Deblurring Gaussian Blur: Continuous and Discrete Approaches

Behjoo Kimia

B.ENG. (Honours)

Computer Vision and Robotics Laboratory

Department of Electrical Engineering

McGill University

Montréal, Québec, Canada

A thesis submitted to the Faculty of Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Engineering

February 1986

Deblurring Gaussian Blur: Continuous and Discrete Approaches

Behjoo Kimia
B ENG (Honours)

Abstract

Gaussian blur or degradation is the convolution of a signal or image against a Gaussian kernel Solutions to the problem of removing Gaussian blur are presented from two approaches First, in the continuous approach, the image is considered to be a function defined on a continuous domain The process of blurring is modelled as continuous convolution of data against the Gaussian kernel. In order to remove this blur a linear deblurring kernel is sought. Although the inverse of a Gaussian cannot be represented exactly as a convolution kernel in the spatial domain, by restricting the blurred data to polynomials of fixed degree, I show that a convolution inverse does exist. These deblurring kernels are the *pseudo-inverses* of the Gaussian convolution operator, and constructive formulas for the deblurring kernels in terms of Hermite polynomials are given. Second, from the discrete approach, the image is modelled as a matrix of discrete values. In contrast to the continuous case, the blurring process is now formulated as multiplication of a data matrix by the blur matrix. However, the resultant system of linear equations can not be solved using typical numerical methods, since the problem is ill-conditioned. The problem is solved by symbolically decomposing the inverse of the blur matrix so that all the numerically ill-conditioned terms are gathered into a diagonal matrix. The result permits exact and stable deblurring provided the extent of blurring is known a priori.

February 1986

Deblurring Gaussian Blur: Continuous and Discrete Approaches

Behjoo Kimia
B ENG (Honours)

Résumé

Le flou gaussien, ou dégradation d'une image, est le résultat de la convolution du signal avec une fonction noyau gaussienne. Des solutions au problème de l'élimination du flou gaussien sont présentées selon deux démarches. Selon la première démarche, l'image est définie comme une fonction continue sur un domaine continu. L'addition de flou est modélisée comme une convolution continue du signal avec le noyau gaussien. Dans le but d'éliminer ce flou, une fonction noyau linéaire est recherchée. Bien que l'inverse d'une convolution gaussienne ne peut pas être représentée de façon exacte comme un noyau de convolution sur l'espace, nous montrons qu'une telle convolution inverse existe si on représente le signal perturbé par des polynômes de degré fixe. Nous définissons ainsi des *pseudo-inverses* des opérateurs de convolution en vue de l'élimination du flou. Les termes de polynômes de Hermite sont explicités à l'aide de formules constructives. Du point de vue discret, l'image est modélisée par une matrice de valeurs. Par opposition au modèle continu, l'addition de flou est représentée par la multiplication d'une matrice de flou par la matrice d'image. Nous obtenons un système d'équations linéaires mal conditionné, et par conséquent, que l'on ne peut pas résoudre à l'aide de méthodes numériques usuelles. Le problème se résoud par une décomposition symbolique de l'inverse de la matrice de flou des manière à réunir les termes mal conditionnés dans une matrice diagonale. Nous pouvons ainsi obtenir une élimination exacte et stable du flou, pourvu que l'amplitude du flou soit connue à priori.

Fevrier 1986

Deblurring Gaussian Blur: Continuous and Discrete Approaches

Behjoo Kimia

B ENG. (Honours)

Acknowledgements

I would like to thank, first and foremost, my research supervisor, Dr. Steven Zucker, for being the perfect advisor. He has provided me with timely direction and focus, encouragement, and never ending enthusiasm. I have enjoyed a precious freedom of thought and choice of research direction while working on this project. I would also like to thank Yvan Leclerc, for his help, interest, and valuable critisism, as well as being a friend. I would like to appreciate the help of the people at the Computer Vision and Robotics Laboratory with different aspects of this thesis; notably, Pierre Parent for help with typesetting the thesis and Vincent Hayward for translating the abstract.

On the non-technical side, I would like to thank my immediate family for endless support they have provided me through the long distance between us. I would like to mention invaluable emotional support that my uncle's family, Morad, Phyllis, Dyan, and Rena Kimia, have provided me throughout my university education; they have really been my family away from home. I also like to thank the people in the Lab. for providing such a nice environment to be in.

The financial assistance of the Natural Sciences and Engineering Research Council through a two year Postgraduate Fellowship is greatly appreciated.

February 1986

Table of Contents

List of Figures

Preface

Gaussian blur is a most common phenomenon. yet to date no exact solution for removing it has been given. Most traditional methods have advocated Fourier transform and ad-hoc techniques. such as "enhancement filters". [Rosenfeld and Kak]. [Pratt]. However. in addition to having problems with numerical stability and singularity. these techniques do not remove the Gaussian blur exactly. In contrast. we will show how to remove Gaussian blur exactly both in the continuous and the discrete domains.

Gaussian blur occurs in many events of nature This is mainly due to an application of the central limit theorem: when a large number of random local degradations combine sequentially. the resulting degradation closely resembles a Gaussian. Natural examples include atmospheric and optical blur. Also. in computerzied tomography. the imaging and reconstruction processes introduce degradations which are approximately Gaussian. [Herman].

The human visual system. as well as other human sensory mechanisms. is a rich source of examples of Gaussian degradation. For instance. the image on the retina is projected by a lens that can only focus on one plane and for only one wavelength of the incoming light. Hence. the image of an object which is not on the focal plane is blurred. Another example illustrating Gaussian blur is the transmission of visual information by the optic tract. The distribution of the axonal diameter of the optic nerves is approximately Gaussian [Fukada]. Since the axonal diameter of the optic nerve directly determines the delay of arrival of information (at the LGN). the information is blurred The kernel of this blur can be approximated by a Gaussian.

These examples pose serious questions in understanding visual perception. Is Gaussian degradation of visual information. which is introduced by various sources. an important aspect of perception. or is it merely the practical and physical limit of the optics and the "wetware"? Furthermore. is such blur an undesirable feature or can it be used to extract useful information about the three dimensional world? Given the finite depth of field of the human eye. Helmholtz has suggested that knowledge of the amount of blur can be used to recover estimates of the depths of objects in the visual field [Helmoholtz], and Pentland has applied these techniques to range-finding practical robots [Pentland]. Further, hyperacuity, the ability to percieve spatial resolution an order of magnitude higher than retinal spacing, can not be explained without seriously considering the effect of blur on the visual information. [Westheimer]

Gaussian degradation manifests itself both in discrete and continuous processes. For instance, the measurement of temperature of an object is a continuous process. In this case the data and the blurred data are both continuous functions and the blurring process is modelled as the continuous convolution of the Gaussian against the data. However, in some other instances, the blurring process has a discrete nature, for example when either or both of the input and output are discrete functions. In such cases, a closer model for blur is the matrix multiplication of the Gaussian blur matrix and the data matrix.

The continuous modelling leaves us with the following integral equation to solve:

$$\begin{aligned} h(x) &= K(x,t) * f(x) \\ &= \int_{\mathbb{R}^n} K(x-\xi,t) f(\xi) d\xi, \end{aligned} \tag{I.1.1}$$

where

$$K(x,t) = \frac{1}{(4\pi t)^{n/2}} e^{-|x|^2/4t} \tag{I.1.2}$$

is the Gaussian kernel, whose extent is parameterized by $t > 0$. It is normalized to have unit mass.

The problem of Gaussian deblurring is to recover the original data $f(x)$, when *only* $h(x)$ and the amount of blurring t are known. We seek a solution in form of a filter $D(x,t)$ such that

$$\begin{aligned} f(x) &= D(x,t) * h(x) \\ &= D(x,t) * K(x,t) * f(x), \end{aligned} \tag{I.1.3}$$

for $f(x)$ among a class of functions. This filter, defined in equation I.5.9, resembles the difference of Gaussians operator (DOG), but with extra side lobes; see figure (0.a). These side lobes are an indication of notions of approximation and order in the theory, and increase in number as the dimension of the space of allowable data increases; see figure I.1.c.

The discrete problem is one of solving the following linear system of equations

$$h = Bf. \qquad (II.2.2)$$

where,

$$B \triangleq \begin{pmatrix} 1 & b & b^4 & b^9 & \dots & b^{(n-1)^2} \\ b & 1 & b & b^4 & \dots & b^{(n-2)^2} \\ b^4 & b & 1 & b & \dots & b^{(n-3)^2} \\ b^9 & b^4 & b & 1 & \dots & b^{(n-4)^2} \\ & \vdots & \vdots & \vdots & \ddots & \vdots \\ b^{(n-1)^2} & b^{(n-2)^2} & b^{(n-3)^2} & b^{(n-4)^2} & \dots & 1 \end{pmatrix}, \qquad (II.2.3)$$

and f is the vector representing the original data, and h is the observed blurred data. Theoretically, the solution to this problem is straight-forward, invert the blur matrix and multiply by h to solve for f. Practically, however, as the dimension of the problem reaches realistic proportions, traditional numerical methods fail miserably due to the accumulation of numerical errors. Therefore, we resort to a *symbolic* solution, where most of the actual inversion is done symbolically for a Gaussian blur matrix. We proceed by decomposing the blur matrix into simpler matrices, which can then be symbolically inverted. Thus, the inverse of the Gaussian blur matrix is decomposed. Further, the decomposition is such that all the ill-conditioned terms are gathered into a diagonal matrix, with other matrices being perfectly well-conditioned.

The thesis is organized into two separate parts, the first of which treats the continuous and the second the discrete approaches to the Gaussian debluring problem. The introductory section to each part gives an independent and in depth introduction.

The two main results in this thesis, the solutions to Gaussian degradation formulated continuously and discretely, are convergent in the following sense. Consider the vector defined by the i^{th} row of the inverse blur matrix. This vector is roughly constant over all i, modulo the appropriate shift. This justifies the definition of a discrete deblurring kernel as the middle row of the inverse blur matrix; see figure (0.b). Observe the similarity between the continuous and discrete deblurring kernels. This is not suprising since the two (continuous and discrete) models are different approximations to the same physical process and must therefore yield similar results.

The different approaches had different advantages, however. In the continuous approach, it becomes clear how the deblurring kernel changes with the order of approximation inherent in the pseudo-inverse calculation In the discrete approach, numerical issues come to the forefront. Both perspectives are therefore necessary to properly understand Gaussian deblurring

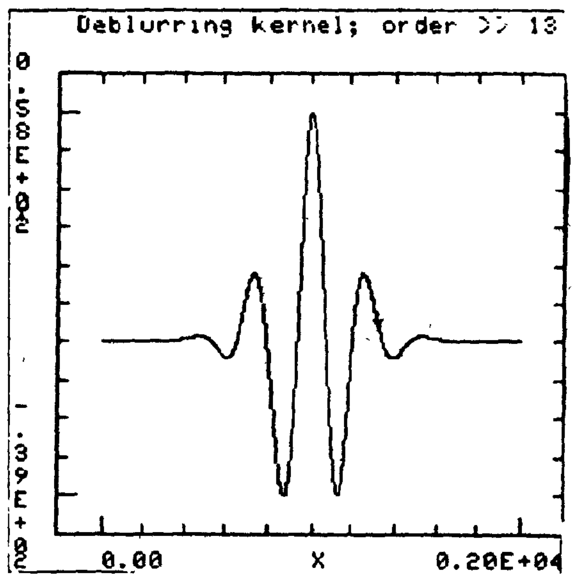

Figure 0.a The continuous deblurring kernel of order 13 with $\sigma = 1.0$, in the graph's x axis units

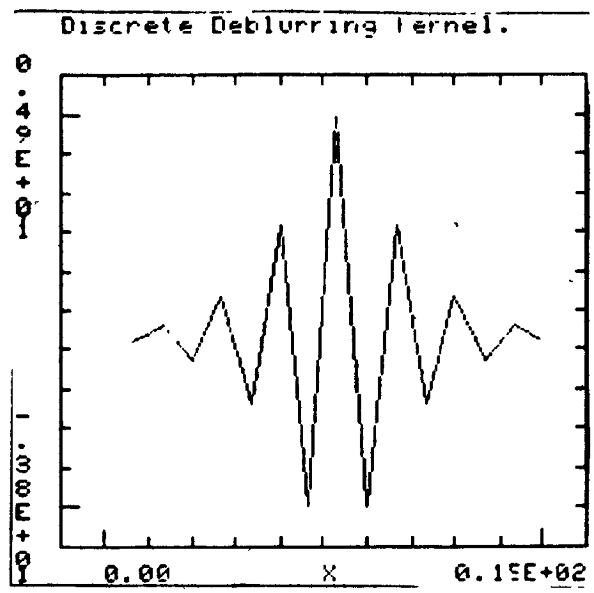

Figure 0.b The discrete deblurring kernel with $h = 0.6$.

PART I – Deblurring Gaussian Blur: Continuous Approach

Abstract I

Gaussian blur, or convolution against a Gaussian kernel, is one of the most common models for image and signal degradation. We are concerned with the inverse of this process, or Gaussian deblurring. As in the process of blurring, we seek a linear deblurring kernel. Although the inverse of a Gaussian cannot be represented exactly as a convolution kernel in the spatial domain, by restricting the space of allowable functions to polynomials of fixed finite degree then a convolution inverse does exist. Constructive formulas for the deblurring kernels are derived in terms of Hermite polynomials. For image polynomials of fixed degree N, the corresponding kernel gives stable deblurring among the class of functions which are Gaussian filtered versions of data well approximated by polynomials of degree N and less. Stated differently, the deblurring kernels are *pseudo-inverses* of the Gaussian convolution operator.

Chapter I.1 Introduction

Given an image or a signal, the realization of any practical system for processing must introduce some amount of degradation. Since almost all of these systems consist of several stages, each of which contributes to the degradation, they often compose into what appears to be a Gaussian degradation. In this paper we shall be concerned with inverting this process, or the deblurring of Gaussian blur.

Our model of blur is as a spatially invariant Gaussian point spread function within a linear system. Formally this leads to convolutions, as follows. Let $f(x)$ denote the original image function, $x \in \mathbb{R}^n$. Then the observable - but blurred - function $h(x)$ is given by:

$$\begin{aligned} h(x) &= K(x,t) * f(x) \\ &= \int_{\mathbb{R}^n} K(x-\xi,t) f(\xi) d\xi, \end{aligned} \tag{I.1.1}$$

where

$$K(x,t) = \frac{1}{(4\pi t)^{n/2}} e^{-|x|^2/4t} \tag{I.1.2}$$

is the Gaussian kernel, whose extent is parameterized by $t > 0$. It is normalized to have unit mass.

The problem of Gaussian deblurring can now be formulated: How can the original data $f(x)$ be reconstructed when *only* $h(x)$ and the amount of blurring t are known? Again, we shall formulate this as a convolution, and we seek a filter $D(x,t)$ such that

$$\begin{aligned} f(x) &= D(x,t) * h(x) \\ &= D(x,t) * K(x,t) * f(x), \end{aligned} \tag{I.1.3}$$

for $f(x)$ among a class of functions.

Our motivation for choosing this problem is two-fold. Firstly, many practical imaging configurations are structured in a manner that introduces blur either optically or for other reasons (e.g. computerized tomography [Herman, 1980]), and the Gaussian is the natural first approximation to this blur. And sensors are becoming far more reliable at higher light levels, leaving deterministic sources of blur more salient. Techniques for reducing this blur are thus of practical importance. There are even applications in physiological optics, such as the de-focusing that automatically takes place for objects outside of the depth of field of an accomodated eye.

Our second motivation is theoretical. It is well known that while the deblurring problem is in general non-invertible from Fourier considerations and unstable, † it is nonetheless possible to achieve acceptable deblurring under certain conditions. One way to accomplish this is by means of a pseudo-inverse ‡ which is an exact inverse under restricted conditions. Although such results have been available in the mathematical literature for some time [John, 1955], they are not widely known within the computational vision and image processing communities. Rather, the image processing community typically formulates the problem purely in discrete terms by applying algebraic pseudo-inverse techniques [e.g., Pratt, 1978]. But this obscures the analytical structure of the process, leaving central notions such as the *order* of the deblurring pseudoinverse implicit. Pseudo-inverses imply notions of approximation, and one would like a formulation in which the degree of this approximation could be made explicit. Then one could understand how the structure of the deblurring kernels changed as a function of the order of approximation.

In this paper, we derive kernels which can be used to deblur a fixed amount of Gaussian blur. They accomplish this inverse process *exactly*, and stably, among polynomials of fixed degree. Our analysis uses Hermite polynomials, a natural choice for reasons that will become clear shortly. The explicit formulas for the deblurring filters are given in the main theorem in chapter I.5. Since the analysis leading to this theorem is technical, we provide

† One must be clear about the fundamental distinction between a *stable* or *unstable problem* (in the Numerical Analysis literature it is usually refered to as a well- or ill-conditioned problem [e.g. see Stewart 1973]) as opposed to a *stable* or *unstable algorithm* for a given problem. Henceforth, stability has a different meaning when applied to a problem or an algorithm.

‡ Also, refered to as a *generalized inverse*.

motivating background in the next chapter.

I.2.1 Blurring and Diffusion

There is a fundamental connection between blurring, deblurring, and the heat equation. It is provided by the structure of the Gaussian distribution, as the following example illustrates. Consider a rod of infinite length onto which an impulse of heat is placed at some position As time evolves, the heat will diffuse and the original impulse will spead out. By basic physics the resulting temperature distribution will approximate a Gaussian whose extent depends on the ellapsed time [see e.g., Feynman, 1963]. By superposition, the model for the temperature distribution along the rod at any time is the initial temperature distribution convolved with a Gaussian. This is the physically realized solution to the heat equation †. The spatial parameter for the Gaussian depends on how much time has evolved, and the diffusion process effectively *blurs* the initial temperature distribution incrementally. In the notation introduced in chapter I.1, if $f(x)$ is the initial temperature distribution, then $h(x,t) = K(x,t) * f(x)$ is the blurred distribution after t units of time. Formally, this is an initial value problem, and can be stated as follows: given $f(x)$ and t, find $h(x,t)$ satisfying

$$\Delta h = \partial h/\partial t, \qquad h(x,0) = f(x). \tag{I.2.1}$$

We, of course, will interpret $f(x)$ as an unblurred image.

† the so-callled "source kernel" [Widder, 1975]

Two basic observations that follow from this formulation of the blurring problem that will be important in the analysis that follows. First, note that the space of initial distributions that can be blurred is a large one; it essentially corresponds to any function for which the convolution integral is defined, and clearly includes some discontinuous ones. Second, suppose that a function $f(x)$ has been blurred for some time, say t_0, resulting in $h(x, t_0)$. This resultant function could subsequently be blurred further, say to t_1, with $t_1 > t_0$. These two blurring operators, each of which may have its own physical justification, results in one composite Gaussian operator. Indeed, by the central limit theorem, other blurring operators compose into approximate Gaussians when iterated.

1.2.2 Deblurring and the Inverse Heat Problem

Since deblurring is the inverse of blurring, the preceeding connection between blurring and diffusion suggests that deblurring can be modelled as a diffusion running backwards in time. Blurring is the forward problem, and deblurring is the inverse problem. Formally, the problem of reconstructing $f(x)$ given $h(x)$ and t is the inverse heat equation problem, since the function $h(x)$ represents a distribution of heat after t units of time, where $f(x)$ is the initial $t = 0$ distribution.

As in the forward or blurring problem, which was modelled as convolutions of the original data against a "blurring kernel" (a Gaussian), our goal now is to find "deblurring kernels", or kernels against which the blurred data can be convolved to yield the deblurred original. However, the mathematics is not straightforward. There are a number of technical differences which make the deblurring problem more difficult than blurring. While the blurring (or heat diffusion) problem can be solved for almost all distributions (i.e., the solution is just a smoothed version of the initial data), the inverse problem is defined only for a restricted class of functions. Running "time" backwards, it is impossible, in general, to reconstruct the original data $f(x)$ from the blurred data $h(x)$. First, not all functions h(x) are blurred versions of some original function $f(x)$. Secondly, the blurring operator is not a one-to-one mapping in a general function space. There exist pairs of distinct functions, $f(x)$ and $\hat{f}(x)$, which yield the same blurred function $h(x)$. Finally, in a general function space the deblurring problem is horribly ill-conditioned. In other words, arbitrary small perturbations in the given function $h(x)$ can lead to large changes in the reconstruction of $f(x)$.

These difficulties are so severe that one might be pessimistic about any progress toward discovering deblurring kernels. However, the deblurring problem can be given a *pseudo-inverse* formulation, which leads to a well-conditioned problem. We formulate the pseudo-inverse problem for polynomial data in chapter 1.3, and present the deblurring kernels for polynomials in (1.5.6). The structure of these kernels is a function of the order of approximation, revealing how the solution to the problem changes as the data become more complex.

Chapter I.3 Pseudoinverse Formulation

Let $\mathcal{T}$ denote the blurring operator. which takes functions in a large normed space Λ into much smoother functions. also in Λ. Although $\mathcal{T}$ is a continuous operator. for many choices of Λ. $\mathcal{T}$ has no continuous inverse defined on its range.

The idea of a pseuoinverse is as follows: Consider a closed subspace $\mathcal{M} \subseteq \Lambda$. The image of $\mathcal{M}$ under $\mathcal{T}$ will also be a closed subspace. and so if Λ is a complete Hilbert space. one can pose the problem

$$\text{Given} \quad h \in \Lambda, \qquad \text{find} \quad f \in \mathcal{M} \qquad \text{minimizing} \quad \|\mathcal{T}f - h\|. \tag{I.3.1}$$

The solution f to this minimization problem is the pseudo-inverse of h under the map $\mathcal{T}$ on Λ relative to the subspace $\mathcal{M}$. and will be denoted $\Lambda_{\mathcal{M}}^{-1}h$.

In our case. we set $\Lambda = \mathcal{L}^2(e^{-x^2}dx)$. an enormous Hilbert space. which contains distributions which are not tempered We set $\mathcal{M} = P_N$, the space of polynomials of degree N and less. In chapter I.4. we will note that M is $\mathcal{T}$-invariant. Since $\mathcal{M}$ is finite dimensional and $\mathcal{T}$ is one-to-one. $\mathcal{T}$ is an isomorphism of $\mathcal{M}$ onto $\mathcal{M}$. Thus the problem of finding the pseudo-inverse of h is equivalent to finding f such that $\mathcal{T}f$ is the orthogonal projection of h onto $\mathcal{M}$. An algorithm for computing f can therefore be constructed by projecting h to h' on $\mathcal{M}$. and then solving the finite dimensional problem $\mathcal{T}f = h'$. Clearly. this process is stable for fixed N.

In chapter I.5. we present the solution to the deblurring problem on $\mathcal{M}$. so that the problem $\mathcal{T}f = h'$ is solved by a convolution

$$f = D_N * h'. \tag{I.3.2}$$

It will become evident that for $h \in \Lambda$, and h' the orthogonal projection of h onto $\mathcal{M} = \mathcal{P}_N$,

$$D_N * h = D_N * h'. \tag{1.3.3}$$

Thus the entire algorithm, projection onto $\mathcal{M}$ and inverting $\mathcal{T}$ on $\mathcal{M}$, can be represented by a single convolution. In fact, the kernels D_N given in chapter 1.5 are unique in having this double property.

Chapter I.4 The Deblurring Problem

Consider the operator Ω_t defined on $\mathcal{L}^2(\mathrm{IR})$ by the equation

$$(\Omega_t f)(y) = \int_{-\infty}^{\infty} \frac{1}{2\sqrt{\pi t}} e^{-x^2/4t} f(y-x)\,dx. \tag{I.4.1}$$

For $t > 0$. Ω_t is a compact symmetric bounded linear operator on $\mathcal{L}^2(\mathrm{IR})$ mapping into $\mathcal{L}^2(\mathrm{IR})$. This operator has many special properties, such as

$$\Omega_t \circ \Omega_s = \Omega_{t+s}. \tag{I.4.2}$$

Also,

$$u(x,t) = (\Omega_t f)(x), \tag{I.4.3}$$

satisfies the *heat equation*

$$\Delta u = u_t, \tag{I.4.4}$$

with

$$u(x,0) = f(x); \tag{I.4.5}$$

see [Bers, John, Schechter]. If we denote the Fourier transform of a function g(x) by $\hat{g}(\omega)$, then Ω_t is a multiplier operator given by

$$(\widehat{\Omega_t f})(\omega) = e^{-w^2 t}\hat{f}(\omega) \tag{I.4.6}$$

By means of this formula, Ω_t can be extended to operate on the class of temperate distributions S' of Fourier transformable distributions [Hörmander, 1983]. In particular, $\Omega_t f$ is defined for any polynomial f.

We will specialize to the case of $t = \frac{1}{4}$, and set

$$\mathcal{T} f = \Omega_{1/4} f = \frac{1}{\sqrt{\pi}} e^{-x^2} * f. \tag{I.4.7}$$

By suitably scaling the spatial parameter $x \in \mathrm{IR}$, Ω_t, $t > 0$, can be seen to be equivalent to $\mathcal{T}$ operating on a rescaled version of f, i.e.,

$$(\Omega_t f)(y) = (\mathcal{T}\tilde{f})(y/2\sqrt{t}), \tag{I.4.8}$$

where

$$\tilde{f}(x) = f(2/\sqrt{t}). \tag{I.4.9}$$

Thus the invertibility of Ω_t is settled by inverting $\mathcal{T}$.

From the Fourier multiplier formula

$$(\widehat{\mathcal{T}f})(\omega) = e^{-\omega^2/4}\hat{f}(\omega), \tag{I.4.10}$$

and the fact that $e^{-\omega^2/4} \neq 0$ for all ω, it is clear that $\mathcal{T}$ is one-to-one on any space of Fourier transformable functions. Further, since the inverse of the multiplier, $e^{\omega^2/4}$, has no inverse Fourier transform, the inverse of $\mathcal{T}$ is not representable as a convolution, nor can be applied to the general space of all Fourier transformable functions. Instead, we can restrict the domain of $\mathcal{T}$, and then represent its inverse as a convolution on the range of $\mathcal{T}$. Many such restricted domains are possible. In the next chapter, we consider $\mathcal{T}$ restricted to the class of polynomials of degree N or less.

Chapter I.5 Polynomial Domains

Let P_N denote the space of polynomials over IR of degree less than or equal to N. The monomials $\{1, x, x^2, \dots x^N\}$ form a basis for P_N. If this basis is orthonormalized with respect to the inner product

$$(f,g) = \int_{-\infty}^{\infty} f(x)g(x)e^{-x^2}dx, \tag{I.5.1}$$

then the basis of Hermite polynomials $\{H_0, H_1, \dots, H_N\}$ result. The Hermites can be represented explicitly:

$$H_n(x) = n! \sum_{m=0}^{[n/2]} (-1)^m \frac{(2x)^{n-2m}}{m!(n-2m)!}, \tag{I.5.2}$$

or by the Rodrigues formula:

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n}(e^{-x^2}), \tag{I.5.3}$$

see, e.g., [Courant and Hilbert, 1962] or [Lebedev, 1965].

Observation I.5.4: $\mathcal{T}$ is closed on P_N.

Proof: we will show that $\mathcal{T} H_n \in P_n$ for $n \leq N$.

$$\begin{aligned}
\sqrt{\pi}(\mathcal{T} H_n)(y) &= \int_{-\infty}^{\infty} e^{-(y-x)^2} H_n(x)dx \\
&= \int_{-\infty}^{\infty} e^{-y^2} e^{2xy}(-1)^n \frac{d^n}{dx^n}(e^{-x^2})dx \\
&= 2y \int_{-\infty}^{\infty} e^{-y^2} e^{2xy}(-1)^{n-1} \frac{d^{n-1}}{dx^{n-1}}(e^{-x^2})dx \\
&= \sqrt{\pi}\ 2y(\mathcal{T} H_{n-1})(y);
\end{aligned}$$

Using $\mathcal{T} H_0 = 1$, we have

$$(\mathcal{T} H_n)(x) = 2^n x^n. \quad (I.5.5)$$

As a result of observation I.5.4, $\mathcal{T}$ is an isomophism of P_N. The inverse of $\mathcal{T}$ on P_N is clearly given by

$$\mathcal{T}^{-1}\left(\sum_{i=0}^{N} a_i x^i\right) = \sum_{i=0}^{N} (a_i/2^i) H_i(x). \quad (I.5.6)$$

Our main result is that $\mathcal{T}^{-1}$ restricted to P_N can be represented by a convolution with an explicit kernel $D_N(x)$:

Theorem I.5.7: For $f \in P_N$ and $g = \mathcal{T} f$, then

$$f = D_N * g \quad (I.5.8)$$

where

$$D_N(x) = e^{-x^2} \sum_{k=0}^{\lfloor N/2 \rfloor} \frac{(-1)^k}{\sqrt{\pi} k! 2^k} H_{2k}(x). \quad (I.5.9)$$

We will give a proof below using direct integration (as opposed to using Fourier transform distributions). Note, however, that $D_N(x)$ is not the unique function representing $\mathcal{T}^{-1}$ on P_N. In general, the kernel can be translated by any function which yields a zero convolution against P_N This includes all functions of the form

$$e^{-x^2} H_n(x), n > N.$$

The stated kernel (I.5.9) is unique among the class of functions of the form $e^{-x^2} P(x)$, where P(x) is a polynomial of degree **N**.

