1989

A computer implementation of an orthonormal expansion method for digital image noise suppression

Hui-Jung Lee

Computer Science Department
Rochester Institute of Technology

A Computer Implementation of an Orthonormal Expansion Method for Digital Image Noise Suppression

by

Hui-Jung Lee

A thesis,
submitted to The Faculty of the
Computer Science Department
in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science.

November 20, 1989

Approved by: 11/20/89
Professor James Heliotis (Chairman)

11/22/89
Professor Andrew Kitchen

21 Nov 89
Professor Peter Anderson

Title of Thesis:

A Computer Implementation of an Orthonormal Expansion Method for Digital Image Noise Suppression

I Hui-Jung Lee **hereby grant permission to the Wallace Memorial Library, of RIT, to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.**

Date: 11/22/1989

ABSTRACT

Images are usually corrupted by noise which comes from various sources: noise in the recording media (e.g. film grain noise), and noise introduced in the transmission channel. Noise degrades the visual quality of images and obscures the detail information in the images. One of the major sources of noise for images recorded on films is film grain noise. An orthonormal expansion algorithm for digital image noise suppression is implemented. The objective is to preserve as much sharpness and produce as few artifacts in the processed image as possible.

The method sections an image into non-overlapping blocks. Each block is treated as a matrix which is decomposed as a sum of outer products of its singular vectors. The coefficient of each outer product is modified by a scaling function and the matrix is reconstructed. The resulting image shows a reduction of noise. The two major problems in the method are: 1. the blocking artifacts due to the sectioned processing, and, 2. the trade-off between the suppression of noise and the loss of sharpness.

By separating the image into the low frequency and the high frequency components and processing only the latter component, the method is able to reduce the blocking artifacts to an invisible level. To obtain the optimal trade-off between the suppression of noise and the loss of sharpness, systematic variations of the coefficient scaling function were used to process the image. The best choice of the scaling function is found to be $[1 - (\sigma_i / a_i)^3]$ which is a little different from the least-square-error estimate, $[1 - (\sigma_i / a_i)^2]$.

ACKNOWLEDGEMENTS

I would like to thank my thesis Chairman, Professor James Heliotis, and my thesis committee, Professor Andrew Kitchen and Professor Peter Anderson, for their proofreading, support, patient encouragement and enthusiasm throughout.

I would also like to thank Professor Don Kreher, who was in my thesis committee but is currently on leave of absence from RIT, for his careful reading of several drafts of this thesis and his helpful suggestions that improve the final version.

Many other people also helped this work. I would like to express my gratitude and appreciation to the Imaging Sceience Laboratory of Kodak Research Laboratories who generously supported this project by providing the computing facilities, the image scanning, and the image printing. I also want to thank the training administration of Eastman Kodak Company who fmancially supported me through the graduate program.

I especially thank my husband, who shared with me the pain and happiness of the years when I was busy with both RIT graduate program and company work. His endless support and encouragement kept me going through it all. Finally, I like to mention my daughter. Though her birth had unavoidably interrupted the completion of this thesis work, her baby smiles, baby cries and innocence have carried me into a motherhood with joy and happiness.

TABLE OF CONTENTS

LIST OF FIGURES *

* NOTE: The fact that evaluation of SVD processes described in this thesis requires the output images be printed on photographic paper which is a better means for showing the processed images without introducing other random noise. The figures (Figure 7- Figure 27) included here are the color reproduction by Kodak ColorEdge color copier from the original photographic prints because not enough were produced. Hopefully, the noise introduced by the color copier will be equally distributed to the figures both processed and unprocessed that the evaluation of the algorithm performance can still be done. A thesis copy with the original photographic prints of Figure 7 to Figure 27 is submitted to RIT Library to meet the school requirement as well as for the purpose of archiving the original prints.

I. INTRODUCTION AND BACKGROUND

1.1 DIGITAL IMAGE PROCESSING

1.1.1 Image Processing

Images can be recorded by different types of sensing and recording systems. Image processing, in its general form, pertains to the alteration and analysis of pictorial information. Basically, three techniques of image processing are available

1. optical image processing,
2. electronic analog image processing, and
3. electronic digital image processing.

Optical image processing uses an arrangement of optics to carry out the desired operations on the images. For example, eyeglasses are a simple form of optical image processing. Others like optical pattern recognition (OPR) systems, use optical interference to identify the location of a given pattern [1].

Electronic analog image processing refers to the alteration of images by electrically modifying the signals. The most common example of this is the television image. The television signal is a voltage level that varies in amplitude to represent brightness throughout the image. By electrically changing this signal, we correspondingly alter the final displayed image appearance.

A digital image is described by a two dimensional array of numbers, each number representing the brightness of the corresponding spatial point of the image. Digital image processing is a form of image processing brought on by the advent of the digital computer. The digital computer is inherently more flexible and is more readily used in operations that involve nonlinearities and/or decision making processes.

Image processing is usually performed to achieve the following three goals:

1. image quality enhancement,
2. image analysis and information extraction, and
3. image coding and compression for transmission and storage.

These three goals can be more flexibly achieved by digital image processing [2-5].

1.1.2 Basic Digital Image Processing System

An image can be recorded in analog (e.g. TV video signals and photographic films) or digital forms. If an image is in an analog form, it has to be digitized by appropriate devices. Once the image is digitized, it can be processed by computer. A basic digital image processing system (see Fig. 1) consists of an input device (a camera and/or a scanner), a computer system, and an output device (a display monitor or an image printer).

In its ordinary sense, digitization consists of sampling the gray level in the image at every one of an M x N array of points. Since the gray level at these points may take any value in a continuous range of real numbers, digital processing rounds off the gray levels to a set of K discrete values. In order for the image reproduced from these numbers to be a good reproduction of the original, M, N and K have to be large. Generally, the finer the sampling and quantization, the better the reproduced image.

Within the digital domain, each point of an image has a numeric location and a numeric brightness. By manipulating these values of brightness within the image, the computer is capable of carrying out complex operations with relative ease. The output image is an array of digital numbers which are functions of the array of the input image.

The display of an output image can take several forms. If the image is to be displayed on a TV monitor, the digital image has to be converted into analog video signals. If a raster monitor is used, the digital image is stored in a frame buffer, and the screen is constantly refreshed by the frame buffer. If the image is to be reproduced as a hard copy picture, the array of digital numbers is used to control an output device to produce different gray levels at different point locations on the printed copy.

