be useful to have some technique that would generate terrain textures with an acceptable quality using elevation and land cover information. In such cases, an artificially generated model can be rendered. Such a model could be the result of a simulation process that tries to predict some terrain parameters, or could also be an arbitrary model built for visualization purposes.

This project explored the use of neural networks to generate artificial terrain textures from terrain models. A simplified model structure containing only elevation data was used. The goal was to be able to generate terrain textures from elevation data using a neural network trained with available terrain models and photographs. Specifically, it was of most interest to determine the quality of the terrain textures that could be obtained from elevation data, and how the input representation could affect the results. Additionally, the generality of this approach was addressed.

The simplified terrain model was expected to produce enough information to address the texture generation problem. The appearance of terrain at a given point is highly influenced by its altitude and slope. This relationship should be similar within a limited geographical area. If such a close relationship exists, a neural network can be used to find an interpolation of the mapping between elevation and color.

Even though elevation is not the only factor that determines the terrain texture, it should play an important role. Other factors such as light exposure, type of soil, and weather have a major impact on the kind of vegetation and rocks found on some terrain. However, their influence on the appearance of the terrain was assumed constant or closely related to elevation or slope for this project.

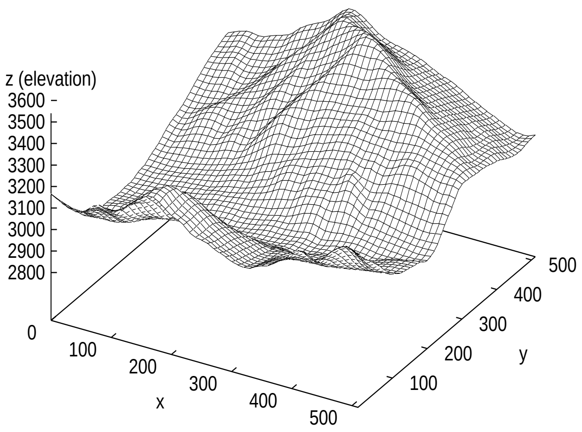
We can overcome the lack of a general model that relates elevation with terrain texture by using neural networks. One feature of neural networks is that they can be trained on a relatively small number of examples of a relationship. Once trained, the network can induce a complete relationship that interpolates in a sensible way. The use of this characteristic should give some insight about how close a relationship there is between elevation and texture, and how realistic the results produced by a trained network are.

### 1.1 Data Description

Two basic types of data were considered in this project: Digital Elevations Models (DEMs) and aerial photographs. As defined by Burrough (1992), "any digital representation of relief over space is known as a DEM." Their uses include:

- Elaboration of topographic maps by government agencies,
- Support of construction projects (especially for volume estimations),
- Display of image simulation models for military purposes and landscape planning,
- Analysis of statistical features, and comparison of different kinds of terrain,
- Elaboration of Geographic Information Systems (GIS).

There are two basic methods to represent DEMs: mathematical methods and image methods (Mark, 1978). The image methods are divided into line models, specified usually as contour lines, and point models which are commonly given as altitude matrices. These matrices or regular grids are the most common form of DEMs, and are particularly popular because of the ease with which they can be handled using a computer (Figure 1.1a). This type of DEM was used in this project to obtain elevation information.




(a)

(b)

Figure 1.1: (a) Block diagram of a DEM, (b) Aerial photograph of the same terrain (taken from one of the data sets used)

Although altitude matrices are the most popular format for DEMs, they have some important disadvantages. First, the terrain elevation is sampled at the same rate in areas of high detail and areas of low detail. The resolution may be too coarse in areas with critical features, while uniform areas may contain redundant information. In addition, the orientation of the axis makes computations along the grid lines simple, while computations at other angles require trigonometric calculations to compute distances and angles.

Altitude matrices are usually obtained from stereoscopic aerial photographs by using a matching process to obtain depth information (Groß, 1994). First, a correspondence between pixels in the stereoscopic photographs is determined. Once the image coordinates are known for a given pixel, the world coordinates of that point can be found by triangulation using the camera geometry. This process is sensitive to distortions and errors during the pixel correspondence phase. Thus, subpixel interpolations or multiple views can be used to provide higher accuracy.

The U.S. Geological Survey (USGS) possesses DEMs covering all of the contiguous United States, Hawaii, and limited portions of Alaska (US Geological Survey, 1990a.) Three distinct digital elevation products are distributed for the contiguous United States:

- 1-Degree DEM. A 1- by 1-degree block with a 3- by 3-arc-second data spacing,
- 7.5-Minute DEM. A 7.5- by 7.5-minute block with a 30- by 30-meter data spacing,
- 30-Minute DEM. Four 15- by 15-Minute blocks with a 2- by 2-arc-second data spacing.

The other type of data used in this project was obtained from terrain images that provide color (texture) information (Figure 1.1b). These images are acquired from aerial or satellite photographs which provide a vertical view of the terrain. When an orthographic projection of the terrain scene cannot be assumed, a geometric recitification of the image is required (Jensen, 1986). This process rectifies the original image to a map coordinate system. Initially, a spatial interpolation is performed by identifying ground control points (GCPs) in the original image and on a reference map. Finally, an intensity (color) interpolation is performed since there is not a one-to-one mapping between the input pixel and output pixel locations.

## 1.2 Neural Network Model

The Multi-Layer Perceptron (MLP) was the neural network model considered for this project. This network is an extension of the Simple Perceptron which consists of a single layer of processing units. An MLP overcomes the limitations of the single layer perceptron, which can only be used to classify inputs that are linearly separable. That is, the training algorithm for a single perceptron converges to a solution where the connection weights specify a hyperplane that divides the input space into two parts.

An MLP is a feed-forward neural network with one or more layers of processing units (nodes) between the inputs and the output layer (Figure 1.2). The units that directly produce the network output are called output units, whereas the other units are called hidden units. Each unit performs a simple linear combination of its inputs, and then applies a function to obtain an activation value for the unit. The linear combination of the inputs is computed using the connection weights associated with each input.

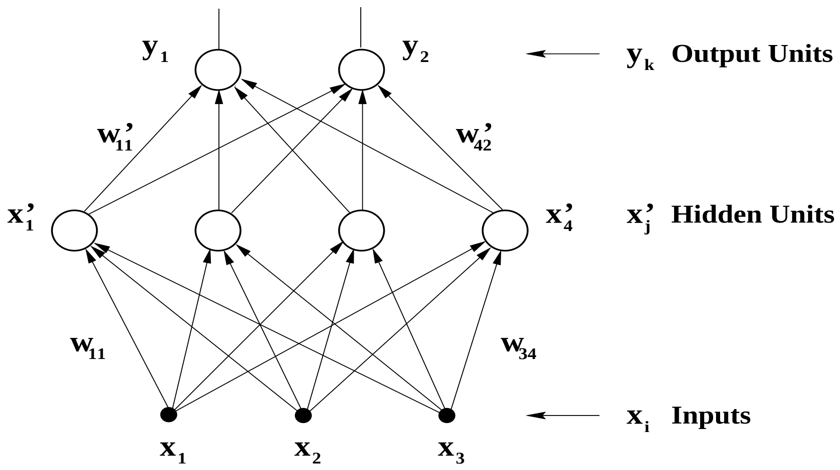



Figure 1.2: A two-layer perceptron showing the notation for units and weights

The MLP, unlike the simple perceptron, uses a continuous activation function. This allows the network to be used not only for classification tasks, but also for function approximation problems in general. This project exploited this feature to find a mapping between elevation and color. MLPs have been referred to as universal approximators (Hornik, 1991; Chen, and Jain, 1994.)

This type of neural network was not used in the past because of the lack of an effective training algorithm to determine adequate connection weights. The invention of the back-propagation learning algorithm showed how to make an MLP learn a particular function. It was invented independently by Bryson and Ho (1969), Werbos (1974), Parker (1985) and Rumelhart, Hinton, and Williams (1986a, b).

The back-propagation algorithm is based on a gradient descent technique that minimizes the output error. Using the notation given in Figure 1.2, the error function to be optimized is given by

$$E(w) = \frac{1}{2}\sum_{\mu}\sum_{l}(d_l^{\mu} - y_l^{\mu})^2 \tag{1.1}$$

where $d_l$ is the desired (target) output value for unit $l$, $y_l$ is the actual unit output, and $\mu$ is an index into the set of training vectors. Since a target or desired output is required for every input during training, back-propagation is classified as a supervised learning algorithm.

Given the cost (error) function (1.1), the gradient descent rule yields the following equation for the input-to-hidden connections

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}}. \tag{1.2}$$

Similarly, we obtain for the hidden-to-output connections

$$\Delta w'_{jk} = -\eta \frac{\partial E}{\partial w'_{jk}}. \tag{1.3}$$

where $\eta$ is called the learning rate, which is a scale factor that indicates how far to move in the direction of the gradient. The learning rate may be different for each layer. A momentum term can also be included to improve the results obtained with equations 1.2 and 1.3 (Hertz, Krogh and Palmer, 1991). This term helps attenuate the oscillation behavior that can take place during learning.

The activation function has to be differentiable in order to be able to apply this gradient descent technique. It is common to use a sigmoid function for the activation function. The functions generally used are differentiable and generate bounded outputs. Figure 1.3 shows two common activation functions.

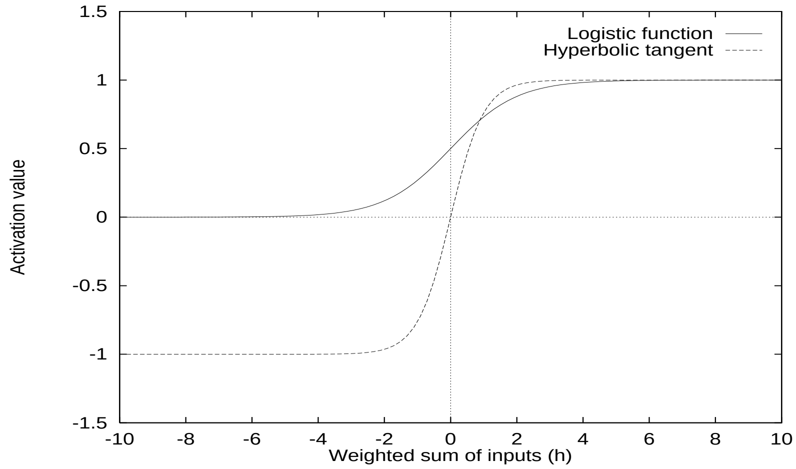



Figure 1.3: Two common activation functions used for MLPs: the logistic function $g(h) = \frac{1}{1+e^{-h}}$ and the hyperbolic tangent $g(h) = tanh(h) = \frac{e^h - e^{-h}}{e^h + e^{-h}}$

The iterative application of the weight update rules needs to be stopped at an appropriate stage. If that process is performed for too long, the network will produce excellent results for the inputs used for training, but poor results for new inputs unless the network is simple enough to not overfit. At that point, the network has lost its generalization feature. This ability to generalize is highly desirable since the training data is usually a relative small group of samples of the input space.

A cross-validation mechanism can be used to avoid losing the generalization feature of the network during training. The input-target pairs available for training are divided into three sets: a training set, a validation test, and a test set. The pairs in the training set are actually used for training. The inputs and targets in the validation set are used at every epoch to measure the error that the network makes with data that is not being used for training. The Root Mean Square Error (RMSE) between each single output and target is used as an error measure. The desired set of connection weights is found when the lowest error of the validation set pairs is detected. At that point, the test set is used to compute an estimation of the network error on a totally new data set that has not been used either for training or validation.

There are several features that make MLPs an appealing technique. They can operate with noisy data. Each unit relies only on local information to compute its activation value. The large number of connections provides a high degree of redundancy that makes the network fault-tolerant. Only a subset of the possible inputs is needed to train the network in many cases. Once it is trained, the network interpolates for unseen input vectors.

However, there are some limitations and disadvantages that need to be considered. The input and output data may need to be preprocessed (normalized, scaled, translated, etc) in order to exploit the information it can provide and obtain the desired results. The training phase is an iterative process that can be computationally expensive. That is, the training data may have to be processed by the network a number of times before a satisfactory solution is reached. Moreover, the learning algorithm is not guaranteed to converge to a global optimal solution. Although this is not a critical problem, several training sessions may be needed to find a good set of weights for a given architecture. Besides, there is not a clear method to know a priori the right architecture and parameters for a specific problem.