It is interesting to compare the form of $D_N(x)$ with standard enhancement filters For example, for $N = 3$,

$$\begin{aligned} D_3(x) &= \frac{2}{\sqrt{\pi}} e^{-x^2}(1 - x^2) \\ &= \frac{1}{\sqrt{\pi}} e^{-x^2} - \frac{1}{2} \frac{d^2}{dx^2}\left(\frac{1}{\sqrt{\pi}} e^{-x^2}\right). \end{aligned} \quad (I.5.10)$$

Thus

$$D_3 * g = [1 - \frac{1}{2}\frac{d^2}{dx^2}]\mathcal{T} g. \qquad (I.5.11)$$

which is a not uncommon high emphasis filter (see, e.g., the papers by E. Mach in [Ratliff, 1965], and [Rosenfeld and Kak, 1976]. In figure I.1, we display plots of D_N for several values of N.

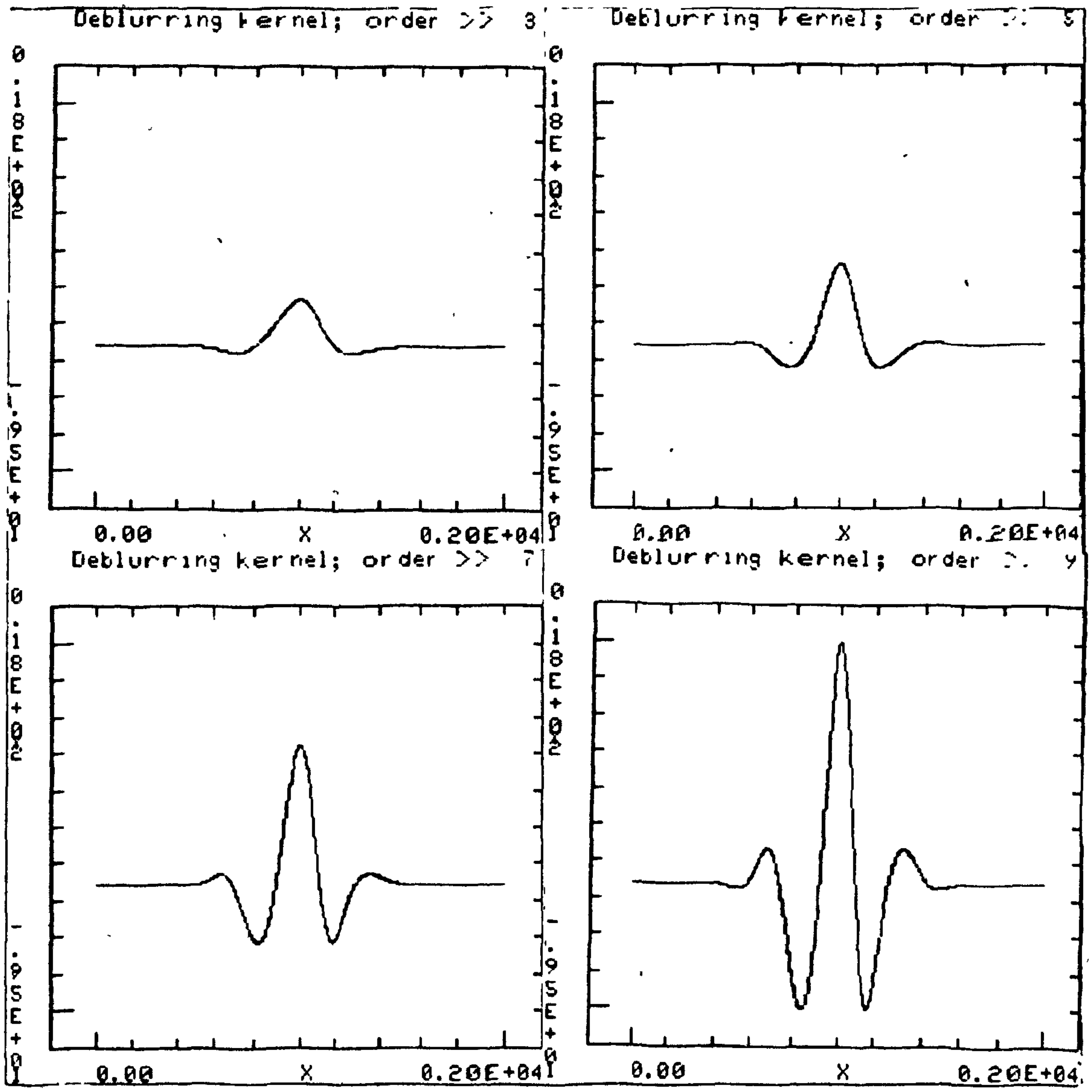

Figure I.1.a One dimensional deblurring kernels, D_N, $N = 3, 5, 7, 9$, drawn to the same scale Note that as the order increases both the magnitude and the number of sign changes increases as well.

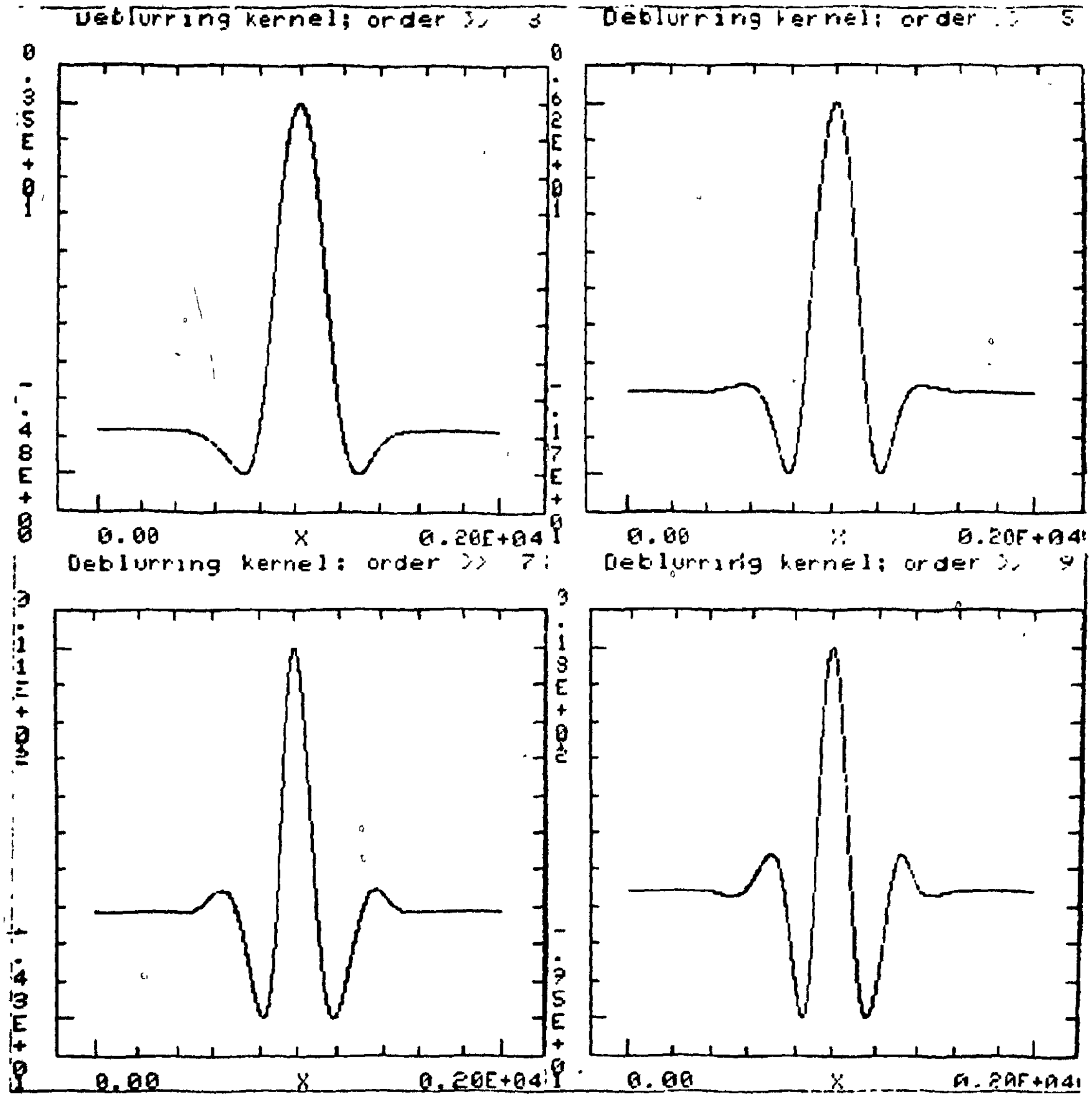

Figure 1.1.b One dimensional deblurring kernels, D_N, $N = 3, 5, 7, 9$, scaled so that the *structure* of each kernel is clear. Note that the number of side lobes is $[\frac{N}{2}]$ on either side of the central peak.

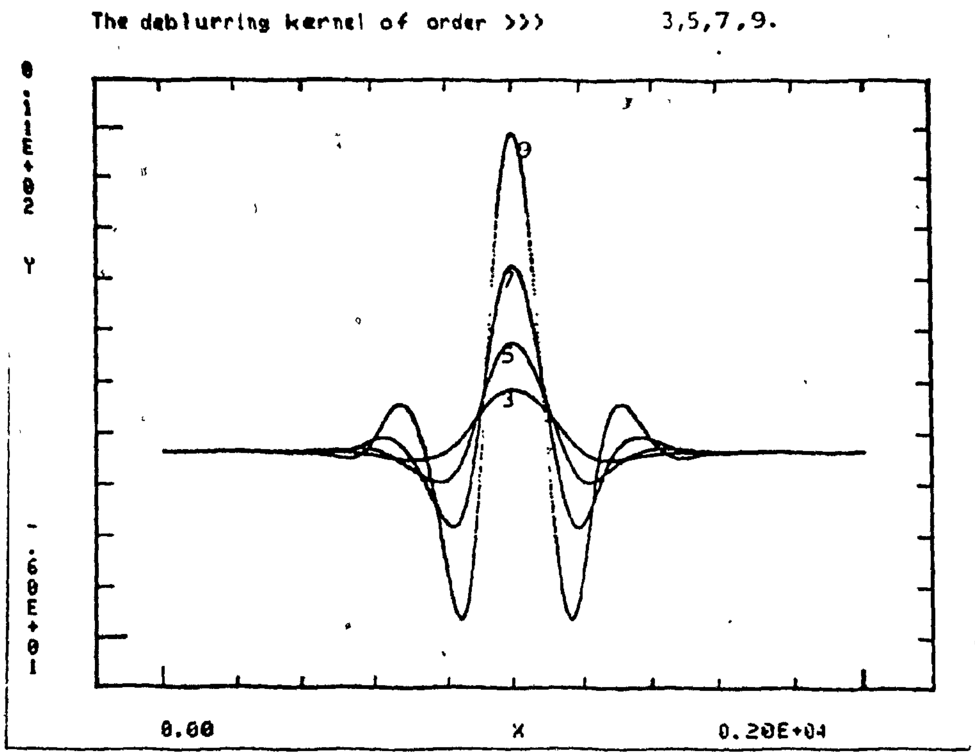

Figure 1.1.c Emergence of side lobes as N increases $N = 3,5,7,9$.

The proof of the theorem depends on several lemmas.

Lemma I.5.12:

$$A_n = \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi}} e^{-x^2} x^n dx = \begin{cases} 0, & \text{n odd} \\ \frac{n!}{2^n (n/2)!}, & \text{n even} \end{cases}$$

Proof: For n odd. $e^{-x^2}x^n$ is an integrable odd function, and so clearly $A_n = 0$. For $n = 2p, p \geq 1$,

$$\begin{aligned} A_{2p} &= \int_{-\infty}^{\infty} e^{-x^2} x^{2p} dx \\ &= -\frac{1}{2} \int_{-\infty}^{\infty} (-2x) e^{-x^2} x^{2p-1} dx \\ &= \frac{2p-1}{2} \int_{-\infty}^{\infty} e^{-x^2} x^{2p-2} dx \\ &= \sqrt{\pi} \frac{2p-1}{2} A_{2p-2}. \end{aligned} \tag{I.5.13}$$

Since $A_0 = 1$,

$$A_{2p} = \frac{(2p-1)(2p-3)\ldots 1}{2^p} = \frac{(2p)!}{2^{2p} p!}. \tag{I.5.14}$$

The Formula holds for $p \geq 0$. ∎

Lemma I.5.15:

$$c_{2k,2p} = \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi}} e^{-x^2} H_{2k}(x) x^{2p} dx = \begin{cases} 0, & p < k \\ \frac{(2p)!}{2^{2p-2k}(p-k)!}, & p \geq k \end{cases}.$$

Proof: For $k \geq 1$, $p \geq 1$,

$$\begin{aligned} \sqrt{\pi} c_{2k,2p} &= \int_{-\infty}^{\infty} e^{-x^2} [(-1)^{2k} e^{x^2} \frac{d^{2k}}{dx^{2k}} (e^{-x^2})] x^{2p} dx \\ &= \int_{-\infty}^{\infty} \frac{d^{2k}}{dx^{2k}} (e^{-x^2}) x^{2p} dx \\ &= -2p \int_{-\infty}^{\infty} \frac{d^{2k-1}}{dx^{2k-1}} (e^{-x^2}) x^{2p-1} dx \\ &= \sqrt{\pi} (2p)(2p-1) c_{2k-2,2p-2}. \end{aligned} \tag{I.5.16}$$

Clearly, $c_{2k,0} = 0, k \geq 1$. Using Lemma 1, $c_{0,2p} = (2p)!/(2^{2p} p!)$, for $p \geq 0$. Combining, $c_{2k,2p} = 0$ for $p < k$, and for $p \geq k$.

$$\begin{aligned} c_{2k,2p} &= (2p)(2p-1)\ldots(2p-2k+1).c_{0,2p-2k} \\ &= \frac{(2p)!}{(2p-2k)!}\frac{(2p-2k)!}{2^{2p-2k}(p-k)!} \\ &= \frac{(2p)!}{2^{2p-2k}(p-k)!}. \end{aligned} \tag{I.5.17}$$

∎

Lemma I.5.18:

For $n \geq k$,

$$d_K = \int_{-\infty}^{\infty} D_N(x)x^k dx = \begin{cases} 0, & \text{k odd} \\ (-1)^{k/2}\frac{k!}{2^k(k/2)!}, & \text{k even} \end{cases}$$

Proof: For **k** odd, we observe from (I.5.9) that $D_N(x)x^k$ is an odd integrable function, and so integrates to zero. For $k = 2p$,

$$\begin{aligned} \int_{-\infty}^{\infty} D_N(x)x^k dx &= \int_{-\infty}^{\infty} e^{-x^2} \sum_{i=0}^{\lfloor N/2 \rfloor} \frac{(-1)^i}{\sqrt{\pi} i! 2^i} H_{2i}(x) x^{2p} dx \\ &= \sum_{i=0}^{\lfloor n/2 \rfloor} \frac{(-1)^i}{i! 2^i} c_{2i,2p} \\ &= \sum_{i=0}^{p} \frac{(-1)^i}{i! 2^i} \frac{(2p)!}{2^{2p-2i}(p-i)!} \\ &= \frac{(2p)!}{2^p p!} \sum_{i=0}^{p} \frac{p!}{i!(p-i)!}(-1)^i (1/2)^{p-i} \\ &= \frac{(2p)!}{2^p p!}(-1/2)^p \\ &= (-1)^p \frac{(2p)!}{2^{2p} p!}. \end{aligned} \tag{I.5.19}$$

∎

Proof of the Theorem: By equation (I.5.5), it suffices to show that $D_N * (2^n x^n) = H_n(x), n \leq N$. We have

$$
\begin{aligned}
(D_N * 2^n x^n)(y) &= \int_{-\infty}^{\infty} 2^n D_N(x)(y-x)^n dx \\
&= \int_{-\infty}^{\infty} 2^n D_N(x) \sum_{k=0}^{n} (-1)^k \binom{n}{k} y^{n-k} x^k dx \\
&= \sum_{k=0}^{n} \frac{2^n n!}{k!(n-k)!} (-1)^k y^{n-k} d_k \\
&= n! \sum_{m=0}^{n} \frac{(-1)^{2m} 2^n}{(2m)!(n-2m)!} (-1)^m \frac{(2m)!}{2^{2m} m!} y^{n-2m} \\
&= n! \sum_{m=0}^{n} \frac{(-1)^m}{m!(n-2m)!} 2^{n-2m} y^{n-2m} \\
&= H_n(y). \qquad \blacksquare
\end{aligned}
\tag{I.5.20}
$$

The theorem above could have been proved using the convolution theorem and by computing the Fourier transform of $D_N(x)$. We will nonetheless compute $\hat{D_N}$ in order to show that the multiplier for D_N approaches, pointwise, the inverse of the multiplier for the operator $\mathcal{T}$ (see (I.4.10)).

Observation I.5.21:

$$\widehat{D_N}(\omega) \to e^{\omega^2/4} \text{ pointwise as } N \to \infty.$$

Proof:

$$\widehat{D_N}(\omega) = \sum_{k=0}^{[N/2]} \frac{(-1)^k}{\sqrt{\pi} k! 2^k} \mathcal{F}[e^{-x^2} H_{2k}(x)](\omega), \tag{I.5.22}$$

where $\mathcal{F}$ stands for the Fourier transform operator. Now,

$$
\begin{aligned}
\mathcal{F}[e^{-x^2} H_{2k}(x)](\omega) &= \mathcal{F}[(-1)^{2k} \frac{d^{2k}}{dx^{2k}} (e^{-x^2})](\omega) \\
&= (i\omega)^{2k} \sqrt{\pi} e^{-\omega^2/4}.
\end{aligned}
\tag{I.5.23}
$$

Thus

$$\widehat{D_N}(\omega) = \sum_{k=0}^{[N/2]} \frac{(-1)^k}{\sqrt{\pi}k!2^k}.(-1)^k \omega^{2k}\sqrt{\pi}e^{-\omega^2/4}$$
$$= e^{-\omega^2/4}\sum_{k=0}^{[N/2]} \frac{1}{k!}(\frac{\omega^2}{2})^k. \tag{I.5.24}$$

Hence

$$\lim_{N\to\infty} \widehat{D_N}(\omega) = e^{-\omega^2/4}e^{\omega^2/2}$$
$$= e^{\omega^2/4}. \tag{I.5.25}$$

∎

As a consequence of observation I.5.21. we see that $D_N(x)$ does <u>not</u> converge pointwise to any function as $N \to \infty$, since otherwise the Fourier transform of that function would be $e^{\omega^2/4}$, which is impossible. $D_N(x)$ does converge in $\mathcal{L}^2(e^{-x^2}dx)$. but that does not imply pointwise convergence to any function. We accordingly have stable deblurring when using the kernels $D_N(x)$, where stability is measured in terms of deviation from a polynomial of degree N. and the $\mathcal{L}^2(e^{-x^2}dx)$ norm is used as the metric.

Chapter I.6 Higher Dimensions

The Gaussian blur operator is given by

$$\mathcal{T} f(x) = \int_{\mathbb{R}^n} \pi^{-n/2} e^{-(x-y)^2} f(y) dy. \tag{I.6.1}$$

Due to the separability of the kernel and Fubini's theorem, $\mathcal{T}$ can be decomposed into n iterated blurrings

$$\mathcal{T} = \mathcal{T}_1 \circ \mathcal{T}_2 \circ \ldots \circ \mathcal{T}_n \tag{I.6.2}$$

$$(\mathcal{T}_i f)(x) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi}} e^{-(x_i - y_i)^2} f(x_1, \ldots, y_i, \ldots, x_n) dy_i \tag{I.6.3}$$

Consider a polynomial in $\mathbb{R}^n$:

$$f(x) = \sum_{|\alpha| \leq N} a_\alpha x^\alpha \tag{I.6.4}$$

$$\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n), \quad \alpha_i \in Z, \quad \alpha_i \geq 0, \tag{I.6.5}$$

$$|\alpha| = \sum \alpha_i, \quad x^\alpha = x_1^{\alpha_1} \ldots x_n^{\alpha_n}. \tag{I.6.6}$$

For fixed x, the function of one real variable

$$g(y_i) = f(x_1, \ldots, y_i, \ldots, x_n) \tag{I.6.7}$$

is a polynomial of degree no greater than N, so

$$D_N * (\mathcal{T} g) = g, \tag{I.6.8}$$

where T is the standard one dimensional blurring operator introduced in chapter I.3. Combining, we find that

$$f(x) = \int_{\mathrm{IR}^n} D_N(y_1)D_N(y_2)\dots D_N(y_n)(Tf)(x-y)dy \qquad (I.6.9)$$

for any polynomial $f(x)$ of form (I.6.4). Thus deblurring of blurred polynomials of degree N can be accomplished by convolution against the kernel

$$D_N{}^*(x) = D_N(x_1)D_N(x_2)\dots D_N(x_n). \qquad (I.6.10)$$

Thus the situation in higher dimensions is similar to the one dimensional case. The deblurring convolution kernel is separable, and will be of the form $e^{-x^2}P(x)$, where $P(x)$ is a polynomial of degree nN in $x \in \mathrm{IR}^n$ Figure I.2 shows a plot of $D_N{}^*$ for $n = 2, N = 3$.

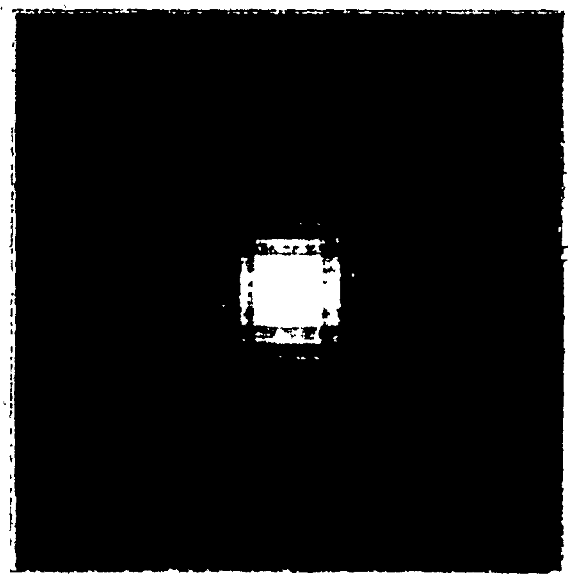

Figure I.2.a Two dimensional deblurring kernel, $D_N(x_1, x_2)$, $N = 3$, displayed as an image. Note the central positive region and the sign changes in the surrounding regions

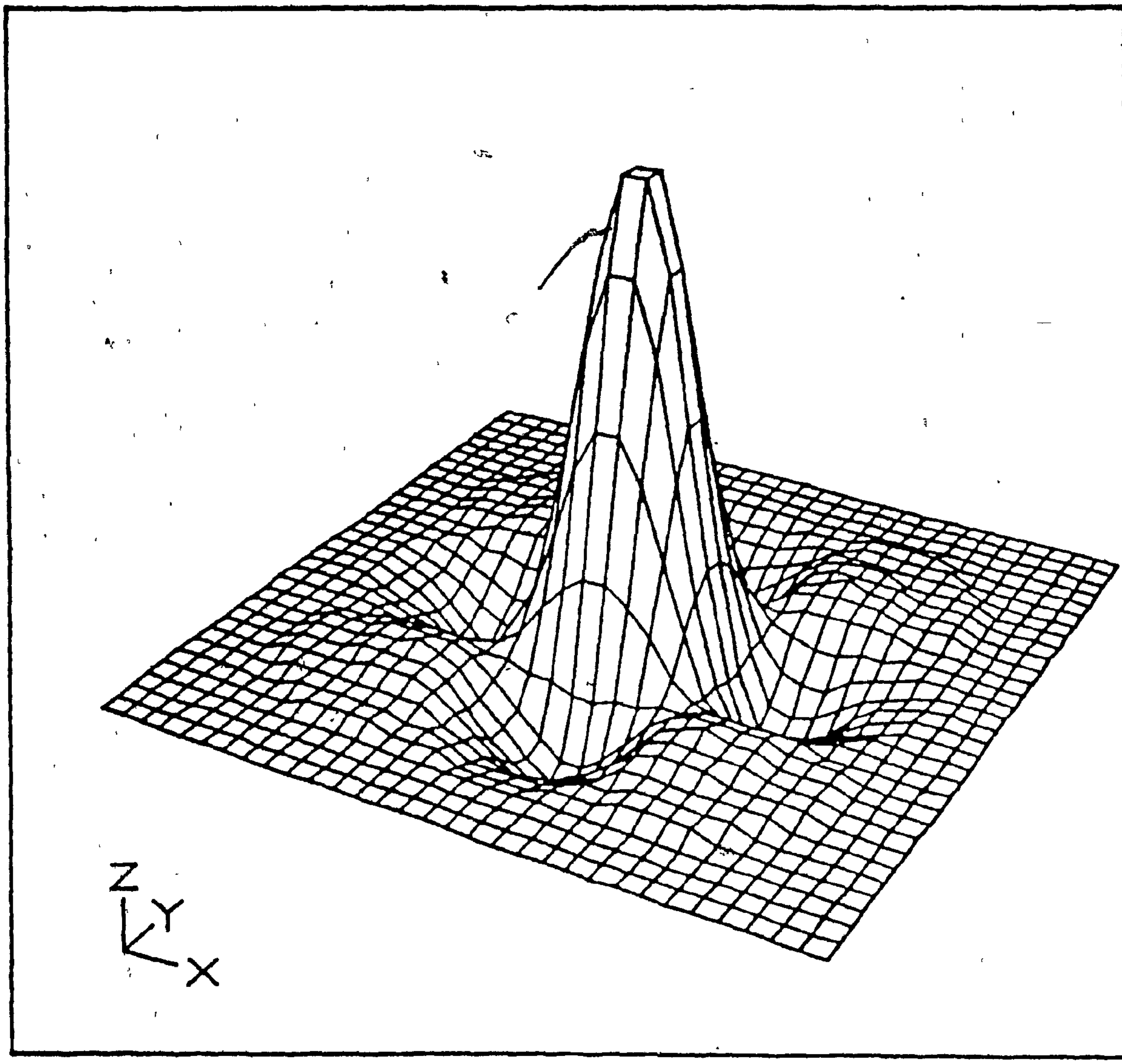

Figure 1.2.b Two dimensional deblurring kernel $D_N(x_1, x_2)$. $N = 3$. Note the sign changes in the kernel surrounding the central positive peak

Chapter I.7 Experiments in Deblurring

In this chapter we illustrate the deblurring operator. It was implemented in the most straightforward fashion, using single-precision arithmetic. The continuous operators, both for blurring and for deblurring, were discretized by point sampling.

Figure I.3.a contains an image of an urban scene (in Pittsburgh); figure I.3.b displays the same image convolved with a Gaussian; and figure I.3.c contains the deblurred image obtained with a 9^{th}-order kernel. † To facilitate reproduction, these images were displayed using dither-matrices on a high-resolution laser (i.e., binary) printer, so the above examples should only be taken as a qualitative indication of the deblurring. Only 32 gray levels are effectively displayed. Informal observations from several members of our laboratory were that the results were much more impressive when viewed on a raster-graphics system monitor.

A more precise representation is shown in figure I.4. This graphs the performance of the operators as a function of both the amount of blurring (σ) and of the order of the deblurring filter (N). Performance is defined as the ratio:

$$\frac{\|f - \mathcal{T}f\| - \|f - \mathcal{T}^{-1}\mathcal{T}f\|}{\|f - \mathcal{T}f\|},$$

where $\|f\|$ is the $F-$norm of the matrix f.

Note that, for this image, the performance of the deblurring filter peaks at order 9 for small σ, but, as the amount of blurring becomes large, the filter becomes much less effective at all orders tested.

---

† An identical $\sigma = 1.0$ in internal units was used for both the blurring and the deblurring kernels.