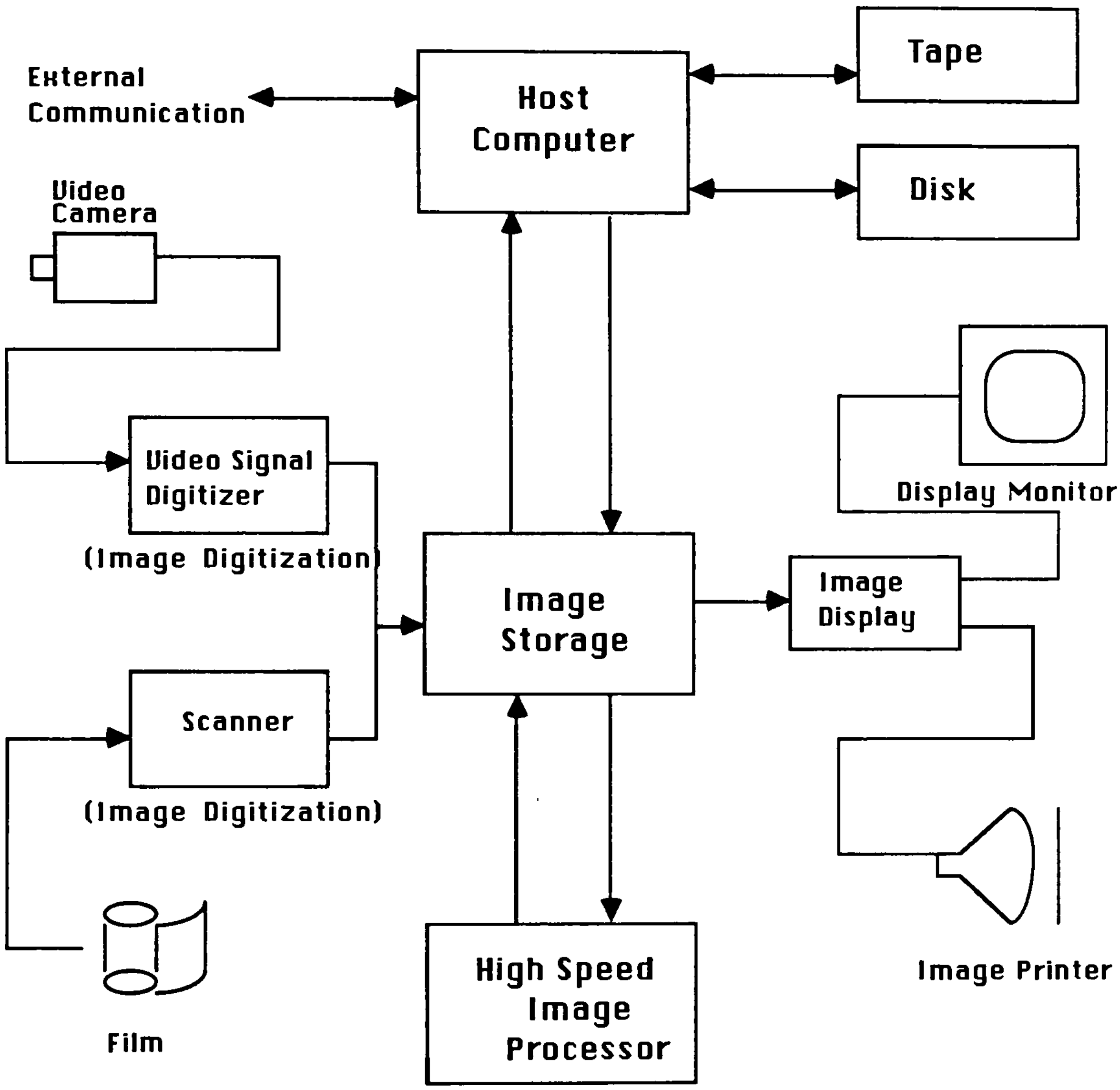


Figure 1. A Basic Digital Image Processing System.

1.1.3 Photographic Image, Scanner, and Digital Image

Photographic films have been the media for recording very high quality images. To convert the images recorded on the film negatives, transparencies, or reflection prints into the digital forms, a scanner with very small aperture can be used to read in the image information point by point. There are two modes of scanning in a scanner (Fig. 2-3):

(1) Transmission mode:

For the images on the film negatives or transparencies, the scanner measures the image signal in the transmission mode. A light source is focused into a small spot on the film. The gray level of a point on the image depends on the amount of light passing through the corresponding point on the film.

(2) Reflection mode:

For the images on reflection prints, the scanner measures the light reflected from the print surface. The amount of light reflected from a point on the print determines the gray level of the corresponding point on the image.

Color images are scanned in a similar fashion as the black and white images, except that color filters are used in the scanning process. Typically, a color image is scanned with three color filters: red, green, and blue filters. The image is scanned three times in registration, each time with a different color filter. The scanner readings after proper calibration are quantized into integers which are stored in the raster scan order. The order determines the location and the integer the gray level.

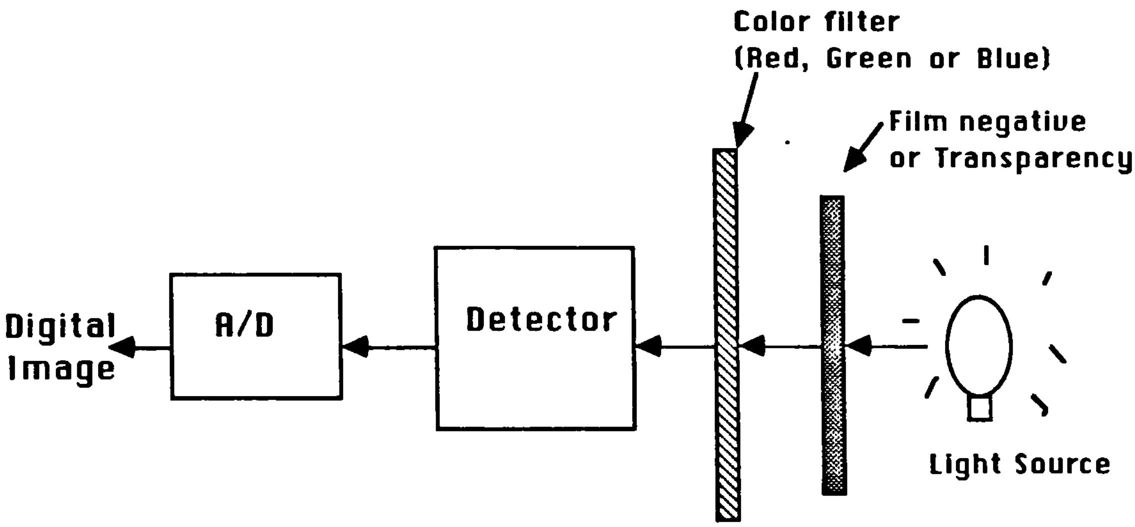


Figure 2. Transmission Mode of Scanning in a Scanner.

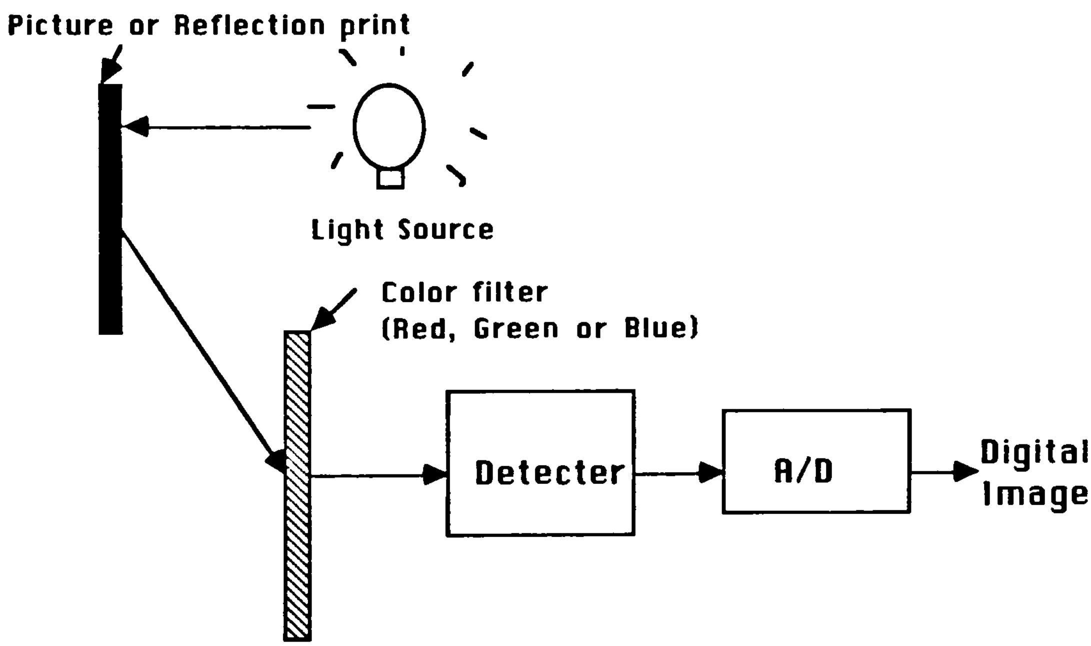


Figure 3. Reflection Mode of Scanning in a Scanner.

1.1.4 Film Characteristics

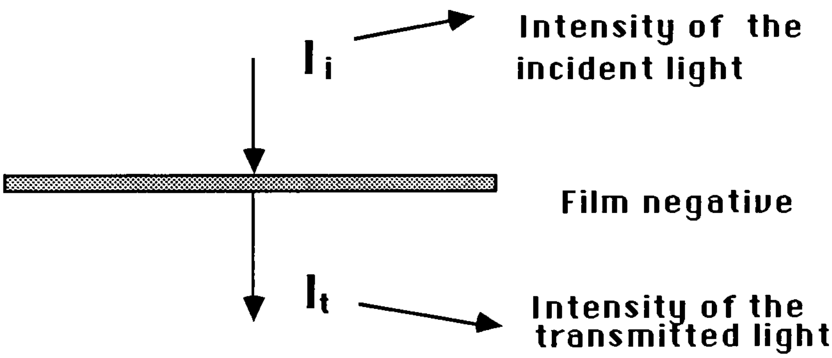
1.1.4.1 Film Structure

Photographic film consists of several layers of light-sensitive material and other chemicals [6]. Each layer has a large number of tiny crystals (grains) of silver halide embedded in a layer of gelatin. The combination of grains and gelatin is often called the photographic emulsion. Color films have emulsion layers containing dye forming chemicals, which are not present in the black and white films. When the emulsion is exposed to light, the number of grains which absorb light is a function of the light intensity. Different areas on a piece of film have different numbers of grains per unit area which absorb light, depending on the light intensity illuminating on that area. These exposed grains are later developed by chemicals, and the final processed film have varying darkness in the image area. The degree of darkness is a measure of the image signal in the original scene.