Some successful applications of MLPs that have been published include: speech recognition (Lippman, 1989), hand-written ZIP code recognition (Le Cun et al., 1989), car navigation (Pomerleau, 1989), sonar target recognition (Gorman and Sejnowski, 1988a, b), phoneme generation from text (Sejnowski and Rosenberg, 1987), and signal prediction and forecasting (Lapedes and Farber, 1988; Farmer and Sidorowich, 1987, 1988).

# Chapter 2

## DATA PREPARATION

This project used four DEM-photograph pairs: *cam1*, *cam2*, *cam3*, and *cam4*. They contain information for a small area in the Colorado State Forest next to the northwest border of the Rocky Mountain National Park in Northern Colorado. Some of the natural features covered include: Michigan River, Diamond Peaks, Seven Utes Mountain, Mount Mahler, Lake Agnes, Nokhu Crags, Snow Lake, Static Peak, Mount Richthofen, and Tepee Mountain. Table 2.1 shows the exact location of the data sets in Universal Transverse Mercator (UTM) coordinates.

The relative location of the four data sets is presented in Figure 2.1. Hidrographic information and the contour lines of the area are also displayed. The elevation data used was obtained from 7.5-minute Digital Elevation Models created by the U.S. Geological Survey. Most of the samples are inside of the DEM *Mt. Richthofen* (No. 4010548). The four data sets partially overlap. In particular, Lake Agnes, the lake with a small island in the middle, is included in all data sets. Some portions of the adjacent DEMs *Fall River Pass* (No. 4010547), *Chambers Lake* (No. 4010557), and *Clark Peak* (No. 4010558) are also covered. These four 7.5-minute DEMs can be obtained from the U.S. Geological Survey (`http://www.usgs.gov/`).

Each DEM-photograph pair was previously registered so both the DEM and the aerial photograph had the same location, area, and sampling rate. That is, each sample on the altitude matrix corresponds to the pixel in the same location on the aerial photograph, and vice versa. The original 7.5-minute DEMs had a data spacing of 30 by 30 meters. After the registration process, both the DEMs and the aerial photographs had a 5- by 5-meter data spacing. This resampling process was performed using a linear interpolation of the original elevation samples in the 7.5-minute DEMs.

Table 2.1: DEM-Photograph Location in Universal Transverse Mercator (UTM) coordinates

| **UTM Coordinates** | *cam1* | *cam2* | *cam3* | *cam4* |
|---|---|---|---|---|
| North | 4,483,020 | 4,482,210 | 4,485,420 | 4,485,390 |
| South | 4,478,760 | 4,478,550 | 4,480,950 | 4,480,980 |
| East | 423,930 | 426,570 | 426,720 | 423,810 |
| West | 419,340 | 422,640 | 422,070 | 419,070 |

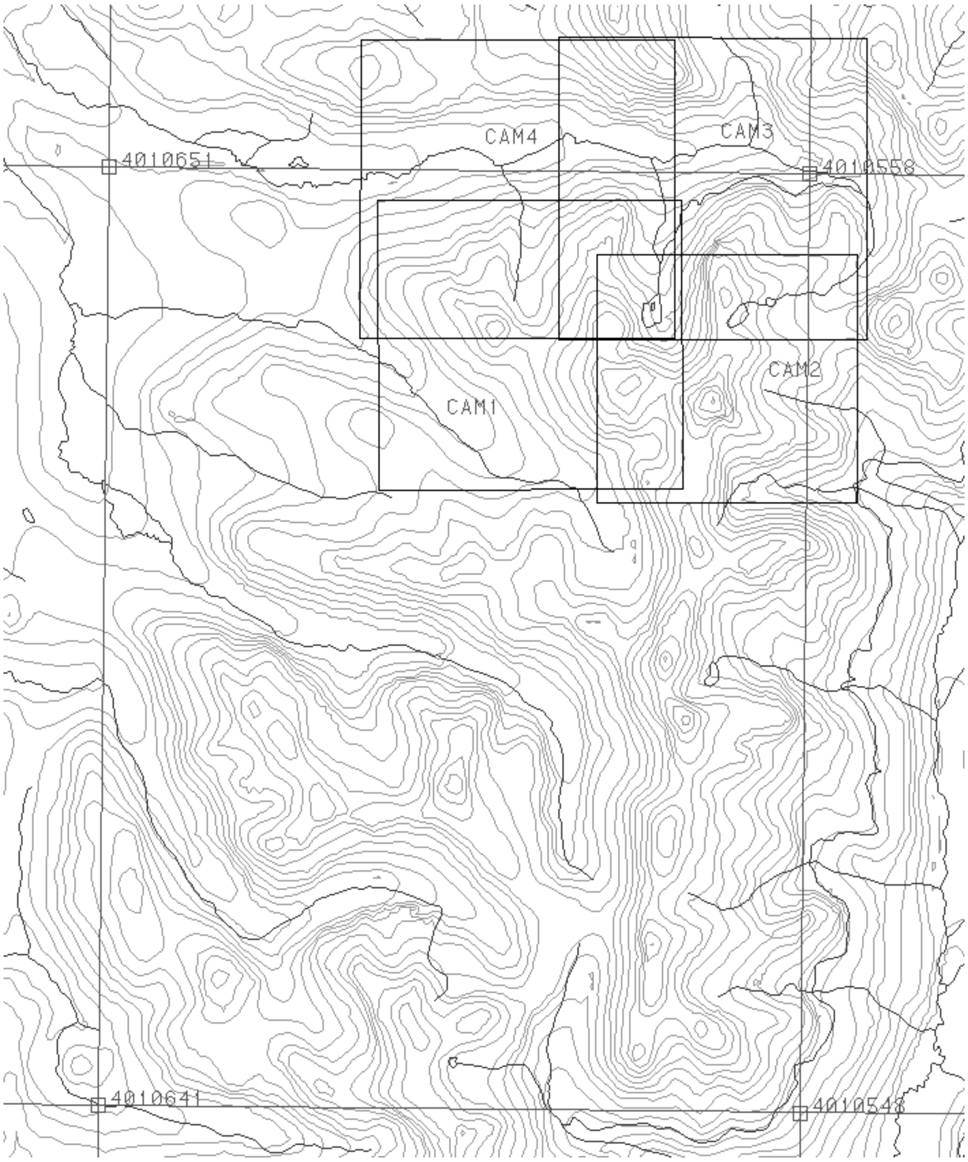
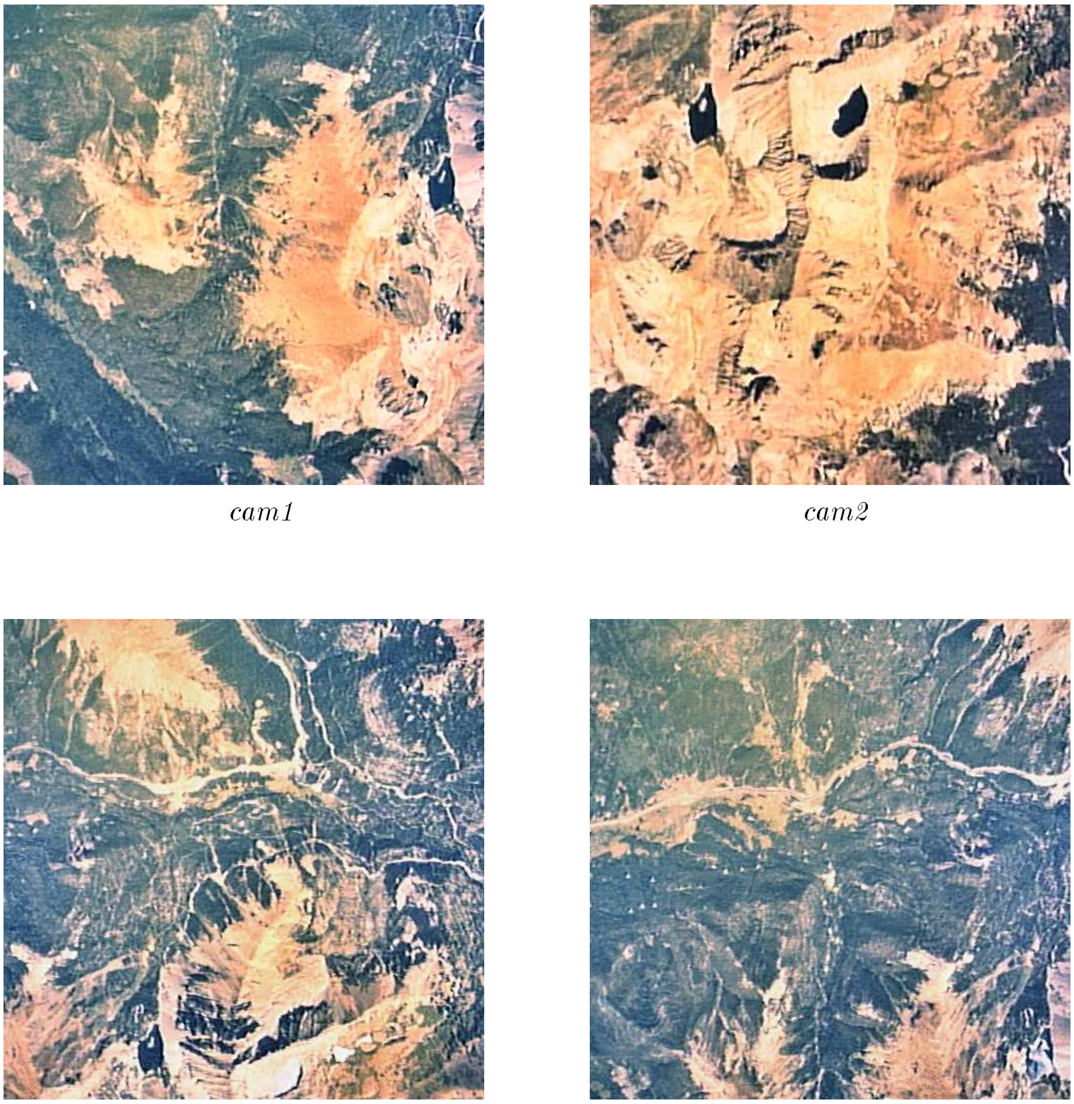



Figure 2.1: Relative location of *cam1*, *cam2*, *cam3*, and *cam4*. The DEM *Mount Richthofen* (No. 4010548) is outlined as a reference.

Figure 2.2 shows the four aerial photographs. *Cam2* and *cam4* are rather dominated by a single type of terrain texture. *Cam2* includes the highest elevations and, therefore, covers terrain with scarce vegetation. *Cam4*, on the other hand, contains the lowest elevations which have little rock texture. *Cam1* and *cam3* show a greater degree of variation in color. Lake Agnes can be easily identified in *cam1*, *cam2* and *cam3*, but is ocluded in *cam4* (lower right corner.) *Cam1* was selected as the primary data set for experimentation. The other three were left to analyze the ability of the network to generalize using new terrain.



*cam1* *cam2*

*cam3* *cam4*

Figure 2.2: Registered aerial photographs.

Figure 2.3 shows the *cam1*, *cam2*, *cam3*, and *cam4* DEMs. They are shown as block diagrams by creating a square grid with adjacent samples. The resolution of the diagrams is thirty times lower than the original DEMs. All four data sets have elevation samples within similar ranges. *Cam3* and *cam4* cover adjacent fragments of the valley north of Lake Agnes and Snow Lake. *Cam1* covers the wavy area to the left of Mount Richthofen. Cam2 covers the area with the steepest slopes including Mount Mahler, Mount Richthofen, and Static Peak.