Figure 1.3.a An image of an urban scene 256x256 resolution displayed using dither matrices with 32 effective gray levels Note that the structure in the left portion of the image contains discernible detail

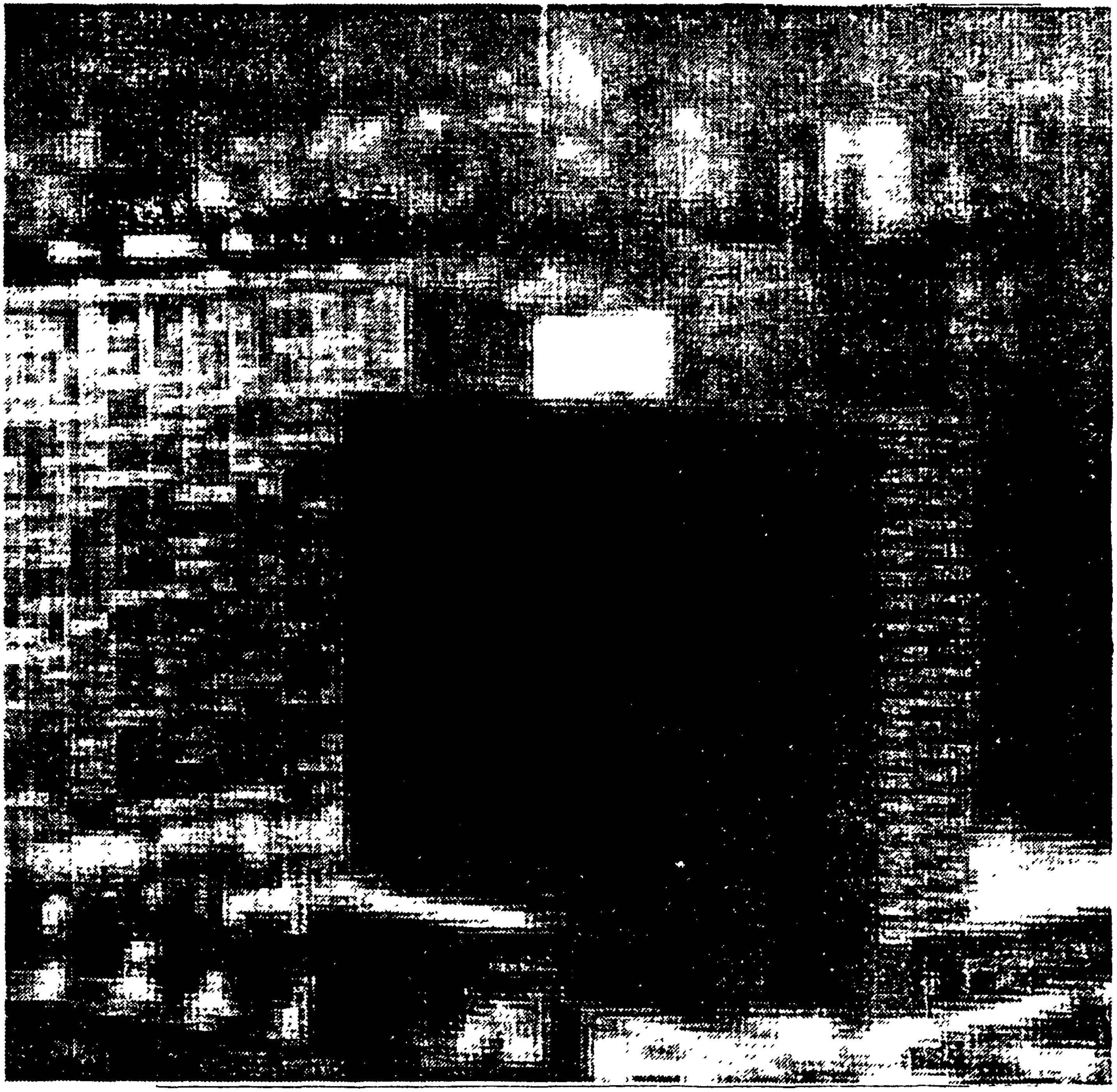

Figure 1.3.b A Gaussian-Blurred version of figure 1 3 a ($\sigma = 1.0$) Note that the detail in the left portion is now smoothed over

Figure 1.3.c An order 9 Gaussian deblurred version of figure 1.3.b. Note that the contrast and detail have been qualitatively improved.

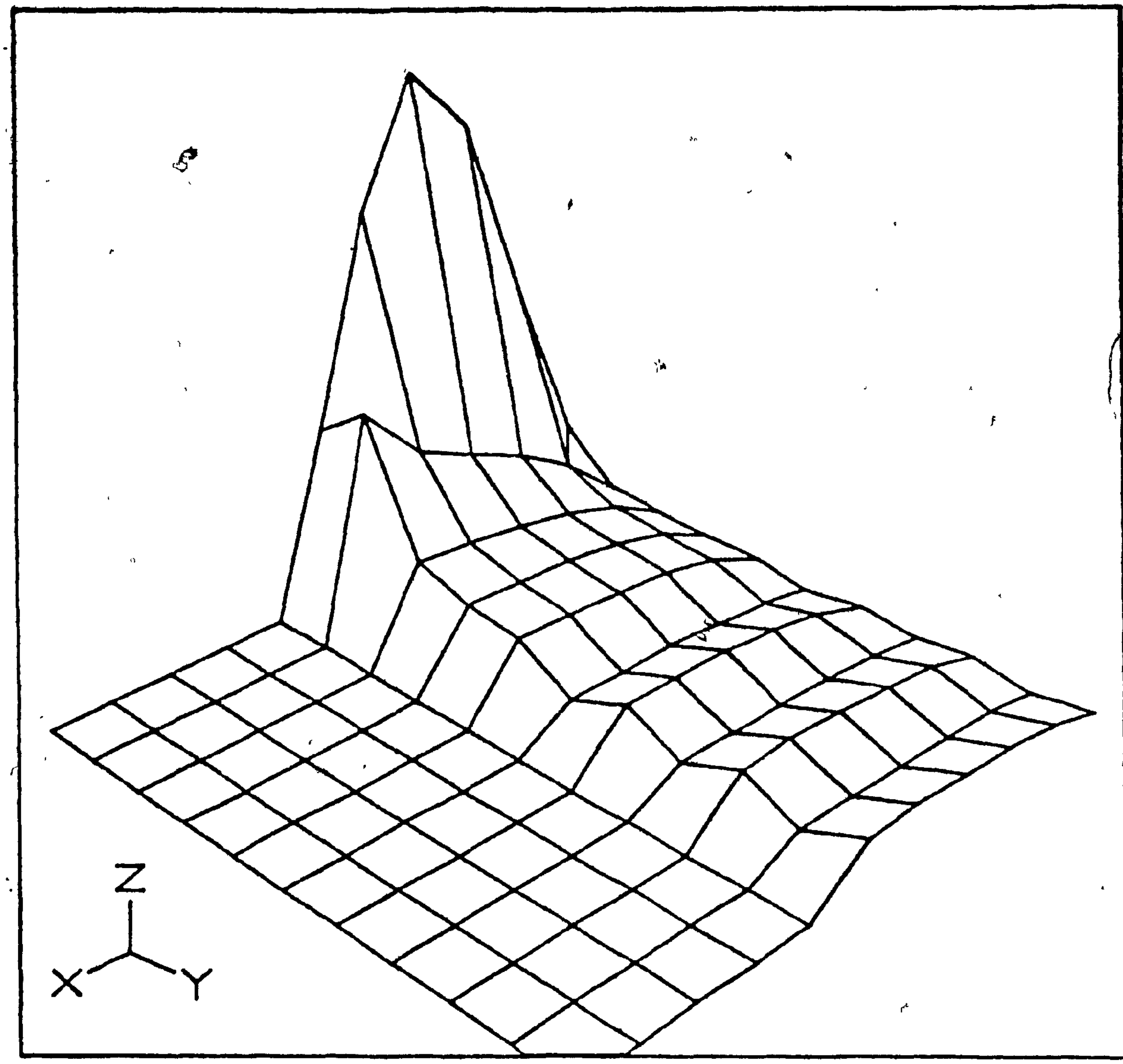

Figure 1.4 An illustration of how the performance of the deblurring operator varies as a function of the order of deblurring kernel (the x-axis $N = 3, 5, 7, 9, 11, 13$), and the amount of blur (the y-axis $\sigma = 1.0, 1.5, 2.0, 2.5, 3.0, 4.0, 5.0, 6.0$) for the image in figure 1.3 a. For small amounts of blurring, the performance of the deblurring filter peeks at order 9, while for higher amounts of blur, it performs much less effectively.

Chapter 1.8 Summary I

Gaussian blur is one of the most common forms of degradation affecting signals and images. It is unfortunately non-invertible in general, but pseudo-inverses are possible In this paper we formulated a precise version of the Gaussian deblurring problem, and obtained formulae for the kernels of deblurring filters in terms of Hermite polynomials One then simply needs to convolve these kernels against (blurred) images to effect deblurring. As the order of the kernel increases, the space on which deblurring is exact increases as well.

The mathematics used in formulating the deblurring kernels were based on the heat equation The connection between blurring and the heat equation is provided by the Gaussian: the spread of any heat distribution is governed by convolutions against a Gaussian kernel. Deblurring then amounts to solving the heat equation backwards in time.

However, backward solutions to the heat equation are notoriously unstable. Nevertheless, we have been able to show that stable deblurring is possible in principle for a class of image functions, and, perhaps more importantly, that some degree of stable deblurring is possible in practice for real images. The example in the paper was obtained using the most straightforward implementation. More serious attention to numerical issues, such as arithmetic precision and quadrature, could possibly lead to even better results.

PART II - Deblurring Gaussian Blur: Discrete Approach

Abstract II

Suppose a function - an image or a signal - is degraded by Gaussian blur; i.e. suppose the function is convolved against a discrete Gaussian kernel. We present a symbolic method to stably remove such Gaussian degradation. The blur is modelled as multiplication by a Toeplitz matrix which is derived from the Gaussian kernel. The problem, therefore, is one of solving a system of linear equations governed by this matrix. We find the inverse by decomposing the blur matrix and then analytically inverting the resulting sub-matrices. Moreover, the decomposition is such that the numerically ill-conditioned terms are gathered into a diagonal form. The result can be combined for exact and stable deblurring.

Chapter II.1 Introduction

Gaussian blur is a common phenomenon, yet to date no exact solution for removing it has been given. Traditional image processing techniques deal with this problem either with ad-hoc practical measures, "enhancement filters" for example [Rosenfeld and Kak], or use Fourier or other methods. However, all of these techniques have problems such as their numerical stability, their singularity, or the introduction of approximations [Pratt]. In contrast, we present a symbolic method to invert Gaussian blur analytically. It is superior to direct numerical techniques as it fully utilizes the special structure of the Gaussian.

Gaussian degradation is the linear process of convolution against a Gaussian blurring kernel. Natural examples include atmospheric and optical blur. The lens of the eye, as another example, blurs images in this fashion [Campbell and Gubisch]. The amount of such blur can be used as a depth cue [Pentland]. Also, to a first approximation, the degradation of computer tomography images is of this kind [Herman]. A vast number of other examples exist, mainly due to an application of the central limit theorem: when a large number of random local degradations combine sequentially, the resulting degradation closely resembles a Gaussian.

II.1.1 Removal of Gaussian Blur is Ill-conditioned

Our goal in this paper is to remove spatially invariant Gaussian degradation of known amount in the discrete domain†. After sampling, a one-dimensional signal is represented

† These conditions will be partly relaxed in chapter II.9

by a vector, and similarly, an image is represented as a matrix of intensity values. The degradation is modelled by discrete convolution of a point-sampled Gaussian against this data. This leads to a Gaussian "blur" matrix B, which we observe to be symmetric Toeplitz. We will show that in the one-dimensional case, the degradation is equivalent to multiplication of the data vector by this Toeplitz matrix. In the two-dimensional case, the separability of the Gaussian kernel allows us to model the degradation as left and right multiplication of the data matrix by two "blur matrices".

Let f denote the true data vector, h the observed degraded data vector, and B the blur matrix. The problem in the one-dimensional discrete domain is then solving the system of linear equations: ‡

$$h = Bf. \tag{II.1.1}$$

Such a task would seem to be straightforward: simply invert the matrix and multiply to obtain the deblurred vector. However, the above matrix is horribly ill-conditioned; a small perturbation of the vector h could lead to a large perturbation of the vector f. This is especially true for a blur matrix with entries close to 1 (which corresponds to a large spatial extent parameter, σ.) Therefore, one can not numerically invert this matrix to solve the system of linear equations.

The problem of solving systems of linear equations, by methods other than inversion, has resulted in a number of *stable algorithms* † which appear to be applicable to Gaussian deblurring. Examples of these algorithms include Gaussian elimination, Crout decomposition, and Cholesky decomposition of positive definite matrices, to name only a few [Stewart]. These algorithms are designed to deal with general systems of linear equations. However, in a particular case, such as Gaussian deblurring, they do not make full use of the special structure of the problem. Moreover, due to accumulation of numerical errors, their performance drops drastically as the dimension of the problem increases to realistic proportions.

‡ The matrix B will be defined formally in chapter II 2

† A stable algorithm is one whose computed output value for some input is the exact output of some nearby input

Our approach to deblurring is to invert the blur matrix analytically. We will accomplish this by means of analytic decompositions and analytic inversions of the resulting submatrices. These decompositions are sketched in the next chapter, in which we lay out the plan for the paper, and are developed in more detail in subsequent chapters. Before beginning, however, we should like to stress one of the advantages of our approach: Analytic inversion does not alter the conditioning of the problem, rather, the decomposition that we derive collects all of the "sensitive" terms into a single diagonal matrix. This diagonal form greatly simplifies the handling of numerical problems.

Chapter II.2 Discrete Deblurring Problem

In this chapter we formulate the discrete deblurring problem as a matrix equation. First, we deal with the one-dimensional case. Let f_i be the original data points, g_i samples of the degrading kernel, and h_i the observed degraded data points. Using discrete convolution as the model of degradation, we have

$$h_j = \sum_{i=-\infty}^{\infty} g_{j-i} f_i; \qquad j = -\infty, \ldots, \infty. \tag{II.2.1}$$

We make two simplifications to arrive at the blur matrix, B. First, when dealing with finite images, we set all the peripheral data to zero, namely, $f_i = 0$ for $i < 1$ and $i > n$, where n is the dimension of our data. Second, we use the fact that the kernel is Gaussian and substitute samples of the Guassian into the matrix. To simplifiy such a substitution, define a constant b of the scaled Gaussian ‡

$$g = e^{-x^2/2\sigma^2}; \qquad b \stackrel{\Delta}{=} e^{-1/2\sigma^2},$$

which implies that samples of the Guassian at i are $g_i = b^{i^2}$.

Then,

$$h_j = \sum_{i=1}^{n} b^{(j-i)^2} f_i \qquad 1 \leq j \leq n.$$

Let,

$$f \stackrel{\Delta}{=} [f_i];$$

‡ For ease of analysis, we will use a scaled Gaussian. Clearly, such a scalar does not affect the result of this paper as we are dealing with a linear system.

$$h \triangleq [h_i];$$

$$B \triangleq [\beta_{ij}];$$

where

$$\beta_{ij} = g_{j-i} = b^{(j-i)^2}.$$

Using this notation, we have

$$\boxed{\mathbf{h} = B\mathbf{f}} \tag{II.2.2}$$

where,

$$B \triangleq \begin{pmatrix} 1 & b & b^4 & b^9 & \dots & b^{(n-1)^2} \\ b & 1 & b & b^4 & \dots & b^{(n-2)^2} \\ b^4 & b & 1 & b & \dots & b^{(n-3)^2} \\ b^9 & b^4 & b & 1 & \dots & b^{(n-4)^2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ b^{(n-1)^2} & b^{(n-2)^2} & b^{(n-3)^2} & b^{(n-4)^2} & \dots & 1 \end{pmatrix}. \tag{II.2.3}$$

Observe that B is symmetric Toeplitz. Also, we will show, by means of the Cholesky decomposition (appendix C) that B is positive-definite.

As a preview of our results, we display the final decompsition of B^{-1}:

Theorem II.6.1:

$$B^{-1} = \hat{L}^T \hat{D} \hat{L}, \tag{II.6.1}$$

where $\hat{L}$ is defined as,

$$\hat{L} \triangleq \begin{pmatrix} 1 & & & & & \\ -b & 1 & & & & \\ b^2 & -b\frac{1-b^4}{1-b^2} & 1 & & & \\ -b^3 & b^2\frac{1-b^6}{1-b^2} & -b\frac{1-b^6}{1-b^2} & 1 & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ (-b)^{n-1} & (-b)^{n-2}\frac{1-b^{2n-2}}{1-b^2} & (-b)^{n-3}\frac{1-b^{2n-2}}{1-b^2}\frac{1-b^{2n-4}}{1-b^4} & (-b)^{n-4}\frac{1-b^{2n-2}}{1-b^2}\frac{1-b^{2n-4}}{1-b^4}\frac{1-b^{2n-6}}{1-b^6} & \dots & 1 \end{pmatrix}; \tag{II.6.2}$$

$$\hat{D} \triangleq \begin{pmatrix} 1 & & & & \\ & \frac{1}{(1-b^2)} & & & \\ & & \frac{1}{(1-b^2)(1-b^4)} & & \\ & & & \frac{1}{(1-b^2)(1-b^4)(1-b^6)} & \\ & & & & \ddots \\ & & & & & \frac{1}{(1-b^2)(1-b^4)\cdots(1-b^{2n-2})} \end{pmatrix} \tag{6.3}$$

The elements of these matrices are defined as

$$\hat{\lambda}_{ij} \triangleq \begin{cases} (-b)^{i-j}\frac{1-b^{2i-2}}{1-b^2}\frac{1-b^{2i-4}}{1-b^4}\cdots\frac{1-b^{2i-2j+2}}{1-b^{2j-2}} & i \geq j \\ 0 & i < j; \end{cases}$$

$$\hat{\delta}_{ij} \triangleq \begin{cases} \frac{1}{(1-b^2)(1-b^4)\cdots(1-b^{2i-2})} & i = j \\ 0 & i \neq j. \end{cases} \blacksquare$$

The full details of this theorem are in chapter II 6.

In the two-dimensional case, we use separability of the kernel to generalize not only to Gaussians with one spatial constant, but to two-dimensional kernels whose one-dimensional profiles are Gaussians of different spatial parameter, σ. In other words, we consider kernels of the form

$$\begin{aligned} g(x,y) &= e^{-x^2/2\sigma_1{}^2} \cdot e^{-y^2/2\sigma_2{}^2} \\ &= g(\sigma_1)\cdot g(\sigma_2). \end{aligned}$$

Then, similar to the one-dimensional case,

$$\begin{aligned} h_{i,j} &= \sum_{k=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} g_{i-k,m-j} f_{k,m} \\ &= \sum_{k=1}^{n_1}\sum_{m=1}^{n_2} g_{i-k}(\sigma_1) g_{j-m}(\sigma_2) f_{k,m} \\ &= \sum_{k=1}^{n_1} g_{i-k}(\sigma_1)\left[\sum_{m=1}^{n_2} f_{k,m} g_{m-j}(\sigma_2)\right]. \end{aligned}$$

Let,

$$H = [h_{i,j}], \quad i = 1,2,\ldots,n_1; \quad j = 1,2,\ldots,n_2$$

$$F = [f_{i,j}]; \quad i = 1,2,\ldots,n_1, \quad j = 1,2,\ldots,n_2$$

$$B^k = [\beta^k_{i,j}] \qquad \text{for} \quad k = 1,2; \quad i,j = 1,2\ldots,n_k$$

where

$$\beta^k_{i,j} = g_{j-i}(\sigma_k) = b_k^{(j-i)^2} \qquad \text{for} \quad k = 1,2, \quad i,j = 1,2\ldots,n_k$$

$$b_k = e^{-1/2\sigma_k^2} \qquad \text{for} \quad k = 1,2.$$

Therefore,

$$\boxed{H = B_1 F B_2} \tag{II.2.4}$$

This problem can be easily solved given that we know how to solve (II.2.2).

II.2.1 Overview of the Solution

There are several steps to inverting the blur matrix B analytically. First, note that B exhibits a regular pattern; one row is a shift of the previous one and also, as we shall show, the elements of each row are powers of one another. However, if one attempts to symbolically invert B, the analytic form of the inverse is elusive: as the dimension of B varies so does the analytic form of the full inverse.

Fortunately, B can be decomposed into less complicated submatrices whose analytic inverse is attainable. Suppose B is decomposed into a lower-triangular matrix L and an upper-triangular matrix R, such that

$$B = LR \tag{II.2.5}$$

Then,

$$B^{-1} = R^{-1}L^{-1}. \tag{II.2.6}$$

We will find these triangular matrices— L and R—(chapter II.3) and will show that they have inverses expressed with exact analytic formulas (chapters II.4 and II.5). This will make it possible to solve the system of equations analytically.

As a by-product of the above LU-decomposition of B, we easily find the Cholesky decompostion of B. This not only aids in algorithms with restricted storage space constraints, but also proves that B is positive-definite (appendix C).

However, the problem at hand is still an ill-conditioned one. In order to effectively deal with the numerical conditioning of this problem, we then derive a further decomposition of B^{-1} (based on the previous decompositions), such that

$$B^{-1} = \hat{L}^T \hat{D} \hat{L}, \tag{II.2.7}$$

where $\hat{L}$ is a lower triangular matrix, and $\hat{D}$ is a diagonal matrix †. Returning to issues of stability and conditioning, we are able to show that $\hat{L}$ is well-conditioned and $\hat{D}$ is ill-conditioned. This decomposition, therefore, unbraids the singularity of B and confines it to a simple diagonal matrix that can be dealt with appropriately. Thus, numerically stable implementations are not only possible, but are practical as well.

Finally, a brief comparison of some of the advantages and disadvantages of numerical versus symbolic methods is made in the chapter II.10.

† Note that T denotes transpose

Chapter II.3 The LU Decomposition of B

In this chapter we prove that B, the Toeplitz matrix derived from the Gaussian kernel, can be decomposed as the product of a lower-triangular-matrix, L, and an upper-triangular-matrix, R.

Theorem II.3.1:

$$B = LR,$$

where

$$B = \begin{pmatrix} 1 & b & b^4 & b^9 & \dots & b^{(n-1)^2} \\ b & 1 & b & b^4 & \dots & b^{(n-2)^2} \\ b^4 & b & 1 & b & \dots & b^{(n-3)^2} \\ b^9 & b^4 & b & 1 & \dots & b^{(n-4)^2} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b^{(n-1)^2} & b^{(n-2)^2} & b^{(n-3)^2} & b^{(n-4)^2} & \dots & 1 \end{pmatrix}, \qquad (II.2.3)$$

$$L \triangleq \begin{pmatrix} 1 & & & & \\ b & (1-b^2) & & & \\ b^4 & b(1-b^4) & (1-b^2)(1-b^4) & & \\ & & & & \\ b^{(n-1)^2} & b^{(n-2)^2}(1-b^{2n-2}) & b^{(n-3)^2}(1-b^{2n-2})(1-b^{2n-4}) & & [(1-b^{2n-2})(1-b^{2n-4}) \quad (1-b^2)] \end{pmatrix}.$$

$$(II.3.2)$$

and,

$$R \triangleq \begin{pmatrix} 1 & b & b^4 & b^9 & \cdot\cdot & b^{(n-1)^2} \\ & 1 & b\frac{1-b^4}{1-b^2} & b^4\frac{1-b^6}{1-b^2} & \cdot & b^{(n-2)^2}\frac{1-b^{2n-2}}{1-b^2} \\ & & 1 & b\frac{1-b^6}{1-b^2} & \cdot\cdot & b^{(n-3)^2}\frac{1-b^{2n-2}}{1-b^2}\frac{1-b^{2n-4}}{1-b^4} \\ & & & \ddots & \vdots & \vdots \\ & & & & & 1 \end{pmatrix}. \qquad (II.3.3)$$

The elements of the matrices B, L, and R are denoted as, β_{ij}, λ_{ij}, and ρ_{ij}, respectively. These are defined as follows.

$$\beta_{ij} \triangleq b^{(i-j)^2}. \qquad (II.3.4)$$

$$\lambda_{ij} \triangleq \begin{cases} b^{(i-j)^2}(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^{2i-2(j-1)}) & i \geq j \\ 0 & i < j \end{cases}; \qquad (II.3.5)$$

$$\rho_{ij} \triangleq \begin{cases} b^{(i-j)^2}\frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4}\dots\frac{1-b^{2j-2(i-1)}}{1-b^{2(i-1)}} & i \leq j \\ 0 & i > j \end{cases}. \qquad (II.3.6)$$

For a proof of this theorem see appendix B.

Chapter II.4

The Inverse of L

In this chapter we derive the inverse of the lower-triangular matrix, L, obtained from the LU-decomposition of B. We proceed by defining a lower-triangular matrix, $\tilde{L}$, and proving it is the inverse of L. This is done by showing

Theorem II.4.1:

$$L\tilde{L} = I. \qquad \blacksquare$$

Define $\tilde{L}$ as,

$$\tilde{L} \triangleq \begin{pmatrix} 1 & & & & \\ \frac{-b}{(1-b^2)} & \frac{1}{(1-b^2)} & & & \\ \frac{b^2}{(1-b^2)(1-b^4)} & \frac{-b}{(1-b^2)^2} & \frac{1}{(1-b^2)(1-b^4)} & & \\ \frac{-b^3}{(1-b^2)(1-b^4)(1-b^6)} & \frac{b^2}{(1-b^2)^2(1-b^4)} & \frac{-b}{(1-b^2)^2(1-b^4)} & \frac{1}{(1-b^2)(1-b^4)(1-b^6)} & \\ & & & & \\ \frac{(-b)^{n-1}}{(1-b^2)(1-b^4)\cdots(1-b^{2n-2})} & \frac{(-b)^{n-2}}{(1-b^2)^2(1-b^4)\cdots(1-b^{2n-4})} & & \cdots & \frac{1}{(1-b^2)(1-b^4)\cdots(1-b^{2n-2})} \end{pmatrix}; \tag{II.4.2}$$

where the elements of $\tilde{L}$ are,

$$\tilde{\lambda}_{ij} \triangleq \begin{cases} \frac{(-b)^{i-j}}{(1-b^2)(1-b^4)\cdots(1-b^{2i-2j})} \cdot \frac{1}{(1-b^2)(1-b^4)\cdots(1-b^{2j-2})} & i \geq j \\ 0 & i < j \end{cases}; \tag{II.4.3}$$

Proof:

We will show the product of the i^{th} row of L and the j^{th} column of $\tilde{L}$ is the ij^{th} element of the identity matrix, I. Clearly, for $j > i$ the product yields zero. So, we consider the case $j \leq i$.