1.1.4.2 Density

The photographic effectiveness of light is measured by the image that can be developed. The developed image on the film, in turn, can be evaluated in terms of its ability to block the passage of light. The most direct measure is either the transmittance or the opacity. The transmittance, **T**, is defined by the ratio $\mathbf{I_t/I_i}$ (Fig.4), where $\mathbf{I_t}$ is the intensity of the transmitted light and $\mathbf{I_i}$ that of the incident light. Thus, the transmittance gives the fraction of the incident light transmitted through the developed film [6]. Opacity is simply the reciprocal of the transmittance $(\mathbf{I_i/I_t})$. The optical density **D** is defined as the logarithm of the opacity. Within a proper range of exposure, the optical density of the developed film is approximately a

linear function of the logarithm of the film exposure (Fig.5), where exposure, **H**, is the light intensity times the exposure time. Different films have a different **D-logH** curve, the more sensitive the film is, the higher density it can develop for the same exposure.



Transmittance $T = \dfrac{I_t}{I_i}$

Density $D = \log(\dfrac{1}{T})$

Figure 4. Measurement of the Transmittance or the Opacity: Opacity is the Reciprocal of the Transmittance. The optical density is defined as the logarithm of the Opacity.

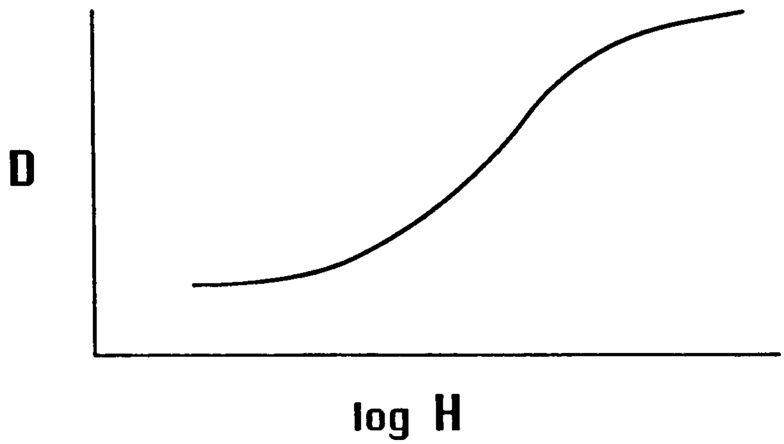


Figure 5. Characteristic Curve of the Negative Film: D-log H curve.

1.2 PROBLEM STATEMENT

1.2.1 Image Noise

Images, generated by different means, usually have various types of noise associated with them. For example, images recorded on film have grain noise; images recorded by solid state sensors have noise due to the random arrival of photons. In order to improve the visual quality of images and help the extraction of the image details, it is very desirable to remove noise in the images. Currently, a large number of images are recorded on films, which suffer from grain noise.

1.2.2 Photographic Grain Noise

When a uniformly exposed and processed film sample is scanned with a small aperture, there is a variation in density as a function of distance resulting from the discrete granular structure of the developed image. The number of grains (or dye clouds in color films) in a given area varies and gives rise to density fluctuations that can be seen as graininess although the individual grains (or dye clouds) are not resolved. In this thesis project, an orthonormal expansion method for digital image noise suppression is implemented. The major source of noise to be removed is film grain noise. The objective is to preserve as much sharpness and produce as few artifacts in the processed image as possible.

1.3 PREVIOUS WORK

Digital image processing has been studied for many years. Extensive discussions can be found in several

books [2-5].

1.3.1 Basic Operations for Digital Image Processing

There are several commonly used basic operations in digital image processing. Convolution, spatial filtering, and edge detection are several examples of such operations.

Convolution is usually done by calculating a weighted average of a picture element (pixel) and its surrounding neighbors. The weighting factors for the center and the neighboring pixels are called convolution coefficients. These coefficients form the so called convolution mask.

Spatial filtering implies the separation of frequency components within a two-dimensional image. The frequency components are spatial frequencies which relate to the rate of changes in gray levels over a certain spatial distance. According to the Fourier transform concept, an image can be decomposed as the sum of sinusoidal functions of different spatial frequencies. Modification of the magnitudes of certain frequency components of an image will change its apperance. Specifically, if the low frequency components are attenuated relative to the high frequency components, then we have a high-pass filter which will enhance the image details as well as noise. On the other hand, if the high frequency components are attenuated relative to the low frequency components, then we have a low-pass filter which will blur the image.

Edge detection is to find the pixel locations in the image where the rate of change in image intensity is large, indicating the end of one region and the beginning of another. Such a discontinuity is called an edge. In certain applications, the orientation and the contrast information of an edge are also computed.

1.3.2 Previous Work on Noise Suppression

The most popular approaches in the past to this problem employ frequency domain techniques [3-5,7-8], or weighted averaging in spatial domain [9-12]. Basically, in all these approaches the idea is to smooth out the intensity fluctuation due to noise. The smoothing operation can be done by low-pass filtering in frequency domain, or by averaging in spatial domain. The major difficulty is that edges in the images tend to be blurred.

II. PROJECT DESCRIPTION

2.1 THEORETICAL AND CONCEPTUAL DEVELOPMENT

Andrews and Patterson [13] proposed an image enhancement method based on singular value decompositions. If an image is treated as a K x K matrix A, then from the singular value decomposition (SVD) theorem, A can be decomposed as a sum of K outer products,

$$A = \sum_{i=1}^{k} a_i U_i V_i^T \qquad (1)$$

where U_i and V_i are, respectively, the i-th left singular vector and the i-th right singular vector, and a_i's are the singular values of A. The matrix $U_i V_i^T$ is called an *eigenimage*. They suggested the following formula for image enhancement:

$$A' = \sum_{i=1}^{k} b_i a_i U_i V_i^T \qquad (2)$$

where b_i's are scaling coefficients, which can be used to provide the enhancement effect. The following functional forms of the b_i's were proposed:

1. b_i is a linear function of i
2. b_i is a power function of a_i

These two functions are usually empirical in nature. However, other scaling functions can also be derived from theoretical criteria such as minimum mean square error. It is known [11, 12, 14] that the scaling function of least square estimation is

$$\frac{\sigma_{s+n}^2 \qquad \sigma_n^2}{\sigma_{s+n}^2} \quad , \qquad (3)$$

where σ_n^2 is the variance of the noise, and σ_{s+n}^2 is the variance of the observed values, i.e. signal plus noise. In this thesis, equation (3) is used to obtain the coefficients b_i in equation (2). (For more details, see equation (9) in section 2.3.) Furthermore, an image can be divided into many small blocks, and singular value decomposition is applied to each block. The computational load will be reduced by significant amount. Since it is the least square estimate of the noiseless signal, the resulting image should have less perceptible noise.

2.2 SINGULAR VALUE DECOMPOSITION (SVD)

2.2.1 Singular Value Decomposition Theorem

Singular Value Decomposition (SVD) is one of the most important decompositions in matrix computations. According to the Singular Value Decomposition theorem [15], if a matrix A is a real valued m x n matrix then there exist U and V which are orthogonal matrices of dimensions mxm and nxn, respectively.

such that

$$U^T A V = D, \qquad (4)$$

where D is a diagonal matrix with elements $a_1 \geq a_2 \geq \ldots \geq a_p \geq 0$, and $p = \min[m,n]$.

($a_1, a_2, \ldots$ a_p are called the singular values of matrix A.)

Recall that U and V are orthogonal means that $U^T U = I$ and $V^T V = I$.

Thus, from equation (4), $$U U^T A V V^T = U D V^T \qquad (5)$$

If U_i is the ith column of U, $1 \leq i \leq m$ and V_i is the ith column of V, then

from equations (5),

$$A = U D V^T$$

$$= \sum_{i}^{p} a_i U_i V_i^T \ . \qquad (6)$$

U_i and V_i are called the left and the right singular vectors of matrix A.

