DEBLURRING GAUSSIAN BLUR

by

Robert Hummel*

and

B. Kimia**

Technical Report No. 115

Robotics Report No. 23

April, 1984

* Courant Institute of Mathematical Sciences, 251 Mercer Street, New York, NY 10021 USA. Supported in part by Office of Naval Research Grant N0014-82-K-0381, and by grants from the Digital Equipment Corporation, the Sloan Foundation, the System Development Foundation, and the IBM Corporation.

** Computer Vision and Robotics Lab., Dept. of Electrical Engineering McGill University, Montreal, Quebec, Canada.

Abstract: Spatially invariant point spread functions are a common model for image and signal degradation. In general, the process of reversing gaussian blur is unstable, and cannot be represented as a convolution filter in the spatial domain. However, if we restrict the domain of allowable functions to polynomials of degree no greater than N, then an inverse filter exists. We use Hermite polynomials to represent kernels which can be used to deblur polynomial data which has been degraded by a known amount of gaussian blur. For fixed N, the corresponding kernel gives stable deblurring among the class of functions which are gaussian filtered versions of data well approximated by polynomials degree N and less.

I. Introduction

Realistic imaging and signal sampling systems necessarily introduce spatial degradation of the original data. Frequently, the degradation process is modeled as a spatially invariant gaussian convolution. Thus if f(x) is the original data, $x \in R^n$, then the observed data is

$$h(x) = \int_{R^n} K(x - x', t)\, f(x')\, dx' ,$$

where

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

is the gaussian kernel, whose extent is parameterized by $t > 0$, and is normalized to have unit mass. How can the oiginal data f(x) be reconstructed when only h(x) and the amount of blurring t is known?

As is well known, in general the data f(x) cannot be reconstructed from h(x). Not all functions h(x) arise as blurred versions of some original data f(x), and even if an ideal function f(x) exists, arbitrarily small inaccuracies in the representation of h(x) can lead to large inaccuracies in the reconstruction of f(x).

Nonetheless, as is also well known, deblurring can in practice be accomplished by high-emphasis filtering. The inherent numerical instabilities in general cause no problems. This apparent contradiction is resolved by the observation that the deblurring process is stable when restricted to a suitable class of functions, and that this class subsumes most naturally occurring signals.

In this paper, we present kernels which can be used to deblur a fixed amount of gaussian blur, and accomplish this inverse process stably among original functions f(x) that are well approximated by polynomials of fixed degree. These kernels will provide exact reconstructions of (sufficiently low order) polynomial original data. The existence and use of these kernels has been presented elsewhere [John]; our analysis is new only in that we emphasize the use of Hermite polynomials, and give explicit formulas in terms of Hermite polynomials.

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. Solving the heat equation backwards in time is an interesting inverse problem with applications in image enhancement, signal and image representation, and perhaps even modeling of neural processes for analyzing data.

In the next section, we formulate the deblurring problem in one dimension. For data in higher dimensions, we shall simply appeal to the separability of the gaussian kernel. Our formulation is strictly in a continuous Euclidean domain. A separate analysis for discrete data is possible [see, e.g., Pratt], but presents different difficulties.

II. Formulating the Deblurring Problem

Consider the opertor Ω_t defined on $L^2(R)$ by the equation

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

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

$$\Omega_t \circ \Omega_s = \Omega_{t+s}$$

and

$$u(x,t) = (\Omega_t f)(x) \quad \text{satisfies} \quad \Delta u = u_t , \quad u(x,0) = f(x);$$

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

$$(\Omega_t f)^{\wedge}(\omega) = e^{-i\omega^2 t} \hat{f}(\omega)$$

By means of this formula, Ω_t can be extended to operate on the Schwartz class $\underline{S}'$ of Fourier transformable distributions [Rudin, Functional Analysis]. In particular, $\Omega_t f$ is defined for any polynomial f.

We will specialize to the case $t = 1/4$, and set

$$Tf = \Omega_{1/4} f = \frac{1}{\sqrt{\pi}} e^{-x^2} * f . \tag{2.1}$$

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

$$(\Omega_t f)(y) = (T\tilde{f})(y/2\sqrt{t}) ,$$

$$\text{where } \tilde{f}(x) = f(2\sqrt{t}\, x).$$

Thus the invertibility of Ω_t is settled by inverting T.