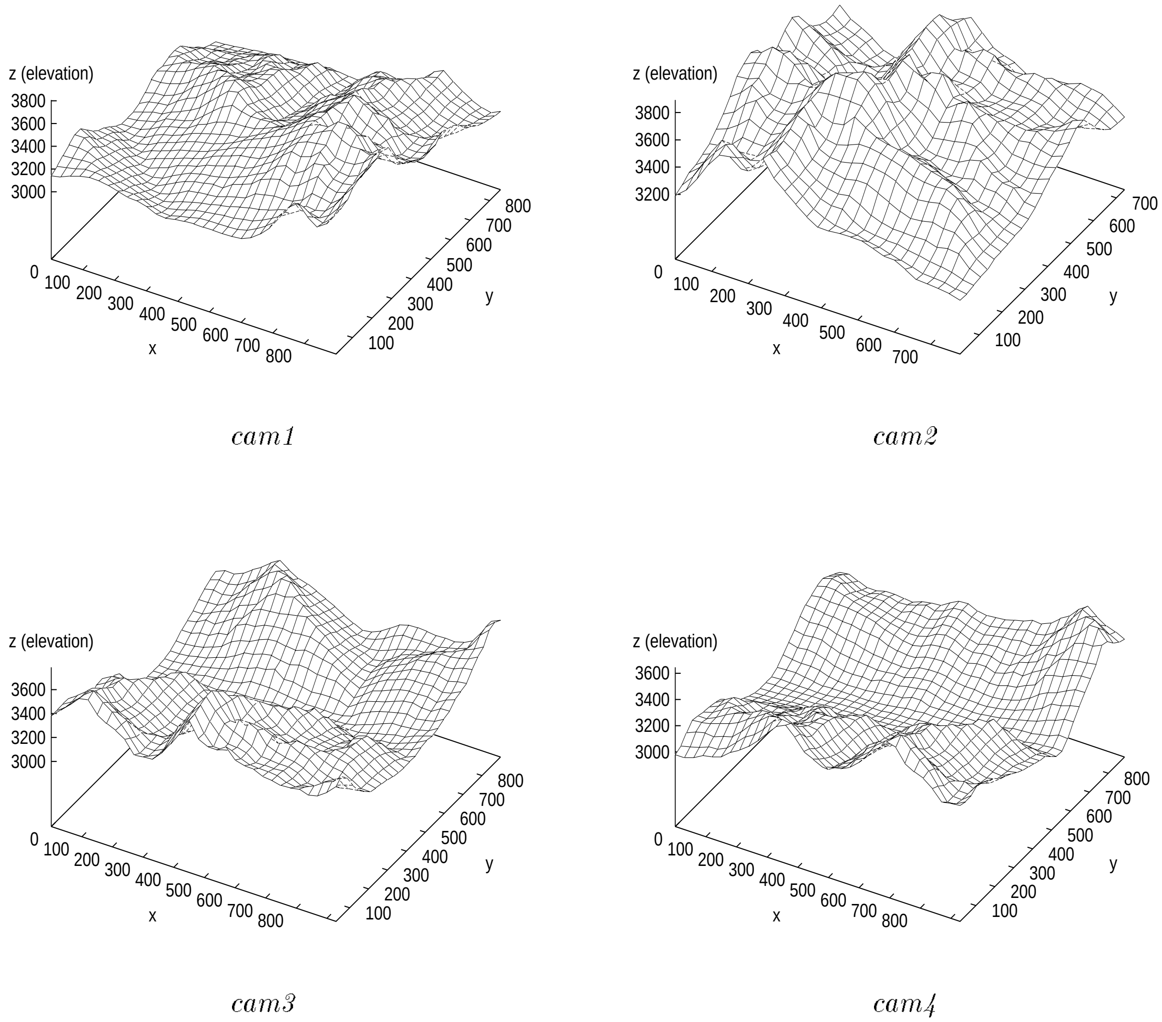



Figure 2.3: Block diagrams of the registered digital elevation models.

## 2.1 Normalization

Initial tests using MLPs to find the elevation-to-color mapping were performed with small pieces of the *cam1* terrain. The input vector was built using a nine-element neighborhood for each elevation sample. The output vector contained three color components at each given pixel. Despite the increase in the dimensionality of the input vector, using a neighborhood was expected to produce better results. The single elevation value was believed to provide insufficient information for a satisfactory estimation of the terrain color at that point. Two points at the same altitude can have extremly different colors depending on vegetation, the presence of water, and terrain features like slope. The use of a neighborhood was expected to help cope with this problem.

Due to the homogeneity present in the small fragments of terrain, the network mapped all inputs to the same outputs. The final result was an image with the same gray value in all its pixels. This problem was expected to be alleviated in part using larger pieces of terrain with more significant differences in elevation. However, a more general solution was selected by transforming the data so that the effect of the width of the input and output ranges could be further reduced.

The four data sets were linearly transformed using the elevation and color means and standard deviations (Equation 2.1). This normalization converts the original data to a distribution with zero mean and unit standard deviation where $x_i$ is an elevation or color sample, $\overline{X}$ is the sample mean and $\sigma_X$ is the sample standard deviation. This transformation enhances the relative differences between samples while preserving the original distribution by translating and scaling each value $x_i$ to a new sample $x_i'$.

$$x_i' = \frac{x_i - \overline{X}}{\sigma_X} \tag{2.1}$$

However, the parameters used for normalization must be obtained from the distribution associated with the terrain to which the neural network is expected to generalize. Using a general distribution guarantees that all inputs that would eventually be fed to the network would be scaled consistently. In order to approximate a general distribution for the type of terrain considered, the four data sets (*cam1*, *cam2*, *cam3*, and *cam4*) were blended to obtain a combined set of elevations and a combined set of pixel colors. These two sets were used to obtain the normalization parameters.

Figure 2.4 shows distributions of the combined data sets. Figure 2.4a shows the elevation histogram of the samples in the four original DEMs. Figures 2.4b, c, and d show the combined color distributions using the Red-Green-Blue (RGB) color model. Table 2.2 shows the parameters of the elevation and color distributions including the mean and standard deviation used for normalization. The parameters for the color distributions in the Hue-Saturation-Value (HSV) color model are also included.

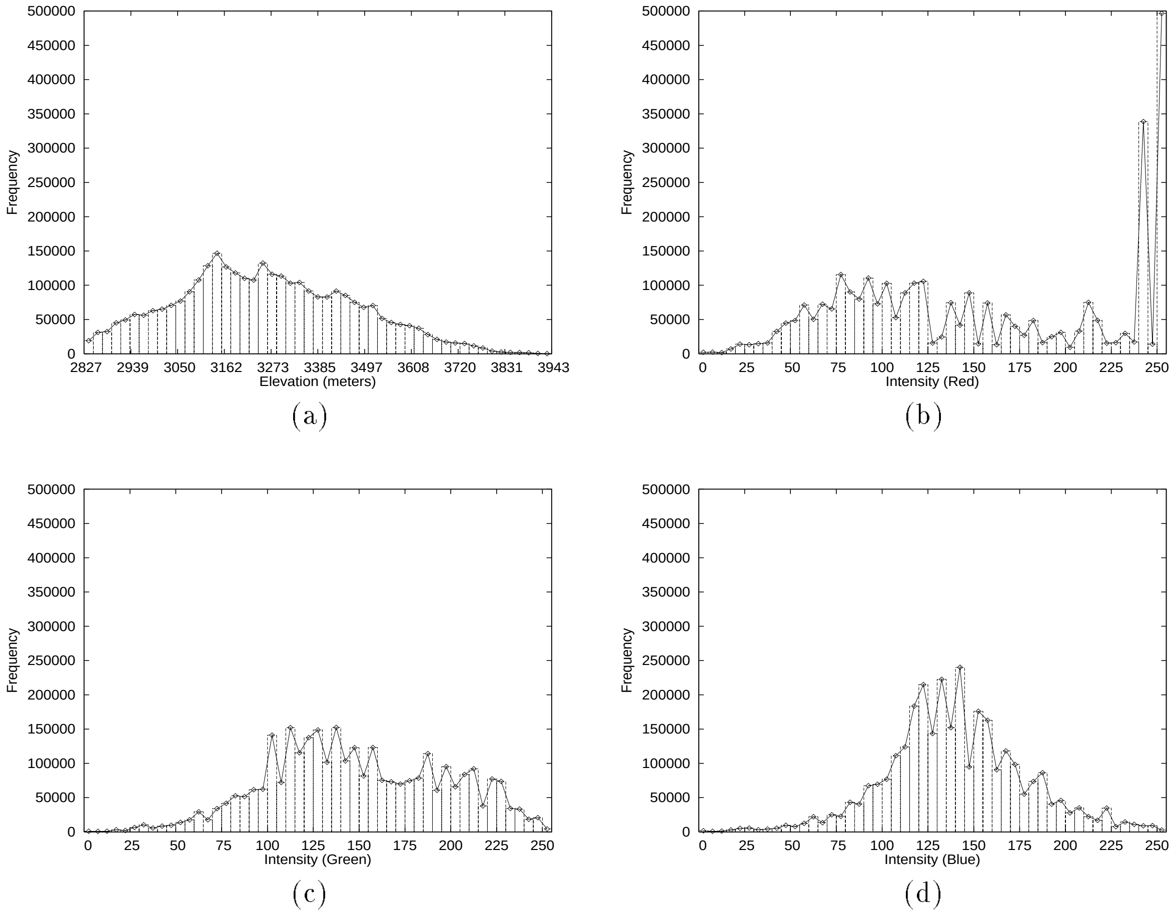



Figure 2.4: Histograms of the combined DEMs and the combined photographs using the RGB color model. (a) Elevation histogram. (b) Red component histogram. (c) Green component histogram. (d) Blue component histogram

The HSV color model was additionally considered to train the network. The original color information was in RGB format. Therefore, the color component had to be transformed to generate the new outputs before training. The shadows present in the

Table 2.2: Distribution parameters of the combined DEMs, and the combined photographs using the RGB and the HSV color models

| **Parameter** | **Elevation** | **Red** | **Green** | **Blue** | **Hue** | **Saturation** | **Value** |
|---|---|---|---|---|---|---|---|
| Lower Bound | - | 0 | 0 | 0 | 0 | 0.000 | 0.000 |
| Upper Bound | - | 255 | 255 | 255 | 360 | 1.000 | 1.000 |
| Minimum | 2827.38 | 0 | 0 | 0 | 0.00 | 0.000 | 0.000 |
| Maximum | 3942.57 | 255 | 255 | 255 | 359.10 | 1.000 | 1.000 |
| Mean | 3260.02 | 156.07 | 149.15 | 140.48 | 121.04 | 0.284 | 0.679 |
| Standard Dev. | 209.08 | 73.29 | 47.69 | 37.42 | 92.75 | 0.147 | 0.230 |

photographs used for training affect the network performance. Using the HSV model, a shadow should affect mainly the value component of the pixel. Using the RGB model, on the other hand, all three color components are affected by a shadow. The same normalization procedure was performed on the transformed color components. Figure 2.5 shows the distributions of the combined data sets using the HSV color model including the distribution of the combined elevation samples again.

The normalized inputs were directly fed to the network without any further transformation. However, the normalized output values needed to be mapped to the proper output range of the network. The exact parameters of this transformation depended on the activation function used in the output units used and the minimum and maximum values of the color components.

Only a portion of the total activation function range was used when transforming the color components. The two asymptotic ends of the sigmoid functions (see Section 1.2) are hardly reached by a network. A extremely high or low weighted sum is required to produce an output close enough to the respective asymptote. The 90% center of the range was actually used when mapping the outputs in order to accelerate the network training and improve performance.