The following is the i^{th} row of the L matrix (represented as a column matrix).

$$L_{row-i}{}^T = \begin{pmatrix} b^{(i-1)^2} \\ b^{(i-2)^2}(1-b^{2i-2}) \\ \vdots \\ b^{(i-j)^2}(1-b^{2i-2})(1-b^{2i-4})\ldots(1-b^{2i-2j+2}) \\ b^{(i-j-1)^2}(1-b^{2i-2})(1-b^{2i-4})\ldots(1-b^{2i-2j}) \\ b^{(i-j-2)^2}(1-b^{2i-2})(1-b^{2i-4})\ldots(1-b^{2i-2j-2}) \\ \vdots \\ b^{(i-(j+k))^2}(1-b^{2i-2})(1-b^{2i-4})\ldots(1-b^{2i-2(j+k)+2}) \\ \vdots \\ (1-b^{2i-2})(1-b^{2i-4})\ldots(1-b^2) \\ \vdots \end{pmatrix}.$$

Similarly, the following is the j^{th} column of $\tilde{L}$.

$$\tilde{L}_{column-j} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \cdot \frac{1}{(1-b^2)(1-b^4)\ldots(1-b^{2j-2})} \\ \frac{-b}{(1-b^2)} \cdot \frac{1}{(1-b^2)(1-b^4)\ldots(1-b^{2j-2})} \\ \frac{(-b)^2}{(1-b^2)(1-b^4)} \cdot \frac{1}{(1-b^2)(1-b^4)\ldots(1-b^{2j-2})} \\ \vdots \\ \frac{(-b)^{(j+k)-j}}{(1-b^2)(1-b^4)\ldots(1-b^{2(j+k)-2j})} \cdot \frac{1}{(1-b^2)(1-b^4)\ldots(1-b^{2j-2})} \\ \vdots \\ \frac{(-b)^{i-j}}{(1-b^2)(1-b^4)\ldots(1-b^{2i-2j})} \cdot \frac{1}{(1-b^2)(1-b^4)\ldots(1-b^{2j-2})} \\ \vdots \end{pmatrix}.$$

The product, then, is

$$
\begin{aligned}
\tilde{L}_{row-i} L_{column-j} = & b^{(i-j)^2}(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^{2i-2j+2})\cdot \\
& \frac{1}{(1-b^2)(1-b^4)\dots(1-b^{2j-2})} + \\
& b^{(i-j-1)^2}(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^{2i-2j})\cdot \\
& \frac{-b}{(1-b^2)}\cdot\frac{1}{(1-b^2)(1-b^4)\dots(1-b^{2j-2})} + \\
& b^{(i-j-2)^2}(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^{2i-2j-2})\cdot \\
& \frac{(-b)^2}{(1-b^2)(1-b^4)}\cdot\frac{1}{(1-b^2)(1-b^4)\dots(1-b^{2j-2})} + \\
& + \\
& b^{(i-(j+k))^2}(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^{2i-2(j+k)+2})\cdot \\
& \frac{(-b)^k}{(1-b^2)(1-b^4)\dots(1-b^{2k})}\cdot\frac{1}{(1-b^2)(1-b^4)\dots(1-b^{2j-2})} + \\
& \vdots \quad + \\
& (1-b^{2i-2})(1-b^{2i-4})\dots(1-b^2)\cdot \\
& \frac{(-b)^{i-j}}{(1-b^2)(1-b^4)\dots(1-b^{2i-2j})}\cdot\frac{1}{(1-b^2)(1-b^4)\dots(1-b^{2j-2})}.
\end{aligned}
$$

Or,

$$
\begin{aligned}
\tilde{L}_{row-i} L_{column-j} = \frac{(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^{2i-2j+2})}{(1-b^2)(1-b^4)\dots(1-b^{2j-2})} \cdot
&\Big\{ b^{(i-j)^2} + \\
&b^{(i-j-1)^2}(1-b^{2i-2j}) \cdot \frac{-b}{(1-b^2)} + \\
&b^{(i-j-2)^2}(1-b^{2i-2j})(1-b^{2i-2j-2}) \cdot \frac{(-b)^2}{(1-b^2)(1-b^4)} + \\
&+ \\
&b^{(i-(j+k))^2}(1-b^{2i-2j})(1-b^{2i-2j-2})\dots(1-b^{2i-2(j+k)+2}) \cdot \\
&\frac{(-b)^k}{(1-b^2)(1-b^4)\dots(1-b^{2k})} + \\
&+ \\
&(1-b^{2i-2j})(1-b^{2i-2j-2})\dots(1-b^2) \cdot \\
&\frac{(-b)^{i-j}}{(1-b^2)(1-b^4)\dots(1-b^{2i-2j})} \Big\}.
\end{aligned}
$$

To sum the above expression, we can name each term explicitly and use induction to arrive at an analytic expression. Define,

$$\Theta_0 \triangleq b^{(i-j)^2};$$

$$\Theta_1 \triangleq -b^{(i-j-1)^2+1} \frac{1-b^{2i-2j}}{1-b^2};$$

$$\Theta_2 \triangleq b^{(i-j-2)^2+2} \frac{1-b^{2i-2j}}{1-b^2} \frac{1-b^{2i-2j-2}}{1-b^4};$$

$$\vdots$$

$$\Theta_k \triangleq (-1)^k b^{(i-j-k)^2+k} \frac{1-b^{2i-2j}}{1-b^2} \frac{1-b^{2i-2j-2}}{1-b^4} \dots \frac{1-b^{2i-2j-2k+2}}{1-b^{2k}};$$

$$\vdots$$

$$\Theta_{i-j} = (-1)^{i-j} b^{i-j} \frac{1-b^{2i-2j}}{1-b^2} \frac{1-b^{2i-2j-2}}{1-b^4} \cdots \frac{1-b^2}{1-b^{2i-2j}}.$$

Now, let's look at the sum of the first two terms in the braces above

$$\begin{aligned}
\Theta_0 + \Theta_1 &= b^{(i-j-1)^2+1}\left[b^{2i-2j-2} - \frac{1-b^{2i-2j}}{1-b^2}\right] \\
&= -b^{(i-j-1)^2+1}\frac{1-b^{2i-2j-2}}{1-b^2}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
\Theta_0 + \Theta_1 + \Theta_2 &= -b^{(i-j-1)^2+1}\frac{1-b^{2i-2j-2}}{1-b^2} + \\
&\quad b^{(i-j-2)^2+2}\frac{1-b^{2i-2j}}{1-b^2}\frac{1-b^{2i-2j-2}}{1-b^4} \\
&= b^{(i-j-2)^2+2}\frac{1-b^{2i-2j-2}}{1-b^2}\frac{1-b^{2i-2j-4}}{1-b^4}.
\end{aligned}$$

This suggests the following:

Lemma II.4.4:

$$\sum_{m=0}^{k} \Theta_m = (-1)^k b^{(i-j-k)^2+k}\frac{1-b^{2i-2j-2}}{1-b^2}\frac{1-b^{2i-2j-4}}{1-b^4}\cdots\frac{1-b^{2i-2j-2k}}{1-b^{2k}}.$$

Proof : We have proved the case for $k = 0, 1, 2$. It remains to show that if the assertion is true for k, then it is also true for $k+1$. So, assume that the above expression holds true for k. Then,

$$\sum_{m=0}^{k+1} \Theta_m = \sum_{m=0}^{k} \Theta_m + \Theta_{k+1}$$

$$= (-1)^k b^{(i-j-k)^2+k} \frac{1-b^{2i-2j-2}}{1-b^2}\frac{1-b^{2i-2j-4}}{1-b^4}\cdots\frac{1-b^{2i-2j-2k}}{1-b^{2k}} +$$

$$(-1)^{k+1} b^{(i-j-k-1)^2+k+1} \frac{1-b^{2i-2j}}{1-b^2}\frac{1-b^{2i-2j-2}}{1-b^4}\cdots\frac{1-b^{2i-2j-2k}}{1-b^{2k+2}}$$

$$= (-1)^{k+1} b^{(i-j-k-1)^2+k+1} \frac{1-b^{2i-2j-2}}{1-b^2}\frac{1-b^{2i-2j-4}}{1-b^4}\cdots\frac{1-b^{2i-2j-2k}}{1-b^{2k}} \cdot \left[-b^{2i-2j-2k-2} + \frac{1-b^{2i-2j}}{1-b^{2k+2}}\right]$$

$$= (-1)^{k+1} b^{(i-j-k-1)^2+k+1} \frac{1-b^{2i-2j-2}}{1-b^2}\frac{1-b^{2i-2j-4}}{1-b^4}\cdots\frac{1-b^{2i-2j-2k-2}}{1-b^{2k+2}}. \quad \blacksquare$$

Returning to the proof of theorem II.4.1, we had,

$$\tilde{L}_{row-i} L_{column-j} = \frac{(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^{2i-2j+2})}{(1-b^2)(1-b^4)\dots(1-b^{2j-2})} \cdot \sum_{m=0}^{i-j} \Theta_m.$$

where

$$\sum_{m=0}^{i-j} \Theta_m = \begin{cases} 1 & i=j \\ \sum_{m=0}^{i-j-1} \Theta_m + \Theta_{i-j} & i>j \end{cases} :$$

In the case of $i > j$, we use the lemma to prove,

$$\sum_{m=0}^{i-j} \Theta_m = \sum_{m=0}^{i-j-1} \Theta_m + \Theta_{i-j}$$

$$= (-1)^{i-j-1} b^{1^2+i-j-1} \frac{1-b^{2i-2j-2}}{1-b^2}\frac{1-b^{2i-2j-4}}{1-b^4}\cdots\frac{1-b^2}{1-b^{2i-2j-2}} + (-1)^{i-j} b^{i-j}$$

$$= 0.$$

In conclusion,

$$\tilde{L}_{row-i} L_{column-j} = \begin{cases} 0 & i<j; \\ 1 & i=j; \\ 0 & i>j; \end{cases}$$

or,

$$\tilde{L}_{row-i} L_{column-j} = I_{ij}.$$

It follows that $\tilde{L}$ is the inverse of L, which proves the theorem.

Chapter II.5 The Inverse of R

This chapter is the analog of the previous chapter; in it we define a matrix $\tilde{R}$ and prove that it is the inverse to R, the upper-triangular obtained from the LU-decomposition of B, by showing

Theorem II.5.1:

$$\tilde{R}R = I.$$

Define $\tilde{R}$ as,

$$\tilde{R} \triangleq \begin{pmatrix}
1 & -b & b^2 & -b^3 & b^4 & \cdots & (-b)^{n-1} \\
 & 1 & -b\frac{1-b^4}{1-b^2} & b^2\frac{1-b^6}{1-b^2} & -b^3\frac{1-b^8}{1-b^2} & \cdots & (-b)^{n-2}\frac{1-b^{2n-2}}{1-b^2} \\
 & & 1 & -b\frac{1-b^6}{1-b^2} & b^2\frac{1-b^8}{1-b^2}\frac{1-b^6}{1-b^2} & \cdots & (-b)^{n-3}\frac{1-b^{2n-2}}{1-b^2}\frac{1-b^{2n-4}}{1-b^4} \\
 & & & 1 & -b\frac{1-b^8}{1-b^2} & \cdots & (-b)^{n-4}\frac{1-b^{2n-2}}{1-b^2}\frac{1-b^{2n-4}}{1-b^4}\frac{1-b^{2n-6}}{1-b^6} \\
 & & & & 1 & \cdots & (-b)^{n-5}\frac{1-b^{2n-2}}{1-b^2}\frac{1-b^{2n-4}}{1-b^4}\frac{1-b^{2n-6}}{1-b^6}\frac{1-b^{2n-8}}{1-b^8} \\
 & & & & & & \vdots \\
 & & & & & & 1
\end{pmatrix} \qquad (II.5.2)$$

where the elements of $\tilde{R}$ are,

$$\tilde{\rho}_{ij} \triangleq \begin{cases} 0 & i > j \\ (-b)^{j-i} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i+2}}{1-b^{2i-2}} & i \leq j. \end{cases} \qquad (II.5.3)$$

We follow the same path as the previous chapters in that the i^{th} row of $\tilde{R}$ times the j^{th} column of R is shown to equal I_{ij}. Clearly, for $i > j$, this product is zero. Hence, we only consider the case $i < j$.

The following is the i^{th} row of $\tilde{R}$ (represented as a column vector)

$$\tilde{R}^T_{row-i} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ -b\frac{1-b^{2i}}{1-b^2} \\ (-b)^2 \frac{1-b^{2i+2}}{1-b^2} \frac{1-b^{2i}}{1-b^4} \\ (-b)^3 \frac{1-b^{2i+4}}{1-b^2} \frac{1-b^{2i+2}}{1-b^4} \frac{1-b^{2i}}{1-b^6} \\ \\ (-b)^k \frac{1-b^{2i+2k-2}}{1-b^2} \frac{1-b^{2i+2k-4}}{1-b^4} \cdots \frac{1-b^{2k+2}}{1-b^{2i-2}} \\ \vdots \\ (-b)^{j-i} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i+2}}{1-b^{2i-2}} \\ \vdots \end{pmatrix}$$

Similarly, the following is the j^{th} column of the R matrix.

$$
R_{column-j} = \begin{pmatrix}
b^{(j-1)^2} \\
b^{(j-2)^2} \frac{1-b^{2j-2}}{1-b^2} \\
\\
b^{(j-i)^2} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2(i-1)}}{1-b^{2(i-1)}} \\
b^{(j-(i+1))^2} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i}}{1-b^{2i}} \\
b^{(j-(i+2))^2} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2(i+1)}}{1-b^{2(i+1)}} \\
b^{(j-(i+3))^2} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2(i+2)}}{1-b^{2(i+2)}} \\
\vdots \\
b^{(j-(i+k))^2} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2(i+k-1)}}{1-b^{2(i+k-1)}} \\
\vdots \\
1 \\
0 \\
0 \\
\vdots
\end{pmatrix}
$$

Therefore, the product is,

$$
\begin{aligned}
\tilde{R}_{row-i} R_{column-j} = {} & 1 \cdot b^{(j-i)^2} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2(i-1)}}{1-b^{2(i-1)}} + \\
& -b \frac{1-b^{2i}}{1-b^2} \cdot b^{(j-(i+1))^2} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i}}{1-b^{2i}} + \\
& (-b)^2 \frac{1-b^{2i+2}}{1-b^2} \frac{1-b^{2i}}{1-b^4} \cdot b^{(j-(i+2))^2} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2(i+1)}}{1-b^{2(i+1)}} + \\
& (-b)^3 \frac{1-b^{2i+4}}{1-b^2} \frac{1-b^{2i+2}}{1-b^4} \frac{1-b^{2i}}{1-b^6} \cdot b^{(j-(i+3))^2} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2(i+2)}}{1-b^{2(i+2)}} + \\
& \qquad \vdots \qquad + \\
& (-b)^k \frac{1-b^{2i+2k-2}}{1-b^2} \frac{1-b^{2i+2k-4}}{1-b^4} \cdots \frac{1-b^{2k+2}}{1-b^{2i-2}} \cdot \\
& \qquad\qquad b^{(j-(i+k))^2} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2(i+k-1)}}{1-b^{2(i+k-1)}} + \\
& \qquad \vdots \qquad + \\
& 1 \cdot (-b)^{j-i} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i+2}}{1-b^{2i-2}} .
\end{aligned}
$$

As before, we need to name each term explicitly and use induction to sum all the terms. Therefore, define

$$\Upsilon_0 \triangleq b^{(j-i)^2} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2(i-1)}}{1-b^{2(i-1)}}.$$

$$\Upsilon_1 \triangleq (-1) b^{(j-i-1)^2+1} \frac{1-b^{2i}}{1-b^2} \cdot \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i}}{1-b^{2i}};$$

$$\Upsilon_2 \triangleq (-1)^2 b^{(j-i-2)^2+2} \frac{1-b^{2i+2}}{1-b^2} \frac{1-b^{2i}}{1-b^4} \cdot \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2(i+1)}}{1-b^{2(i+1)}};$$

$$\Upsilon_3 \triangleq (-1)^3 b^{(j-i-3)^2+3} \frac{1-b^{2i+4}}{1-b^2} \frac{1-b^{2i+2}}{1-b^4} \frac{1-b^{2i}}{1-b^6} \cdot \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2(i+2)}}{1-b^{2(i+2)}};$$

$$\vdots$$

$$\begin{aligned} \Upsilon_k \triangleq (-1)^k b^{(j-i-k)^2+k} & \frac{1-b^{2i+2k-2}}{1-b^2} \frac{1-b^{2i+2k-4}}{1-b^4} \cdots \frac{1-b^{2k+2}}{1-b^{2i-2}} \cdot \\ & \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2(i+k-1)}}{1-b^{2(i+k-1)}}; \end{aligned}$$

$$\vdots$$

$$\Upsilon_{j-i} \triangleq (-1)^{j-i} b^{j-i} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i+2}}{1-b^{2i-2}}.$$

Then, we have

$$\bar{R}_{row-i} R_{column-j} = \sum_{m=0}^{j-i} \Upsilon_m.$$

Let us consider sum of two terms.

$$
\begin{aligned}
\Upsilon_0 + \Upsilon_1 &= b^{(j-i)^2} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2(i-1)}}{1-b^{2(i-1)}} + \\
&\quad - b^{(j-i-1)^2+1} \frac{1-b^{2i}}{1-b^2} \cdot \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i}}{1-b^{2i}} \\
&= b^{(j-i-1)^2+1} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i+2}}{1-b^{2i-2}} \cdot \\
&\qquad \left[b^{2j-2i-2} - \frac{1-b^{2j-2i}}{1-b^2} \right] \\
&= -b^{(j-i-1)^2+1} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i+2}}{1-b^{2i-2}} \cdot \frac{1-b^{2j-2i-2}}{1-b^2} .
\end{aligned}
$$

Similarly.

$$
\begin{aligned}
\Upsilon_0 + \Upsilon_1 + \Upsilon_2 &= -b^{(j-i-1)^2+1} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i+2}}{1-b^{2i-2}} \cdot \frac{1-b^{2j-2i-2}}{1-b^2} + \\
&\quad b^{(j-i-2)^2+2} \frac{1-b^{2i+2}}{1-b^2} \frac{1-b^{2i}}{1-b^4} \cdot \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2(i+1)}}{1-b^{2(i+1)}} \\
&= -b^{(j-i-2)^2+2} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i+2}}{1-b^{2i-2}} \cdot \frac{1-b^{2j-2i-2}}{1-b^2} \\
&\qquad \left[b^{2j-2i-4} - \frac{1-b^{2j-2i}}{1-b^4} \right] \\
&= b^{(j-i-2)^2+2} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i+2}}{1-b^{2i-2}} \cdot \frac{1-b^{2j-2i-2}}{1-b^2} \frac{1-b^{2j-2i-4}}{1-b^4} ,
\end{aligned}
$$

which leads us to the following lemma

Lemma II.5.4:

$$\sum_{m=0}^{k} \Upsilon_m = (-1)^k b^{(j-i-k)^2+k} \frac{1-b^{2j-2}}{1-b^2}\,\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2i+2}}{1-b^{2i-2}}\cdot$$
$$\frac{1-b^{2j-2i-2}}{1-b^2}\,\frac{1-b^{2j-2i-4}}{1-b^4}\cdots\frac{1-b^{2j-2i-2k}}{1-b^{2k}}.$$

Proof : We have proved the case for $k = 0, 1, 2$. It remains to show that the above assertion holds for $k+1$ if it is true for k. Assume, therefore, it is true for k. Then,

$$\begin{aligned}
\sum_{m=0}^{k+1} \Upsilon_m = \sum_{m=0}^{k} \Upsilon_m + \Upsilon_{k+1} \\
&= (-1)^k b^{(j-i-k)^2+k} \frac{1-b^{2j-2}}{1-b^2}\,\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2i+2}}{1-b^{2i-2}}\cdot \\
&\qquad \frac{1-b^{2j-2i-2}}{1-b^2}\,\frac{1-b^{2j-2i-4}}{1-b^4}\cdots\frac{1-b^{2j-2i-2k}}{1-b^{2k}}+ \\
&\quad (-1)^{k+1} b^{(j-i-k-1)^2+k+1} \frac{1-b^{2i+2k}}{1-b^2}\,\frac{1-b^{2i+2k-2}}{1-b^4}\cdots\frac{1-b^{2k+4}}{1-b^{2i-2}}\cdot \\
&\qquad \frac{1-b^{2j-2}}{1-b^2}\,\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2i-2k}}{1-b^{2i+2k}}
\end{aligned} \tag{II.5.5}$$

Rearrange Υ_{k+1} as

$$
\begin{aligned}
\Upsilon_{k+1} &= (-1)^{k+1} b^{(j-i-k-1)^2+k+1} \frac{1-b^{2j-2}}{1-b^2}\,\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2i+2}}{1-b^{2i-2}}\cdot \\
&\qquad \frac{1-b^{2i+2k}}{1-b^2}\,\frac{1-b^{2i+2k-2}}{1-b^4}\cdots\frac{1-b^{2k+4}}{1-b^{2i-2}}\cdot \\
&\qquad \frac{1-b^{2j-2i}}{1-b^{2i}}\,\frac{1-b^{2j-2i-2}}{1-b^{2i+2}}\cdots\frac{1-b^{2j-2i-2k}}{1-b^{2i+2k}} \\
&= (-1)^{k+1} b^{(j-i-k-1)^2+k+1} \frac{1-b^{2j-2}}{1-b^2}\,\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2i+2}}{1-b^{2i-2}}\cdot \\
&\qquad \frac{1-b^{2i+2k}}{1-b^2}\,\frac{1-b^{2i+2k-2}}{1-b^4}\cdots\frac{1-b^{2i}}{1-b^{2k+2}}\cdot \\
&\qquad \frac{1-b^{2j-2i}}{1-b^{2i}}\,\frac{1-b^{2j-2i-2}}{1-b^{2i+2}}\cdots\frac{1-b^{2j-2i-2k}}{1-b^{2i+2k}} \\
&= (-1)^{k+1} b^{(j-i-k-1)^2+k+1} \frac{1-b^{2j-2}}{1-b^2}\,\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2i+2}}{1-b^{2i-2}}\cdot \\
&\qquad \frac{1-b^{2j-2i-2}}{1-b^2}\,\frac{1-b^{2j-2i-4}}{1-b^4}\cdots\frac{1-b^{2j-2i-2k}}{1-b^{2k}}\cdot\frac{1-b^{2j-2i}}{1-b^{2k+2}}.
\end{aligned}
$$

Substituting back into the sum expression (II.5.5), we get

$$
\begin{aligned}
\sum_{m=0}^{k+1} \Upsilon_m = {} & (-1)^k b^{(j-i-k)^2+k} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i+2}}{1-b^{2i-2}} \cdot \\
& \qquad \frac{1-b^{2j-2i-2}}{1-b^2} \frac{1-b^{2j-2i-4}}{1-b^4} \cdots \frac{1-b^{2j-2i-2k}}{1-b^{2k}} + \\
& (-1)^{k+1} b^{(j-i-k-1)^2+k+1} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i+2}}{1-b^{2i-2}} \cdot \\
& \qquad \frac{1-b^{2j-2i-2}}{1-b^2} \frac{1-b^{2j-2i-4}}{1-b^4} \cdots \frac{1-b^{2j-2i-2k}}{1-b^{2k}} \cdot \frac{1-b^{2j-2i}}{1-b^{2k+2}} \\
= {} & (-1)^{k+1} b^{(j-i-k-1)^2+k+1} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i+2}}{1-b^{2i-2}} \cdot \\
& \qquad \frac{1-b^{2j-2i-2}}{1-b^2} \frac{1-b^{2j-2i-4}}{1-b^4} \cdots \frac{1-b^{2j-2i-2k}}{1-b^{2k}} \\
& \qquad \left[-b^{2j-2k-2k-2} + \frac{1-b^{2j-2i}}{1-b^{2k+2}} \right] \\
= {} & (-1)^{k+1} b^{(j-i-k-1)^2+k+1} \frac{1-b^{2j-2}}{1-b^2} \frac{1-b^{2j-4}}{1-b^4} \cdots \frac{1-b^{2j-2i+2}}{1-b^{2i-2}} \cdot \\
& \qquad \frac{1-b^{2j-2i-2}}{1-b^2} \frac{1-b^{2j-2i-4}}{1-b^4} \cdots \frac{1-b^{2j-2i-2k}}{1-b^{2k}} \\
& \qquad \frac{1-b^{2j-2i-2k-2}}{1-b^{2k+2}}.
\end{aligned}
$$

which is exactly what we set out to prove in lemma II.5.4.

Now, returning to theorem II.5.1, if $i = j$,

$$\tilde{R}_{row-i} R_{column-j} = 1.$$

For $i < j$,

$$\begin{aligned}\sum_{m=0}^{j-i} \Upsilon_m &= \sum_{m=0}^{j-i-1} \Upsilon_m + \Upsilon_{j-i} \\ &= (-1)^{j-i-1} b^{j-i} \frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2i+2}}{1-b^{2i-2}} + \\ &\quad (-1)^{j-i} b^{j-i} \frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2i+2}}{1-b^{2i-2}} \\ &= 0.\end{aligned}$$

To summarize,

$$\tilde{R}_{row-i} R_{column-j} = \begin{cases} 0 & i < j \, . \\ 1 & i = j \, ; \\ 0 & i > j \, . \end{cases}$$

or,

$$\tilde{R}_{row-i} R_{column-j} = I_{ij}.$$

It follows that $\tilde{R}$ is the inverse of R, thus completing the proof of theorem II.5.1.

Chapter II.6 The $L^T DL$ Decomposition of the Inverse of B

In this chapter, we give a decomposition of B^{-1} which we derive from the LU-decomposition of B. We will define three matrices: $\hat{L}$ a lower-triangular matrix, $\hat{D}$ a diagonal matrix, and $\hat{R}$ an upper-triangular matrix. Then, we will show that

$$B^{-1} = \hat{R}\hat{D}\hat{L}.$$

Furthermore, we will find that,

$$\hat{R}^T = \hat{L}$$

so that,

Theorem II.6.1:

$$B^{-1} = \hat{L}^T \hat{D}\hat{L}. \quad \blacksquare \qquad (II.6.1)$$

Proof: We proceed as follows. From the LU-decomposition of B we get L and R, which in turn gives a decomposition of B^{-1}. Then, we decompose the lower-triangular matrix $\tilde{L}$ (recall this is the inverse of L) into another lower-triangular matrix $\hat{L}$ and a diagonal matrix $\hat{D}$. Define

$$\hat{L} \triangleq \begin{pmatrix} 1 & & & & & \\ -b & 1 & & & & \\ b^2 & -b\frac{1-b^4}{1-b^2} & 1 & & & \\ -b^3 & b^2\frac{1-b^6}{1-b^2} & -b\frac{1-b^6}{1-b^2} & 1 & & \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ (-b)^{n-1} & (-b)^{n-2}\frac{1-b^{2n-2}}{1-b^2} & (-b)^{n-3}\frac{1-b^{2n-2}}{1-b^2}\frac{1-b^{2n-4}}{1-b^4} & (-b)^{n-4}\frac{1-b^{2n-2}}{1-b^2}\frac{1-b^{2n-4}}{1-b^4}\frac{1-b^{2n-6}}{1-b^6} & \cdots & 1 \end{pmatrix} \tag{II.6.2}$$

$$\hat{D} \triangleq \begin{pmatrix} 1 & & & & \\ & \frac{1}{(1-b^2)} & & & \\ & & \frac{1}{(1-b^2)(1-b^4)} & & \\ & & & \frac{1}{(1-b^2)(1-b^4)(1-b^6)} & \\ & & & & \ddots \\ & & & & & \frac{1}{(1-b^2)(1-b^4)\cdots(1-b^{2n-2})} \end{pmatrix} \tag{II.6.3}$$

The elements of these matrices are defined as

$$\hat{\lambda}_{ij} \triangleq \begin{cases} (-b)^{i-j}\frac{1-b^{2i-2}}{1-b^2}\frac{1-b^{2i-4}}{1-b^4}\cdots\frac{1-b^{2i-2j+2}}{1-b^{2j-2}} & i \geq j \\ 0 & i < j; \end{cases} \tag{II.6.4}$$

$$\hat{\delta}_{ij} \triangleq \begin{cases} \frac{1}{(1-b^2)(1-b^4)\cdots(1-b^{2i-2})} & i = j \\ 0 & i \neq j \end{cases}; \tag{II.6.5}$$

respectively. Then,

$$
\begin{aligned}
(\hat{D}\hat{L})_{ij} &= \sum_{k=1}^{n} \hat{\delta}_{ik}\hat{\lambda}_{kj} \\
&= \hat{\delta}_{ii}\hat{\lambda}_{ij} \\
&= \begin{cases} \frac{(-b)^{i-j}}{(1-b^2)(1-b^4)\cdots(1-b^{2i-2j})} \cdot \frac{1}{(1-b^2)(1-b^4)\cdots(1-b^{2j-2})} & i \geq j \\ 0 & i < j \end{cases} \\
&= \tilde{\lambda}_{ij}.
\end{aligned}
$$

Therefore,

$$L^{-1} = \tilde{L} = \hat{D}\hat{L}.$$

Also, define,

$$\hat{R} \triangleq \tilde{R} = R^{-1}.$$

At this point we note that $\hat{R}^T = \hat{L}$. Using this decomposition of $\tilde{L}$, we conclude

$$
\begin{aligned}
B^{-1} &= R^{-1}L^{-1} \\
&= \hat{L}^T\hat{D}\hat{L}.
\end{aligned}
$$

Note the special property of this decomposition: namely, that $\hat{L}$ is a well-conditioned matrix. This implies that we have isolated the ill-conditioning into a diagonal matrix - $\hat{D}$- thereby greatly simplifying the handling of possible numerical problems. In this sense, then, the L^TDL decomposition might be said to be optimal.

Chapter II.7 Examples

In this chapter we provide examples of discrete deblurring. The images were obtained in two different ways: (1) by simulated blur, and (2) by a realistic optical source of blur.

The simulated blur was implemented using discrete convolution against a Gaussian. Then, we use the $\hat{L}^T\hat{D}\hat{L}$ decomposition of B^{-1} to deblur them. Recall that

$$h = Bf. \tag{II.2.2}$$

Therefore, to obtain f from h, we use

$$f = \hat{L}^T\hat{D}\hat{L}h. \tag{II.7.1}$$

For images the problem is obtaining F from H,

$$H = B_1 F B_2 \tag{II.2.4}$$

which is readily solved as

$$F = (\hat{L}_1^T\hat{D}_1\hat{L}_1) \quad H \quad (\hat{L}_2^T\hat{D}_2\hat{L}_2). \tag{II.7.2}$$

The original and blurred images of SQUARE, CORNER, and CITY are displayed in the (.a) and (.b) part of figures II.1, II.2, and II.3, respectively. The SQUARE image was chosen to illustrate deblurring of images of geometric structures. The CORNER image examines a subportion of SQUARE at higher resolution. Finally, the CITY image carries the demonstration to an image of a natural scene very rich in structure. The deblurred images are found in the (.c) part of the figure. Note that the reconstruction is perfect in the case of spatially invariant Gaussian blur of known spatial extent and high accuracy representation.

Since the problem with Gaussian deblurring is (in part) numerical, in the next series of experiments we varied the accuracy of the initial given information The SQUARE, CORNER, and CITY images were accurately convolved against a Gaussian and then represented by 4, 6, 8, 10, 12, 14, 16, 18, 24, and 30 bits/pixel The deblurring value is the best possible for each representation, and was chosen by comparing the results of deblurring with several values. The results are shown in figures II.4, II.5, and II.6 respectively Note that with 4 bits of accuracy some minimal deblurring is achieved But as the accuracy is increased, our method gives progressively better results. At 30 bits, the reconstruction is virtually perfect.

A plot of performance versus representation accuracy for the CITY series is shown in figure II.7. Accuracy is in bits/pixel. Performance is the normalized norm difference between the (known) original, F, and the reconstructed image, F_R.

$$\text{performance} = \frac{\|F - F_R\|}{\|F\|}, \tag{II.7.3}$$

where,

$$\|F\| = [\sum_{i,j} F_{ij}^2]^{1/2}. \tag{II.7.4}$$

Our last example is the EYECHART image with several different deblurring values. In this case, the source of blur is optical: the camera lens was defocused before the image was scanned, and the result digitized to 8 bits/pixel quantization. The results in figure II.8 show the well-known trade off between reconstruction of the image and avoiding noise amplification The best possible deblurring value depends on the accuracy of the given degraded image.