From the Fourier multiplier formula

$$(Tf)^{\hat{}}(\omega) = e^{-\omega^2/4}\,\hat{f}(\omega) , \tag{2.2}$$

and the fact that $e^{-\omega^2/4} \neq 0$ for all ω, it is clear that 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 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 T, and then respesent its inverse as a convolution on the range of T. Many such restricted domains are possible. In the next section, we consider T restricted to the class of polynomials of degree N or less.

III. Polynomial Domains

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

$$(f,g) = \int_{-\infty}^{\infty} f(x)\, g(x)\, e^{-x^2}\, dx\,, \tag{3.1}$$

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

$$H_n(x) = n! \sum_{m=0}^{n/2} (-1)^m \frac{(2x)^{n-2m}}{m!(n-2m)!}\,, \tag{3.2}$$

or by the Rodrigues formula:

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n}(e^{-x^2})\,, \tag{3.3}$$

see, e.g., [Courant & Hilbert] or [Lebedev].

<u>Observation 1</u>: T is closed on P_N.

<u>Proof</u>: We will show that $TH_n \in P_N$ for $n < N$.

$$\sqrt{\pi}\,(TH_n)(y) = \int_{-\infty}^{\infty} e^{-(y-x)^2} H_n(x)\, dx$$

$$= \int_{-\infty}^{\infty} e^{-y^2} e^{2xy} (-1)^n \frac{d^n}{dx^n}(e^{-x^n})\, dx$$

$$= 2y \int_{-\infty}^{\infty} e^{-y^2} e^{2xy} (-1)^{n-1} \frac{d^{n-1}}{dx^{n-1}}(e^{-x^2})\, dx$$

$$= \sqrt{\pi} \cdot 2y(TH_{n-1})(y)$$

Thus

$$(TH_n)(x) = 2^n x^n . \qquad \bullet \tag{3.4}$$

As a result of Observation 1, T is an isomorphism of P_N . The inverse of T on P_N is clearly given by

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

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

<u>Theorem</u>: For $f \in P_N$ and $g = Tf$, then

$$f = K_N * g \tag{3.5}$$

where

$$K_N(x) = e^{-x^2} \sum_{k=0}^{N/2} \frac{(-1)^k}{\sqrt{\pi}\, k!\, 2^k} H_{2k}(x) . \tag{3.6}$$

We will give a proof below using direct integration (as opposed to using Fourier transform distributions). Note, however, that $K_N(x)$ is not the unique function representing 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 (3.6) 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 $K_N(x)$ with standard enhancement filters. For example, for N = 3,

$$K_3(x) = \frac{2}{\sqrt{\pi}} e^{-x^2} (1 - x^2)$$

$$= \frac{1}{\sqrt{\pi}} e^{-x^2} - \frac{1}{2} \frac{d^2}{dx^2} (\frac{1}{\sqrt{\pi}} e^{-x^2}) .$$

Thus

$$K_3 * g = [1 - \frac{1}{2} \frac{d^2}{dx^2}] Tg ,$$

which is a not uncommon high emphasis filter (see, e.g., the papers by E. Machin [Ratliff], and [Rosenfeld and Kak]. In Figure 1, we display plots of K_N for several values of N.

The proof of the theorem depends on several lemmas.

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

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

$$\sqrt{\pi}\, A_{2p} = \int_{-\infty}^{\infty} e^{-x^2} x^{2p} \, dx$$

$$= -\frac{1}{2} \int (-2x) e^{-x^2} x^{2p-1} \, dx$$

$$= \frac{2p-1}{2} \int_{-\infty}^{\infty} e^{-x^2} x^{2p-2} \, dx = \sqrt{\pi} \, \frac{2p-1}{2} A_{2p-2} \, .$$

Since $A_0 = 1$,

$$A_{2p} = \frac{(2p-1)(2p-3) \ldots 1}{2^m} = \frac{(2p)!}{2^{2m} p!} \, .$$

The formula holds for $p \geq 0$.