## 2.2 Training, Validation, and Test Sets

*Cam1* was used as the primary data set for experimentation. It contains approximately 1,600,000 elevation and color samples. A training set, a validation set, and a test

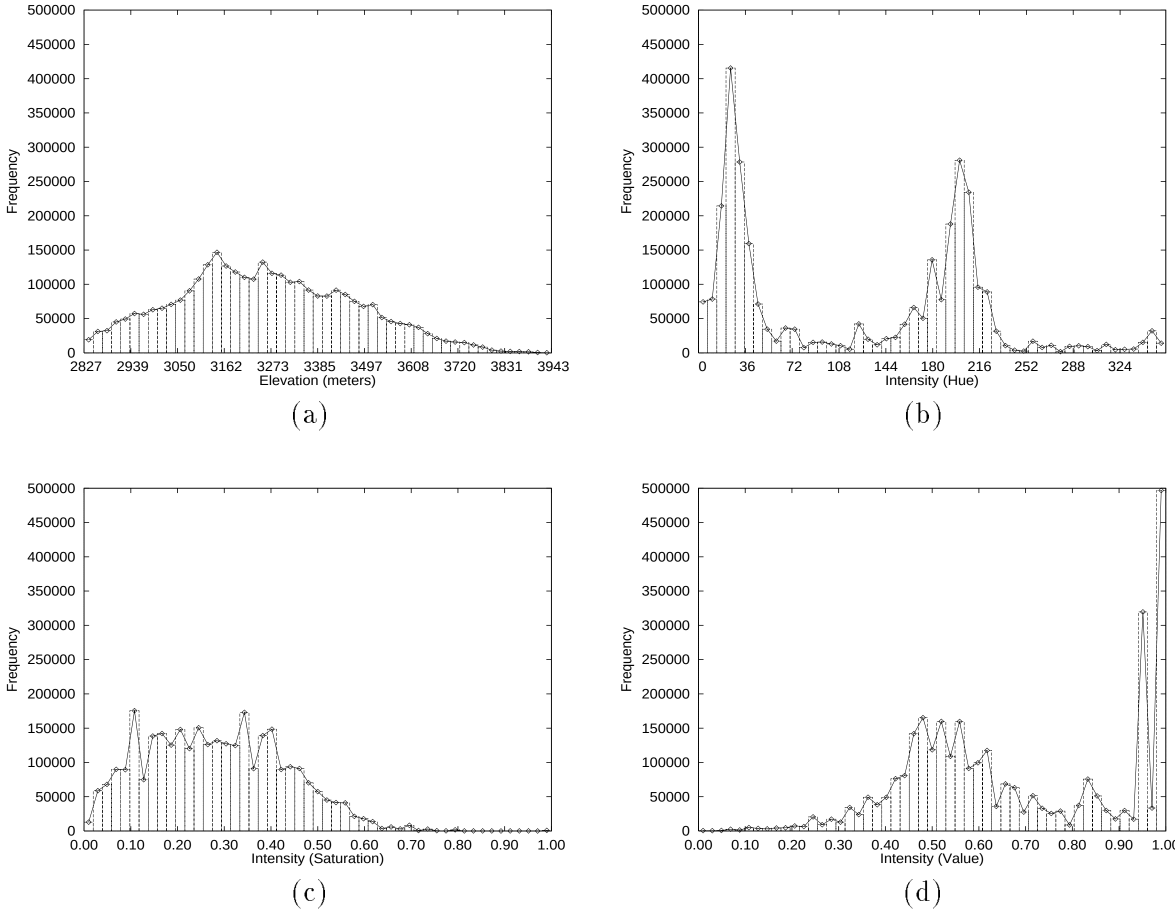



Figure 2.5: Histograms of the combined DEMs and the combined photographs using the HSV color model. (a) Elevation histogram. (b) Hue component histogram. (c) Saturation component histogram. (d) Value component histogram

set were created by sampling the data set at random. These three sets are used in the cross-validation mechanism described in chapter 1.

The elevation-color pairs were divided in different proportions to create the three sets. The validation set was created with 5% of the sample pairs. Different sizes were used for the training set. Preliminary tests showed that the validation error decreased asymptotically with the percentage of pairs used for training. That is, there is little difference in the results for high percentages of samples. Therefore, a reduced number of elevation/color pairs could be used to train the network with a satisfactory performance. Based on these observations, the percentage of samples taken to build the training set was subject to experimentation. The actual values used are described in the next chapter.

Finally, the test set was constructed with all the pairs that were not used for the training set or the validation set.

# Chapter 3

# EXPERIMENT DESCRIPTION

## 3.1 Parameter Search and Percentage of Data Sampled

One of the limitations of most neural networks is the lack of a clear theoretical base to select adequate values for their parameters. Useful values are generally expected within certain ranges, but specific values in those ranges can generate networks with important differences in performance. Moreover, adequate values are generally problem dependent, and previous experiences are not always useful. Therefore, some exploration of the parameter space is required to ensure that an acceptable performance is achieved.

However, the number of scenarios defined by the combination of several parameter values easily becomes too expensive. Therefore, the number of parameters to be explored, and the number of values that each of them can take, need to be reduced. This reduction of the parameter space does not guarantee an optimal configuration, but certainly makes more manageable the selection of parameter values.

An initial approach to the selection of appropriate parameter values was performed on a 128-node CNAPS Server II from Adaptive Solutions, Inc. Even though the training time is greatly reduced by this machine, the lack of floating-point arithmetic affects the results. This limitation should not be important for classification applications, but it is prejudicial for function approximation purposes. This constraint led to the use of a single-processor workstation, reducing the number of scenarios to be studied due to the lower computing capacity.

Four basic network parameters were fixed: number of layers, momentum term, activation function, and ratio between the learning rate in the hidden layer and the learning rate in the output layer. A two-layer network was used since it is the minimum number of layers required to overcome the limitations of a single perceptron. The parameter values selected for experimentation were chosen based on the preliminary results obtained with the CNAPS Server. The momentum term was set to zero, and the ratio between the hidden layer learning rate and the output layer learning rate was set to four. The traditional logistic function (3.1) was used as the activation function for both layers:

$$g(h) = \frac{1}{1+e^{-h}} \tag{3.1}$$

The number of units in the hidden layer and the learning rate were the two parameters explored. Three different numbers of hidden units were considered: 25, 35, and 45. Two learning rates were used: 0.01 and 0.1. These values define six different scenarios that were explored.

Each experiment defined by the combination of hidden units and learning rate was replicated five times. Experiment replications allow computation of a better estimation of the error measures obtained during training. Replications also give some idea of the variability of the results. This replicated parameter search was performed with five different training sets. All of them were taken from *cam1*, but contain different amounts of training information as explained in the next section.

Preliminary experiments showed the necessity for reducing the number of elevation-color pairs used for training. The execution time dramatically increased in proportion to the number of pairs. However, the test error did not decrease proportionally. Therefore, it was desirable to use as few pairs for training as possible, but using enough samples to find a proper elevation-to-color mapping.

Five different percentages of sample pairs from *cam1* were used to perform the parameter search: 1%, 5%, 10%, 25%, and 40%. That is, when 1% of the pairs were used for training, each sample was selected with probability 0.01. The eight closest neighbors of each selected sample were used to build an input vector with nine components. This input vector was expected to produce better results since it provides more global information that just the single sample.

### 3.2 Including the Surface Normal Vector

A DEM describes a piece of terrain as a two-dimensional elevation function which can also be visualized as a three-dimensional surface. The normal vector is one of the attributes of a surface. This vector gives information about the surface orientation at a given point. The effect of adding the normal vector to the input vectors was studied. This inclusion increased the length of the input vector by four components (three normal components and the magnitude value of the vector.)

The inclusion of the normal vector in the input vector may look redundant. As mentioned before, the input vector was initially built using a 9-element neighborhood (the elevation sample and its eight adjacent neighbors.) Therefore, the normal could be estimated from these nine inputs by the network if it is important to estimate the outputs. However, the use of additional information should yield better results since it implies some amount of preprocessing.

The normal vector was estimated using the elevation gradient vector, $\vec{G} = \langle g_x, g_y \rangle$. The normal vector, $\vec{n}$, is computed by taking the cross product of the two gradient components (equation 3.2).

$$\vec{n} = \vec{g_x} \times \vec{g_y} \tag{3.2}$$

The gradient components were computed using an adjustable gradient filter proposed by Goss (1994). This filter represents an improvement over the fixed-response filters commonly used to approximate the gradient vector. The filter is generated by truncating (limiting in space) an ideal gradient filter and multiplying it by a window function (*Kaiser window*). The width of the window function can be adjusted using a parameter $\alpha$ which controls how quickly it tapers off to zero at the edges. Different $\alpha$ values produce different levels of smoothing in the gradient.

For a filter with an even number of non-zero elements, $M$ ($M/2$ element to the right and left of the center element zero), the convolution with the elevation data ($x_{i,j}$) that estimates the gradient vector $\langle g_x, g_y \rangle$ is given by:

$$g_x = \sum_{m=-M/2}^{M/2} \left( \frac{I_0(\beta)}{I_0(\alpha)} \frac{(-1)^m}{-m} \right) x_{i+m,j} \tag{3.3}$$

$$g_y = \sum_{m=-M/2}^{M/2} \left( \frac{I_0(\beta)}{I_0(\alpha)} \frac{(-1)^m}{-m} \right) x_{i,j+m} \tag{3.4}$$

where $I_0(\beta)$ is the order zero modified Bessel function of $\beta$, and $\beta$ is calculated using $\alpha$, $m$, and $M$, as follows:

$$\beta = \alpha \sqrt{1 - \left( \frac{2m}{M-1} \right)^2} \tag{3.5}$$

A filter with 7 elements and an $\alpha$ value of four was used to compute the gradient information (filter values $[0.1086, -0.3167, 0.8964, 0.0000, -0.8964, 0.3167, -0.1086]$). This configuration preserves the high frequency detail in the elevation grid much better than the traditional center-difference method used for gradient estimation (Goss, 1994).

## 3.3 Using the HSV Color Model

Shadows are one of the problems that may affect the generation of terrain textures. Aerial photographs are taken under special atmospheric conditions, and during specific periods of time. However, shadows have to be tolerated to some extent. A small amount of clouds could still be present or the sun may not be right on its zenith. The transformation of the output (color) values to the HSV color model is expected to reduce the effects that shadows can have on the neural network. Shadows act as a very strong source of noise since points at similar elevations can have important color differences due to shadows.

The HSV (Hue-Saturation-Value) is a color model that is based on the intuitive appeal of the artist's model of tint, shade, and tone. The value of $H$ ranges from zero to 360. The values of $S$ and $V$ go from zero to one. A hue value with saturation and value one represents the pure pigment used as starting point in mixing colors. Decreasing $S$ corresponds to adding white to the original pigment. Decreasing $V$, on the other hand, corresponds to adding black to the pigment.

The $V$ component is expected to be the main difference between two points with similar terrain cover, but one covered with shadow. The presence of shadows on *cam1* is especially evident on the highest peaks. In most peaks, one of the slopes was receiving light directly from the sun, while the other was receiving reflected light. Two of the three color components (hue and saturation) are expected to be similar for points in either of the two slopes. In the RGB color model, the three color components are expected to be different for those points.

A set of experiments were run using the HSV color model. The training pairs were constructed with the transformed color values. Foley et al (1994) describe the algorithms required to convert from the RGB color model to the HSV model. The same structure of the input vector described in Section 3.2 was used (a 9-element neighborhood plus the normal vector information). 5% of the samples in *cam1* was taken at random for training, and the same parameter search for the learning rate and the number of hidden units was performed.

### 3.4 Network Generalization

Once trained, a neural network is expected to generate a sensible approximation of similar inputs. *Cam2*, *cam3*, and *cam4* were used to test the capacity of the neural network to generalize terrain textures. These experiments attempted to determine if the elevation-to-color mapping found by the neural network can be applied to nearby terrain. It is important to determine this generalization capacity since this would be the intended use of this technique. The trained network should be used to generate an approximation of a texture from a terrain model that does not necessarily exist. Previous experiments used *cam1* to train and test the neural network.

The input vector described in Section 3.2 was used for these DEMs. This vector has the highest dimensionality of the ones tried, and it contains the most information about the data sets. The network used was the one that gave the lowest test error after being trained with data from *cam1*. Section 4.2 discusses the architecture and parameters of this network.

The neural network is expected to color unseen elevation samples by interpolating from the pairs that were used for training. However in some cases, the network had to color elevations that were not in the range of the training data. Table 3.1 shows the elevation ranges of the DEMs used. *Cam2* has higher elevations and *cam4* has lower elevations than the ones from *cam1* used for training. Furthermore, *cam2* has steeper slopes that the one in *cam1*. In those cases, the neural network also had to extrapolate points.

Table 3.1: Elevation ranges of the DEMs used

| **DEM** | **Minimum Elevation** | **Maximum Elevation** |
|---|---|---|
| *cam1* | 2875.00 m | 3886.31 m |
| *cam2* | 3109.00 m | 3942.56 m |
| *cam3* | 2898.00 m | 3788.43 m |
| *cam4* | 2827.38 m | 3652.38 m |

# Chapter 4

# RESULTS

This chapter contains the results of the experiments described in the previous chapter. Each section matches one section in chapter 3 where all the details of the experiments are presented. Section 4.1 contains the results of the parameter search performed with an input vector built with a 9-element neighborhood. The results obtained by including the normal vector and its magnitude in the input vector are presented in section 4.2. Section 4.3 presents the results of transforming the color information to the HSV color model. Finally, the results of the generalization experiments are included in section 4.4.

## 4.1 Parameter Search and Percentage of Data Sampled

Figure 4.1 presents the test and training errors for the two learning rates ($\eta = 0.01$ and $\eta = 0.1$) and the five different percentages of samples (1%, 5%, 10%, 25%, and 40%) considered. All experiments used a 9-element neighborhood of elevation samples as input vector, and the color outputs in RBG format.The results obtained with $\eta = 0.01$ (Figures 4.1a and 4.1b) are similar to the ones observed in preliminary experiments. Both the test and the training error decrease as the percentage of samples used increases. However, the behavior of the test error for $\eta = 0.1$ was significantly different. In this case, the test error shows a minimum around 5%. That minimum has the lowest test error of all the scenarios considered.