There are some technical points about the deblurring algorithm that are relevent to successful implementations. First, the fractional elements of $\hat{L}$ should not be computed in a straightforward way Rather, factors should be cancelled (which is possible in every case), in order to avoid numerically unstable computation of these elements This is especially true for values of b close to 1 (corresponding to a large amount of blur). Second, elements of $\hat{D}$, the matrix which is the ill-conditioned part of this problem, should be handled with extreme care. Observe that the elements of $\hat{D}$ increase very rapidly in magnitude moving down the diagonal. Therefore, one should use specialized "long-word" arithmetic functions to deal with them. This is affordable since the complexity increases linearly with size. Also, note the elements of $\hat{D}$ are highly correlated, the next element on the diagonal is easily computed from the present element. This fact can be utilized to obtain high accuracy.

The accuracy of deblurring is directly affected by numerical quantization: for a given word length stable deblurring only appears possible up to a cutoff blur parameter b_0. For instance for Fortran REAL*8 accuracy (inner products are also computed REAL*8), the cutoff point is approximately $b_0 = 0.85$. For values of b less b_0, the algorithm always performs perfectly, given complete information about the blurred image (i.e. no truncation occured after blurring.) However, slightly beyond this point, i.e. $b > b_0$, the performance drops drastically. Thus, for a given numerical accuracy, one can determine how much blur can be removed reliably. Note, however, that the upper limit of deblurring value in our examples was decided by the accuracy in the representation of the degraded image rather than the internal accuracy in representation and arithmetic. The plot in figure II.7 supports this point.

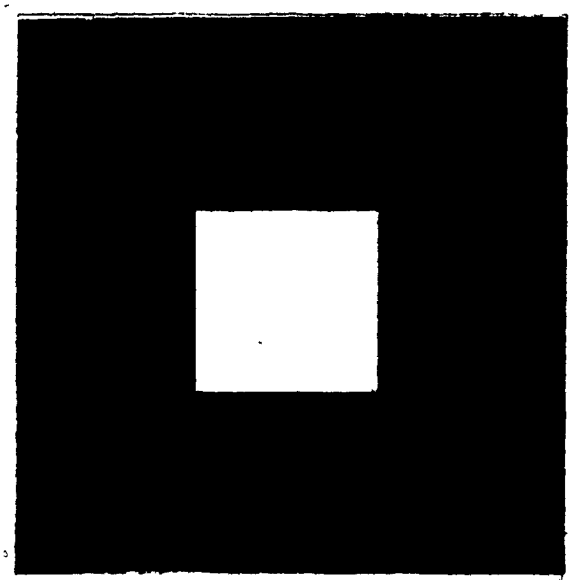

Figure II.1.a The original image, SQUARE at 40x40 spatial resolution, 6 bits/pixel. The image was chosen to illustrate the consequences of deblurring in detail for a structured image.

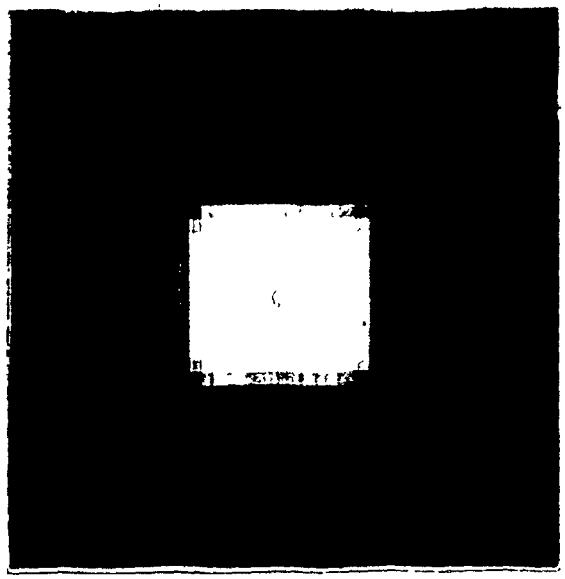

Figure II.1.b The image SQUARE blurred with $k = 0.80$

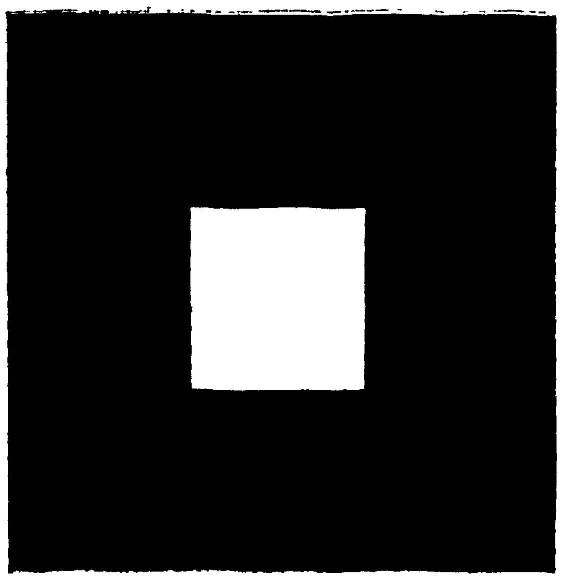

Figure II.1.c The deblurred image, using $k = 0.80$. Again note that the deblurring is effectively perfect.

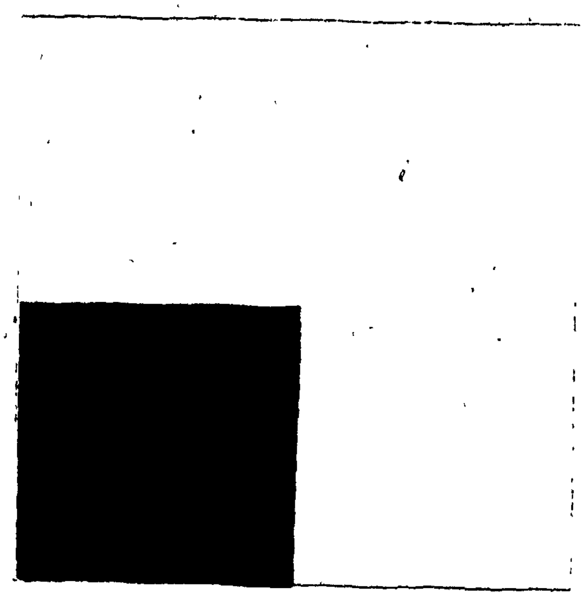

Figure II.2.a The original image CORNER at 100x100 spatial resolution. 6 bits/pixel

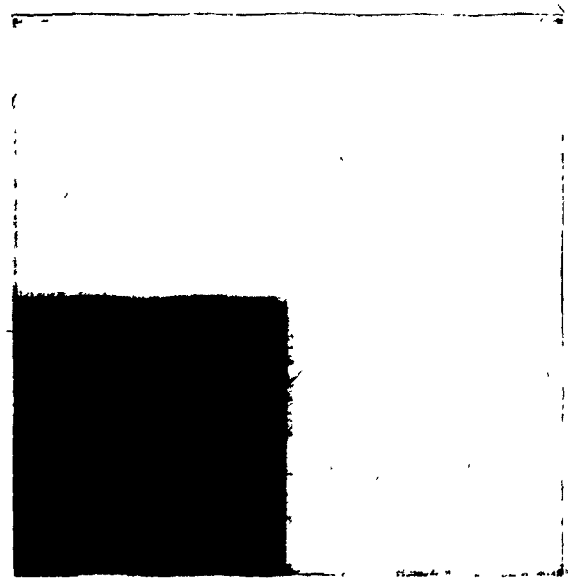

Figure II.2.b The image CORNER blurred with $b = 0.80$

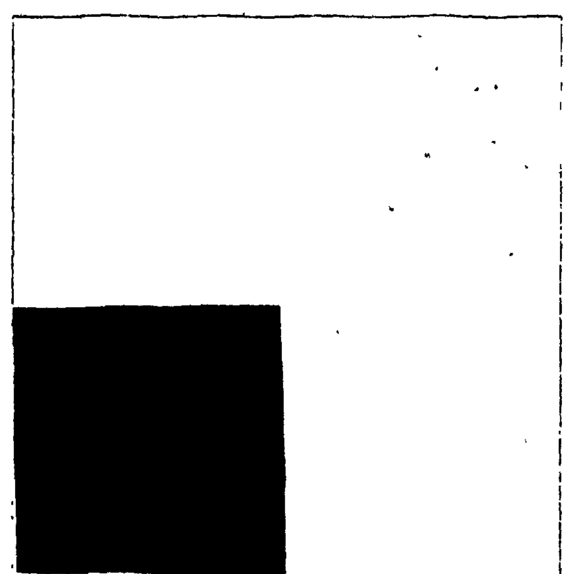

Figure II.2.c The deblurred image, using $b = 0.80$. Note that with high accuracy representation of the blurred image, the representation is effectively perfect

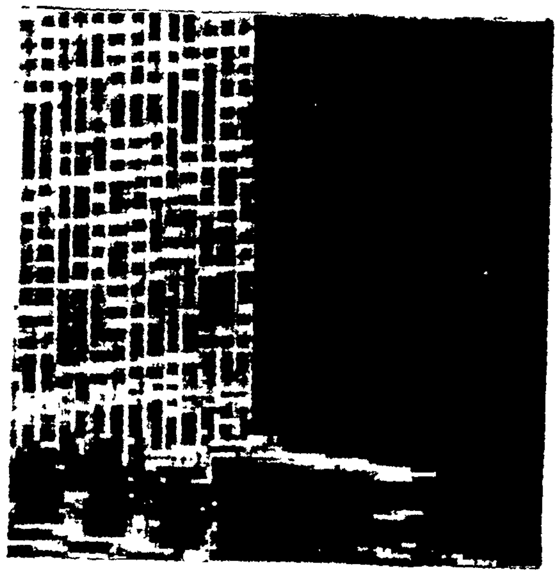

Figure II.3.a The original image CITY at 100x100 spatial resolution 6 bits/pixel This image is chosen to demonstrate deblurring of a known structure at higher resolution

Figure II.3.b The image CITY blurred with $h = 0.80$

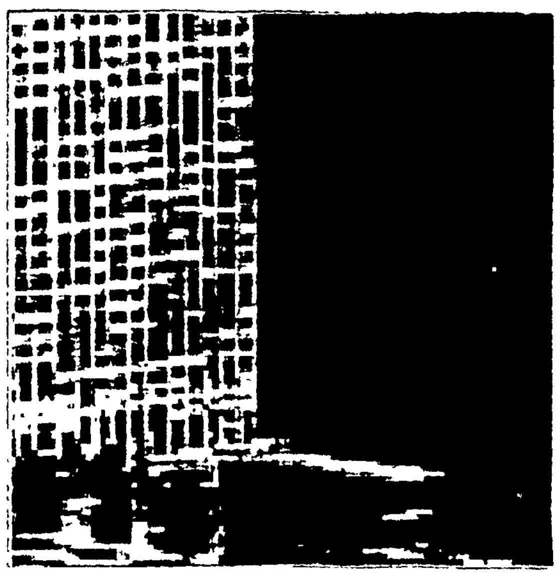

Figure II.3.c The deblurred image using $h = 0.80$ Note that the performance of our deblurring method is not affected by the contents of the image Note that deblurring of CITY which is very rich in structure is again virtually perfect

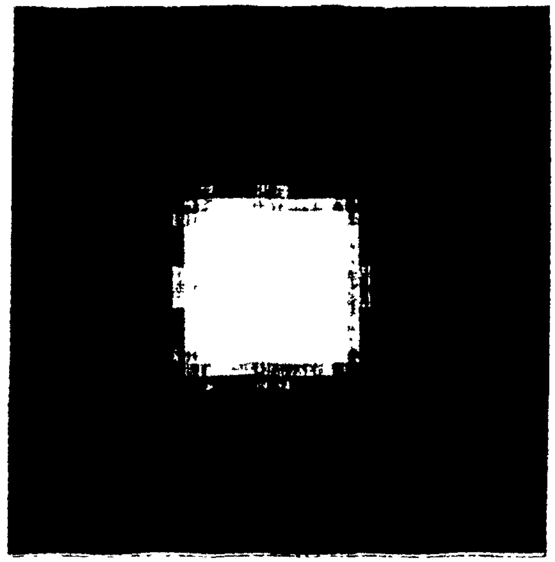

Figure II.4.a The image SQUARE deblurred using $h = 0.20$ with 4 bit representation of the blurred image This series of experiments (4 5 and 6) was undertaken to illustrate the relationship between the numerical representation and the amount of stable Gaussian deblurring See text

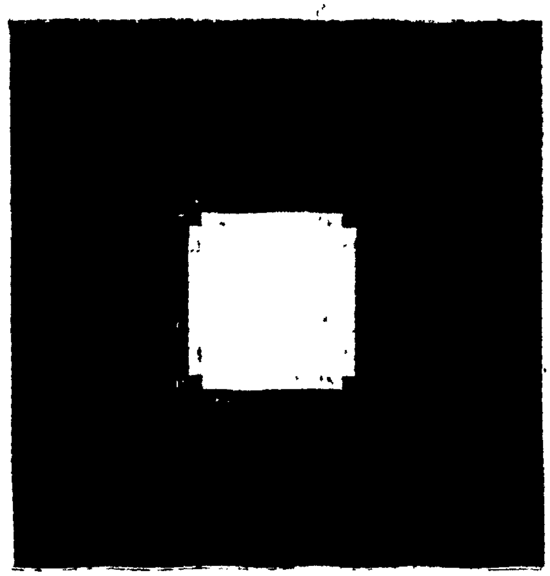

Figure II.4.b The image SQUARE deblurred using $t = 0.35$ with 6 bit representation of the blurred image

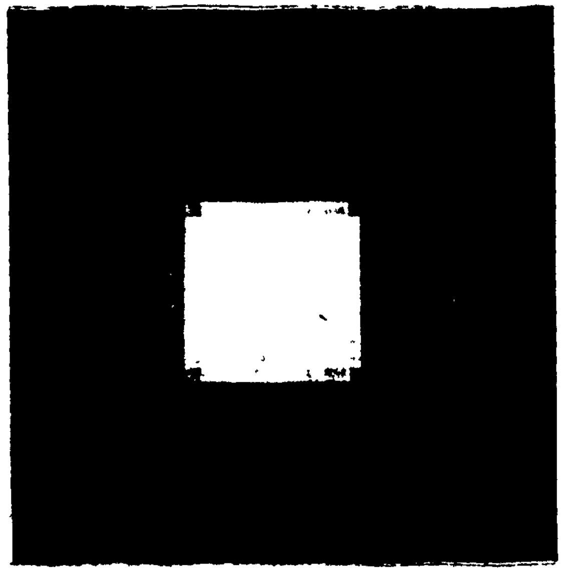

Figure II.4.c The image SQUARE deblurred using $h = 0.45$ with 8 bit representation of the blurred image

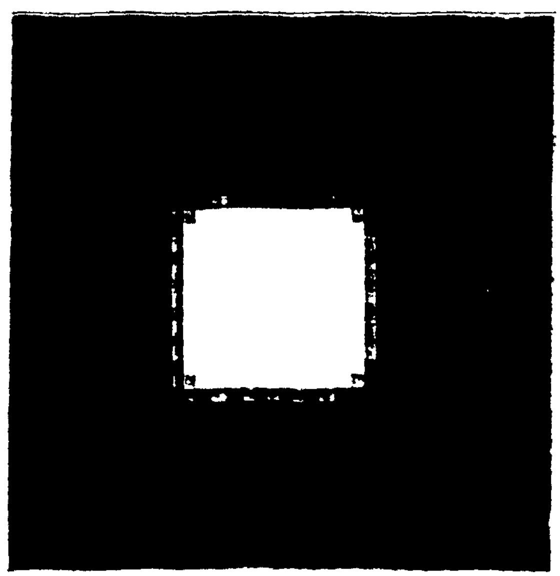

Figure II.4.d The image SQUARE deblurred using $h = 0.55$ with 10 bit representation of the blurred image

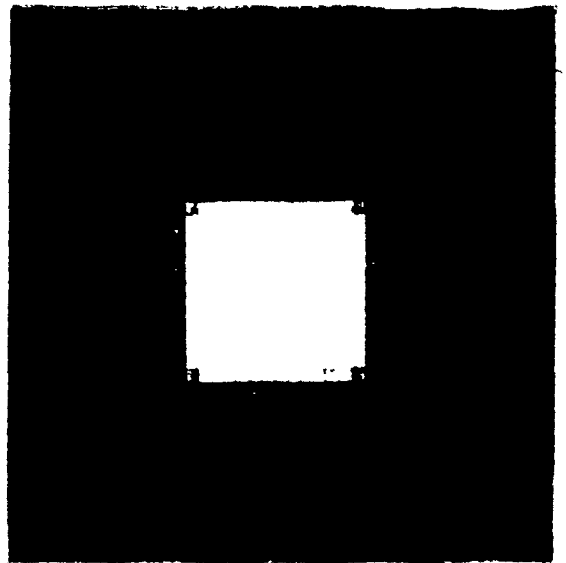

Figure II.4.e The image SQUARE deblurred using $k = 0.60$ with 12 bit representation of the blurred image

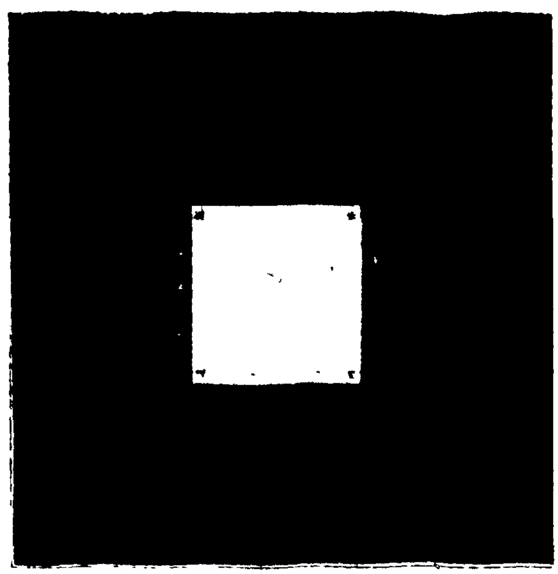

Figure II.4.f The image SQUARE deblurred using $\ell = 0.65$ with 14 bit representation of the blurred image

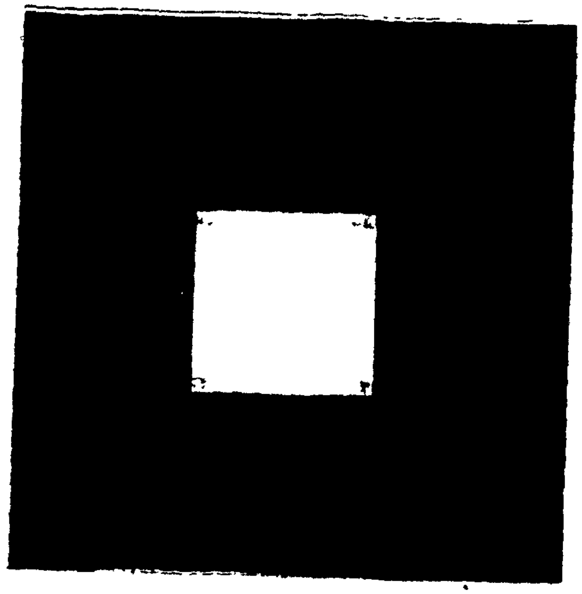

Figure II.4.g The image SQUARE deblurred using $k = 0.65$ with 16 bit representation of the blurred image

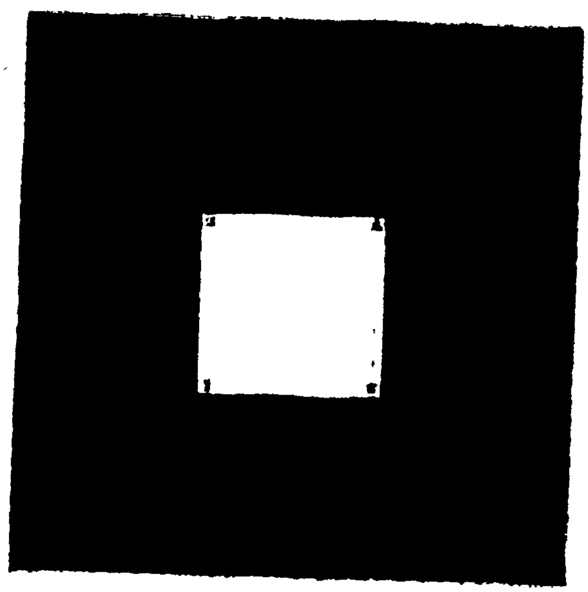

Figure II.4.h The image SQUARE deblurred using $k = 0.70$ with 18 bit representation of the blurred image

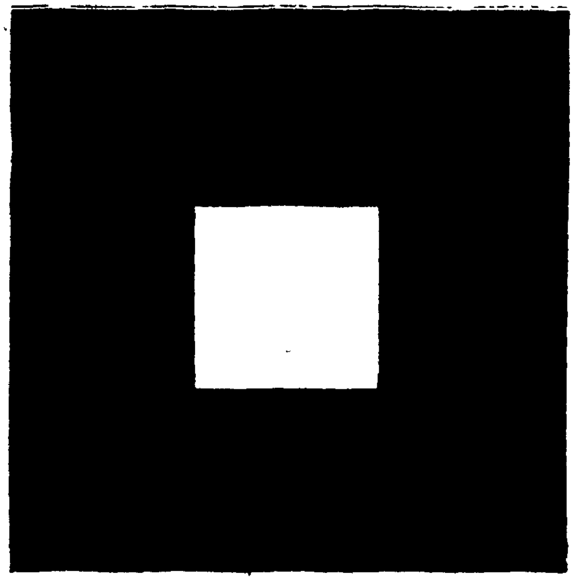

Figure II.4.i The image SQUARE deblurred using $h = 0.75$ with 24 bit representation of the blurred image.

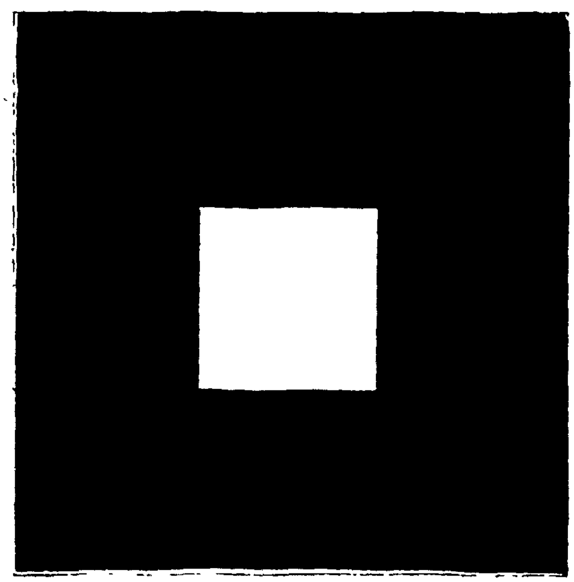

Figure II.4.j The image SQUARE deblurred using $h = 0.80$ with 30 bit representation of the blurred image

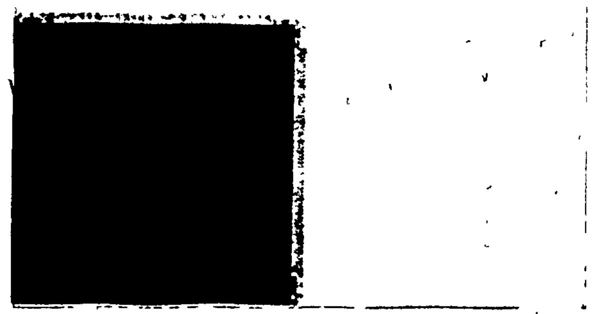

Figure II.5.a The image CORNER deblurred using $b = 0.30$ with 4 bit representation of the blurred image

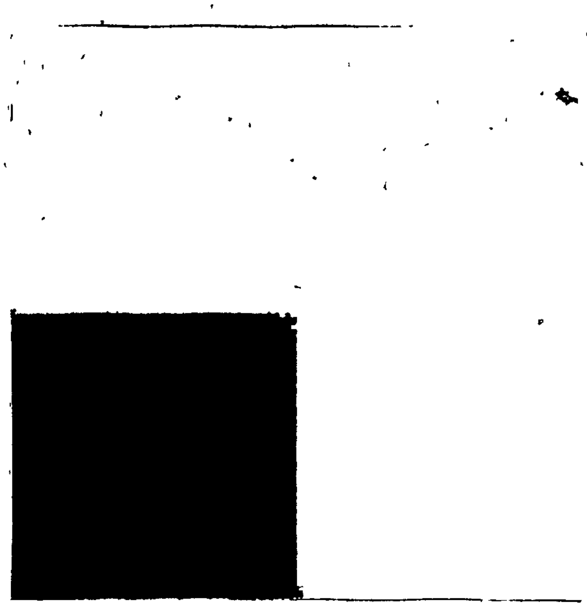

Figure II.5.b The image CORNER deblurred using $b = 0.40$ with 6 bit representation of the blurred image

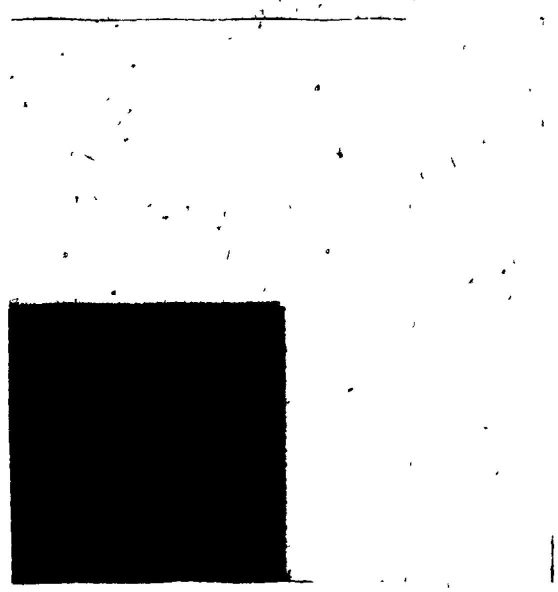

Figure II.5.c The image CORNER deblurred using $t = 0.45$ with 8 bit representation of the blurred image

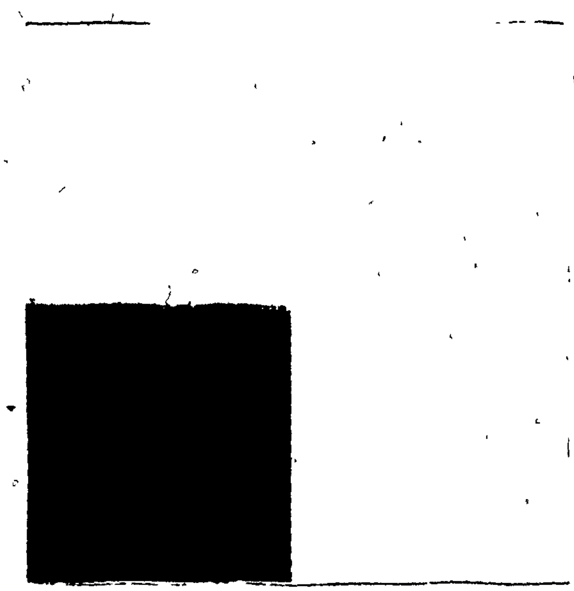

Figure II.5.d The image CORNER deblurred using $t = 0.50$ with 10 bit representation of the blurred image

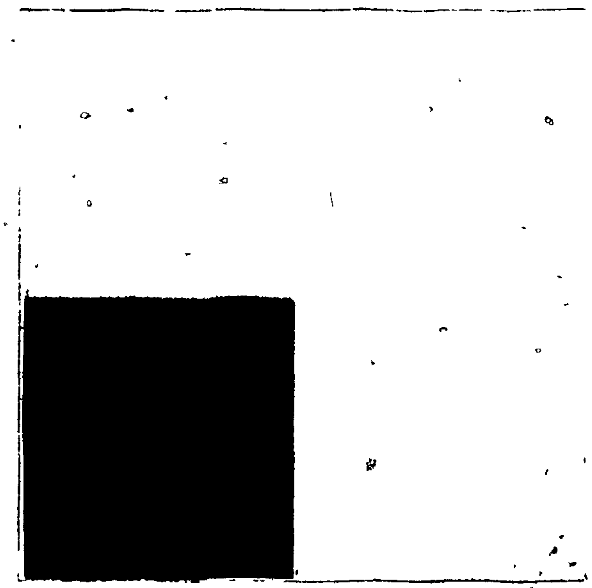

Figure II.5.e The image CORNER deblurred using $k = 0.50$ with 12 bit representation of the blurred image

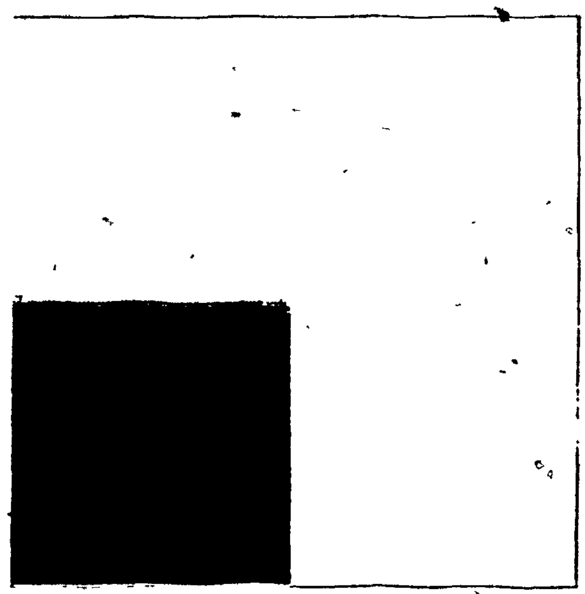

Figure II.5.f The image CORNER deblurred using $k = 0.50$ with 14 bit representation of the blurred image

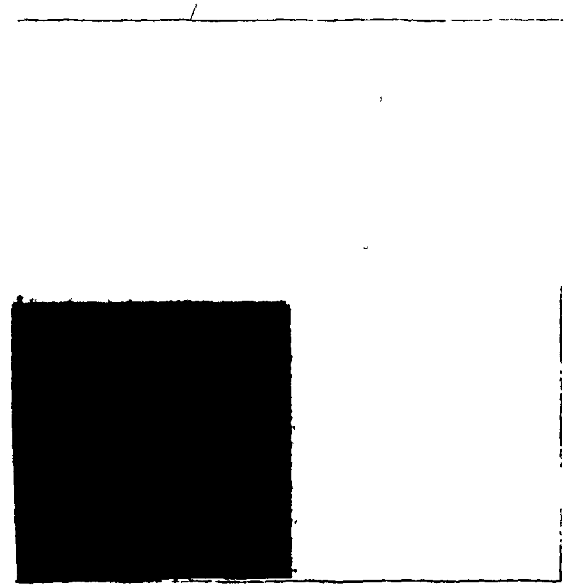

Figure II.5.g The image CORNER deblurred using $l = 0.50$ with 16 bit representation of the blurred image.