<u>Lemma 2</u>:

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

<u>Proof</u>: For $k \geq 1$, $p \geq 1$,

$$\sqrt{\pi} \, c_{2k,2p} = \int_{-\infty}^{\infty} e^{-x^2} \left[(-1)^{2k} e^{x^2} \frac{d^{2k}}{dx^{2k}} (e^{-x^2}) \right] x^{2p} \, dx$$

$$= \int_{-\infty}^{\infty} \frac{d^{2k}}{dx^{2k}} (e^{-x^2}) \, x^{2p} \, dx$$

$$= - \, 2p \int_{-\infty}^{\infty} \frac{d^{2k-1}}{dx^{2k-1}} (e^{-x^2}) \, x^{2p-1} dx$$

$$= \sqrt{\pi} \, (2p)(2p-1) \, c_{2k-2,2p-2}$$

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 $p \geq k$,

$$c_{2k,2p} = (2p)(2p-1)\ \dots\ (2p-2k+1) \cdot c_{0,2p-2k}$$

$$= \frac{(2p)!}{(2p-2k)!} \frac{(2p-2k)!}{2^{2p-2k}(p-k)!} = \frac{(2p)!}{2^{2p-2k}(p-k)!}$$

Lemma 3: For N > k,

$$d_k = \int_{-\infty}^{\infty} K_N(x)\, x^k\, dx = \begin{cases} 0 & , \quad k \text{ odd} \\ (-1)^{k/2} \dfrac{k!}{2^k (k/2)!} & , \quad k \text{ even} \end{cases}$$

Proof: For k odd, we observe from (3.6) that $K_N(x)x^k$ is an odd integrable function, and so integrates to zero. For k = 2p,

$$\int_{-\infty}^{\infty} K_N(x)x^k dx = \int_{-\infty}^{\infty} e^{-x^2} \sum_{i=0}^{N/2} \frac{(-1)^i}{\sqrt{\pi}\ i!\ 2^i} H_{2i}(x)\ x^{2p}\ dx$$

$$= \sum_{i=0}^{N/2} \frac{(-1)^i}{i!2^i} c_{2i,2p}$$

$$= \sum_{i=0}^{p} \frac{(-1)^i}{i!2^i} \frac{(2p)!}{2^{2p-2i}(p-i)!}$$

$$= \frac{(2p)!}{2^p p!} \sum_{i=0}^{p} \frac{p!}{i!(p-i)!} (-1)^i \left(\frac{1}{2}\right)^{p-i}$$

$$= \frac{k!}{2^p p!} \left(- \frac{1}{2}\right)^p = (-1)^p \frac{k!}{2^k p!}$$

Proof of the Theorem: By equation (3.4), it suffices to show that $K_N * (2^n x^n) = H_n(x)$, $n < N$. We have

$$(K_N * 2^n x^n)(y) = \int_{-\infty}^{\infty} 2^n K_N(x)(y-x)^n dx$$

$$= \int_{-\infty}^{\infty} 2^n K_N(x) \sum_{k=0}^{n} (-1)^k \binom{n}{k} y^{n-k}\ x^k\ dx$$

$$= \sum_{k=0}^{n} \frac{2^n n!}{k!(n-k)!} (-1)^k \cdot y^{n-k} \cdot d_k$$

$$= n! \sum_{m=0}^{n/2} \frac{(-1)^{2m} 2^n}{(2m)!(n-2m)!} (-1)^m \frac{(2m)!}{2^{2m} m!} y^{n-2m}$$

$$= n! \sum_{m=0}^{N/2} \frac{(-1)^m}{m!(n-2m)!} 2^{n-2m} y^{n-2m} = H_n(y) .$$

The theorem above could have been proved using the convolution theorem and by computing the Fourier transform of $K_N(x)$. We will nonetheless compute $\hat{K}_N$ in order to show that the multiplier for K_N approaches, pointwise, the inverse of the multiplier for the operator T (see (2.2)).

Observation 2: $\hat{K}_N(\omega) \to e^{\omega^2/4}$ pointwise as $N \to \infty$.

Proof: $\hat{K}_N(\omega) = \sum_{k=0}^{N/2} \frac{(-1)^k}{\sqrt{\pi}\ k!2^k} \underline{F}[e^{-x^2} H_{2k}(x)](\omega)$, where $\underline{F}$ stands for the Fourier tansform operator. Now,

$$\underline{F}[e^{-x^2} H_{2k}(x)](\omega) = \underline{F}[(-1)^{2k} \frac{d^{2k}}{dx^{2k}} (e^{-x^2})](\omega)$$