The behavior in Figure 4.1b may be a symptom of overfitting. As the percentage of samples increases, the network is trained with a larger number of pairs before checking for overfitting. If the addition of more samples is not providing new information, there is a greater chance that the network overfits the training pairs and loses its ability to generalize.

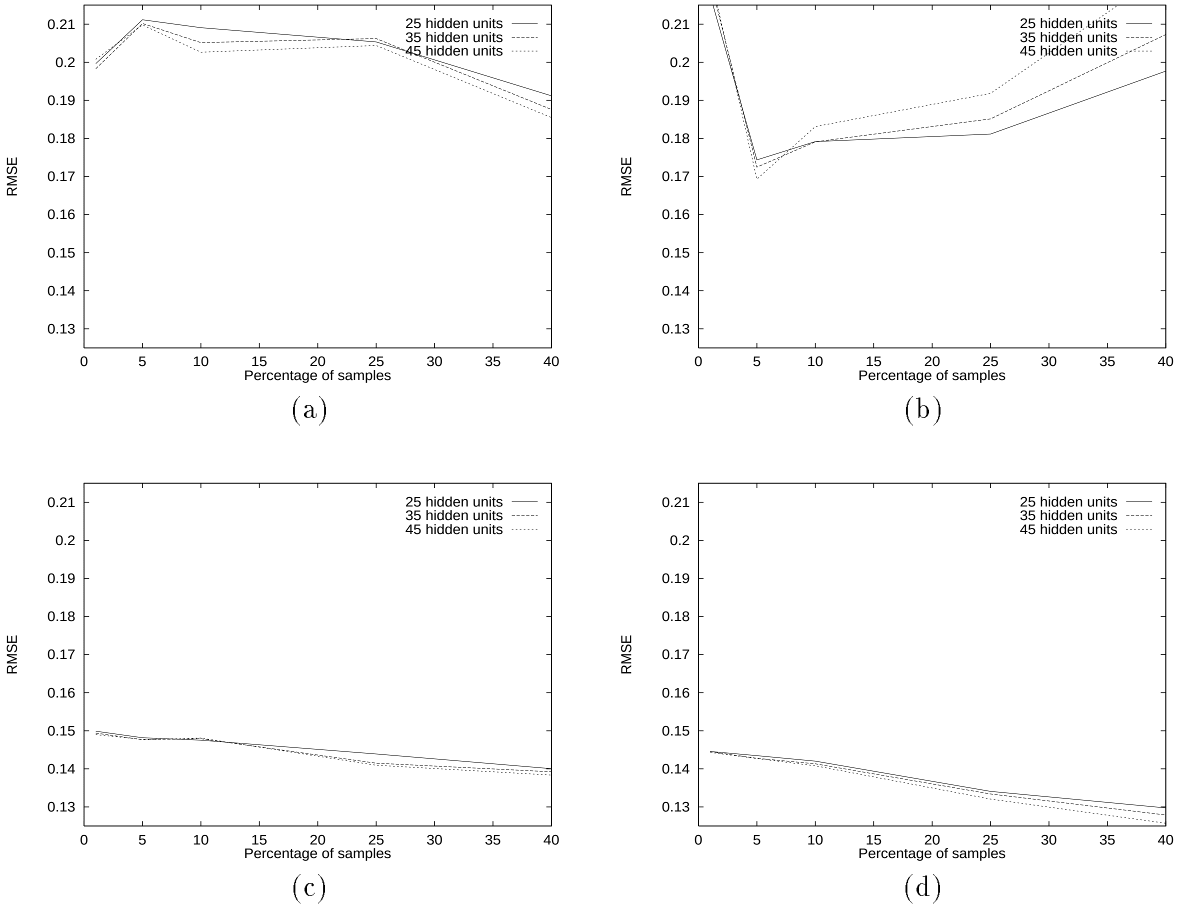



Figure 4.1: Test and training errors (RMSE) for training sets with different sizes. (a) Test error with $\eta = 0.01$ (b) Test error with $\eta = 0.1$ (c) Training error with $\eta = 0.01$ (d) Training error with $\eta = 0.1$

Based on the previous result, the percentage of samples used in training was fixed to 5% for all the experiments. This might not necessarily be the best value since the parameter search was not exhaustive. However, this percentage yielded the best results and helped reduce the training time to reasonable values. Training using 5% of the samples typically required between three and five hours depending on the actual network parameters and architecture, and the load on the HP 9000/715/50 workstations used.

The search for good values for the learning rate and the number of hidden units was performed independently for each of the different input vectors considered. The network configuration that produced the best results for the nine-element neighborhood considered at this point had $\eta = 0.1$ and 45 hidden units. These values were consistent with the number of hidden units and the learning rate that were found best for most of the other scenarios. The test error during training for this scenario was 0.1678.

Figure 4.2 shows the results obtained with this input vector (nine-element neighborhood of elevation samples). The output texture, the target texture (*cam1*) and an error image are displayed as a flat image, and mapped onto the terrain. The error image corresponds to the distance between pixels in RGB space. A darker gray means a greater distance between pixels, i.e. a higher error magnitude.

The regions that have a homogeneous land cover in the output and the target texture have a close resemblance. The first difference that can be noted is the lack of high-frequency detail. The output texture does not show the spots at low elevations that are not covered with vegetation. There is also a smooth transition between rock and vegetation that is much sharper in the target texture. Moreover, the network seems to define several thresholds on the elevation where the terrain color changes smoothly. The smoothing effect observed in the output is directly associated with the noise removal properties of neural networks and the use of a neighborhood as an input vector.

The colors in the target and the output textures also have a close resemblance especially in the areas covered with vegetation. These areas have a bluish appearance on both textures, but the red component has lower values in the target texture. The bluish appearance of the target texture can be explained by the multiple factors that influence perception of distant vegetation. In particular, there are atmospheric effects (light scattering and absorption) that distort color.

The color approximation of lake Agnes is especially poor. There is not a sharp edge between water and ground, and the lake shore is totally missing in some parts of the lake. Besides, the vegetation and water color are similar. However, these two colors are also very alike in the target texture. In this case, the lake is mainly recognized by its shape. Moreover, the small island inside the lake is not present in the output. This situation is caused by the lack of elevation samples of that feature in the original 7.5-minute DEM.

As shown on the error image (Figures 4.2e and 4.2f), the areas with the highest error are along the tree line. These errors are expected since the tree line is not defined by

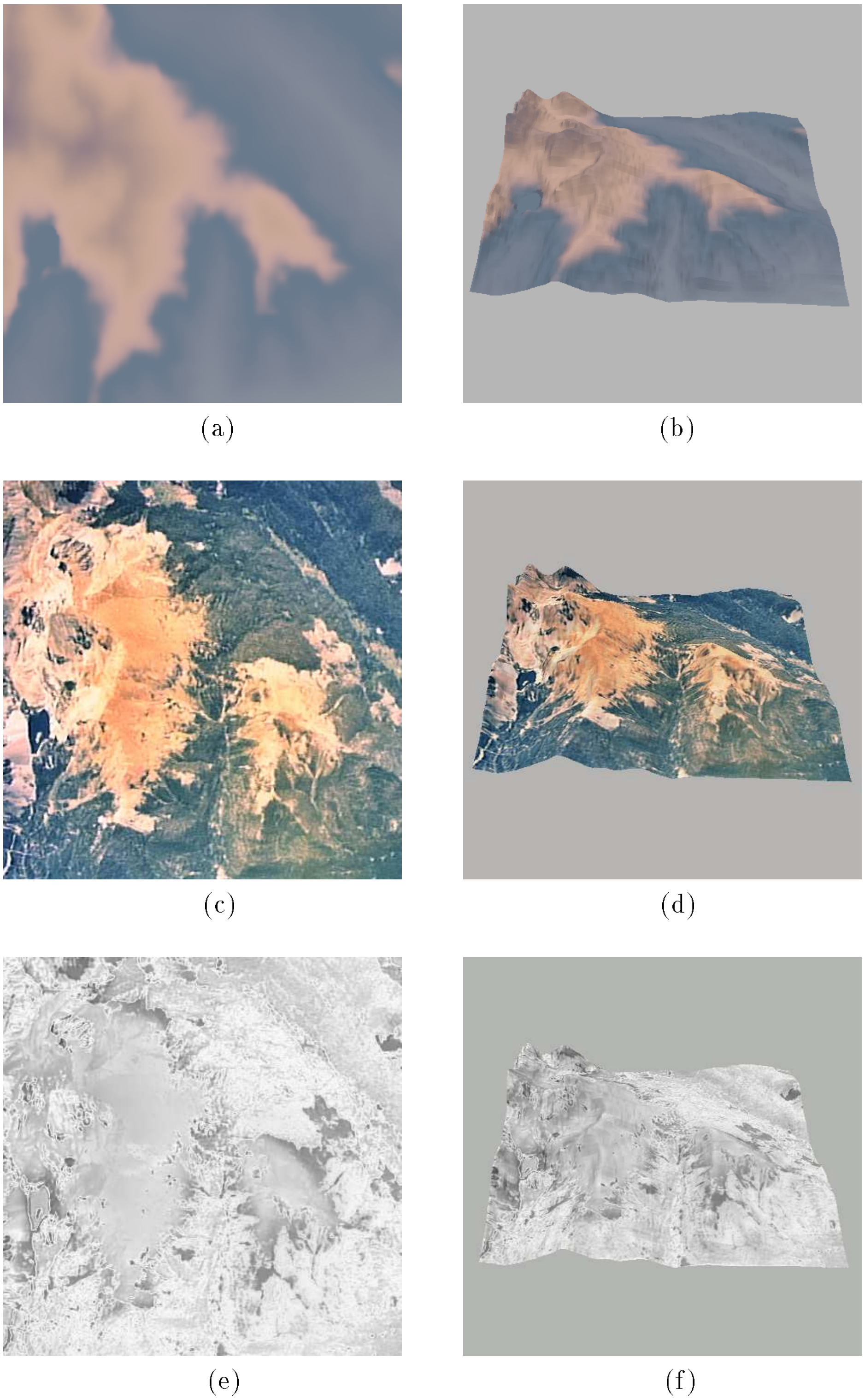



Figure 4.2: Results obtained using a 9-element neighborhood of elevation samples. (a) output image. (b) Output image mapped onto the terrain. (c) Original photograph (target image). (d) Original photograph mapped onto the terrain (e) Error image (difference between output and target images) (f) Error image mapped onto the terrain

a hard elevation threshold. That is, there are points with similar elevation that fall in areas covered with vegetation while others are covered by rock. This problem is directly associated with the fact that the color should be a function of many variables, not just elevation.