2.2.2 Computing Singular Value Decomposition

The following shows how the Singular Value Decomposition can be computed from the eigenvectors and eigenvalues of AA^T and $A^T A$:

$$AA^T = (U D V^T)(V D^T U^T) = U D D^T U^T \qquad (7)$$

$$A^T A = (V D^T U^T)(U D V^T) = V D^T D V^T \qquad (8)$$

From equations (7) and (8), U_i's and V_i's are the eigenvectors of matrices AA^T and A^TA respectively, and the eigenvalues of AA^T are the squares of the corresponding singular values of matrix A (see [16]).

2.2.3 An Example of SVD of a Matrix

$$A = \begin{pmatrix} 0.96 & 1.72 \\ 2.28 & 0.96 \end{pmatrix}$$

$$B = A\,A^T = \begin{pmatrix} 0.96 & 1.72 \\ 2.28 & 0.96 \end{pmatrix} \begin{pmatrix} 0.96 & 2.28 \\ 1.72 & 0.96 \end{pmatrix}$$

$$= \begin{pmatrix} 3.88 & 3.84 \\ 3.84 & 6.12 \end{pmatrix}$$

$$C = A^T A = \begin{pmatrix} 0.96 & 2.28 \\ 1.72 & 0.96 \end{pmatrix} \begin{pmatrix} 0.96 & 1.72 \\ 2.28 & 0.96 \end{pmatrix}$$

$$= \begin{pmatrix} 6.12 & 3.84 \\ 3.84 & 3.88 \end{pmatrix}$$

To find eigenvalue x of matrix B:

$$\begin{vmatrix} 3.88 - x & 3.84 \\ 3.84 & 6.12 - x \end{vmatrix} = 0$$

Solving the above quadratic equation, we find the eigenvalues as

$$x_1 = 9$$
$$x_2 = 1$$

Then, for the eigenvalue $x_1=9$, the eigenvector U_1 can be found by solving the following equations:

$$\begin{pmatrix} 3.88-9 & 3.84 \\ 3.84 & 6.12-9 \end{pmatrix} \begin{pmatrix} u_x \\ u_y \end{pmatrix} = 0$$

If we normalize the eigenvector U_1, i.e., let $u_x{}^2 + u_y{}^2 = 1$ then

$$U_1 = \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix}$$

Similarly, for the eigenvalue $x_2=1$, the eigenvector U_2 is

$$U_2 = \begin{pmatrix} -0.8 \\ 0.6 \end{pmatrix}$$

Following the same procedure for matrix C, we can find eigenvectors V_1 and V_2 as:

$$V_1 = \begin{pmatrix} 0.8 \\ 0.6 \end{pmatrix} \qquad V_2 = \begin{pmatrix} 0.6 \\ -0.8 \end{pmatrix}$$

The singular values σ_1 and σ_2 of matrix A are the square roots of the eigenvalues x_1 and x_2, therefore,

$$\sigma_1 = 3$$
$$\sigma_2 = 1$$

The matrices U and V are

$$U = \begin{pmatrix} 0.6 & -0.8 \\ 0.8 & 0.6 \end{pmatrix} \qquad V = \begin{pmatrix} 0.8 & 0.6 \\ 0.6 & -0.8 \end{pmatrix}$$

Now,

$$U^T A V = \begin{pmatrix} 0.6 & 0.8 \\ -0.8 & 0.6 \end{pmatrix} \begin{pmatrix} 0.96 & 1.72 \\ 2.28 & 0.96 \end{pmatrix} \begin{pmatrix} 0.8 & 0.6 \\ 0.6 & -0.8 \end{pmatrix}$$

$$= \begin{pmatrix} 2.4 & 1.8 \\ 0.6 & -0.8 \end{pmatrix} \begin{pmatrix} 0.8 & 0.6 \\ 0.6 & -0.8 \end{pmatrix}$$

$$= \begin{pmatrix} 3 & 0 \\ 0 & 1 \end{pmatrix} = D ,$$

conforming equation (4).

$$U^TU = \begin{pmatrix} 0.6 & 0.8 \\ -0.8 & 0.6 \end{pmatrix} \begin{pmatrix} 0.6 & -0.8 \\ 0.8 & 0.6 \end{pmatrix}$$

$$= \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix} = I ,$$

and

$$V^TV = \begin{pmatrix} 0.8 & 0.6 \\ 0.6 & -0.8 \end{pmatrix} \begin{pmatrix} 0.8 & 0.6 \\ 0.6 & -0.8 \end{pmatrix}$$

$$= \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix} = I .$$

The outer products of $U_1V_1^T$ and $U_2V_2^T$ can be used to reconstruct the matrix A as can be seen in the following steps:

$\sigma_1 * U_1 V_1^T + \sigma_2 * U_2 V_2^T$

$$= 3.0 * \begin{pmatrix} 0.6 \\ 0.8 \end{pmatrix} \begin{pmatrix} 0.8 & 0.6 \end{pmatrix} + 1.0 * \begin{pmatrix} -0.8 \\ 0.6 \end{pmatrix} \begin{pmatrix} 0.6 & -0.8 \end{pmatrix}$$

$$= 3.0 * \begin{pmatrix} 0.48 & 0.36 \\ 0.64 & 0.48 \end{pmatrix} + 1.0 * \begin{pmatrix} -0.48 & 0.64 \\ 0.36 & -0.48 \end{pmatrix}$$

$$= \begin{pmatrix} 1.44 & 1.08 \\ 1.92 & 1.44 \end{pmatrix} + \begin{pmatrix} -0.48 & 0.64 \\ 0.36 & -0.48 \end{pmatrix}$$

$$= \begin{pmatrix} 0.96 & 1.72 \\ 2.28 & 0.96 \end{pmatrix} = A .$$

2.2.4 Why Use the SVD for Digital Image Noise Suppression

If an image or part of it is treated as an array, from equation (1), the SVD decomposes it into its basic two dimensional components. Each component represents certain features of the image, and the magnitude of a singular value represents the 'energy' of that component. Andrews and Patterson [13] argued that the use of SVD in processing of images is the introduction of generalized filtering concepts in which various weight functions are applied to the eigenimages to provide an enhancement. The motivation for such a procedure might be the fact that the basis images are perfectly matched to the original image, hence modifying certain eigenimages will change specific structure inherent to original image alone. The following examples illustrate the idea.

Example 1:

A small image of a triangle pattern in front of a uniform background:

0.000	0.000	0.000	0.000	0.000
0.000	8.500	0.000	0.000	0.000
0.000	8.500	8.500	0.000	0.000
0.000	8.500	8.500	8.500	0.000
0.000	0.000	0.000	0.000	0.000

After SVD operations, the singular values (in descending order) are:

the first singular value = 19.099
the second singular value = 6.816
the third singular value = 4.717
the fourth singular value = 0.000
the fifth singular value = 0.000

The U matrix is :

0.000	0.000	0.000	0.000	1.000
-0.328	0.737	-0.591	0.000	0.000
-0.591	0.328	0.737	0.000	0.000
-0.737	-0.591	-0.328	0.000	0.000
0.000	0.000	0.000	1.000	0.000

The V matrix is:

0.000	0.000	0.000	0.000	1.000
-0.737	0.591	-0.328	0.000	0.000
-0.591	-0.328	0.737	0.000	0.000
-0.328	-0.737	-0.591	0.000	0.000
0.000	0.000	0.000	1.000	0.000

The corresponding eigenimages are:

The first eigenimage is

0.000	0.000	0.000	0.000	0.000
0.000	4.617	3.702	2.055	0.000
0.000	8.319	6.671	3.702	0.000
0.000	10.373	8.319	4.617	0.000
0.000	0.000	0.000	0.000	0.000

The second eigenimage is

0.000	0.000	0.000	0.000	0.000
0.000	2.969	-1.648	-3.702	0.000
0.000	1.321	-0.733	-1.648	0.000
0.000	-2.381	1.321	2.969	0.000
0.000	0.000	0.000	0.000	0.000

The third eigenimage is

0.000	0.000	0.000	0.000	0.000
0.000	0.914	-2.055	1.648	0.000
0.000	-1.140	2.562	-2.055	0.000
0.000	0.507	-1.140	0.914	0.000
0.000	0.000	0.000	0.000	0.000

The fourth eigenimage is

0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000

The fifth eigenimage is

0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000

The reconstructed image is:

0.000	0.000	0.000	0.000	0.000
0.000	8.500	0.000	0.000	0.000
0.000	8.500	8.500	0.000	0.000
0.000	8.500	8.500	8.500	0.000
0.000	0.000	0.000	0.000	0.000

which is of course, identical to the original image.