$$= (i\omega)^{2k} \sqrt{\pi} \cdot e^{-\omega^2/4} .$$

Thus

$$\hat{K}_N(\omega) = \sum_{k=0}^{N/2} \frac{(-1)^k}{\sqrt{\pi}\ k!\ 2^k} \cdot (-1)^k \omega^{2k} \sqrt{\pi}\ e^{-\omega^2/4}$$

$$= e^{-\omega^2/4} \sum_{k=0}^{N/2} \frac{1}{k!} \left(\frac{\omega^2}{2}\right)^k$$

$$\to e^{-\omega^2/4} e^{\omega^2/2} = e^{\omega^2/4} .$$

As a consequence of Observation 2, we see that $K_N(X)$ 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. $K_N(x)$ does converge in $L^2(e^{-x^2}dx)$, but that does not imply pointwise convergence to any function. We accordingly have stable deblurring when using the kernels $K_N(x)$, where stability is measured in terms of deviation from a polynomial of degree N, and the $L^2(e^{-x^2}dx)$ norm is used as the metric.

IV. Higher Dimensions

The gaussian blur operator is given by

$$Tf(x) = \int_{R^n} \pi^{-n/2} e^{-(x-y)^2} f(y)\, dy \,. \qquad (4.1)$$

Due to the separability of the kernel and Fubini's theorem, T can be decomposed into n iterated blurrings:

$$T = T_1 \circ T_2 \circ \ldots \circ T_n \qquad (4.2)$$

$$(T_i f)(x) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi}} e^{-(x_i-y_i)^2} f(x_1 \ldots, y_i , \ldots, x_n) dy_i \qquad (4.3)$$

Consider a polynomial in R^n:

$$f(x) = \sum_{|\alpha| \leq N} a_\alpha x^\alpha \qquad (4.4)$$

$$\alpha = (\alpha_1,\ldots,\alpha_n) \,, \quad \alpha_i \in Z \,, \quad \alpha_i \geq 0 \,,$$

$$|\alpha| = \Sigma\, \alpha_i \,, \quad x^\alpha = x_1^{\alpha_1} \ldots x_n^{\alpha_n} \,.$$

For fixed x, the function of one real variable

$$g(y_i) = f(x_1,\ldots, y_i , \ldots, x_n)$$

is a polynomial of degree no greater than N, so

$$K_N * (Tg) = g \,,$$

where T is the standard one dimensional blurring operator (2.1). Combining, we find that

$$f(x) = \int_{R^n} K_N(y_1)\ K_N(y_2)\ \ldots\ K_N(y_n)\ (Tf)(x-y)\ dy \tag{4.5}$$

for any polynomial f(x) of form (4.4). Thus deblurring of blurred polynomials of degree N can be accomplished by convolution against the kernel

$$K_N^*(x) = K_N(x_1) \cdot K_N(x_2)\ \ldots\ K_N(x_n). \tag{4.6}$$

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 $x \in R^n$. Figure 2 shows a plot of $K_N^*(x)$ for n = 2, N = 3.

V. Some Experimental Results and Comments

In Figure 3, we show a 512 by 512 pixel array digitized to 256 gray levels, focused and imaged as carefully as possible. The two dimensional blurring operator T was applied to obtain Figure 4, using an interpixel distance of 0.2 in both dimensions. A VICOM Systems, Inc., image processing computer was used to perform the convolution, using 12 bits of significance in the calculation. The convolution was performed in each dimension separately, since the kernel is separable, using coefficients obtained by integrating 0.2 intervals of the gaussian function $1/\sqrt{\pi}\ (e^{-x^2})$. In a similar manner, deblurring kernels $K_N^*(x)$ were applied to obtain Figures 5, 6 and 7. Again, a 0.2 interpixel distance was used, so that the deblurring extent matched the blurring operator.

We observe that as N increases, successively better deblurrings are obtained. In no case, however, is the reconstruction visually indistinguishable from the original (Figure 3). Further, as N increases, the fluctuations in $K_N(x)$ become more violent, and will lead to numerical inaccuracies in the deblurring process.

This inability to accurately reconstruct, and the artifacts that appear as a result of the deblurring process, reflect the fact that natural image data is not represented well locally by Nth order polynomials. This does not preclude the possibility that some invertible transformation of the image data <u>is</u> well represented locally by polynomials, and so will perform better. Classes of signals other than natural images may admit better approximations by polynomials. Perhaps a treatment of the same deblurring problem using a class of analytic functions different than Nth order polynomials is possible, and will lead to better performance on image data.