### 4.2 Including the Surface Normal Vector

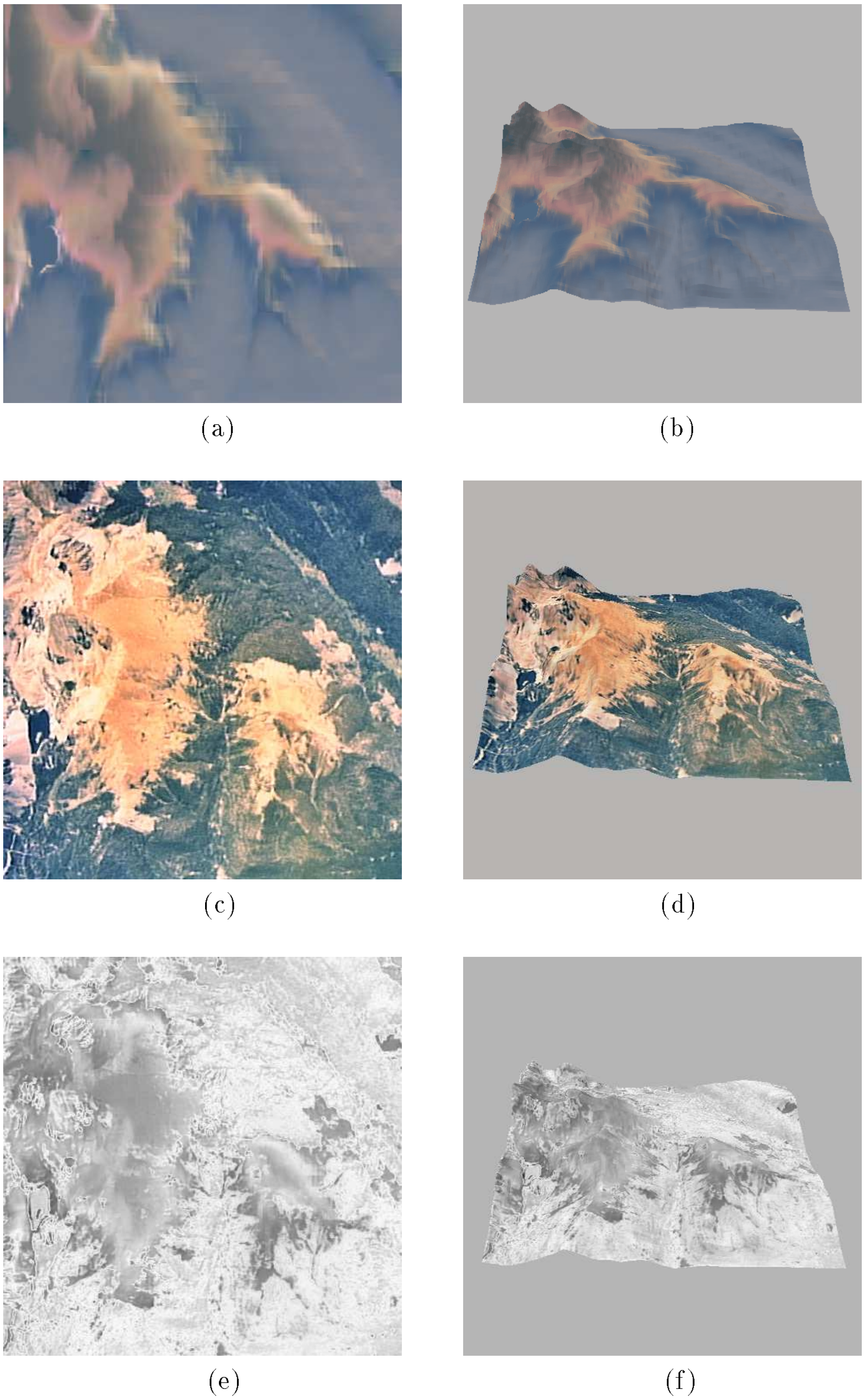
The best results obtained with this extended input vector (9-element neighborhood of elevations samples, three normal vector components, and normal magnitude) are presented in Figure 4.3. The network configuration used to get this result had $\eta = 0.1$ and 35 hidden units (all other parameters remaining the same). The test error was 0.1956.

The output texture has some artifacts that give it a blocky appearance. The regularity of those artifacts made very unlikely that they were caused by the neural network. Therefore, the elevation values were checked at those points where major color discontinuities were detected. Figure 4.4 shows how significant elevation discontinuities were found in the DEM.

The distance between discontinuities shows that this problem was originally in the 7.5-minute DEMs. It was not created by the linear interpolation of elevation samples that was performed during the registration step of the DEMs and the aerial photographs. This problem can be approached by smoothing the elevation map (Dudgeon and Mersereau, 1984). Another approach is to apply a non-linear interpolation that guarantees continuity (Press et al, 1992).

There are several features where the improvement over the previous result can be perceived. The tree line is much better defined. Also, the network produced a similar color difference between the two slopes of Mount Mahler on the left. An analogous shadow pattern is generated, but the color generated on the right side of the mount is darker than desired. The error image (Figures 4.3e and 4.3f) shows how the difference between the output and the target textures concentrates on the highest elevations.

There is an important improvement in the appearance of Lake Agnes. About half of the lake shore is now clearly defined. The border of the lake seems to be partially



(a) (b) (c) (d) (e) (f)

Figure 4.3: Results obtained using a 9-element neighborhood of elevation samples and the surface normal vector. (a) output image. (b) Output image mapped onto the terrain. (c) Original photograph (target image). (d) Original photograph mapped onto the terrain (e) Error image (difference between output and target images) (f) Error image mapped onto the terrain

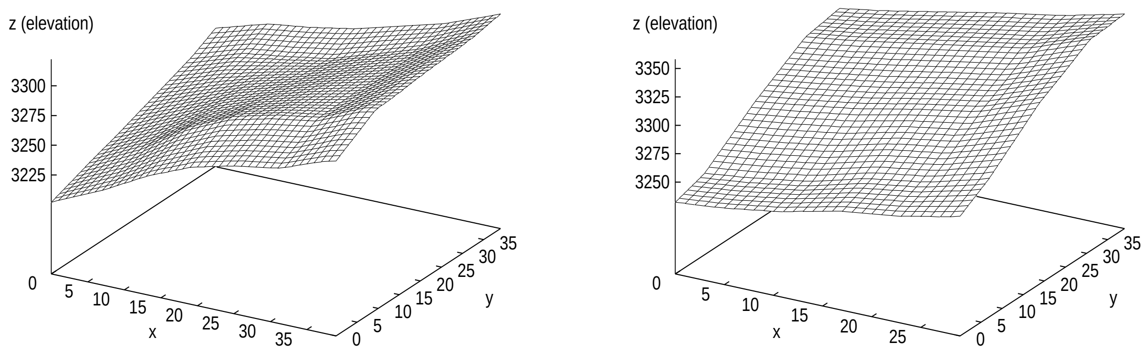



Figure 4.4: Slope discontinuities in the DEMs that affect the elevation-to-color mapping

surrounded by steep slopes. The inclusion of the normal vector generates this feature on the output texture. However, there are other portions of the lake, especially the upper left side, where its boundaries are still poorly approximated. The water color and the island inside the lake remain as problems that do not benefit from the inclusion of the normal vector.

In general, the inclusion of the normal in the input vector improved the appearance of the the output texture despite the increase in the test error (from 0.1678 to 0.1956) due to a greater color difference in the output texture. This difference is mainly associated with the red component. The red values at low elevations are higher and at high elevation are lower that they should be.

Figure 4.5 shows how the network generates color distributions for the red and blue components that resemble the original distributions. The location of the highest peaks are approximated with an acceptable degree of accuracy despite the differences in magnitude and the differences in color ranges. However, the network produces a poor result for the red component. The original distribution reflects the great number of samples that have a very high red component which correspond to rock. The output distribution, on the other hand, has a peak around the middle. This error accounts for most of the color differences in the output texture.

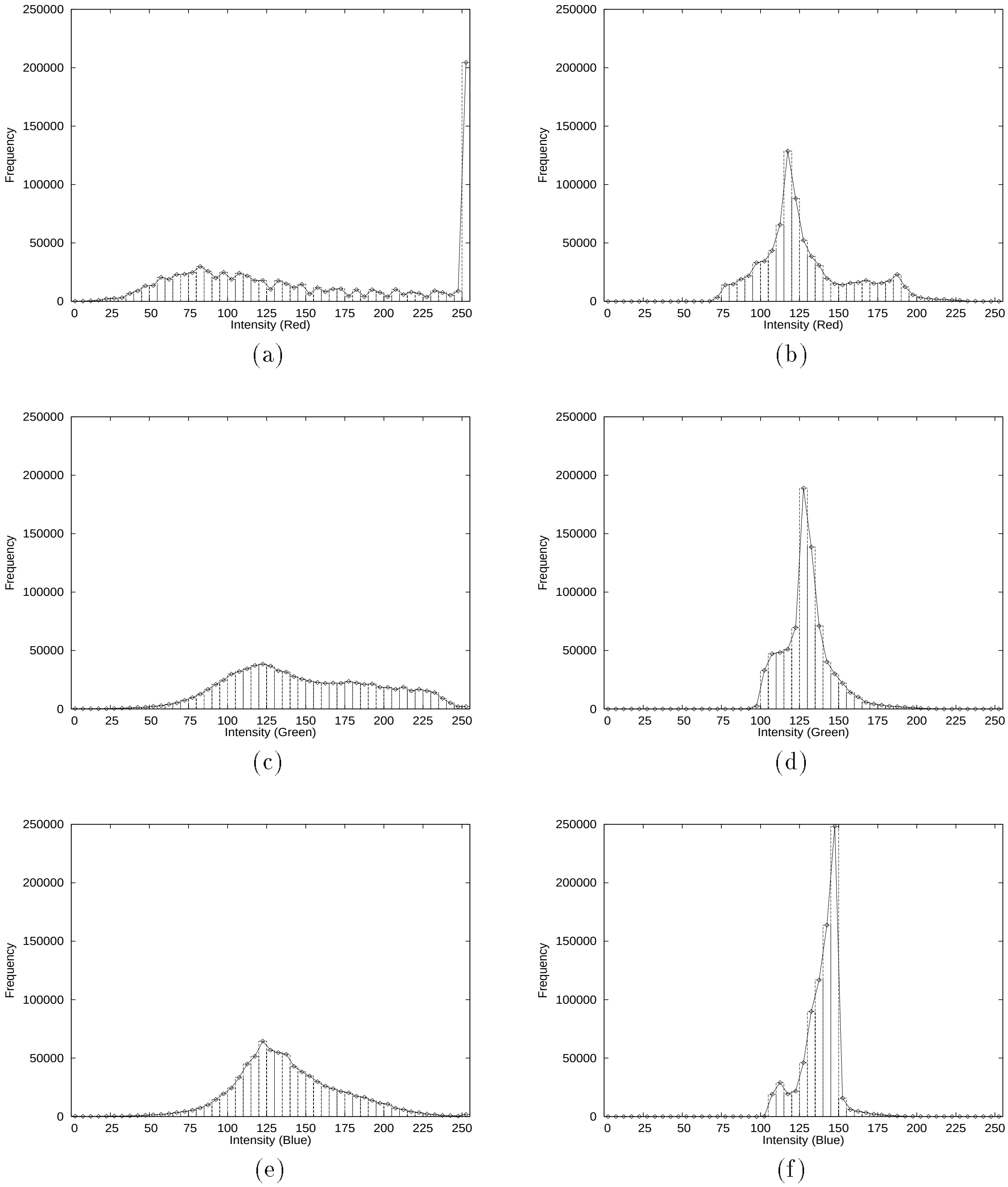



Figure 4.5: Color distributions of the target and output textures for *cam1*. (a) Red component (target) (b) Red component (output) (c) Green component (target) (d) Green component (output) (e) Blue component (target) (f) Blue component (output)

After finding how the normal vector improves the results, a reduced version of this input vector was tried. The nine input components corresponding to the nine-element neighborhood were reduced to only the center element. The components corresponding to the normal vector remained in the same position. Since the neural network was not implemented on a parallel machine, the reduction of the input vector means an important decrease in computation time.

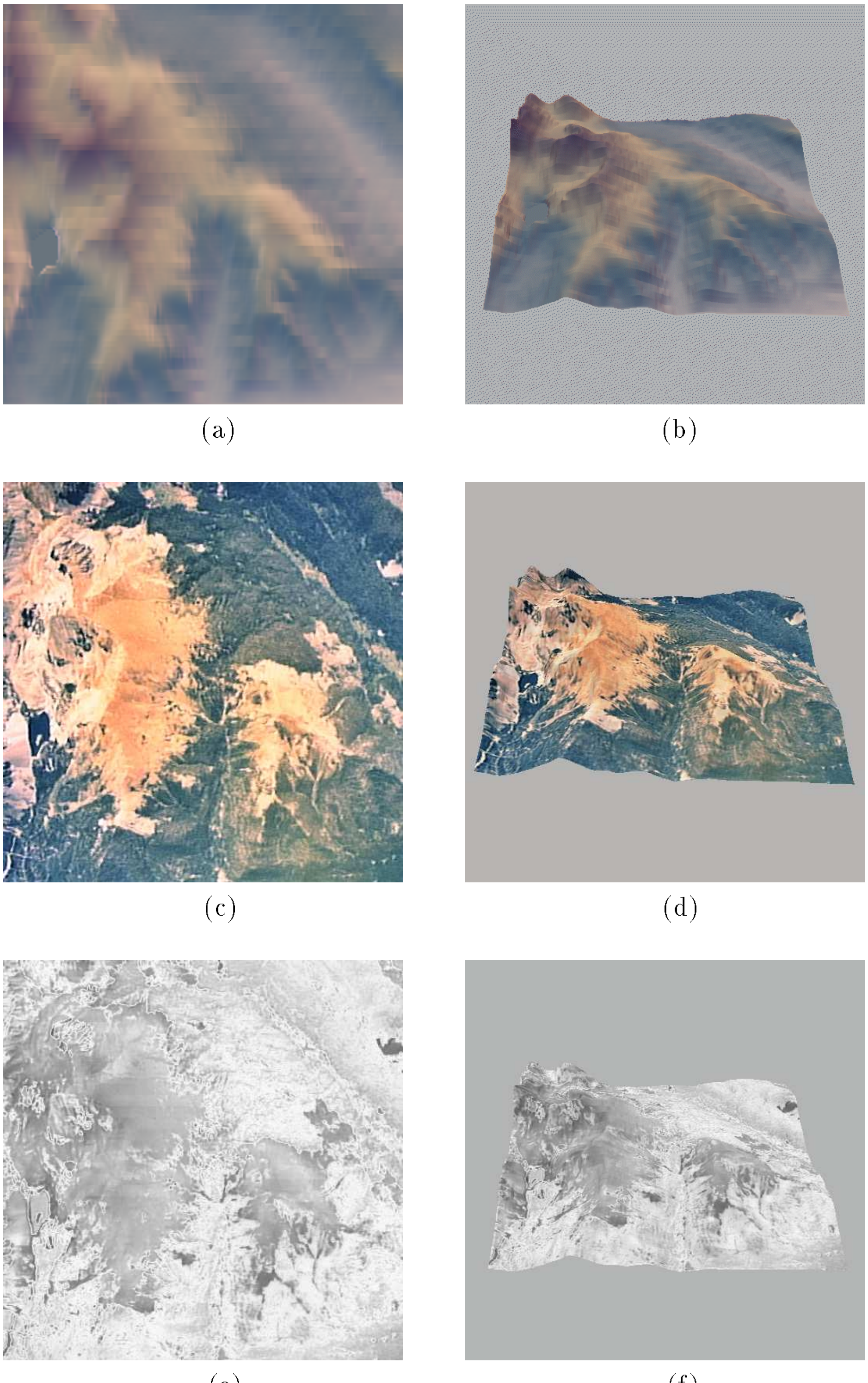
Figure 4.6 shows the results obtained. The output texture looks similar to the result obtained using all nine elements of the neighborhood. However, the blocky appearance is more obvious. The use of a neighborhood partially smooths the discontinuities present in the DEM. This approach would rely much more on smoothing the elevation values previous to processing.