Example 2:

The image of triangle pattern in example 1 is corrupted with noise:

0.100	0.000	0.000	0.000	0.000
0.000	8.500	0.000	0.200	0.000
0.000	8.600	8.400	0.000	0.100
0.000	8.500	8.500	8.300	0.000
0.000	0.100	0.000	0.000	0.000

After SVD operations, the singular values (in descending order) are:

the first singular value = 19.082
the second singular value = 6.653
the third singular value = 4.671
the fourth singular value = 0.100
the fifth singular value = 0.000

The U matrix is:

0.000	0.000	0.000	-1.000	0.000
-0.334	0.739	-0.585	0.000	0.012
-0.593	0.317	0.740	0.000	0.000
-0.732	-0.594	-0.333	0.000	0.000
-0.004	0.009	-0.007	0.000	-1.000

The V matrix is:

0.000	0.000	0.000	-1.000	0.000
-0.742	0.596	-0.307	0.000	0.000
-0.587	-0.359	0.725	0.000	-0.012
-0.322	-0.719	-0.616	0.000	0.012
-0.003	0.005	0.016	0.000	1.000

The corresponding eigenimages are:

The first eigenimage is

0.000	0.000	0.000	0.000	0.000
0.000	4.733	3.745	2.053	0.020
0.000	8.404	6.649	3.646	0.035
0.000	10.377	8.210	4.502	0.043
0.000	0.055	0.044	0.024	0.000

The second eigenimage is

0.000	0.000	0.000	0.000	0.000
0.000	2.929	-1.763	-3.536	0.023
0.000	1.257	-0.757	-1.517	0.010
0.000	-2.354	1.417	2.841	-0.019
0.000	0.035	-0.021	-0.043	0.000

The third eigenimage is

0.000	0.000	0.000	0.000	0.000
0.000	0.838	-1.981	1.682	-0.043
0.000	-1.061	2.508	-2.129	0.055
0.000	0.477	-1.127	0.957	-0.025
0.000	0.009	-0.022	0.019	0.000

The fourth eigenimage is

0.100	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000

The fifth eigenimage is

0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000

If we throw away the eigenimages with singular values less than 0.5 (thresholding technique), we obtain the reconstructed image:

0.000	0.000	0.000	0.000	0.000
0.000	8.500	0.000	0.200	0.000
0.000	8.600	8.400	0.000	0.100
0.000	8.500	8.500	8.300	0.000
0.000	0.100	0.000	0.000	0.000

Compared with the original noisy image, the reconstructed image has less noise - the noise in the upper left corner is suppressed. In this case, the matrix size is small and the noise suppression effect is correspondingly weak. In the experiments to be described later, the matrix size is increased to 16 by 16, and the effect of noise suppression is better. Another problem with this example is that the thresholding scheme is not a continuous operation. Noise below the threshold is removed but noise above the threshold remains. In the following section, scaling instead of thresholding would be used in the experiments.

2.3 DESIGN OF EXPERIMENTS

In section 2.1, the conceptual foundation of using SVD for noise suppression was presented. The entire input image was treated as a big matrix which was decomposed into the sum of outer products of its singular vectors (see equation 1). The coefficient of each term was scaled by a function shown in equation (3), and the output noise-suppressed image was reconstructed by equation (2). Since a typical digital image has several hundreds of pixels in either dimension, the SVD of an entire image will take many days of computer time, not mentioning the memory size problem.

In order to speed up the computation, an input image is divided into many small non-overlapping k by k blocks. Each block is treated as a matrix which is then decomposed as the sum of the outer products of its left and right singular vectors, as shown in equation (1). As discussed in section 2.1, the scaling factors, b_i , as suggested by the least-squared-error estimation, should have the following form:

$$b_i = \begin{cases} \dfrac{a_i^2 - \sigma_i^2}{a_i^2}, & \text{if } a_i^2 > \sigma_i^2 \\ 0, & \text{if } a_i^2 <= \sigma_i^2 \end{cases} \tag{9}$$

where σ_i's are constants to be estimated from the noise source and properly adjusted to produce the images of the best visual quality. The basic idea underlying equation (9) is to scale each term according to the amount of the contribution of the signal to the singular value of that term. This is so because a_i^2 represents the signal plus noise power. The numerator a_i^2 σ_i^2 therefore represents the signal power.

The proposed implementation for the above noise suppression algorithm will be done according to the following sequence of experiments.

1. Select a reasonable block size, considering the computing time and algorithm performance.
2. Let noise standard deviation measured on the grain patch be σ_n. Assuming that the noise power

presents in each eigenimage is a constant, i.e.,

$$\sigma_1 = \sigma_2 = \ . \ . \ . \ = \sigma_k = C \ .$$

The constant C is then changed from σ_n to 10 σ_n to see the effect of noise suppression .

3. $\sigma_1, \sigma_2, \ . \ . \ . \ , \sigma_k$ are set to be the averaged singular values of the image of pure noise source, and the image is processed using equations (1), (2) and (9).

4. Since the input image is sectioned into non-overlapping blocks which are processed separately, the block boundaries may show up in the output image. This artifact is called the blocking artifact (e.g., see Fig. 13). One approach to solving this problem is to apply the SVD only to the high frequency components of the input image (see Fig. 6). A low-pass filter, Gaussian convolution mask, is convolved with the input image to smooth out the noisy image. The mask is generated by truncating the Gaussian function at four standard deviations in both x and y directions. The high-pass image is obtained by subtracting the low-pass image from the input noisy image. The standard deviation of the Gaussian convolution mask determines the frequency bandwidth of the low-pass filter. The smaller the standard deviation is, the wider the frequency bandwidth will be and the more the noise will pass through. The choice of the bandwidth of a low-pass filter determines the visibility of the noise. In general, the low-pass filter smooths out the noise, but also smears the image details. The optimal trade-off is determined experimentally. Experiments therefore include:

   a. the determination of the optimal low-pass filter, e.g., select standard deviation = 2, or 3, or 4 pixels for Gaussian convolution mask, and,

   b. the determination of the optimal scaling factors.

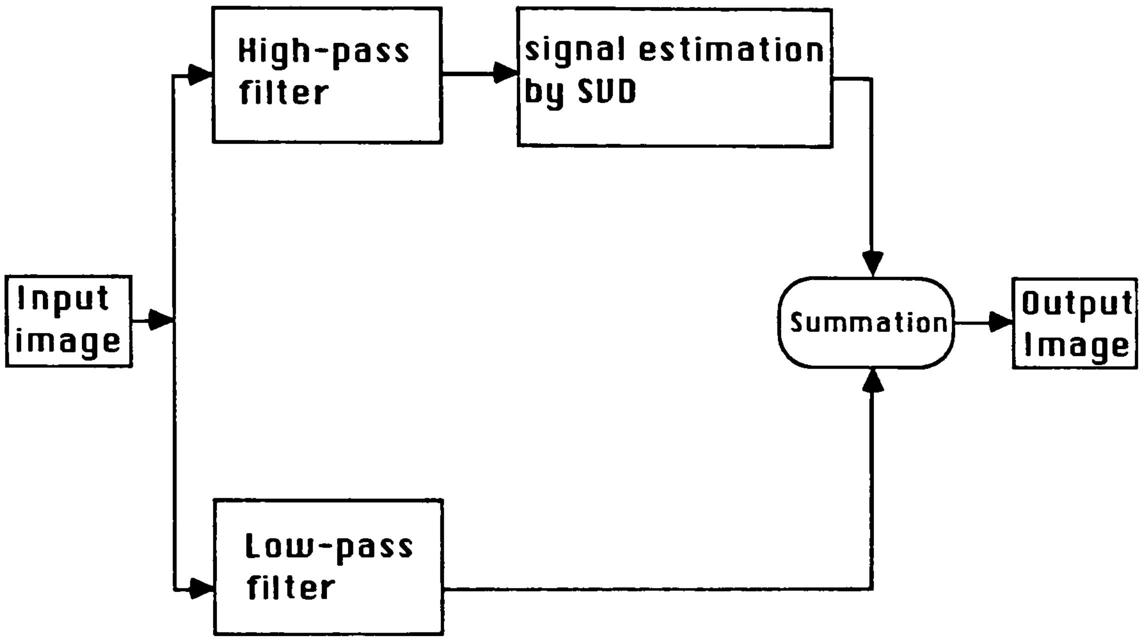


Figure 6. SVD Operations with High Frequency Components of Input Images.