The real issue of this paper is the search for an intermediate representation of signal data in a fashion that preserves only that information essential to the interpretation of the data. For imagery, this means that reconstruction of a visually similar picture should be possible from a good representation. This paper has shown that a blurred version of data is a suitable representation when the original data is well approximated locally by polynomials of bounded degree. While this may not describe general images, it may nonetheless be the key to alternate representations involving gaussian blurrings.

The supervision of this work by Professor Steven Zucker of McGill University is gratefully acknowledged.

References

Bers, John, and Schechter. Partial Differential Equations, American Mathematical Society, Providence, RI 02904 (1964).

Couarnt and Hilbert, Methods of Mathematical Physics, Wiley, New York, 1962.

John, F., "Numerical solution of the Equation of Heat conduction for preceding times," Annali di Matematica 17 (1955) 129-142.

Lebedev, N., in Special Functions and their Applications, (R. A. Silverman, Ed.), Prentice-Hall, 1965.

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

Ratcliff, F., Mach Bands: quantitative Studies on Neural Networks on the Retina, Holden-Day, 1965.

Rosenfeld and Kak, Digital Picture Processing, Academic Press, 1976.

Rudin, W., Functional Analysis, McGraw-Hill, 1973.

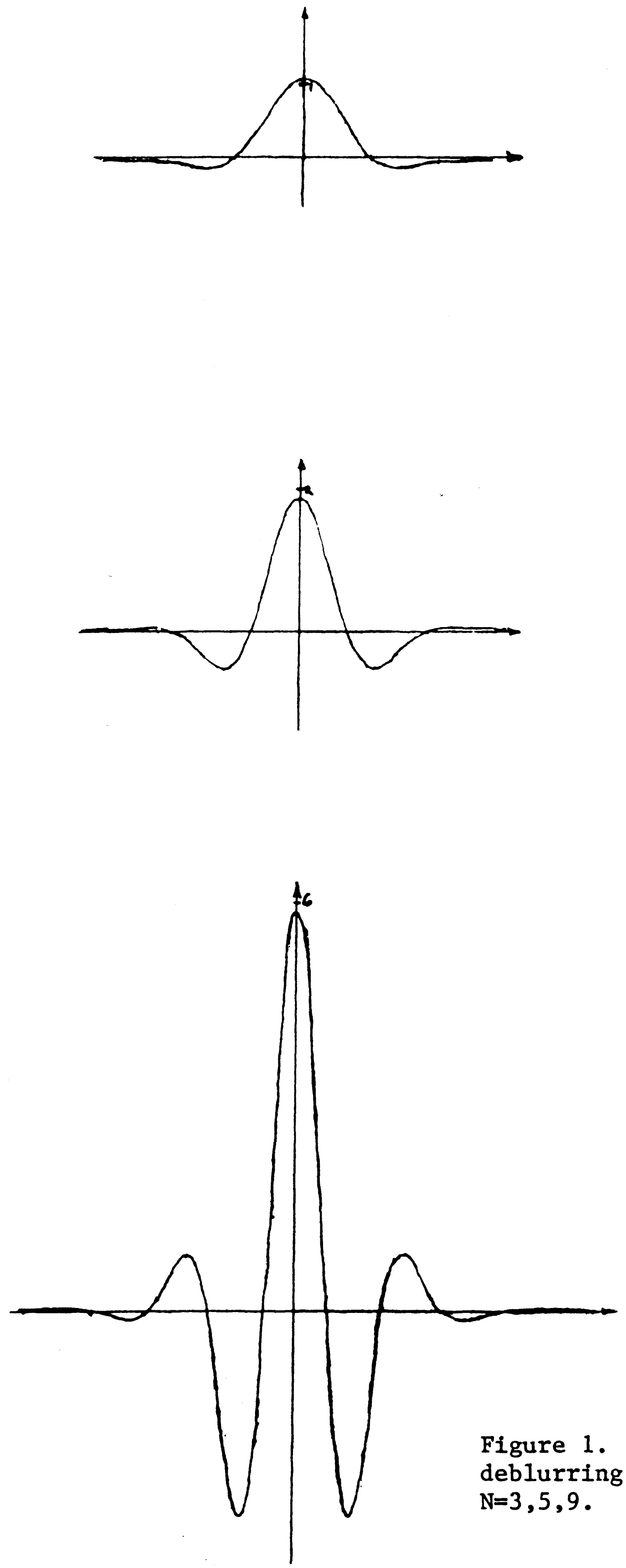

Figure 1. One dimensional deblurring kernels $K_N(x)$, N=3,5,9.