This reduced input vector generates a greater range of terrain colors. For instance, those areas with a small slope and low elevation have a lighter color. This coloration helps visualize the terrain, but does not necessarily corresponds to the target texture. Besides, the difference in color between the two slopes of the mountains does not show up in the output texture. This feature should be closely related to the normal vector of the terrain, but the inclusion of a neighborhood seems to play an important role in its generation.

The detail around Lake Agnes confirms the importance of the normal vector in the generation of this area. The problem with the water color remains since it is related to the close similarity in the water and the terrain colors in the target texture. The absence of island inside the lake also remains since it is caused by undersampling and similarities in elevation.

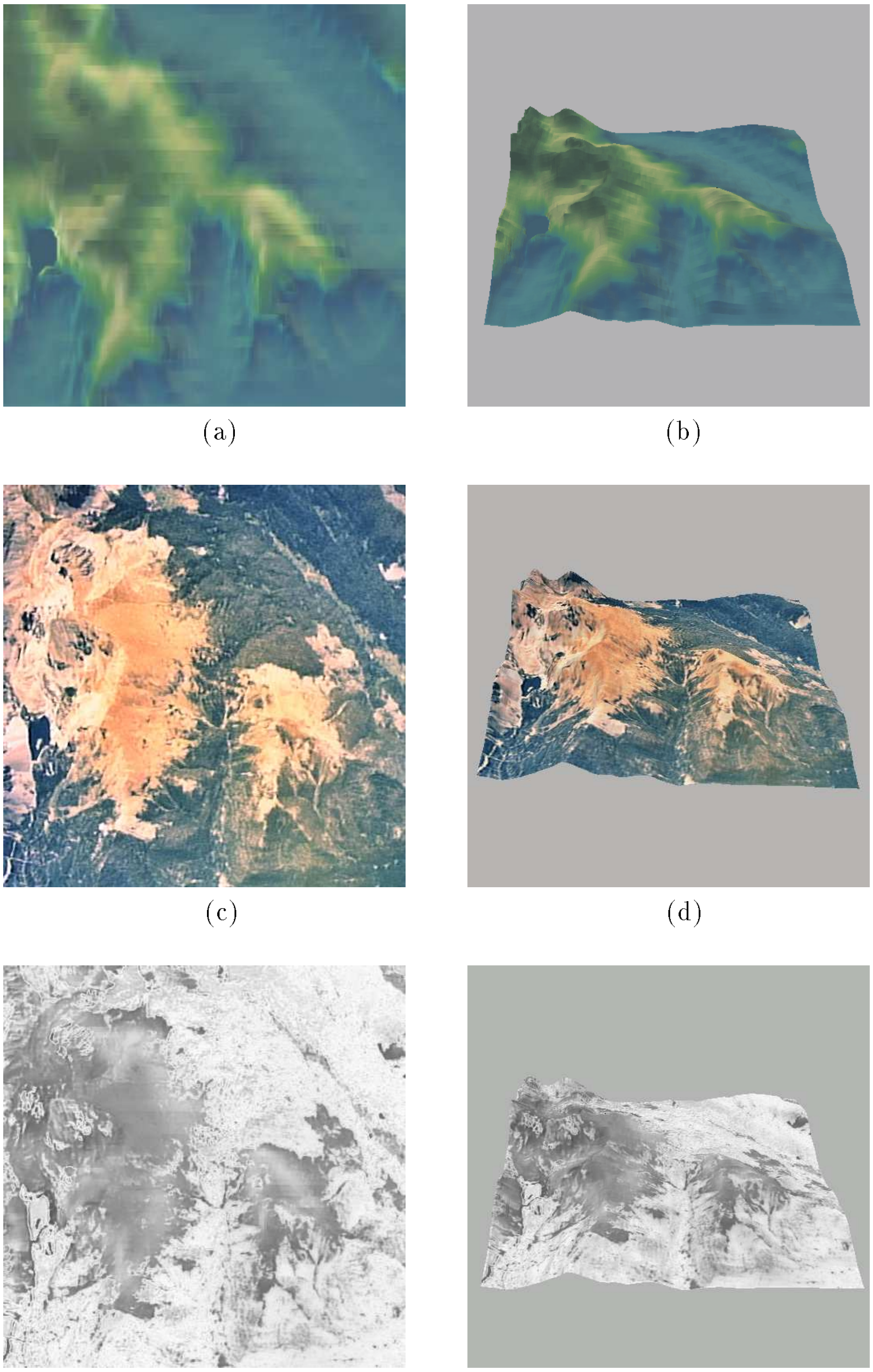
### 4.3 Using the HSV Color Model

Figure 4.7 presents the best results obtained using the HSV color model. The network parameters used were: $\eta =$ and 25 hidden units. Once again, 5% of the samples in *cam1* were used for training, the learning rate for the hidden layer was four times larger, and the momentum term was set to zero. The input vector was composed of a 9-element neighborhood, the three normal vector components, and the normal vector magnitude. The test error was 0.207248.



(a) (b) (c) (d) (e) (f)

Figure 4.6: Results obtained using the single elevation sample and the surface normal vector. (a) output image. (b) Output image mapped onto the terrain. (c) Original photograph (target image). (d) Original photograph mapped onto the terrain (e) Error image (difference between output and target images) (f) Error image mapped onto the terrain



(a) (b) (c) (d) (e) (f)

Figure 4.7: Results obtained using a 9-element neighborhood of elevation samples and the surface normal vector (HSV outputs). (a) output image. (b) Output image mapped onto the terrain. (c) Original photograph (target image). (d) Original photograph mapped onto the terrain (e) Error image (difference between output and target images) (f) Error image mapped onto the terrain

Despite the differences in color, the general appearance of the output texture is very similar to the result obtained with the RGB color model. However, there is a high error in the areas covered by rock. Those areas do not have a redish color as in the target texture, but a bright green color instead. Nevertheless, the areas covered with vegetation have a better approximation. The error image contains dark areas (high error) on the highest elevations, but light gray tones (low error) on the lower elevations. The approximation of the vegetation texture is significantly better than the previous results obtained. The target texture is clearly dominated by two hue values: green and red, and the network is finding a better relationship between elevation and color for those points with a green hue.

This error difference between the areas covered with vegetation and those covered with rock was unexpected since the use of the HSV should have yielded better results for higher elevations (with a redish coloration) where shadows are present. However, the assumption that the color of those rock areas covered with shadows differ mainly on the $V$ component was found incorrect. The inspection of *cam1* gives indication that in most cases the opposite holds. The $V$ component remains almost unchanged, while the hue and the saturation vary. This unexpected feature should be related to the remote sensing process of the terrain.

### 4.4 Network Generalization

Figure 4.8 shows the results obtained for *cam2*, *cam3* and *cam4*. The original terrain textures are also presented for comparison. All textures were mapped onto the DEMs and rendered as seen from the west. The network used corresponds to the one described at the beginning of section 4.2 ($\eta = 0.1$ and 35 hidden units, trained with 5% of *cam1*).

The output textures for *cam2*, *cam3*, and *cam4* have visually the same quality of the texture obtained for *cam1*. Snow Lake, which is east of lake Agnes, is rendered with the same color as the surrounding rock. It is recognized by its homogeneous coloration and its edges. The lake shore has steep slopes that cause a shadow effect that give a 3D appearance to the lake. The two lakes can be recognized in the output textures, even though they cannot always be recognized in the respective target texture.

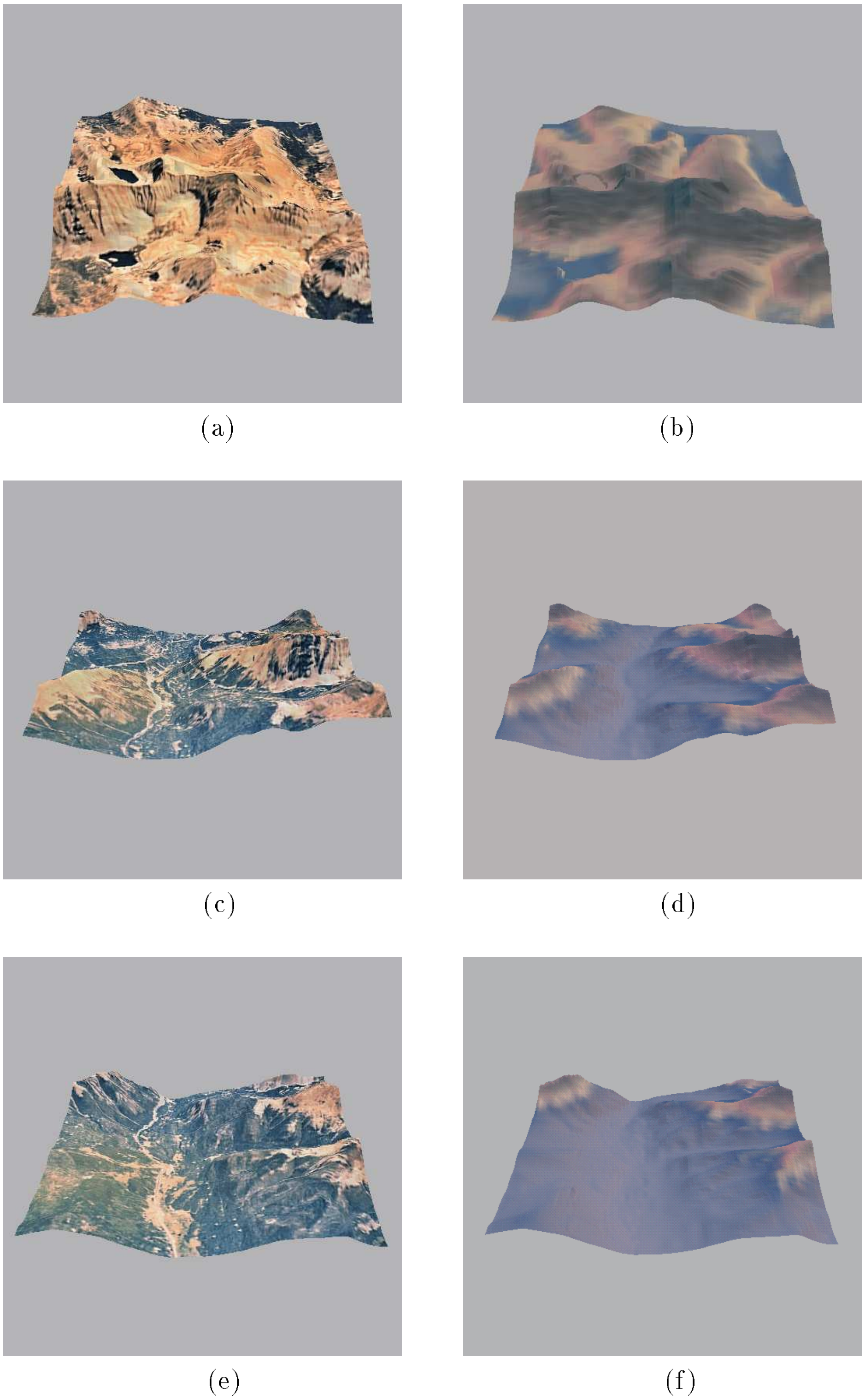



Figure 4.8: Results obtained using a 9-element neighborhood of elevation samples and the surface normal vector from *cam1*. (a) *cam2*, (b) Network approximation of *cam2*, (c) *cam3*, (d) Network approximation of *cam3*, (e) *cam4*, (f) Network approximation of *cam4*

Figure 4.9 shows the error (RMSE) between the target and the output textures for each of the color components. The error values are relatively similar for all DEMs. The error in the output texture obtained for *cam2* is slightly higher. This data set contains terrain that is at high elevations and is mostly covered by rock. Moreover, the highest elevations are actually above the range of *cam1*. Therefore, the network extrapolates at these points and should be prone to produce outputs with a higher error.