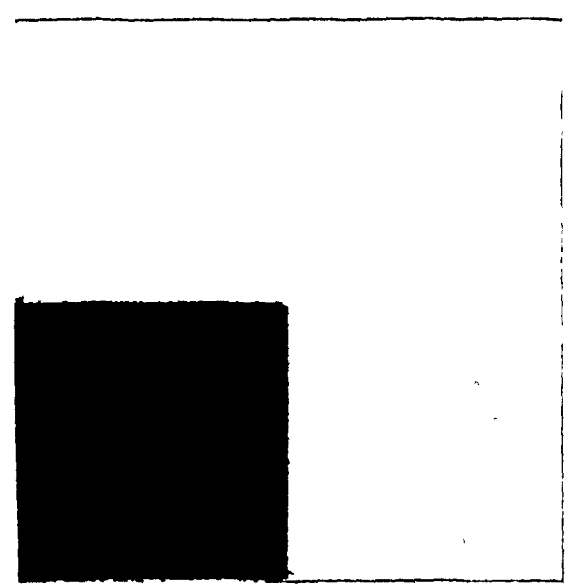

Figure II.5.h The image CORNER deblurred using $l = 0.50$ with 18 bit representation of the blurred image

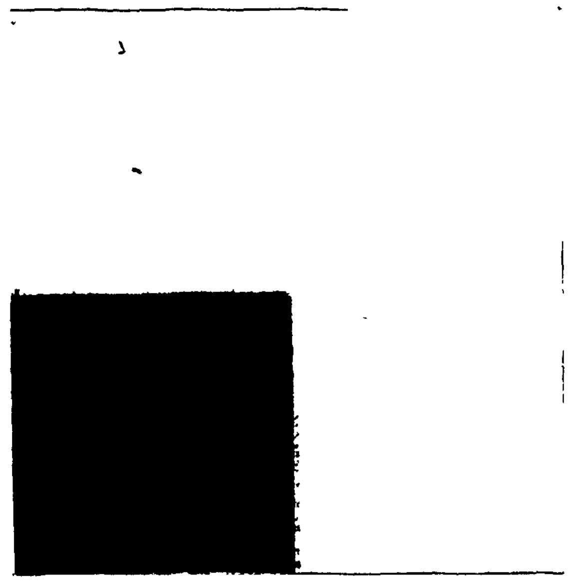

Figure II.5.i The image CORNER deblurred using $b = 0.50$ with 24 bit representation of the blurred image

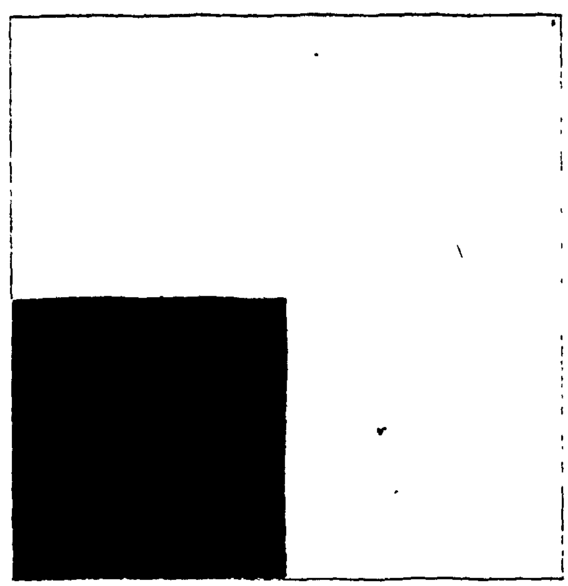

Figure II.5.j The image CORNER deblurred using $b = 0.80$ with 30 bit representation of the blurred image

Figure II.6.a The image CITY deblurred using $b = 0.20$ with 4 bit representation of the blurred image

Figure II.6.b The image CITY deblurred using $b = 0.35$ with 6 bit representation of the blurred image.

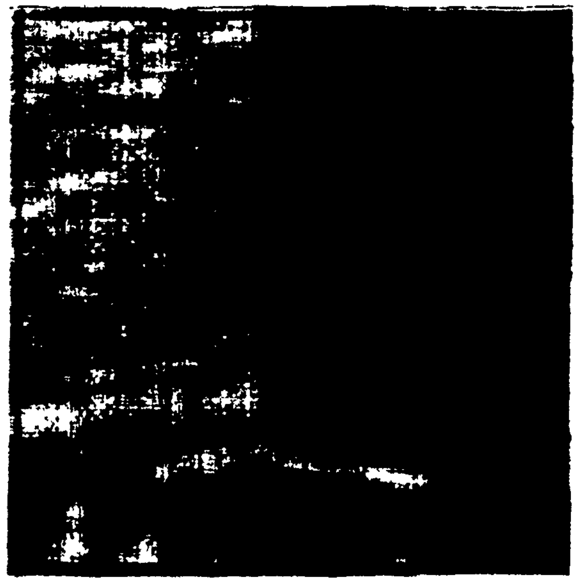

Figure II.6.c The image CITY deblurred using $b = 0.45$ with 8 bit representation of the blurred image

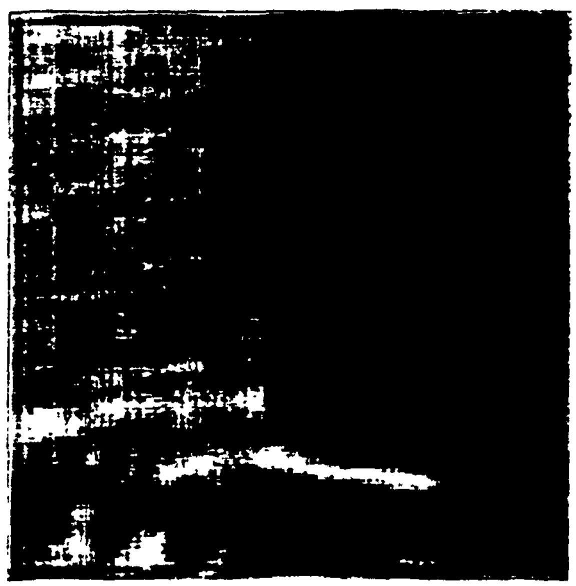

Figure II.6.d The image CITY deblurred using $b = 0.50$ with 10 bit representation of the blurred image

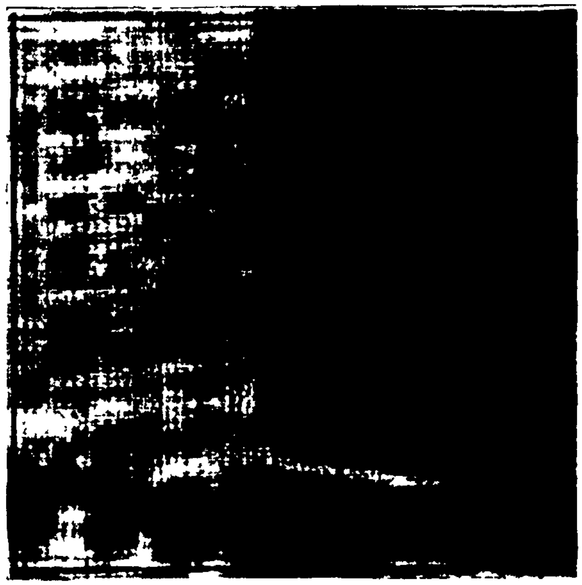

Figure II.6.e The image CITY deblurred using $b = 0.60$ with 12 bit representation of the blurred image

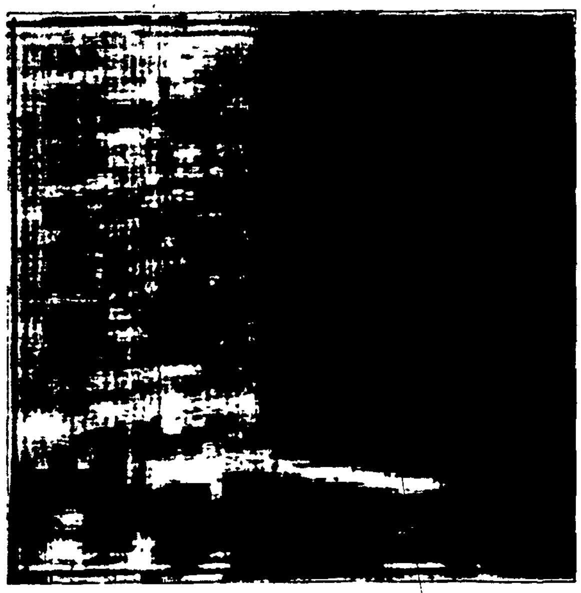

Figure II.6.f The image CITY deblurred using $b = 0.60$ with 14 bit representation of the blurred image

Figure II.6.g The image CITY deblurred using $b = 0.65$ with 16 bit representation of the blurred image

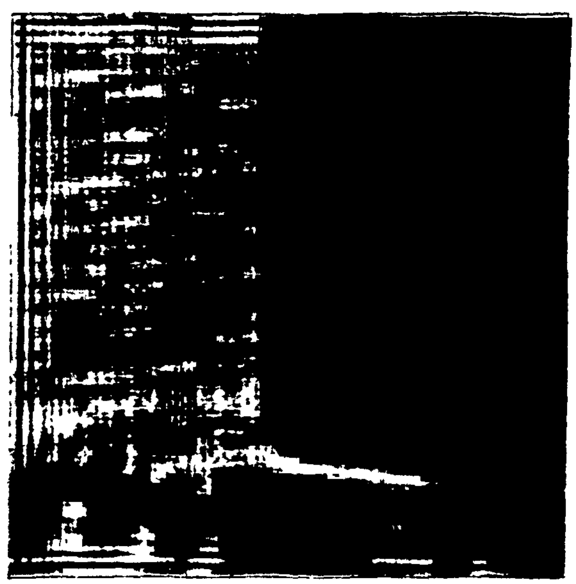

Figure II.6.h The image CITY deblurred using $b = 0.70$ with 18 bit representation of the blurred image

Figure II.6.i The image CITY deblurred using $b = 0.70$ with 24 bit representation of the blurred image

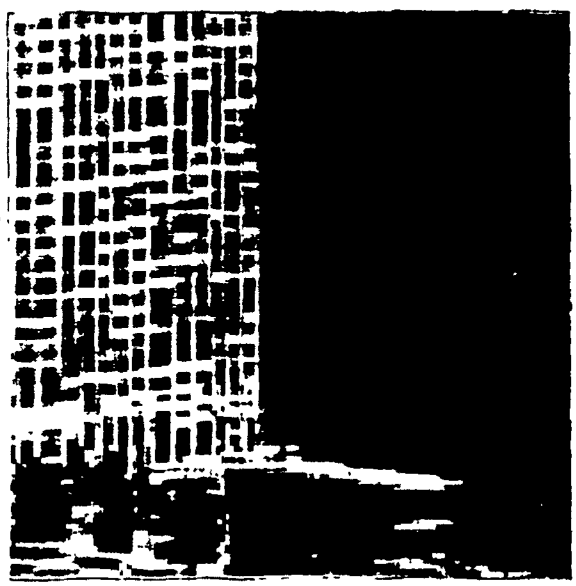

Figure II.6.j The image CITY deblurred using $b = 0.80$ with 30 bit representation of the blurred image

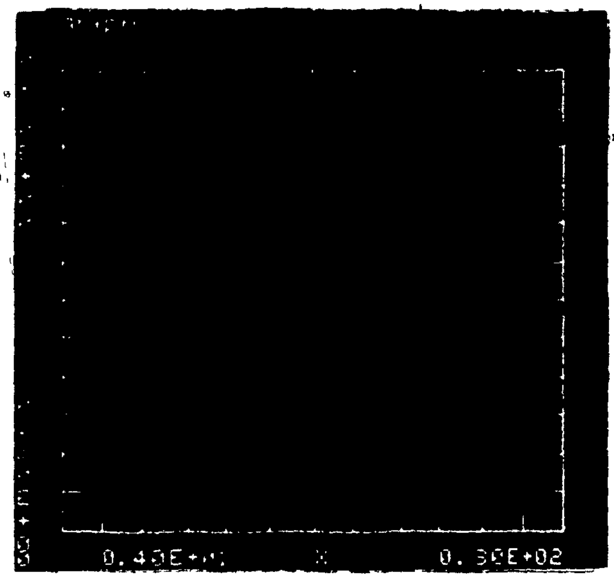

Figure II.7 Performance versus accuracy of the degraded image for CITY Accuracy in bits/pixel in the x-axis Perfomance the normalized norm difference between the original image and the reconstructed image is the y-axis, see text

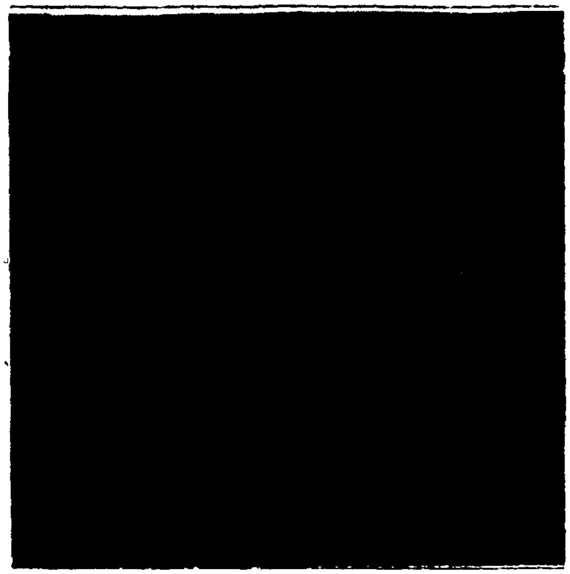

Figure II.8.a The image EYECHART obtained from a camera with the picture out of focus The image is scanned at 200x200 spatial resolution 8 bits/pixel

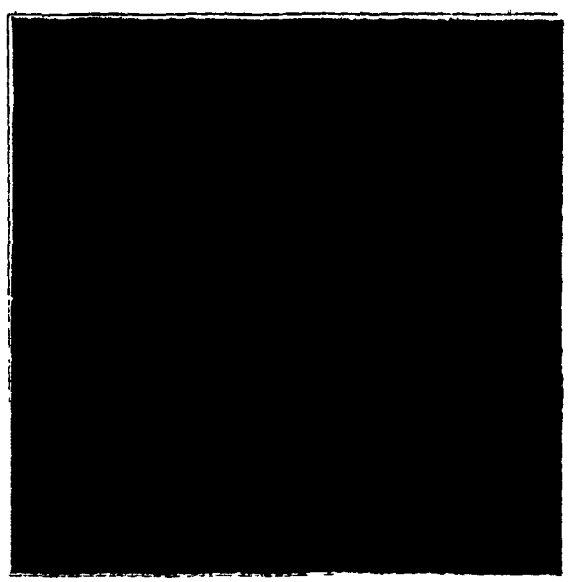

Figure II.8.b The EYECHART image deblurred with $k = 0\ 30$

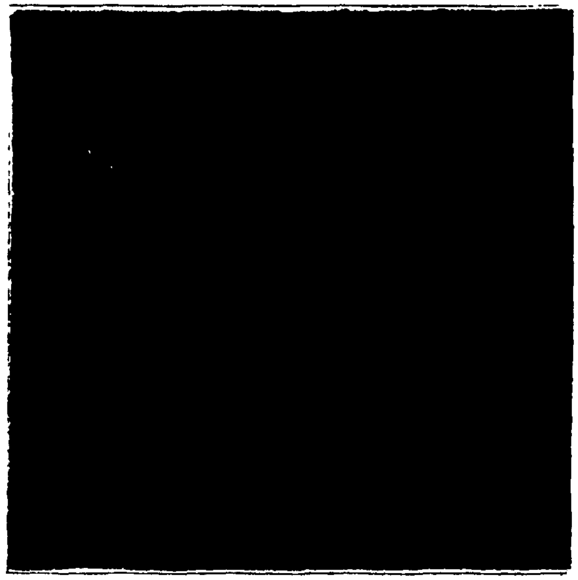

Figure II.8.c The EYECHART image deblurred with $b = 0\ 40$

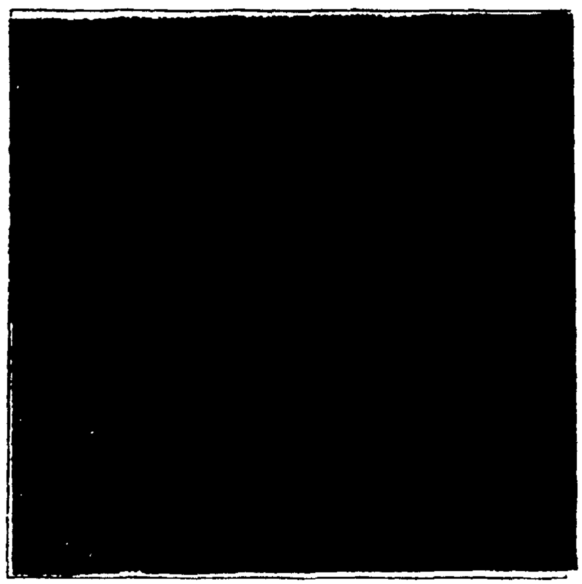

Figure II.8.d The EYECHART image deblurred with $b = 0.45$ Note the trade-off between the reconstruction and noise amplification Increasing the accuracy of representation of the degraded image would improve the results as is suggested by experiments II 4, II 5 and II 6

Chapter II.8 Discrete vs. Continuous Deblurring

There are a number of interesting similarities as well as differences between the method presented for deblurring continuous Gaussian blur in [Kimia and Zucker] and [Hummel, Kimia, and Zucker], and the method presented here for removing discrete Gaussian blur.

The main result of the above papers was to conclude that in order to remove continuous Gaussian blur, the image should be convolved against the deblurring kernel

$$\dot{D}_N(x) = e^{-x^2} \sum_{k=0}^{\lfloor N/2 \rfloor} \frac{(-1)^k}{\sqrt{\pi} k! 2^k} H_{2k}(x), \qquad (I.5.9)$$

where N is the order of polynomial approximation of the original image and $H_{2k}(x)$, is the Hermite polynomial of order $2k$. Figure (II 9.a) shows the deblurring kernel for $N = 9$. Similarly, one can obtain a discrete deblurring kernel by ploting a row of B^{-1} (Figure II.9.b) Note that the plots of the deblurring kernels resemble each other closely

The discrete method, however, does not have some of the problems associated with continuous convolutions The shrinking of the boundary, for example, is non-existant for discrete deblurring and the discrete algorithm permits more stable implementations in high-accuracy, low noise situations.

Figure (II.10) shows a comparison of the two methods on the image CITY. At low numerical accuracy (6 bits/pixel) the results appear better with continuous deblurring. However, at high numerical accuracy, the results are significantly better with discrete deblurring. The stability at lower numerical accuracy is a result of smoothing inherent in the low order approximation of the continuous deblurring

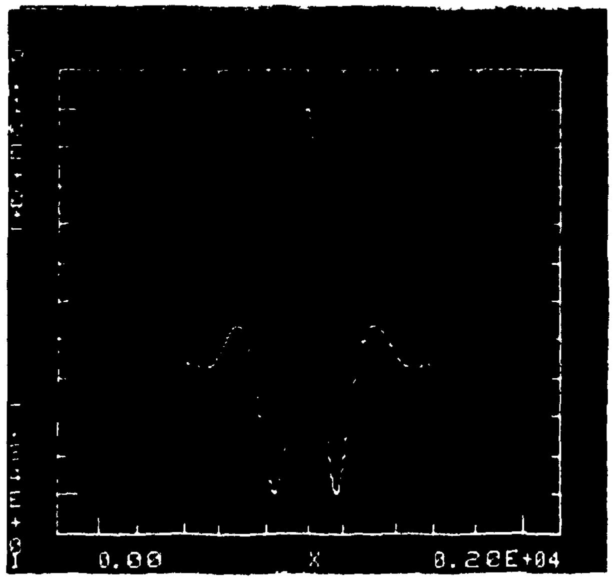

Figure II.9.a A plot of the continuous deblurring kernel

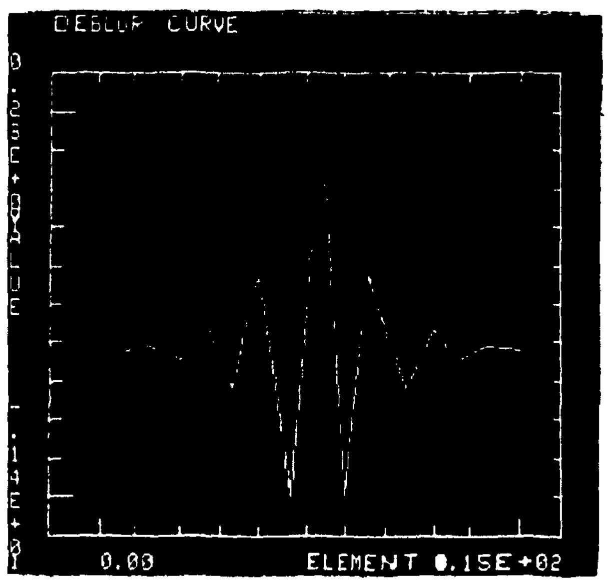

Figure II.9.b A plot of the discrete deblurring kernel Note the similarity in structure to the continuous deblurring kernel in figure II 9 a

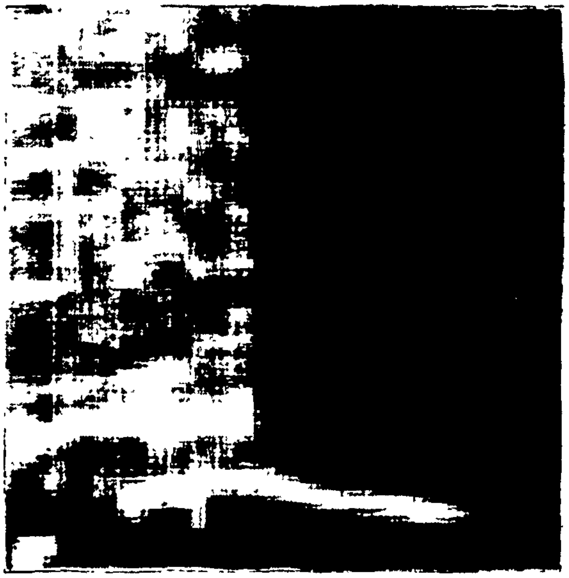

Figure II.10.a The CITY image deblurred using the continuous method with 6 bits/pixel accuracy with equivalent blur and deblur sigma. σ = 1.5, and deblur kernel order 5. Compare this result to II 6 b where discrete deblurring is performed on CITY with the same numerical accuracy. See text for discussion

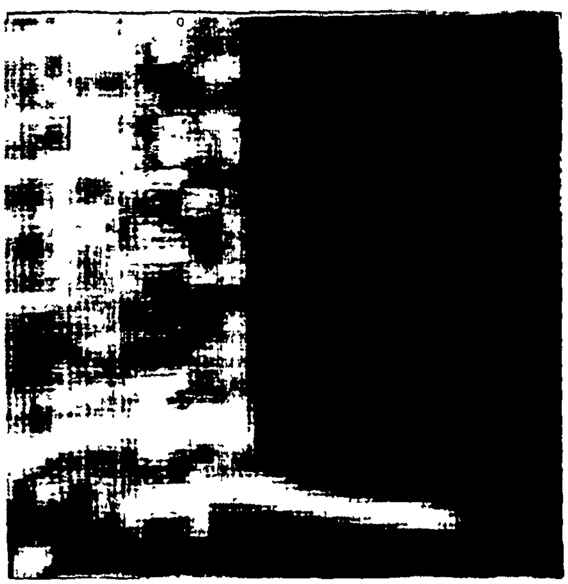

Figure II.10.b The CITY image deblurred using the continuous method, as above, but with higher accuracy

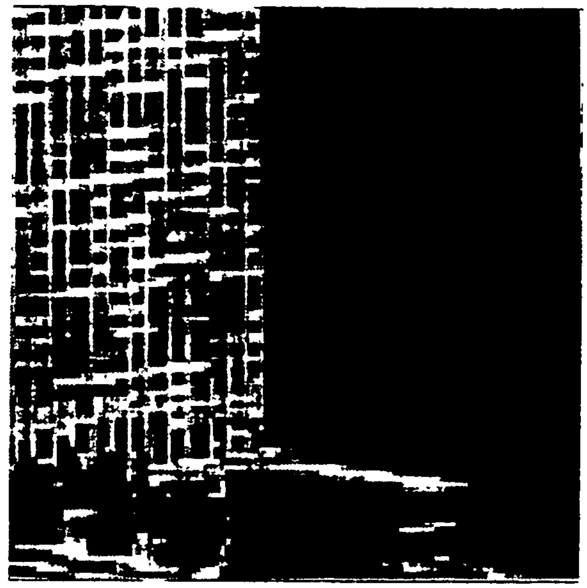

Figure II.10.c The CITY image deblurred using the discrete method. The deblur parameter is $b = 0.80$ which is equivalent to the above sigma Note that with the same accuracy as II 10 b, the results are significantly better.

Chapter II.9 Extension to Stably Decomposable Kernels

Degradations other than Gaussian can also be represented as the linear convolution of the image against some kernel Thusfar, we have shown how the problem is structured for a Gaussian kernel. Now, we would like to extend this result to a larger class of kernels. The extension is not completely straightforward, however, since for a Gaussian kernel we were able to model the degradation as the left and right multiplication of the image matrix by degradation matrices and this is not always possible. Rather, discrete convolution is represented, in general, as a single multiplication of the image vector by the degradation kernel (matrix). In this chapter, we show how to put a subclass of degradation kernels into the left and right matrix form, so that the techniques developed for removing Gaussian degradation can be applied.

Consider the class of kernels for which it is possible to model the degradation with left and right degradation matrices. That is, let the matrix F represent the image. Then, a kernel K belongs to this class if

$$\exists\ \tilde{B}_L, \tilde{B}_R \quad \text{such that} \quad \tilde{B}_L F \tilde{B}_R{}^T = F * K \quad \forall\ F, \tag{II.9.1}$$

where $\tilde{B}_L$ and $\tilde{B}_R$ are the left and right degradation matrices, respectively. A kernel that satisfies (9.1) will be refered to as a *decomposable kernel.*

Note that, with the image represented as a matrix, neighbourhood relations are preserved In appendix D, we discuss the necessary and sufficient conditions for a kernel to be decomposable.

The next step to extend the result to such kernels is to obtain blur matrices from the left and right degradation matrices $\tilde{B}_L$ and $\tilde{B}_R$. Hence, define matrices V_L and V_R,

$$\tilde{B}_L = V_L B_L \quad : \quad \tilde{B}_R = B_R V_R, \tag{II.9.2}$$

where B_L is the left blur matrix, V_L is the "residual", and similarly for the right side. Then, we have

$$H = V_L B_L F B_R V_R. \tag{II.9.3}$$

The problem, therefore, is recovering F from H in (II.9.3), and this can be done in two steps. First, solve for Z in

$$H = V_L Z V_R. \tag{II.9.4}$$

Second, solve for F in

$$Z = B_L F B_R. \tag{II.9.5}$$

The first step can be achieved using conventional methods for solving a linear system of equations, provided matrices V_L and V_R are not ill-conditioned. Let us refer to decomposable kernels that yield well-conditioned matrices V_L and V_R as *stably decomposable* kernels. The second step employs algorithms based on theorem (II.6.1), which recover F accurately.

We have therefore shown how to remove degradations governed by stably decomposable kernels. Given a kernel K representing some degradation, we can use the conditions of appendix D to determine whether K is decomposable. Then, we can test for the conditioning of V_L and V_R to determine whether K is stably decomposable. If K is indeed a stably decomposable kernel, then the method of this paper can be applied effectively.