III. IMPLEMENTATION AND EXPERIMENTAL RESULTS

3.1 CHARACTERIZATION OF NOISE

3.1.1 Preparation of Uniformly Exposed Grain Patch

When a patch of film is exposed with a uniform field of light, and then chemically processed and developed, we have a piece of developed film containing only the film grain noise without any other signal. This uniformly exposed piece of developed film is usually called a grain patch, because it is used to characterize the grain noise for the type of film. The uniform exposure is usually done with an instrument called a sensitometer.

Eastman Kodak VR disc film generation III was used for the grain patch. A grain patch prepared from a given type of film can be scanned to provide a digital image of pure grain noise, which we will call the noise image. The noise image used in this project has the size of 400 by 400 pixels. The exposure of the grain patch was chosen to produce a negative density about 1.3 which represents a density level with medium noise variation (see next section for mean density values).

3.1.2 Measurement of Noise Characteristics

(1) Standard Deviation

The film grain noise is usually modeled as having a Gaussian distribution in the optical density space.

Therefore, the standard deviation of the noise image is the most often used parameter for characterizing the film grain noise. The following show the measurements of the noise image we use:

	RED	GREEN	BLUE
MEAN DENSITY (Status M)	1.055	1.3096	1.7778
STANDARD DEVIATION	0.0395	0.0409	0.0787

(Status M refers to a set of standardized spectral responses.)

(2) Singular Values

Since we propose to use the singular value decomposition for grain noise suppression, SVD was also applied to the noise image to estimate the distribution of the singular values of pure grain noise. (The SVD subroutine was taken from the book listed as reference [17].) The following are the measurements of the noise image used (the block size was chosen to be 16 x 16 as discussed in the next section):

RED SINGULAR VALUES: (16X16 SVD, AVERAGE OF 625 BLOCKS)

0.3617922	0.2949340	0.2438512	0.2009201
0.1642079	0.1327216	0.1065034	0.8368029E-01
0.6510691E-01	0.5162595E-01	0.4033887E-01	0.3101235E-01
0.2300261E-01	0.1599143E-01	0.9144974E-02	0.3016687E-02

THE VARIANCES OF RED SINGULAR VALUES:

0.1732443E-02	0.1129662E-02	0.7863222E-03	0.5989745E-03
0.4206558E-03	0.3350226E-03	0.2614392E-03	0.1752377E-03
0.1069710E-03	0.6478580E-04	0.4199932E-04	0.2993453E-04
0.2049530E-04	0.1615168E-04	0.1108895E-04	0.5519035E-05

GREEN SINGULAR VALUES:

0.3443733	0.2855030	0.2426044	0.2054947
0.1749011	0.1468179	0.1217875	0.1000684
0.8056251E-01	0.6349075E-01	0.4948781E-01	0.3710322E-01
0.2676791E-01	0.1769640E-01	0.9684031E-02	0.3054531E-02

THE VARIANCES OF GREEN SINGULAR VALUES:

0.1185573E-02	0.6290656E-03	0.4819608E-03	0.3640908E-03
0.2940365E-03	0.2486450E-03	0.1768705E-03	0.1453669E-03
0.1158462E-03	0.8431292E-04	0.5829547E-04	0.3992549E-04
0.2872598E-04	0.1983629E-04	0.1320877E-04	0.5872684E-05

BLUE SINGULAR VALUES:

0.6870323	0.5659634	0.4743122	0.4025620

0.3383419	0.2837123	0.2353573	0.1927931
0.1549102	0.1230870	0.9445686E-01	0.6984116E-01
0.5036274E-01	0.3321106E-01	0.1841496E-01	0.5944110E-02

THE VARIANCES OF BLUE SINGULAR VALUES:

0.5160239E-02	0.3207810E-02	0.2055362E-02	0.1613628E-02
0.1134218E-02	0.9122007E-03	0.7212870E-03	0.5734316E-03
0.4595025E-03	0.3740422E-03	0.2462001E-03	0.1696793E-03
0.1196575E-03	0.7052891E-04	0.5056098E-04	0.2232813E-04

It is clear that the variation of singular values is very small (less than 1%). Therefore, their averages alone can be used.

3.2 EXPERIMENTS AND RESULTS

A high quality, professionally photographed picture was used as the original image. The image was scanned by an Optronics scanner. The resulted digital image is very smooth and clean with 1136 by 1600 pixels (Fig. 7). To simulate the film grain noise, a uniformly exposed piece of developed film was scanned to produce the noise-only image. Adding the noise-only image to the original clean image produced the noise corrupted image shown in Figure 8. The graininess of this image did not look noisy enough for the experiment. Therefore, the noise amplitudes were increased by a factor of 2 and added to the original clean image to create a more noisy image as shown in Figure 9, which is used as the input image for all the experiments.

The input image is partitioned into non-overlapping blocks. The size of each block is 16 pixels by 16 pixels. The selection of the block size is a compromise between the computation time and the algorithm performance. A large size block will take a long time to process, while a small size block does not give good noise suppression. On the other hand, there is a range of block size which will give similar results. The choice has been made based on the consideration of efficiency and performance.

The noise suppressed digital images are printed directly on photographic paper by a CRT output scanner. Since the grain noise of photographic paper is very small (in most cases, invisible), the photographic prints show the processed images faithfully without any added noise from the printing process. It should be pointed out that if one is to print the noise-reduced digital images on film and then print on paper, one has to choose films with very fine grains. Otherwise, additional grain noise will be introduced, complicating the evaluation procedure.

3.2.1 Experiments with Scaling Factors using Constant Noise Contribution

Assuming that the noise contribution to each singular value is constant, the standard deviation of the noise-only image was used as the constant and subtracted from each singular value to calculate the scaling factors used in the SVD process, i.e. $b_i = (a_i^2 \quad c^2) / a_i^2$, $c = \sigma_n$, where σ_n is the standard deviation of the grain patch pure noise. The processed output image shown in Figure 10 appears to has as much graininess as the input image shown in Figure 9. Other experiments, therefore, were done to see how the graininess could be removed by increasing the magnitude of the noise constant. Figure 11 and Figure 12 showed output images with scaling factors $b_i = (a_i^2 - c^2) / a_i^2$, $c = 2\sigma_n$ and $b_i = (a_i^2 \quad c^2) / a_i^2$, $c = 3\sigma_n$ respectively. The graininess still can be seen in Figure 11 and Figure 12. If the scaling factor is

modified to be $b_i = (a_i^2 - c^2) / a_i^2$, $c = 10\sigma_n$, the blocking artifacts showed up seriously in the processed output image as shown in Figure 13 even though most noise was removed .

3.2.2 An Experiment with Scaling Factors using Averaged Noise Singular Values

From the experiments described in section 3.2.1, it seems that the scaling factors using constant noise contribution tend to remove too much signal in the eigenimages which have small singular values, and too little noise in the eigenimages which have large singular values. A possible improvement is to use averaged singular values of the noise-only image to replace the noise constant in the scaling factor. The scaling factor now become: $b_i = (a_i^2 - \sigma_i^2) / a_i^2$, where σ_i is the averaged singular value of the grain patch pure noise. The resulted output image is shown in Figure 14. The blocking artifact is still objectionable. The only visible improvement seems to be in the region of the parsley leaves.

3.2.3 Experiments with Low-pass / High-pass Filters and Different Scaling Factors

A close examination of the blocking artifact seems to indicate that it is caused by the brightness discontinuities in the image, which are created by the different modification of the eigenimages in the neighboring blocks. If the continuity of the slowly changing (low frequency) component in the brightness variations is preserved, the blocking artifact could be significantly reduced. Based on this idea, it was proposed in section 2.3 that the input image be filtered through a low-pass and a high-pass filter to separate the low frequency and the high frequency components of the image. The low-pass filtered image contains only low frequency noise, which is less visible. The SVD noise suppression algorithm is applied to high-pass filtered image only (see Figure 6).

(A) The Determination of the Optimal Low-pass Filter Size

The low-pass filter used in the experiments is a Gaussian filter with different standard deviations. Since the low-pass filtered image will not be processed by the SVD noise suppression algorithm, it should have as little perceptible noise as possible. Different sizes of the Gaussian convolution mask are applied to the input noisy image to determine the size of the optimal low-pass filter mask. Figures 15, 16, and 17 show the images processed through low-pass filters with mask sizes 17 x 17 pixels, 25 x 25 pixels, and 33 x 33 pixels respectively. Figure 17, even though the most blurry , has the least perceptible noise. Therefore, the 33 x 33 mask size will be used in the experiments for determining the optimal scaling factors to be described later.