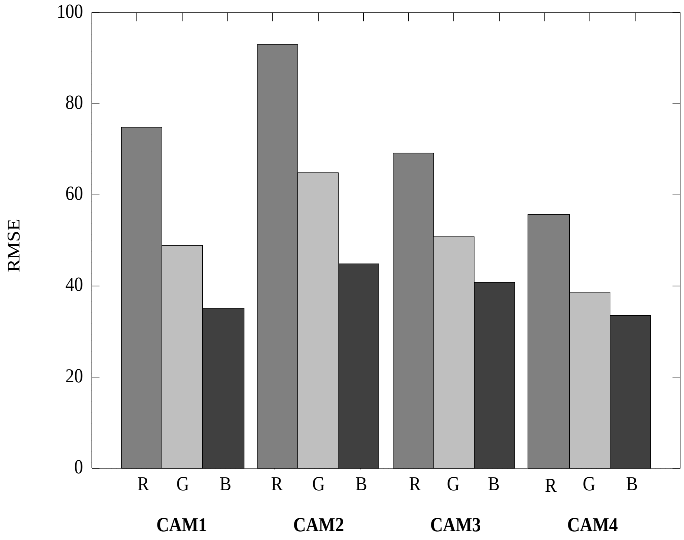



Figure 4.9: RMSE between target and output textures using network trained with *cam1* data

The RMSE value for each data set seems to be associated with the amount of rock. This may be caused by insufficient training samples from those zones in *cam1* covered with rock. Those areas have the highest values for the red component which has the highest RMSE. Using a target texture for training with a more even distribution of terrain types should help reduce this problem.

# Chapter 5

## DISCUSSION

This project shows how MLPs can be used to approximate a mapping between terrain elevation and color. This type of neural network has been proved to be capable of producing arbitrarily accurate approximations to an arbitrary function. Even though the color of a given terrain location does not only depend on its elevation, the relationship between these two is strong enough to produce a result that clearly resembles the original terrain.

The input representation was an important factor for the generation of results. The normal vector was found to be essential to improve the quality of the output textures obtained. Features like shadows, lakes and the tree line are significantly sharper when the normal vector is included. The magnitudes of the connection weights in the hidden layer also confirm the important role that the normal plays. Similar results might be obtained by replacing the three normal components and its magnitude with the two gradient components of the elevation.

However, there are several limitations associated with this approach. The noise removal feature of neural networks contributes to the smooth appearance of the output textures. This characteristic is intrinsic to neural networks, therefore image enhancement and restoration techniques will be required to improve the final image (Jain, 1989; Gonzalez and Woods, 1992). In addition, a trained network will only produce textures with visible differences if the elevation maps have significant differences.

Another limitation of this approach is associated with the availability of adequate training data. The DEM and the aerial photograph used for training should provide enough samples of all the terrain features that want to be approximated. This limitation is contributing to the color errors of rock and lakes in the data sets used. Moreover, the training data should have an elevation range wide enough so the network does not have to extrapolate color values when a new DEM is processed.

The applicability of a trained network is also limited to terrain that has features similar to the training data. In other words, a network can only generate terrain it has been taught. A network trained with terrain that has some particular elevation range and land cover should produce poor results if used to approximate texture of terrain with different features. Moreover, some features that are not related to elevation, such as man-made structures, are not likely to appear in the output textures.

The training time can be seen as a drawback of the back-propagation algorithm used to train MLPs. The training times ranged between three and five hours for most experiments as run in this project. However, this might not be a problem if more computing resources are available or if a parallel machine can be used. One of the positive features of this approach is its generalization capability. The error measures obtained when applying the trained network to new DEMs show that this technique is useful to approximate terrain textures within the constraints previously mentioned.

However, some important problems remain to be solved. To remove the blocky appearance of some of the textures, the DEMs should be smoothed before processing them. Nevertheless, this effect could persist to some degree depending on the severity of the discontinuities in the original data. The error in the estimation of the red component is the main cause of color differences. This problem could be related to an insufficient number of samples with a high red intensity used during training.

Even though the use of the HSV color model did not proved useful for removing shadows, it helped improve the results for areas covered with vegetation. It would be desirable to find a way to take advantage of this feature, and obtain the rock approximation generated with the RGB color model at the same time.

An insufficient number of samples is also causing the differences in the coloration of lakes. A greater number of samples corresponding to the lakes should be used. Alternatively, some artificial training pairs could be used to make the network learn an appropriate color when the normal vector is zero.

Finally, the artificial textures produced by the neural network might be improved by using additional terrain information. In particular, larger input vector could include land cover information. The land cover maps distributed by the USGS includes terrain classifications such as: forest land, water, wetland, barren land and tundra (US Geological Survey, 1990b). Maps with more specific classifications (level II) are also available. The inclusion of this information should sharpen the terrain texture (e.g. the tree line,) and help attenuate the blocky appearance obtained so far.

However, the inclusion of this information also has some drawbacks. A larger input vector imposes a grater number of restrictions on the terrain models that can be applied to a trained network. Also, the generation of terrain models to be applied to a trained network might become a problem. Some of the additional information might not be available or could not be estimated. Furthermore, a larger input vector implies a greater amount of resources needed for storing and processing the terrain models.

# REFERENCES


Bryson, A. E. and Ho Y., 1969. *Applied Optimal Control.* New York: Blaisdel.

Burrough, P. A.. (1992). *Principles of Geographical Information Systems for Land Resources Assessment.* Oxford University Press, New York. ch. 3

Chen, D. S. and Jain, R. C., 1994. A Robust Back Propagation Learning Algorithm for Function Approximation. *IEEE Transactions on Neural Networks.* Vol. 5, No. 3. pp. 467-479

Cohen, D. and Gotsman, C., 1994. Photorealistic Terrain Imaging and Flight Simulation. *IEEE Computer Graphics and Applications.* March 1994, pp. 10-12

Dudgeon, D. E, Mersereau, R. M., 1984. *Multidimensional Digital Signal Processing.* Prentice-Hall, Englewood Cliffs, New Jersey. Ch. 3

Farmer, D. and Sidorowich, 1987. Predicting Chaotic Time Series. *Physical Review Letters.* Vol. 59, pp. 845-848

Farmer, D. and Sidorowich, 1988. Exploiting Chaos to Predict the Future and Reduce Noise. In *Evolution, Learning, and Cognition.* Ed. W. C. Lee, pp.277-330. Singapore: World Scientific

Foley, J. D., Dam, A. V., Feiner, S. K., Huges, J. F., Philips, R. L., 1994. *Introduction to Computer Graphics.* Addison-Wesley, Reading. pp. 402-422

Fowler, R. J. and Little, J. J., 1979. Automatic Extraction of Irregular Network Digital Terrain Models. *Computer Graphics.* August, pp. 199-207

Gonzalez, Rafael C. and Woods, Richard E., 1992. *Digital Image Processing.* Addison-Wesley, Reading, Massachusetts. Ch. 4, 5

Gorman, R. P., Sejnowski, T. J., 1988a. Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets. *Neural Networks.* Vol 1, pp. 75-89

Gorman, R. P., Sejnowski, T. J., 1988b. Learned Classification of Sonar Targets Using a Massively-Parallel Network. *IEEE Transactions on Acoustics, Speech, and Signal Processing.* Vol. 36, pp. 1135-1140

Goss, M. E., 1994. An Adjustable Gradient Filter for Volume Visualization Image Enhancement. *Proceedings of Graphics Interface '94.* Banff, Alberta. pp. 67-74

Groß, M., 1994. *Visual Computing.* Springer-Verlag, Berlin. pp. 176-200

Hertz, J., Krogh, A. and Palmer, R. G., 1991. *Introduction to the Theory of Neural Computation.* Addison-Wesley, Redwood, Massachusetts. p. 123

Hornik, k., 1991. Approximation Capabilities of Multilayer Feedforward Networks. *Neural Networks.* Vol. 4, No. 4. pp. 251-257

Jain, Anil K., 1989. *Fundamentals of Digital Image Processing.* Prentice-Hall, Englewood Cliffs, New Jersey. Ch. 7, 8

Jensen, R. J., 1986. *Introductory Digital Image Processing: A Remote Sensing Perspective.* Prentice-Hall, Englewood Cliffs, New Jersey. pp. 104-115

Lapedes, A. and Farber, R., 1988. How Neural Nets Work. In *Neural Information Processing Systems* (Denver 1987), ed. D. Z. Anderson, pp. 442-456. New York: American Institute of Physics.

Le Cun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubard, W., Jackel, L. D., 1989. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation.* Vol. 1. pp. 541-551

Lippmann, R. P., 1987. An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine.* April. pp. 4-22

Lippmann, R. P., 1989. Review of Neural Networks for Speech Recognition. *Neural Computation.* Vol. 1. pp. 1-38

Mark, D. M., 1978. Concepts of Data Structure for Digital Terrain Models. In *Proc. DTM Symp. American Society of Photogrammetry.* St. Louis, Missouri. pp. 24-31

McCullagh, M. J., 1982. Mini/Micro Display of Surface Mapping and Analysis Techniques. *Cartographica.* Vol. 19, No. 2. pp. 136-144

Miller, G. S. P., 1986. The Definition and Rendering of Terrain Maps. *Computer Graphics.* Vol. 20, No. 4. pp. 39-45

Parker,D. B., 1985. Learning Logic. Technical Report TR-47, Center for Computational Research in Economics and Management Science, Massachusetts Institute of Technology, Cambridge, MA.

Pomerleau, D.A., 1989. ALVINN: An Autonomous Land Vehicle in a Neural Network. In *Advances in Neural Information Processing Systems I* (Denver 1988), ed. D. S. Touretzky, pp. 305-313. San Mateo: Morgan Kaufmann.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., 1992. *Numerical Recipes in C.* 2nd edition, Cambridge University Press, New York. Ch. 3

Rumelhart, D. E., McClelland, J. L., and Williams, R. J., 1986a. Learning Representations by Back-Propagating Errors. *Nature.* Vol. 323, pp.533-536

Rumelhart, D. E., McClelland, J. L., and Williams, R. J., 1986b. Learning Internal Representations by Error Propagation. *Parallel Distributed Processing.* Vol. 1, chap. 8

Scarlatos, L. L., 1990. A Refined Triangulation Hierarchy for Multiple Levels of Terrain Detail. In *Proceeding of the Image V Conference.* Phoenix, Arizona. June. pp. 19-22

Sejnowski, T.J., Rosenberg, C. R., 1987. Parallel Networks that Learn to Pronounce English Text. *Complex Systems.* Vol. 1, pp. 145-168

Tarvydas, A., 1984. Terrain Approximation by triangular Facets. *ASP-ACSM Convention.* Technical Papers, Vol 1. pp. 524-533

Taylor, D. C. and Barrett, W. A. (1994). An Algorithm for Continuous Resolution Polygonalizations of a Discrete Surface. *Proceedings of Graphics Interface '94.* Banff, Alberta. pp. 33-41

U.S. Geological Survey, 1990a. *Digital Elevation Models: Data Users Guide.* Reston, Virginia

U.S. Geological Survey, 1990b. *Land Use and Land Cover from* 1:250,000- *and* 1:100,000-*Scale Maps.* Reston, Virginia

Werbos, P., 1974. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph. D. Thesis, Harvard University.