Chapter II.10 Summary II

We have considered the problem of inverting discrete Gaussian blur. This was accomplished by deriving a decomposition of the blur matrix that isolates the numerically ill-conditioned part of the problem. There are several advantages to this method compared with direct numerical methods. First, typical numerical methods for solving the matrix equation (II.1.1) fail when the dimension of the problem reaches realistic proportions. Second, such methods employ general purpose algorithms that do not make use of the special structure of the Gaussian. In contrast, we have utilized the full structure of the problem. Third, since we carried out a large part of the inversion process symbolically, much of the overall numerical error is avoided. Fourth, our symbolic method confines the ill-conditioned part of the problem to a diagonal matrix. This allows for accurate methods since the complexity of very accurate computations increases linearly with the dimension of the problem. Also, the partitioning of the problem into complex well-conditioned and simple ill-conditioned parts has made it possible to predict the numerical accuracy needed to deal with any amount of blur. Finally, the method can be extended to a larger class of kernels, which we refered to as stably decomposable ones.

More practically, our method is effective with problems characterized by low noise and high representation accuracy. In such situations, the method removes the blur completely and exactly. Given the improvements constantly taking place in imaging technology, particularly the increased reliability of sensors at higher light levels, images to which this method can be successfully applied should become more common.

PART III – Concluding Remarks

Chapter III.1 Speculation on results

The last two parts have demonstrated how to remove Gaussian blur in the continuous and the discrete domains. I will now take the liberty to speculate about possible implications of these results towards interpreting the physiology of the human visual system.

Recently, physiologists have noticed that the receptive fields of some cortical cells, which were thought to be accurately modelled by a difference of Gaussians, exhibit side lobes. Gabor functions are now used to approximate the receptive fields of these cortical cells [Marcelja]. Interestingly, our deblurring kernels are extremely close to an even Gabor function in form, although their analytical expression is different. Figures III.1.a shows an even (Cosine) Gabor function, and Figures III.1.b and III.1.c show the continuous and the discrete deblurring kernels, respectively. Note the similarity, both qualitative and quantitative, between the Gabor function and the deblurring kernels

What functional interpretation can be ascribed to these cortical cells given their observed receptive field structure? To answer this, note that side lobes have been observed in the receptive fields of simple cells which respond to thin oriented stimuli at a specific location [Movshon]. Such simple cells are generally thought of as contributing to line and contour "detection". One might speculate, then, that these cells perform a combined function of responding to very thin oriented stimuli while simultaneously deblurring in the orthogonal direction to the prefered orientation. The primary task of deblurring in such an hypothesis, then, is to localize blurred line segments. This assertion conforms to the location specificity of a simple cell. I should stress the belief that several different functional roles may be simultaneously assigned to any given physiological observation, and therefore, this spectulation does not prevent other interpretations of the observed receptive fields.

Blurring and deblurring may also contribute to visual hyperacuity We have shown that the information in the original and the blurred data are equivalent and interchangable. However, given the choice of sampling the original or the blurred data, it is preferable to sample the blurred data. This is because blurring spreads the information, say about an edge in the scene, to several retinal receptors; this information might have been missed otherwise. Deblurring, then, provides a way of reconstructing the inter-receptor information. In the framework of an elaborate theory, [Zucker and Hummel, 1985] have shown how hyperacuity might be achieved by deblurring visual information in the smallest channel. According to this theory, the visual information is first processed by a differential operator (Laplacian), separated into its negative and positive parts, and then transmitted to the cortex. A deblurring of the information in the smallest channel is needed for a possible (implicit) reconstruction of the visual image which provides visual precision better than the retinal spacing.

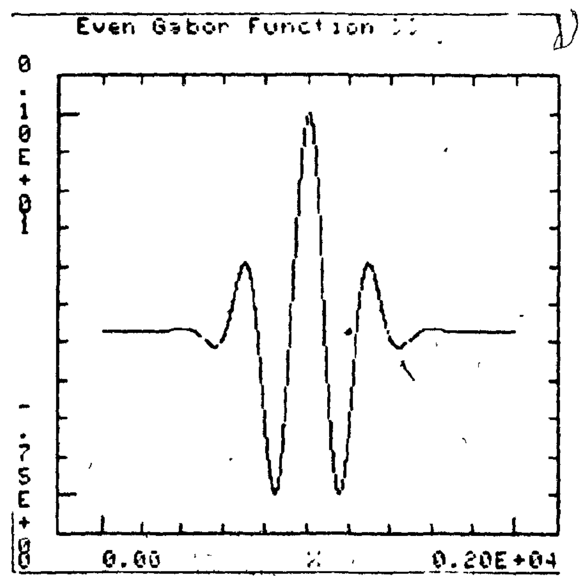

Figure III.1.a A Cosine Gabor function with $\sigma = 1.0$, in the graph's x axis unit, with frequency of the cosine term, $\omega = 0.02$

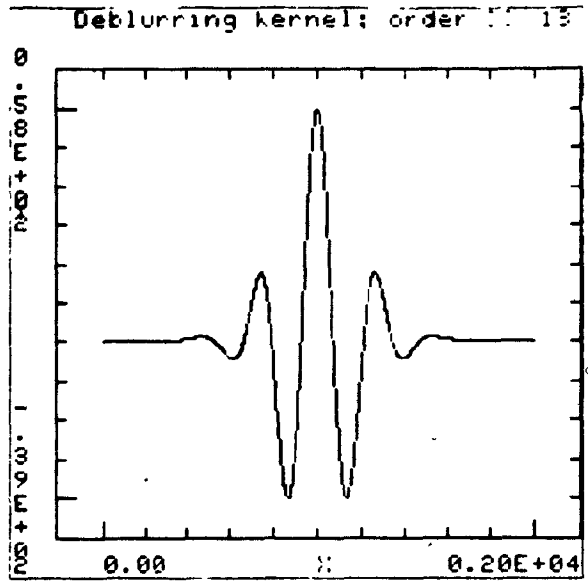

Figure III.1.b The continuous deblurring kernel of order 13 with $\sigma = 1.0$, in the graph's x axis unit

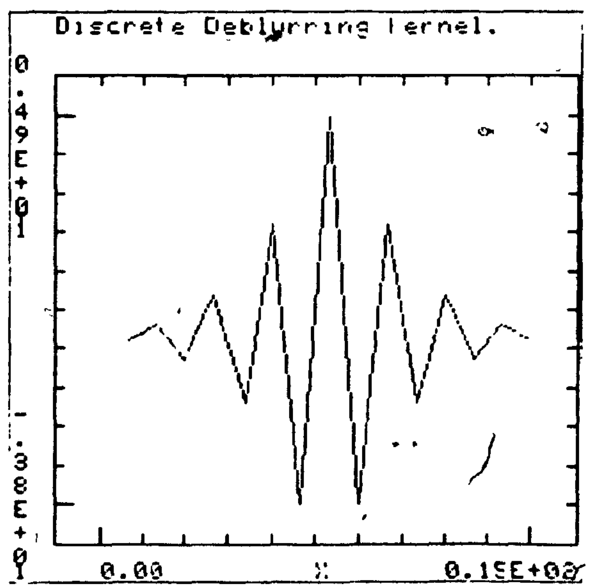

Figure III.1.c The discrete deblurring kernel with $k = 0.60$.

Chapter III.2 Conclusion

In summary, in this thesis I attempted to model and solve the Gaussian blur problem. Gaussian blur was approached from two directions; one continuous and the other discrete. In the continuous approach, we used continuous convolution of the Gaussian against continuous data to model the blurring process. Then, the problem of removing Gaussian blur is one of solving an integral equation. The solution to this problem is formulated as continuous convolution of a deblurring kernel (I.5.9) against the blurred data. In the discrete approach, we modelled the Gaussian blur process as matrix multiplications. The solution now amounts to solving a linear system of equations which are, however, ill-conditioned. We solve the equations by a series of symbolic decompositions and inversions. The result is that the inverse blur matrix is decomposed into two well-conditioned triangular matrices and one diagonal matrix which contains all the ill-conditioned terms.

The solutions for both the continuous and the discrete domains are convergent in form: both the continuous and the discrete solutions present kernels which resemble a difference of Gaussians (DOG), but with extra side lobes. I speculate that it is the form of these kernels with their side lobes that provides one possible explanation for observed side-lobes of receptive fields in the visual cortex. Furthermore, deblurring can be used to recover the depth of objects in the visual field. Also, the phenomenon of hyperacuity would not be possible without blurring, and deblurring provides a possible method for reconstructing the inter-receptor visual detail. It would be interesting to see whether the conceptual framework of deblurring and further or simultaneous processing of visual information, in a combined form, could provide constraints for deepening our understanding of physiological data.

References

Bers, John, Schechter, *Partial Differential Equations*, American Mathematical Society, Providence, RI 02904, 1964

Campbell, F.W., Gubisch, R.W., Optical Quality of the Human Eye, *J. Physiol. (Lond.)*, 1966, **186**, 558-578.

Courant, R., and Hilbert, D., *Methods of Mathematical Physics*, Wiley, New York, 1962.

Feynman R., Leighton, R., Sands, M., *The Feynman Lectures on Physics.*, Addison-Wesley, 1963

Fukada Y., Receptive Field Organization of Cat Optic Nerve Fibers with Special Reference to Conduction Velocity, *Vision Res.*, 1971, **11**, 209-226.

Helmholtz H. von, Treatise on Physiological Optics, J.P.C. Southall, ed., Dover (reprint), 1962.

Herman, G., *Image Reconstruction from Projections*, Academic Press, New York, 1980.

Hörmander, L., *The Analysis of Linear Partial Differential Operators I*, Springer, New York., 1983

Hummel, R.A., Kimia, B., Zucker, S.W., *Deblurring Gaussian Blur*, TR 83-15R, McGill University, Montreal, Canada, *Computer Vision, Graphics, and Image Processing*, to appear.

John, F., Numerical Solution of the Equation of Heat Conduction for Preceeding Times, *Annali de Matematica*, 1955, **17**, 129-142.

Kimia, B., and Zucker, S.W., Deblurring Gaussian Blur, *IEEE Conf. Computer Vision and*

Pattern Recognition. Washington. D.C.. June. 1983

Lebedev. N. N.. *Special Functions and their Applications*. R A. Silverman(Ed.). Prentice-Hall Inc.. Englewood Cliffs. N J.. 1965.

Marcelja. S.. Mathematical description of the responses of simple cortical cells. *J. Opt. Soc. Amer.*. 1980. **70**. 1297 - 1300

Movshon. J.. Thompson. I.. and Tolhurst. D.. Spatial summation in the receptive fields of simple cells in the cat's striate cortex. *J. Physiol. (London)*. 1978. **283**. 53-77.

Pentland. P.A.. A New Sense for Depth of Field. *Proceedings of the Ninth IJCAI*. 988-994. 1985

Pratt. W.. *Digital Image Processing*. Wiley-Interscience. 1978.

Ratliff. F.. *Mach Bands: Quantitative Studies on Neural Networks in the Retina*. Holden-Day. San Francisco. 1965

Rosenfeld. A.. and Kak. A . *Digital Picture Processing*. Academic Press. New York. 1976.

Stewart. G. W.. *Introduction to Matrix Computations*.Academic Press. New York. 1973.

Westheimer. The Spatial Sense of the Eye. *Invest. Ophthal. Visual Sci.*. 1979. **18**. 893-912.preface.tex

Widder. D.. *The Heat Equation*. New York. Academic Press. 1975.

Zucker. S.W.. Hummel. R.A.. *Receptive Fields and the Representation of Visual Information*. TR 85-16R. McGill University. Montreal. Canada. 1985.

Appendix A. A Second Proof of Theorem I.5.7

In this appendix, we outline a natural and more intuitive proof of the main theorem (I.5.7) in order to derive the deblurring kernel $D_N(x)$.

Proof: We require that the deblurring kernel $D(x)$satisfy

$$D(x) * 2^n x^n = H_n(x); \qquad n \leq N. \tag{A.1}$$

Equivalently,

$$\int_{-\infty}^{\infty} D(\xi)[2^n(x-\xi)^n]d\xi = H_n(x). \tag{A.2}$$

By expanding both sides we discover that this requirement translates into constraints on the monomial moments $d_k = \int_{-\infty}^{\infty} D(x)x^k dx$ of $D(x)$, namely

$$\sum_{k=0}^{n} \frac{2^n n!(-1)^k}{(n-k)!k!}[\int_{-\infty}^{\infty} D(\xi)\xi^k d\xi]x^{n-k} = n! \sum_{m=0}^{[n/2]} \frac{(-1)^m(2x)^{n-2m}}{m!(n-2m)!}. \tag{A.3}$$

Equating powers of x,

$$d_{2m+1} = 0, \qquad m = 0, 1, \ldots, [\frac{n-1}{2}] \tag{A.4}$$

and

$$d_{2m} = \frac{(-1)^m(2m)!}{2^{2m}m!}, \qquad m = 0, 1, \ldots, [\frac{n}{2}]. \tag{A.5}$$

These equations should hold for every $n, n = 0, 1, \ldots, N$. The conditions for $n = N$ subsume all others.

The above constraints on d_k can be translated to yield formulae for coefficients of an expansion of $D(x)$ of the form

$$D(x) = e^{-x^2} \sum_{k=0}^{N} a_k H_k(x). \tag{A.6}$$

We restrict the upper limit of the above sum not only because it provides sufficient constraints for a unique kernel, but this kernel turns out to be the pseudo-inverse operator on P_N, the space of polynomials of order N. Also, it is interesting to note that it yields the most stable kernel. Adding any combination of higher order Hermite polynomials $H_n(x) : n = N+1, \ldots,$ to $D(x)$ will yield another legitimate deblurring kernel.

The a_k's will satisfy the linear system of equations

$$\sum_{k=0}^{N} a_k\left[\int_{-\infty}^{\infty} e^{-x^2} H_k(x) x^l dx\right] = d_l, \qquad l = 0,1,\ldots,N. \tag{A.7}$$

We first consider odd $l = 2m+1$ for which $d_l = 0$. For $k = 2p, p = 0,1,\ldots,[\frac{N}{2}]$, H_k is a sum of even power polynomials and according to lemma 1.5.12, the integral vanishes. However, when $k = 2p+1, p = 0,1,\ldots,[\frac{N-1}{2}]$, we have

$$\sum_{p=0}^{[\frac{N-1}{2}]} a_{2p+1}\left[\int_{-\infty}^{\infty} e^{-x^2} H_{2p+1}(x) x^{2m+1} dx\right] = 0, \qquad m = 0,1,\ldots,[\frac{N-1}{2}]. \tag{A.8}$$

We use the following lemma to prove this linear system of equations is non-singular.

Lemma A.9:

$$c_{2k+1,2p+1} = \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi}} e^{-x^2} H_{2k+1}(x) x^{2p+1} dx = \begin{cases} 0, & p < k \\ (-1)^k \frac{(2p-2k+2)(2p+1)!}{2^{2p-2k+1}(p-k+1)!}, & p \geq k \end{cases}.$$

Proof: For $k \geq 1$, $p \geq 1$,

$$\begin{aligned} \sqrt{\pi} c_{2k+1,2p+1} &= \int_{-\infty}^{\infty} e^{-x^2}\left[(-1)^{2k+1} e^{x^2} \frac{d^{2k+1}}{dx^{2k+1}}(e^{-x^2})\right] x^{2p+1} dx \\ &= -\int_{-\infty}^{\infty} \frac{d^{2k+1}}{dx^{2k+1}}(e^{-x^2}) x^{2p+1} dx \\ &= (2p+1)\int_{-\infty}^{\infty} \frac{d^{2k}}{dx^{2k}}(e^{-x^2}) x^{2p} dx \\ &= -\sqrt{\pi}(2p+1)(2p) c_{2k-1,2p-1}. \end{aligned} \tag{A.10}$$

Clearly, $c_{2k+1,1} = 0$, $k > 1$. Using lemma 1.5.12,

$$c_{1,2p+1} = 2\frac{(2p+2)!}{2^{2p+2}(p+1)!}, p \geq 0. \tag{A.11}$$

Combining, $c_{2k+1,2p+1} = 0$ for $p < k$, and for $p \geq k$,

$$\begin{aligned} c_{2k+1,2p+1} &= (-1)(2p+1)(2p)\ldots(-1)(2p-2k+3)(2p-2k+2).c_{1,2p-2k+1} \\ &= (-1)^k \frac{(2p+1)!}{(2p-2k+1)!}.2\frac{(2p-2k+2)!}{2^{2p-2k+2}(p-k+1)!} \\ &= (-1)^k(2p-2k+2)\frac{(2p+1)!}{2^{2p-2k+1}(p-k+1)!}. \end{aligned} \tag{A.12}$$

∎

The lemma implies none of the $c_{2p+1,2p+1}$ is zero. Hence, the solution to

$$\sum_{p=0}^{[\frac{N-1}{2}]} a_{2p+1}c_{2p+1,2m+1} = 0 \qquad m = 0,1,\ldots,[\frac{N-1}{2}]. \tag{A.13}$$

is unique and clearly.

$$a_{2p+1} = 0, \qquad p = 0,1,\ldots,[\frac{N-1}{2}]. \tag{A.14}$$

For even $l = 2m$.

$$\sum_{k=0}^{N} a_k[\int_{-\infty}^{\infty} e^{-x^2}H_k(x)x^{2m}dx] = \frac{(-1)^m(2m)!}{2^{2m}m!}, \qquad m = 0,1,\ldots,[\frac{N}{2}]. \tag{A.15}$$

The above integral vanishes for n odd. Thus,

$$\sum_{p=0}^{[\frac{N}{2}]} a_{2p}[\int_{-\infty}^{\infty} e^{-x^2}H_{2p}(x)x^{2m}dx] = \frac{(-1)^m(2m)!}{2^{2m}m!}, \qquad m = 0,1,\ldots,[\frac{N}{2}]. \tag{A.16}$$

which becomes, using lemma 2 of section 5,

$$\sum_{p=0}^{m} a_{2p}[\sqrt{\pi}\frac{(2m)!}{2^{2m-2p}(m-p)!}] = \frac{(-1)^m(2m)!}{2^{2m}m!}, \qquad m = 0,1,\ldots,[\frac{N}{2}], \tag{A.17}$$

Canceling and rearranging,

$$\sqrt{\pi}\sum_{p=0}^{m} \frac{m!}{(m-p)!}a_{2p}2^{2p} = (-1)^m, \qquad m = 0,1,\ldots,[\frac{N}{2}], \tag{A.18}$$

which can be easily recognized in the form

$$\sum_{p=0}^{m} \binom{m}{p} y^p x^{m-p} = (x+y)^m, y = 2, x = -1. \tag{A.19}$$

Then, it is evident that

$$a_{2p} = \frac{(-1)^p}{\sqrt{\pi}p!2^p}. \tag{A.20}$$

In summary, then,

$$D(x) = D_N(x) = e^{-x^2} \sum_{p=0}^{[\frac{N}{2}]} \frac{(-1)^p}{\sqrt{\pi}p!2^p} H_{2p}(x). \qquad \blacksquare \qquad (I.5.9)$$

Appendix B. Proof of the LU Decomposition of B

In this appendix we present the proof for the theorem (II.3.1).

$$B = \mathcal{L}R. \tag{II.3.1}$$

Proof: We will show that the i^{th} row of the $\mathcal{L}$ matrix, $\mathcal{L}_i$, multiplied by the j^{th} column of the R matrix, R_j is the ij^{th} element of the blur matrix, B.

Assume $i \leq j$. Denote $\mathcal{L}_i R_j$ by Λ. Then,

$$\Lambda = \Big(b^{(i-1)^2} \quad b^{(i-2)^2}(1-b^{2i-2}) \quad b^{(i-3)^2}(1-b^{2i-2})(1-b^{2i-4}) \quad \ldots$$
$$\left[(1-b^{2i-2})(1-b^{2i-4})\ldots(1-b^2)\right] \ldots \Big) \cdot \begin{pmatrix} b^{(j-1)^2} \\ b^{(j-2)^2}\frac{1-b^{2j-2}}{1-b^2} \\ b^{(j-3)^2}\frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4} \\ \vdots \\ b^{(j-i)^2}\frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4} \vdots \frac{1-b^{2j-2(i-1)}}{1-b^{2(i-1)}} \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Rewriting it with a few more terms explicit, we get,

$$\begin{aligned}
\Lambda = {} & b^{(i-1)^2}\ b^{(j-1)^2} \\
& + b^{(i-2)^2}(1-b^{2i-2})\ b^{(j-2)^2}\frac{1-b^{2j-2}}{1-b^2} \\
& + b^{(i-3)^2}(1-b^{2i-2})(1-b^{2i-4})\ b^{(j-3)^2}\frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4} \\
& + \qquad \vdots \\
& + \left[b^4(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^6)\right]\ b^{(j-i+2)^2}\frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2(i-3)}}{1-b^{2(i-3)}}. \\
& + \left[b(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^4)\right]\ b^{(j-i+1)^2}\frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2(i-2)}}{1-b^{2(i-2)}}. \\
& + \left[(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^2)\right]\ b^{(j-i)^2}\frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2(i-1)}}{1-b^{2(i-1)}}.
\end{aligned} \tag{B.1}$$

$$\begin{aligned}
\Lambda = {} & b^{(i-1)^2}\ b^{(j-1)^2} \\
& + b^{(i-2)^2}(1-b^{2i-2})\ b^{(j-2)^2}\frac{1-b^{2j-2}}{1-b^2} \\
& + b^{(i-3)^2}(1-b^{2i-2})(1-b^{2i-4})\ b^{(j-3)^2}\frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4} \\
& + \qquad \vdots \\
& + \left[b^9(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^8)\right]\ b^{(j-i+2)^2}\frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2(i-4)}}{1-b^{2(i-4)}} \\
& + \left[b^4(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^6)\right]\ b^{(j-i+2)^2}\frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2(i-3)}}{1-b^{2(i-3)}} \\
& + \left[b(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^4)\right]\ b^{(j-i+1)^2}\frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2(i-2)}}{1-b^{2(i-2)}} \\
& + b^{(j-i)^2}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2(i-1)}).
\end{aligned}$$

Recall that our intention is to prove $\Lambda = b^{(j-i)^2}$. Hence, it seems reasonable to expand the last part of the previous formula based on its form. Therefore, we define

$$\Pi \triangleq b^{(j-i)^2}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2(i-1)}),$$

and try to expand it.

$$
\begin{aligned}
\Pi = b^{(j-i)^2}[& - b^{2j-2i+2}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2i+4}) + \\
& - b^{2j-2i+4}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2i+6}) + \\
& \vdots \\
& - b^{2j-6}(1-b^{2j-2})(1-b^{2j-4}) + \\
& - b^{2j-4}(1-b^{2j-2}) + \\
& - b^{2j-2} + \\
& 1].
\end{aligned}
$$

$$
\begin{aligned}
\Pi = & - b^{j^2-2(i-1)j} b^{i^2-2i+2}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2i+4}) + \\
& - b^{j^2-2(i-1)j} b^{i^2-2i+4}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2i+6}) + \\
& - b^{j^2-2(i-1)j} b^{i^2-2i+6}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2i+8}) + \\
& \vdots \\
& - b^{j^2-2(i-1)j} b^{i^2-6}(1-b^{2j-2})(1-b^{2j-4}) + \\
& - b^{j^2-2(i-1)j} b^{i^2-4}(1-b^{2j-2}) + \\
& - b^{j^2-2(i-1)j} b^{i^2-2} + \\
& b^{(j-i)^2}.
\end{aligned}
$$

Substituting Π back into the expression (B.1) for Λ we get

$$
\begin{aligned}
\Lambda = {} & b^{(i-1)^2}\ b^{(j-1)^2} \\
& + b^{(i-2)^2}(1-b^{2i-2})\ b^{(j-2)^2}\ \frac{1-b^{2j-2}}{1-b^2} \\
& + b^{(i-3)^2}(1-b^{2i-2})(1-b^{2i-4})\ b^{(j-3)^2}\ \frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4} \\
& + \qquad \vdots \\
& + \left[b^9(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^8)\right]\ b^{(j-i+3)^2}\ \frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2(i-4)}}{1-b^{2(i-4)}} \\
& + \left[b^4(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^6)\right]\ b^{(j-i+2)^2}\ \frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2(i-3)}}{1-b^{2(i-3)}} \\
& + \left[b(1-b^{2i-2})(1-b^{2i-4})\dots(1-b^4)\right]\ b^{(j-i+1)^2}\ \frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4}\cdots\frac{1-b^{2j-2(i-2)}}{1-b^{2(i-2)}} \\
& + \Big\{-b^{j^2-2(i-1)j}b^{i^2-2i+2}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2i+4})+ \\
& \quad -b^{j^2-2(i-1)j}b^{i^2-2i+4}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2i+6})+ \\
& \quad -b^{j^2-2(i-1)j}b^{i^2-2i+6}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2i+8})+ \\
& \qquad \vdots \\
& \quad -b^{j^2-2(i-1)j}b^{i^2-6}(1-b^{2j-2})(1-b^{2j-4})+ \\
& \quad -b^{j^2-2(i-1)j}b^{i^2-4}(1-b^{2j-2})+ \\
& \quad -b^{j^2-2(i-1)j}b^{i^2-2}+ \\
& \quad b^{(j-i)^2}\Big\}.
\end{aligned}
$$

Now, rewrite the first few lines of this expression having in mind we want to somehow come up with $b^{(j-i)^2}$!

$$
\begin{aligned}
\Lambda =\ & b^{j^2-2j}\ b^{i^2-2i+2} \\
& + b^{j^2-4j}\ b^{i^2-4i+8}(1-b^{2j-2})\frac{1-b^{2i-2}}{1-b^2} \\
& + b^{j^2-6j}\ b^{i^2-6i+18}(1-b^{2j-2})(1-b^{2j-4})\frac{1-b^{2i-2}}{1-b^2}\,\frac{1-b^{2i-4}}{1-b^4} \\
& + \\
& + b^{j^2-2(i-3)j} b^{i^2-6i+18}\left[(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2(i-4)})\right]\frac{1-b^{2i-2}}{1-b^2}\,\frac{1-b^{2i-4}}{1-b^4}\,\frac{1-b^{2i-6}}{1-b^2} \\
& + b^{j^2-2(i-2)j}\ b^{i^2-4i+8}\left[(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2(i-3)})\right]\frac{1-b^{2i-2}}{1-b^2}\,\frac{1-b^{2i-4}}{1-b^4} \\
& + b^{j^2-2(i-1)j}\ b^{i^2-2i+2}\left[(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2(i-2)})\right]\frac{1-b^{2i-2}}{1-b^2} \\
& + \Big\{ - b^{j^2-2(i-1)j}\ b^{i^2-2i+2}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2i+4})+ \\
& \quad - b^{j^2-2(i-1)j}\ b^{i^2-2i+4}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2i+6})+ \\
& \quad - b^{j^2-2(i-1)j}\ b^{i^2-2i+6}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2i+8})+ \\
& \qquad \vdots \\
& \quad - b^{j^2-2(i-1)j}\ b^{i^2-6}(1-b^{2j-2})(1-b^{2j-4})+ \\
& \quad - b^{j^2-2(i-1)j}\ b^{i^2-4}(1-b^{2j-2})+ \\
& \quad - b^{j^2-2(i-1)j}\ b^{i^2-2}+ \\
& \quad b^{(j-i)^2}\Big\}.
\end{aligned}
$$

Now, pair up expressions, starting with the top and bottom expressions (excluding $b^{(j-i)^2}$ of course; it is a special case!) and terminating with the center ones. Some reorganization of the expressions, powers of b in particular, is essential for the proof to continue successfully.

$$
\begin{aligned}
\Lambda = \; & b^{(j-i)^2} + \\
& - b^{j^2-2(i-1)j} b^{i^2-2}\left[1 - b^{(2i-4)(j-1)}\right] + \\
& - b^{j^2-2(i-1)j} b^{i^2-4}(1 - b^{2j-2})\left[1 - b^{(2i-6)(j-2)}\frac{1 - b^{2i-2}}{1 - b^2}\right] + \\
& - b^{j^2-2(i-1)j} b^{i^2-6}(1 - b^{2j-2})(1 - b^{2j-4})\left[1 - b^{(2i-8)(j-3)}\frac{1 - b^{2i-2}}{1 - b^2}\frac{1 - b^{2i-4}}{1 - b^4}\right] + \\
& \vdots \\
& - b^{j^2-2(i-1)j} b^{i^2-2i+6}(1 - b^{2j-2})(1 - b^{2j-4})\ldots(1 - b^{2j-2(i-4)}). \\
& \qquad\qquad \left[1 - b^{4(j-i+3)}\frac{1 - b^{2i-2}}{1 - b^2}\frac{1 - b^{2i-4}}{1 - b^4}\frac{1 - b^{2i-6}}{1 - b^6}\right] + \\
& - b^{j^2-2(i-1)j} b^{i^2-2i+4}(1 - b^{2j-2})(1 - b^{2j-4})\ldots(1 - b^{2j-2(i-3)}). \\
& \qquad\qquad \left[1 - b^{2(j-i+2)}\frac{1 - b^{2i-2}}{1 - b^2}\frac{1 - b^{2i-4}}{1 - b^4}\right] + \\
& - b^{j^2-2(i-1)j} b^{i^2-2i+2}\left[(1 - b^{2j-2})(1 - b^{2j-4})\ldots(1 - b^{2j-2(i-2)})\right]\left(1 - \frac{1 - b^{2i-2}}{1 - b^2}\right).
\end{aligned}
$$

Now, define Δ_n as the $n+1^{\text{th}}$ row of the last expression:

$$
\begin{aligned}
\Delta_1 &\triangleq -b^{j^2-2(i-1)j}b^{i^2-2}\left[1 - b^{(2i-4)(j-1)}\right] \\
\Delta_2 &\triangleq -b^{j^2-2(i-1)j}b^{i^2-4}(1-b^{2j-2})\left[1 - b^{(2i-6)(j-2)}\frac{1-b^{2i-2}}{1-b^2}\right] \\
\Delta_3 &\triangleq -b^{j^2-2(i-1)j}b^{i^2-6}(1-b^{2j-2})(1-b^{2j-4})\left[1 - b^{(2i-8)(j-3)}\frac{1-b^{2i-2}}{1-b^2}\,\frac{1-b^{2i-4}}{1-b^4}\right] \\
&\vdots \\
\Delta_{i-3} &\triangleq -b^{j^2-2(i-1)j}b^{i^2-2i+6}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2(i-4)}). \\
&\qquad\left[1 - b^{4(j-i+3)}\frac{1-b^{2i-2}}{1-b^2}\,\frac{1-b^{2i-4}}{1-b^4}\,\frac{1-b^{2i-6}}{1-b^6}\right] \\
\Delta_{i-2} &\triangleq -b^{j^2-2(i-1)j}b^{i^2-2i+4}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2(i-3)}). \\
&\qquad\left[1 - b^{2(j-i+2)}\frac{1-b^{2i-2}}{1-b^2}\,\frac{1-b^{2i-4}}{1-b^4}\right] \\
\Delta_{i-1} &\triangleq -b^{j^2-2(i-1)j}b^{i^2-2i+2}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2(i-2)})\left(1-\frac{1-b^{2i-2}}{1-b^2}\right).
\end{aligned}
$$

In general,

$$
\begin{aligned}
\Delta_{i-k} &\triangleq -b^{j^2-2(i-1)j}b^{i^2-2i+2k}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2(i-k-1)}). \\
&\qquad\left[1 - b^{(k-1)(2j-2i+2k)}\frac{1-b^{2i-2}}{1-b^2}\,\frac{1-b^{2i-4}}{1-b^4}\cdots\frac{1-b^{2i-2k}}{1-b^{2k}}\right].
\end{aligned}
$$

Then,

$$
\begin{aligned}
\Lambda &= b^{(j-i)^2} + \Delta_1 + \Delta_2 + \Delta_3 + \ldots + \Delta_{i-3} + \Delta_{i-2} + \Delta_{i-1}. \\
&= b^{(j-i)^2} + \sum_{k=1}^{i-1} \Delta_{i-k}.
\end{aligned}
$$

Hence, the problem reduces to showing the sum term above is zero. The following lemma gives an explicit expression for the sum.