Figure 18 shows the high-pass filtered image before the SVD noise suppression algorithm is applied. The grainy noise is clearly visible in the image. Figure 19 shows the same image after the SVD noise suppression algorithm is applied and it appears that most of the noise has been removed. (One word about the color of these high-pass filtered images is worth mentioning here: image regions where there are no color edges have little high frequency color information, and therefore, those regions in the high-pass filtered image have near zero values for red, green, blue signals resulting in grayish colors.)

Figures 20, 21, and 22 show the output images which are the combination of the low-pass filtered image with the corresponding noise suppressed high-pass image, each having different Gaussian convolution mask size: 17 x 17, 25 x 25, and 33 x 33 pixels respectively. In all three images the SVD noise suppression algorithm used the following scaling factor: $b_i = (a_i^2 - \sigma_i^2) / a_i^2$, where σ_i is the averaged singular value of the grain patch pure noise. The trade-off between sharpness and noise suppression can be easily seen in these three images. Figure 20 has the most noticeable noise but also the

highest sharpness, while Figure 22 has the opposite result.

(B) The Determination of the Optimal Scaling Factors

In all the experiments described so far, the noise suppression algorithm has used scaling factors of the following form: $b_i = (a_i^2 \quad \sigma_i^2) / a_i^2$, where σ_i represents the noise contribution to the observed singular value of the image, a_i. The physical meaning of the above scaling factors can be made more clear by rewriting them in the following form: $b_i = 1 \quad (\sigma_i / a_i)^2$. If σ_i / a_i is much smaller than 1, then the scaling factor b_i is very close to 1, which means the ith eigenimage has very little noise in it and this eigenimage component should not be modified much. If σ_i / a_i is close to 1, then the scaling factor b_i is very close to 0, which means the ith eigenimage has very little signal in it and the contribution of this eigenimage component to the reconstructed image should be significantly reduced. From the above discussion, several forms of the scaling factor seem to have the potential for improving the trade-off between sharpness and noise suppression.

1) $b_i = g_i [1 \quad (\sigma_i / a_i)^n]$, $n > 2$; (12)

2) thresholding: $b_i = 0.0$, if $a_i < 1.2\,\sigma_i$; $b_i = 1.0$ otherwise. (13)

Using equation (12) with $n = 4$, $g_i = 1.0$. the resulted output image shown in Figure 23 seems to have sharper details but also more visible artifacts. Using equation (12) with $n = 4$, $g_i = 1.2$, the resulted output image shown in Figure 24 seems to be even sharper in details and also worst in terms of visible artifacts. Using equation (13), the resulted output image shown in Figure 25 obviously has many more artifacts due to the discontinous nature of the thresholding. The best compromise from all the

experiments is to use equation (12) with $n = 3$, $g_i = 1.0$. The resulted output image is shown in Figure 26. To get a direct comparison between the noisy input image and the noise suppressed output image, Figure 9 and Figure 26 are shown side by side in Figure 27. It is clear that the algorithm proposed in this thesis is quite effective in suppressing the noise, but the loss of sharpness due to the algorithm is also quite noticeable.

IV CONCLUSION AND DISCUSSION

In this thesis, a digital image noise suppression algorithm based on Singular Value Decomposition has been proposed. An input image is partitioned into non-overlapping blocks. The size of each block is 16 x 16 pixels. Each block is treated as a matrix, A, which is operated by Singular Value Decompositon:

$$A = \sum_{i=1}^{k} a_i \; U_i \; V_i^T , \qquad \text{where } a_i\text{'s are the singular values of } A.$$

The algorithm for image noise suppression modifies each term by multiplying a_i with a scaling factor b_i, and then reconstruct the matrix by summning up all the modifed terms, i.e.

$$A' = \sum_{i=1}^{k} b_i \; a_i \; U_i \; V_i^T .$$

One of the major problems encountered early in the experiments was the blocking artifact due to the brightness discontinuities created by the algorithm because each block of the image was processed independently. The solution proposed in this thesis to eliminate the blocking artifact was to separate the low frequency and the high frequency components of the input image, and apply the SVD noise suppression algorithm only on the high frequency component. The output image is the sum of the low frequency component and the noise suppressed high frequency component of the input image.

An experiment was conducted to determine the optimal mask size for the low-pass filter, so that the low-pass filtered image has the least perceptible noise and still preserves enough brightness continuity to avoid the blocking artifact. It is concluded that a Gaussian filter with mask size of 33 x 33 pixels and

standard deviation of 4 pixels produces the best result.

Many experiments were conducted to determine the optimal scaling factors for modifying the coefficients of the eigenimages, so that the processed image has the best visual quality in terms of the trade-off between the preserved sharpness and the residual noise. It is concluded that the following scaling factors produce the best result: $b_i = (a_i^3 - \sigma_i^3) / a_i^3$, where σ_i is the averaged singular value of the pure noise source.

The effectiveness of the proposed noise suppression algorithm can be seen in Figure 27 which shows the comparison between the input noisy image and the output noise suppressed image. The loss of image details, although not serious, is obviously noticeable. Further experiments need to be done to increase the sharpness of the noise suppressed image. For example, techniques like the unsharp masking [10] can be used to enhance the image sharpness.

Figure 7. The original image of a high quality, professionally photographed picture which was scanned by an Optronics scanner. The image size is 1136 by 1600 pixels.

Figure 8. The noise corrupted image which was produced by adding the film grain noise to the original clean image.

Figure 9. The noise corrupted image which was produced by adding two times of the film grain noise to the original clean image. The grain noise amplitudes were increased by a factor of 2. This image is used as the input image for all the experiments.

Figure 10. The output image which is processed by assuming that the noise contribution to each singular value is constant. The scaling factor used is $b_i = (a_i^2 - c^2) / a_i^2$, $c = \sigma_n$, where σ_n is the standard deviation of the pure noise grain patch.

Figure 11. The output image which is processed by assuming that the noise contribution to each singular value is constant. The scaling factor used is $b_i = (a_i^2 \quad c^2) / a_i^2$, $c = 2\,\sigma_n$, where σ_n is the standard deviation of the pure noise grain patch.

Figure 12. The output image which is processed by assuming that the noise contribution to each singular value is constant. The scaling factor used is $b_i = (a_i^2 - c^2) / a_i^2$, $c = 3\sigma_n$, where σ_n is the standard deviation of the pure noise grain patch.

Figure 13. The output image which is processed by assuming that the noise contribution to each singular value is constant. The scaling factor used is $b_i = (a_i^2 \quad c^2) / a_i^2$, $c = 10\, \sigma_n$, where σ_n is the standard deviation of the pure noise grain patch.

Figure 14. The output image which is processed by applying the following scaling factor: $b_i = (a_i^2 - \sigma_i^2) / a_i^2$, where σ_i is the averaged singular value of the pure noise source.

Figure 15. The low-pass filtered image which is to be combined with the noise suppressed high-pass image (see Figure 6). The Gaussian convolution mask size is 17 x 17 pixels with standard deviation equal to 2 pixels.

Figure 16. The low-pass filtered image which is to be combined with the noise suppressed high-pass image (see Figure 6). The Gaussian convolution mask size is 25 x 25 pixels with standard deviation equal to 3 pixels.

Figure 17. The low-pass filtered image which is to be combined with the noise suppressed high-pass image (see Figure 6). The Gaussian convolution mask size is 33 x 33 pixels with standard deviation equal to 4 pixels.

Figure 18. The high-pass filtered image with noticeable grainy noise before SVD process, where the high-pass filtered image was produced by subtracting the low-pass filtered image from the noisy input image (see Figure 6). The Gaussian convolution mask size used for the low-pass filtered image is 33 x 33 pixels with standard deviation equal to 4 pixels.

Figure 19. The high-pass filtered image with grainy noise removed after SVD process, where the high-pass filtered image was produced by subtracting the low-pass filtered image from the noisy input image (see Figure 6). The Gaussian convolution mask size used for the low-pass filtered image is 33 x 33 pixels with standard deviation equal to 4 pixels.

Figure 20. The output image which is the combination of the low-pass filtered image with the noise suppressed high-pass image (see Figure 6). The Gaussian convolution mask size used for the low-pass filtered image is 17 x 17 pixels with standard deviation equal to 2 pixels. The SVD noise suppression algorithm here used the following scaling factor: $b_i = (a_i^2 - \sigma_i^2) / a_i^2$, where σ_i is the averaged singular value of the pure noise source.