Lemma B.2:

$$
\begin{aligned}
\sum_{m=1}^{k-1} \Delta_{i-m} = {} & b^{j^2-2(i-1)j} b^{i^2-2i+2k} (1-b^{2j-2})(1-b^{2j-4}) \ldots (1-b^{2j-2i+2k}) \cdot \\
& \Big[\frac{1-b^{2i-2k}}{1-b^2} + b^{2j-2i+2k} \frac{1-b^{2i-2k-2}}{1-b^2} \frac{1-b^{2i-2k}}{1-b^4} + \ldots + \\
& b^{(m-1)(2j-2i+2k)} \frac{1-b^{2i-2k+2(m-1)}}{1-b^2} \frac{1-b^{2i-2k-2(m-2)}}{1-b^4} \cdots \frac{1-b^{2i-2k}}{1-b^{2m}} + \ldots \\
& b^{(k-2)(2j-2i+2k)} \frac{1-b^{2i-4}}{1-b^2} \frac{1-b^{2i-6}}{1-b^4} \cdots \frac{1-b^{2i-2k}}{1-b^{2k-2}} \Big].
\end{aligned}
\tag{B.2}
$$

Proof: We prove this lemma by using induction. Let's have a look at the sum for some values of k. This will give us an idea of what the general expression for the sum is (i.e. above was not pulled from a hat!), as well as providing the start point for the induction

$$
\begin{aligned}
\Delta_{i-1} &= -b^{j^2-2(i-1)j} b^{i^2-2i+2} (1-b^{2j-2})(1-b^{2j-4}) \ldots (1-b^{2j-2(i-2)}) \Big(1 - \frac{1-b^{2i-2}}{1-b^2}\Big) \\
&= b^{j^2-2(i-1)j} b^{i^2-2i+4} (1-b^{2j-2})(1-b^{2j-4}) \ldots (1-b^{2j-2(i-2)}) \frac{1-b^{2i-4}}{1-b^2}.
\end{aligned}
$$

$$
\begin{aligned}
\Delta_{i-1}+\Delta_{i-2} &= b^{j^2-2(i-1)j}b^{i^2-2i+4}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2(i-2)})\frac{1-b^{2i-4}}{1-b^2} \\
&\quad - b^{j^2-2(i-1)j}b^{i^2-2i+4}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2(i-3)}). \\
&\qquad\qquad \left[1-b^{2(j-i+2)}\frac{1-b^{2i-2}}{1-b^2}\frac{1-b^{2i-4}}{1-b^4}\right] \\
&= b^{j^2-2(i-1)j}b^{i^2-2i+4}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2(i-3)}). \\
&\quad \left[(1-b^{2j-2(i-2)})\frac{1-b^{2i-4}}{1-b^2}-1+b^{2(j-i+2)}\frac{1-b^{2i-2}}{1-b^2}\frac{1-b^{2i-4}}{1-b^4}\right] \\
&= -b^{j^2-2(i-1)j}b^{i^2-2i+4}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2(i-3)}). \\
&\quad \left[(1-\frac{1-b^{2i-4}}{1-b^2})+b^{2j-2i+4}\frac{1-b^{2i-4}}{1-b^2}(1-\frac{1-b^{2i-2}}{1-b^4})\right] \\
&= -b^{j^2-2(i-1)j}b^{i^2-2i+4}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2(i-3)}). \\
&\quad \left[-b^2\frac{1-b^{2i-6}}{1-b^2}-b^{2j-2i+8}\frac{1-b^{2i-4}}{1-b^2}\frac{1-b^{2i-6}}{1-b^4}\right] \\
&= b^{j^2-2(i-1)j}b^{i^2-2i+6}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2(i-3)}). \\
&\quad \left[\frac{1-b^{2i-6}}{1-b^2}+b^{2j-2i+6}\frac{1-b^{2i-4}}{1-b^2}\frac{1-b^{2i-6}}{1-b^4}\right].
\end{aligned}
$$

Now, proceed to sum more expressions.

$$
\begin{aligned}
\Delta_{i-1}+\Delta_{i-2}+\Delta_{i-3} = {} & b^{j^2-2(i-1)j} b^{i^2-2i+6}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2(i-3)}). \\
& \left[\frac{1-b^{2i-6}}{1-b^2} + b^{2j-2i+6}\frac{1-b^{2i-4}}{1-b^2}\frac{1-b^{2i-6}}{1-b^4}\right] + \\
& - b^{j^2-2(i-1)j} b^{i^2-2i+6}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2(i-4)}). \\
& \left[1 - b^{4(j-i+3)}\frac{1-b^{2i-2}}{1-b^2}\frac{1-b^{2i-4}}{1-b^4}\frac{1-b^{2i-6}}{1-b^6}\right] \\
= {} & b^{j^2-2(i-1)j} b^{i^2-2i+6}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2(i-4)}). \\
& \Bigg\{(1-b^{2j-2(i-3)})\left[\frac{1-b^{2i-6}}{1-b^2} + b^{2j-2i+6}\frac{1-b^{2i-4}}{1-b^2}\frac{1-b^{2i-6}}{1-b^4}\right] + \\
& - \left[1 - b^{4(j-i+3)}\frac{1-b^{2i-2}}{1-b^2}\frac{1-b^{2i-4}}{1-b^4}\frac{1-b^{2i-6}}{1-b^6}\right]\Bigg\} \\
= {} & b^{j^2-2(i-1)j} b^{i^2-2i+6}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2(i-4)}). \\
& [-(1-\frac{1-b^{2i-6}}{1-b^2}) - b^{2j-2i+6}\frac{1-b^{2i-6}}{1-b^2}(1-\frac{1-b^{2i-4}}{1-b^4}) + \\
& - b^{4j-4i+12}\frac{1-b^{2i-4}}{1-b^2}\frac{1-b^{2i-6}}{1-b^4}(1-\frac{1-b^{2i-2}}{1-b^6})] \\
= {} & b^{j^2-2(i-1)j} b^{i^2-2i+8}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2(i-4)}). \\
& [\frac{1-b^{2i-8}}{1-b^2} + b^{2j-2i+8}\frac{1-b^{2i-6}}{1-b^2}\frac{1-b^{2i-8}}{1-b^4} + \\
& + b^{4j-4i+16}\frac{1-b^{2i-4}}{1-b^2}\frac{1-b^{2i-6}}{1-b^4}\frac{1-b^{2i-8}}{1-b^6}].
\end{aligned}
$$

We now use induction in order to obtain the sum (B.2) for $i-1$ terms. Assume,

$$
\begin{aligned}
\sum_{m=1}^{k-1} \Delta_{i-m} = {} & \Delta_{i-1}+\Delta_{i-2}+\ldots+\Delta_{i-k+1} \\
= {} & b^{j^2-2(i-1)j} b^{i^2-2i+2k}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2i+2k}). \\
& [\frac{1-b^{2i-2k}}{1-b^2} + b^{2j-2i+2k}\frac{1-b^{2i-2k-2}}{1-b^2}\frac{1-b^{2i-2k}}{1-b^4} + \ldots + \\
& b^{(m-1)(2j-2i+2k)}\frac{1-b^{2i-2k+2(m-1)}}{1-b^2}\frac{1-b^{2i-2k-2(m-2)}}{1-b^4}\cdots\frac{1-b^{2i-2k}}{1-b^{2m}} + \ldots \\
& b^{(k-2)(2j-2i+2k)}\frac{1-b^{2i-4}}{1-b^2}\frac{1-b^{2i-6}}{1-b^4}\cdots\frac{1-b^{2i-2k}}{1-b^{2k-2}}],
\end{aligned}
$$

and consider.

$$\sum_{m=1}^{k} \Delta_{i-m} = \Delta_{i-1} + \Delta_{i-2} + \ldots + \Delta_{i-k}.$$

We will prove that our proposition is true for k.

$$\begin{aligned}
\sum_{m=1}^{k} \Delta_{i-m} &= \sum_{m=1}^{k-1} \Delta_{i-m} + \Delta_{i-k} \\
&= b^{j^2-2(i-1)j} b^{i^2-2i+2k}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2i+2k}). \\
&\quad \Big[\frac{1-b^{2i-2k}}{1-b^2} + b^{2j-2i+2k}\frac{1-b^{2i-2k-2}}{1-b^2}\frac{1-b^{2i-2k}}{1-b^4} + \ldots + \\
&\quad b^{(m-1)(2j-2i+2k)}\frac{1-b^{2i-2k+2(m-1)}}{1-b^2}\frac{1-b^{2i-2k-2(m-2)}}{1-b^4}\ldots\frac{1-b^{2i-2k}}{1-b^{2m}} + \ldots + \\
&\quad b^{(k-2)(2j-2i+2k)}\frac{1-b^{2i-4}}{1-b^2}\frac{1-b^{2i-6}}{1-b^4}\ldots\frac{1-b^{2i-2k}}{1-b^{2k-2}}\Big] + \\
&- b^{j^2-2(i-1)j} b^{i^2-2i+2k}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2(i-k-1)}). \\
&\qquad \Big[1 - b^{(k-1)(2j-2i+2k)}\frac{1-b^{2i-2}}{1-b^2}\frac{1-b^{2i-4}}{1-b^4}\ldots\frac{1-b^{2i-2k}}{1-b^{2k}}\Big] \\
&= b^{j^2-2(i-1)j} b^{i^2-2i+2k}(1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^{2j-2i+2k+2}). \\
&\quad \Big\{\Big[\frac{1-b^{2i-2k}}{1-b^2} + b^{2j-2i+2k}\frac{1-b^{2i-2k-2}}{1-b^2}\frac{1-b^{2i-2k}}{1-b^4} + \ldots + \\
&\quad b^{(m-1)(2j-2i+2k)}\frac{1-b^{2i-2k+2(m-1)}}{1-b^2}\frac{1-b^{2i-2k-2(m-2)}}{1-b^4}\ldots\frac{1-b^{2i-2k}}{1-b^{2m}} + \ldots + \\
&\quad b^{(k-2)(2j-2i+2k)}\frac{1-b^{2i-4}}{1-b^2}\frac{1-b^{2i-6}}{1-b^4}\ldots\frac{1-b^{2i-2k}}{1-b^{2k-2}}\Big] + \\
&\quad \Big[-b^{2j-2i+2k}\frac{1-b^{2i-2k}}{1-b^2} - b^{4j-4i+4k}\frac{1-b^{2i-2k-2}}{1-b^2}\frac{1-b^{2i-2k}}{1-b^4} + \ldots + \\
&\quad - b^{m(2j-2i+2k)}\frac{1-b^{2i-2k+2(m-1)}}{1-b^2}\frac{1-b^{2i-2k-2(m-2)}}{1-b^4}\ldots\frac{1-b^{2i-2k}}{1-b^{2m}} + \ldots + \\
&\quad - b^{(k-1)(2j-2i+2k)}\frac{1-b^{2i-4}}{1-b^2}\frac{1-b^{2i-6}}{1-b^4}\ldots\frac{1-b^{2i-2k}}{1-b^{2k-2}}\Big] + \\
&\quad - \Big[1 - b^{(k-1)(2j-2i+2k)}\frac{1-b^{2i-2}}{1-b^2}\frac{1-b^{2i-4}}{1-b^4}\ldots\frac{1-b^{2i-2k}}{1-b^{2k}}\Big]\Big\}.
\end{aligned}$$

Now, grouping similar powers of b inside the braces we get.

$$
\begin{aligned}
\sum_{m=1}^{k}\Delta_{i-m} = {} & b^{j^2-2(i-1)j}b^{i^2-2i+2k}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2i+2k+2})\cdot \\
& \Bigg\{\left(\frac{1-b^{2i-2k}}{1-b^2}-1\right)+ \\
& b^{2j-2i+2k}\frac{1-b^{2i-2k}}{1-b^2}\left(\frac{1-b^{2i-2k+2}}{1-b^4}-1\right)+\dots+ \\
& b^{(m-1)(2j-2i+2k)}\cdot \\
& \quad \frac{1-b^{2i-2k+2(m-2)}}{1-b^2}\,\frac{1-b^{2i-2k-2(m-3)}}{1-b^4}\cdots\frac{1-b^{2i-2k}}{1-b^{2m-2}}\left(\frac{1-b^{2i-2k+2(m-1)}}{1-b^{2m}}-1\right) \\
& b^{(k-2)(2j-2i+2k)}\frac{1-b^{2i-6}}{1-b^2}\,\frac{1-b^{2i-8}}{1-b^4}\cdots\frac{1-b^{2i-2k}}{1-b^{2k-4}}\left(\frac{1-b^{2i-4}}{1-b^{2k-2}}-1\right)+ \\
& b^{(k-1)(2j-2i+2k)}\frac{1-b^{2i-4}}{1-b^2}\,\frac{1-b^{2i-6}}{1-b^4}\cdots\frac{1-b^{2i-2k}}{1-b^{2k-2}}\left(\frac{1-b^{2i-2}}{1-b^{2k}}-1\right)\Bigg\} \\
= {} & b^{j^2-2(i-1)j}b^{i^2-2i+2k+2}(1-b^{2j-2})(1-b^{2j-4})\dots(1-b^{2j-2i+2k+2})\cdot \\
& \Bigg\{\frac{1-b^{2i-2k-2}}{1-b^2}+ \\
& b^{2j-2i+2k+2}\frac{1-b^{2i-2k}}{1-b^2}\,\frac{1-b^{2i-2k-2}}{1-b^4}+\dots+ \\
& b^{(m-1)(2j-2i+2k+2)}\cdot \\
& \quad \frac{1-b^{2i-2k+2(m-2)}}{1-b^2}\,\frac{1-b^{2i-2k-2(m-3)}}{1-b^4}\cdots\frac{1-b^{2i-2k}}{1-b^{2m-2}}\,\frac{1-b^{2i-2k-2}}{1-b^{2m}}+\dots+ \\
& b^{(k-2)(2j-2i+2k+2)}\frac{1-b^{2i-6}}{1-b^2}\,\frac{1-b^{2i-8}}{1-b^4}\cdots\frac{1-b^{2i-2k}}{1-b^{2k-4}}\,\frac{1-b^{2i-2k-2}}{1-b^{2k-2}}+ \\
& b^{(k-1)(2j-2i+2k+2)}\frac{1-b^{2i-4}}{1-b^2}\,\frac{1-b^{2i-6}}{1-b^4}\cdots\frac{1-b^{2i-2k}}{1-b^{2k-2}}\,\frac{1-b^{2i-2k-2}}{1-b^{2k}}\Bigg\}.
\end{aligned}
$$

This proves the proposed induction and hence the lemma. ∎

We apply this lemma to the sum (B.2) with $i-2$ terms.

$$
\begin{aligned}
\sum_{m=1}^{i-2} \Delta_{i-m} &= b^{j^2-2(i-1)j} b^{i^2-2}(1-b^{2j-2}) \cdot \\
&\qquad \left\{1 + b^{2j-2} + \ldots + b^{(m-1)(2j-2)} + \ldots + b^{(i-4)(2j-2)} + b^{(i-3)(2j-2)}\right\} \\
&= b^{j^2-2(i-1)j} b^{i^2-2}\left(1-b^{(i-2)(2j-2)}\right).
\end{aligned}
$$

Then,

$$
\begin{aligned}
\sum_{m=1}^{i-1} \Delta_{i-m} &= \sum_{m=1}^{i-1} \Delta_{i-m} + \Delta_1 \\
&= b^{j^2-2(i-1)j} b^{i^2-2}\left(1-b^{(i-2)(2j-2)}\right) - b^{j^2-2(i-1)j} b^{i^2-2}\left[1-b^{(2i-4)(j-1)}\right] \\
&= 0.
\end{aligned}
$$

Finally,

$$
\begin{aligned}
\Lambda &= b^{(j-i)^2} + \sum_{k=1}^{i-1} \Delta_{i-k} \\
&= b^{(j-i)^2}.
\end{aligned}
$$

∎

It remains to prove this for the case $i > j$, which brings us to the following lemma.

Lemma B.2: $\mathcal{L}_i R_j = \mathcal{L}_j R_i$.

Proof: Assume $i > j$. Then,

$$
\begin{aligned}
\mathcal{L}_i R_j = \Big(& b^{(i-1)^2} \quad b^{(i-2)^2}(1-b^{2i-2}) \quad b^{(i-3)^2}(1-b^{2i-2})(1-b^{2i-4}) \quad \ldots \\
& \left[b^{(i-j)^2}(1-b^{2i-2})(1-b^{2i-4})\ldots(1-b^{2i-2(j-1)})\right] \ldots \\
& \left[(1-b^{2i-2})(1-b^{2i-4})\ldots(1-b^2)\right] \ldots \Big).
\end{aligned}
$$

$$\begin{pmatrix} b^{(j-1)^2} \\ b^{(j-2)^2}\frac{1-b^{2j-2}}{1-b^2} \\ b^{(j-3)^2}\frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4} \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

Hence,

$$\begin{aligned}
\mathcal{L}_i R_j = {} & b^{(i-1)^2}.b^{(j-1)^2} + \\
& b^{(i-2)^2}(1-b^{2i-2}).b^{(j-2)^2}\frac{1-b^{2j-2}}{1-b^2} + \\
& b^{(i-3)^2}(1-b^{2i-2})(1-b^{2i-4}).b^{(j-3)^2}\frac{1-b^{2j-2}}{1-b^2}\frac{1-b^{2j-4}}{1-b^4} + \\
& \qquad\qquad \vdots + \\
& b^{(i-j)^2}(1-b^{2i-2})(1-b^{2i-4})\ldots(1-b^{2i-2(j-1)}).1 \\
= {} & b^{(j-1)^2}.b^{(i-1)^2} + \\
& b^{(j-2)^2}(1-b^{2j-2}).b^{(i-2)^2}\frac{1-b^{2i-2}}{1-b^2} + \\
& b^{(j-3)^2}(1-b^{2j-2})(1-b^{2j-4}).b^{(i-3)^2}\frac{1-b^{2i-2}}{1-b^2}\frac{1-b^{2i-4}}{1-b^4} + \\
& \qquad\qquad \vdots + \\
& (1-b^{2j-2})(1-b^{2j-4})\ldots(1-b^2).b^{(i-j)^2}\frac{1-b^{2i-2}}{1-b^2}\frac{1-b^{2i-4}}{1-b^4}\cdots\frac{1-b^{2i-2(j-1)}}{1-b^{2j-2}} \\
= {} & \mathcal{L}_j R_i. \qquad \blacksquare
\end{aligned}$$

The above lemma proves that the matrix product $\mathcal{L}\,R$ is symmetric. Therefore, the proof which has been presented for the case of $i \leq j$, is valid for all i, j.

In conclusion, this section has demonstrated that the product of $\mathcal{L}$ and R, lower-triangular and upper-triangular matrices, respectively, is actually the blur matrix B.

Appendix C. The Cholesky Decomposition of B

In this appendix we decompose the blur matrix, B, into the product of a lower triangular matrix, $\mathcal{L}'$, and its transpose (Cholesky decompostion). This new decomposition may offer new possibilities in reducing storage space and computation time. Also, it proves that B is a positive definite matrix. Moreover, later decompsitions can also be derived from it.

In order to obtain $\mathcal{L}'$, we resort to the results of chapter II.3, namely, the LU-decomposition of B. To do this, we introduce an auxillary diagonal matrix D as defined below. The elements of D, $\mathcal{L}'$, and, R' are denoted by δ_{ij}, λ'_{ij}, and ρ'_{ij}, respectively.

$$D \triangleq \begin{pmatrix} 1 & & & & \\ & \frac{1}{\sqrt{1-b^2}} & & & \\ & & \frac{1}{\sqrt{1-b^2}}\frac{1}{\sqrt{1-b^4}} & & \\ & & & \frac{1}{\sqrt{1-b^2}}\frac{1}{\sqrt{1-b^4}}\frac{1}{\sqrt{1-b^6}} & \\ & & & & \ddots \\ & & & & & \frac{1}{\sqrt{1-b^2}}\frac{1}{\sqrt{1-b^4}}\cdots\frac{1}{\sqrt{1-b^{2n-2}}} \end{pmatrix}, \tag{C.1}$$

and,

$$\delta_{ij} \triangleq \begin{cases} \frac{1}{\sqrt{1-b^2}}\frac{1}{\sqrt{1-b^4}}\cdots\frac{1}{\sqrt{1-b^{2i-2}}} & i=j \\ 0 & i \neq j \end{cases}. \tag{C.2}$$

We showed $B = \mathcal{L}R$, which yields

$$\begin{aligned} B &= \mathcal{L}DD^{-1}R \\ &= \mathcal{L}'R', \end{aligned}$$

where

$$\mathcal{L}' \triangleq \mathcal{L}D,$$

and,

$$R' \triangleq D^{-1}R.$$

Proving the following lemma, will give us the Cholesky decomposition of B.

Lemma C.1: $\mathcal{L}'^T = R'$.

Proof:

$$\lambda'_{ij} = \sum_{k=1}^{n} \lambda_{ik}\delta_{kj} = \lambda_{ij}\delta_{jj}. \tag{C.3}$$

Hence, using the expression for λ_{ij} (II.3.5) we have,

$$\lambda'_{ij} = \begin{cases} b^{(i-j)^2} \frac{(1-b^{2i-2})(1-b^{2i-4}) \ldots (1-b^{2i-2(j-1)})}{\sqrt{1-b^2}\sqrt{1-b^4} \ldots \sqrt{1-b^{2j-2}}} & i \geq j \\ 0 & i < j \end{cases}.$$

Similarly,

$$\rho'_{ij} = \sum_{k=1}^{n} \delta'_{ik}\rho_{kj} = \delta'_{ii}\rho_{ij};$$

where,

$$D^{-1} = \left[\delta'_{ij}\right] = diag\left[\frac{1}{\delta_{jj}}\right].$$

Then,

$$\rho'_{ij} = \begin{cases} b^{(i-j)^2} \frac{1-b^{2i-2}}{1-b^2} \frac{1-b^{2i-4}}{1-b^4} \ldots \frac{1-b^{2i-2(j-1)}}{1-b^{2(i-1)}} \sqrt{1-b^2}\sqrt{1-b^4} \ldots \sqrt{1-b^{2i-2}} & i \geq j \\ 0 & i < j \end{cases}.$$

Therefore,

$$\rho'_{ji} = \lambda'_{ij};$$

$$R' = \mathcal{L}'^T. \quad \blacksquare$$

This proves the assertion that

$$B = \mathcal{L}'\mathcal{L}'^T.$$

Appendix D. Decomposable Kernels

In this appendix, we establish the necessary and sufficient conditions for a discrete kernel to be decomposable. First, recall that a discrete kernel is called *decomposable* if the convolution of the image against this kernel can be modelled as multiplication of the image, which is represented as a matrix, F, by two degradation matrices. Specifically, a kernel K is decomposable if

$$\exists\ \tilde{B}_L, \tilde{B}_R \quad \text{such that} \quad \tilde{B}_L F \tilde{B}_R{}^T = F * K \qquad \forall\ F, \tag{II.9.1}$$

where $\tilde{B}_L$ and $\tilde{B}_R$ are the left and right degradation matrices, respectively.

Since (II.9.1) holds for all F, let

$$F_{ij} = \delta_{i-i', j-j'}, \qquad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n, \tag{D.1}$$

for some i', j'.

Denote columns of $\tilde{B}_L$ and $\tilde{B}_R$ by $\tilde{B}_{L_i}$ and $\tilde{B}_{R_i}$, respectively. Then, omitting details, we conclude

$$\tilde{B}_L F \tilde{B}_R{}^T = a_{i'} c_{j'}^T. \tag{D.2}$$

Also,

$$\begin{aligned} [F * K]_{ij} &= = \sum_{pq} F_{pq} K_{i-p, j-q} \\ &= K_{i-i', j-j'}. \end{aligned} \tag{D.3}$$

Therefore, K is decomposable if and only if

$$\tilde{B}_{L_{i,i'}} \tilde{B}_{R_{j,j'}} = K_{i-i', j-j'} \qquad \forall\ i, j, i', j'. \tag{D.4}$$

The above equivalent condition (D.4) provides the following necessary and sufficient condition for a decompsable kernel:

$$K_{pq} K_{rs} = K_{ps} K_{rq} \qquad \forall\ p, q, r, s. \tag{D.5}$$

To prove necessity, assume K is decomposable, and let $i, j, k, l, i', j', k', l'$, be such that the following hold.

$$\begin{aligned} p &= i - i' \\ q &= j - j' \\ r &= k - k' \\ s &= l - l' \end{aligned} \qquad (D.6).$$

Then,

$$\begin{aligned} K_{pq}K_{rs} &= \tilde{B}_{L_{i,i'}}\tilde{B}_{R_{j,j'}}.\tilde{B}_{L_{k,k'}}\tilde{B}_{R_{l,l'}} \\ &= \tilde{B}_{L_{i,i'}}\tilde{B}_{R_{l,l'}}.\tilde{B}_{L_{k,k'}}\tilde{B}_{R_{j,j'}} \\ &= K_{ps}K_{rq}, \end{aligned} \qquad (D.7)$$

which concludes the necessity of (D.5).

Next, we show that condition (D.5) is also sufficient. In other words, for any given kernel that satisfies (D 5), there exist left and right blur matrices satisying (9.1). Let,

$$\tilde{B}_{L_{i,i}} = 1, \qquad i = 1, 2, \ldots, m. \qquad (D.8)$$

Then,

$$\tilde{B}_{R_{j,j'}} = K_{0,j-j'}, \qquad j = 1, 2, \ldots, n. \qquad (D.9)$$

Consequently,

$$\begin{aligned} \tilde{B}_{L_{i,i'}} &= = \frac{K_{i-i',j-j'}}{\tilde{B}_{R_{j-j'}}} \\ &= \frac{K_{i-i',j-j'}}{K_{0,j-j'}} \end{aligned} \qquad (D.10).$$

This is a meanigful assignment since,

$$\frac{K_{i-i',j-j'}}{K_{0,j-j'}} = \frac{K_{i-i',l-l'}}{K_{0,l-l'}}, \qquad \forall j, j', l, l', \qquad (D.11)$$

from (D.5). The matrices $\tilde{B}_L$ and $\tilde{B}_R$ satisfy (D.4) and therefore, K is decomposable. This proves the sufficiency of (D.5).

In summary, we have proved that (D.5) is a necessary and sufficient condition for a kernel to be decomposable.