Figure 21. The output image which is the combination of the low-pass filtered image with the noise suppressed high-pass image (see Figure 6). The Gaussian convolution mask size used for the low-pass filtered image is 25 x 25 pixels with standard deviation equal to 3 pixels. The SVD noise suppression algorithm here used the following scaling factor: $b_i = (a_i^2 - \sigma_i^2) / a_i^2$, where σ_i is the averaged singular value of the pure noise source.

Figure 22. The output image which is the combination of the low-pass filtered image with the noise suppressed high-pass image (see Figure 6). The Gaussian convolution mask size used for the low-pass filtered image is 33 x 33 pixels with standard deviation equal to 4 pixels. The SVD noise suppression algorithm here used the following scaling factor: $b_i = (a_i^2 - \sigma_i^2) / a_i^2$, where σ_i is the averaged singular value of the pure noise source.

Figure 23. The output image which is the combination of the low-pass filtered image with the noise suppressed high-pass image (see Figure 6). The Gaussian convolution mask size used for the low-pass filtered image is 33 x 33 pixels with standard deviation equal to 4 pixels. The SVD noise suppression algorithm here used the following scaling factor: $b_i = (a_i^{\,4} - \sigma_i^{\,4}) / a_i^{\,4}$, where σ_i is the averaged singular value of the pure noise source.

Figure 24. The output image which is the combination of the low-pass filtered image with the noise suppressed high-pass image (see Figure 6). The Gaussian convolution mask size used for the low-pass filtered image is 33 x 33 pixels with standard deviation equal to 4 pixels. The SVD noise suppression algorithm here used the following scaling factor: $b_i = 1.2 \; (a_i^{\,4} - \sigma_i^{\,4}) \, / \; a_i^{\,4}$, where σ_i is the averaged singular value of the pure noise source.

Figure 25. The output image which is the combination of the low-pass filtered image with the noise suppressed high-pass image (see Figure 6). The Gaussian convolution mask size used for the low-pass filtered image is 33 x 33 pixels with standard deviation equal to 4 pixels. The SVD noise suppression algorithm here used the following thresholding formula for the scaling factor b_i :

$b_i = 0.0$, if $a_i < 1.2\,\sigma_i$; $b_i = 1.0$ otherwise, where σ_i is the averaged singular value of the pure noise source.

Figure 26. The output image which is the combination of the low-pass filtered image with the noise suppressed high-pass image (see Figure 6). The Gaussian convolution mask size used for the low-pass filtered image is 33 x 33 pixels with standard deviation equal to 4 pixels. The SVD noise suppression algorithm here used the following scaling factor: $b_i = (a_i^3 \quad \sigma_i^3) / a_i^3$, where σ_i is the averaged singular value of the pure noise source.

(a) (b)

Figure 27. (a) The noise corrupted input image (same as Figure 9).
(b) The noise suppressed output image (same as Figure 26).

REFERENCES

1. D. Cassasent, "Pattern recognition: a review," IEEE Spectrum, Vol. 18, No. 3, pp. 28-33, 1981.

2. G.A. Baxes, *Digital Image Processing*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1984.

3. A. Rosenfeld and A.C. Kak, *Digital Picture Processing*, Academic Press, Inc., New York, 1976.

4. W.K. Pratt, *Digital Image Processing*, John Wiley & Sons, New York, 1978.

5. H.C. Andrews and B.R. Hunt, *Digital Image Restoration*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1977.

6. B.H. Carroll, G.C. Higgins, and T.H. James, *Introduction to Photographic Theory*, John Wiley & Sons, New York, 1980.

7. W.K. Pratt, "Generalized Wiener filtering computation techniques," IEEE Trans. on Computers, Vol. C-21, No. 7, pp. 636-641, 1972.

8. H. Thiry, "Some qualitative and quantitative results on spatial filtering of granularity," Applied Optics, Vol. 3, No. 1, pp. 39-43, 1964.

9. T. Peli and J.S. Lim, "Adaptive filtering for image enhancement," Optical Engineering, Vol. 21, No. 1, pp. 108-112, 1982.

10. R.G. Powell, B.E. Bayer, "A method for the digital enhancement of unsharp, grainy photographic images," Proceedings of the International Conference on Electronic Image Processing, University of York, U.K., pp. 179-181, July, 1982.

11. J.S. Lee, "Digital image enhancement and noise filtering by use of local statistics," IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI-2, No. 2, pp. 165-168, 1980.

12. R. Bernstein, "Contrast enhancement of noisy images by signal adaptive two-dimensional filtering," Proceedings of the International Conference on Digital Signal Processing, Florence, Italy, pp. 519-523, September, 1984.

13. H.C. Andrews and C.L. Patterson, "Singular value decomposition and digital image processing," IEEE Trans. on Acoustics, Speech, and Signal Processing, Vol. ASSP-24, No. 1, pp. 26-53, 1976.

14. A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, Inc, New York, 1965.

15. G.H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Maryland, 1983, pp. 16-20.

16. S. Lipschutz, *Theory and Problems of Linear Algebra*, Schaum's Outline Series, McGraw-Hill Book Company, New York, 1968, p. 200.

17. G.E. Forsythe, M.A. Malcolm, and C.B. Moler, *Computer Methods for Mathematical Computations*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632, 1977, pp. 229-235.

GLOSSARY

Brightness: The value associated with a pixel representing its gray value from black to white.

Convolution: The mathematical operation whereby image processes are implemented. See Section 1.3.1 (page 12).

Convolution coefficient: A numeric value defining the weight that a pixel within a convolution kernel takes on in an image process. See Section 1.3.1 (page 12).

Convolution kernel: The size of the pixel array used in the calculation of an output pixel in an image process. See Section 1.3.1 (page 12).

Density: A deposit (e.g. of silver or dye) formed in a photographic emulsion as a result of exposure and processing. The optical density is defined as the logarithm of the opacity.

Digitization: The act of sampling and quantizing an analog video signal.

Display: The means by which an image is viewed. A television monitor is a typical display device.

Edge enhancement: Any operation that accentuates edge details within an image.

Film grain noise: The grains of silver compose the developed image. The silver grains are randomly distributed with respect to their size and shape, they are randomly located in distance from one another in the film emulsion. They behave randomly under condition of exposure and development. This inherent randomness in silver grain formation results
in a type of noise referred to as film grain noise.

Gray level: The brightness value assigned to a pixel. A value may range from black, through the grays, to white.

high-pass filter: An image operation that enhances high spatial frequencies or attenuates low frequencies in an image. This operation is used to bring out details difficult to see in the original.

Image analysis: Any image operation intended to numerically tabulate some aspect of an image.

Image coding: Any image operation used to reduce the amount of data required to describe the content of an image.

Image operation: Any algorithm for evoking a quality enhancement, analysis, or coding upon an image.

Image process: Any method for implementing an image operation.

Low-pass filter: Any operation that enhances low spatial frequencies or attenuates high frequencies in an image. This operation is used to bring out elements of an image difficult to see in the original.

Pixel: The fundamental picture element of a digital image. Also, the coordinate used for defining the horizontal spatial location of a pixel within an image.

Quantization: The act of converting an analog pixel brightness to a digital quantity.

Raster scan: In raster scan display, the refresh memory is arranged as two- dimensional array. The entry at a particular row and column stores the brightness and/or color value of the corresponding(x,y) position on the screen in the simple one-to-one relationship, each screen location and menory location is referenced by an x-coordinate and a y-coordinate. The top row of memory corresponds to the top scan line, etc., the image refreshing is done by a sequential raster scan through the buffer by scan line.

Registration: In many image processing application it is necessary to form a pixel-by-pixel comparison of two images of the same object field obtained from different sensors, or of two images of an object field taken from the same sensor at different times. To form this comparison it is necessary spatially to register the images and thereby correct for relative translational shifts magnification differences, and rotational shifts, as well as geometrical and intensity distortions of each image.

Sampling: The chopping of the analog video signal into discrete pixels but not including the quantization process.

Spatial: Pertaining to the two-dimensional nature of an image.

Spatial filtering: The set of image operations allowing the attenuation or accentuation of spatial frequencies within an image. Such operations include low-pass and high-pass filtering.

Spatial frequency: The concept dealing with the rate of brightness change in an image. Brightness fluctuations occurring in close proximity to one another represent high spatial frequencies, whereas regions of relatively constant brightness represent low spatial frequencies.

Weighted average: The mathematical operation used in spatial convolution to compute the result of each output pixel based on an input pixel and its eight neighbors. Each pixel and its neighbors are multiplied by their respective convolution coefficients as defined in the convolution mask. The results are summed, yielding the weighted average.