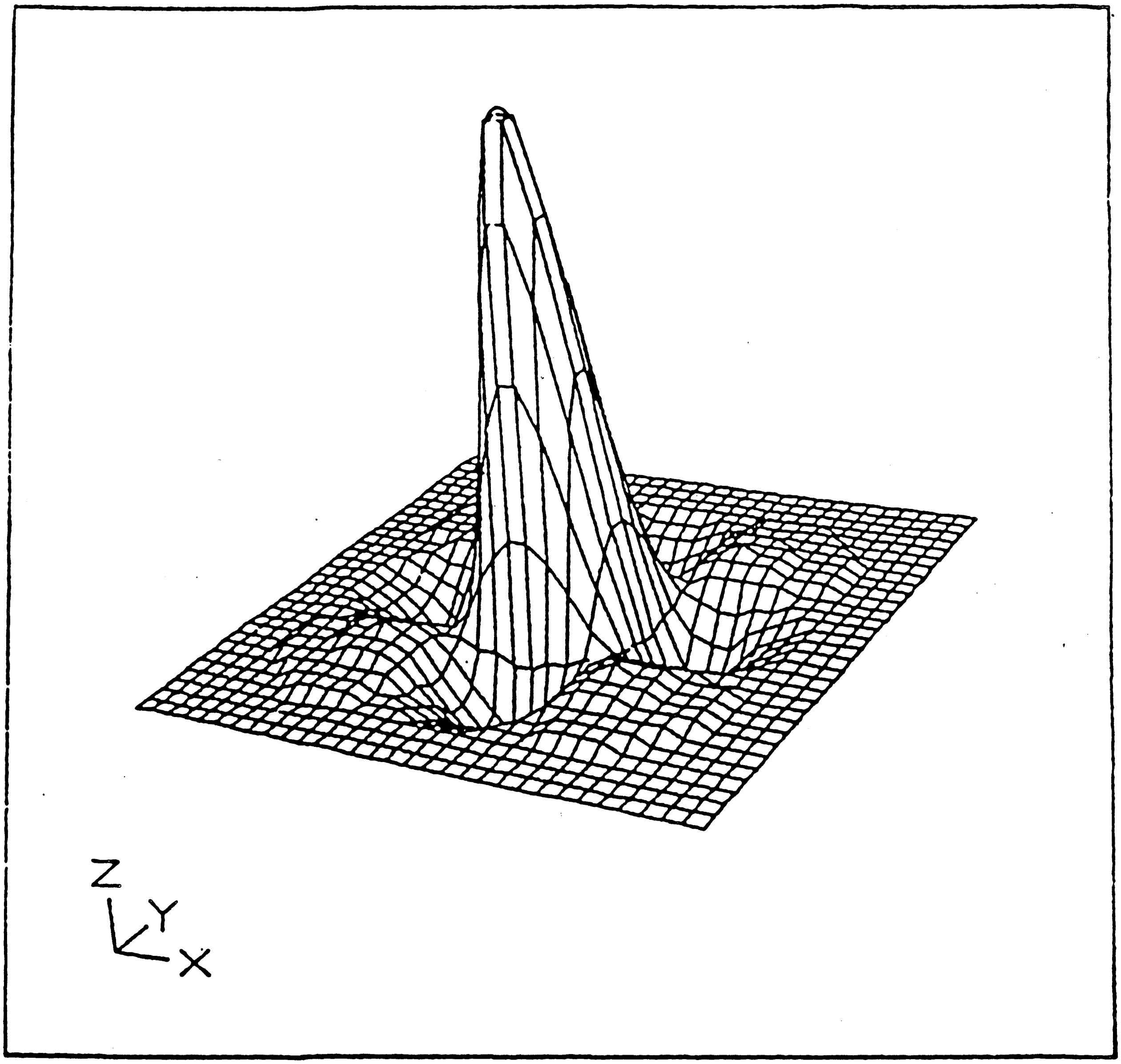

Figure 2. Two dimensional deblurring kernel, N = 3.

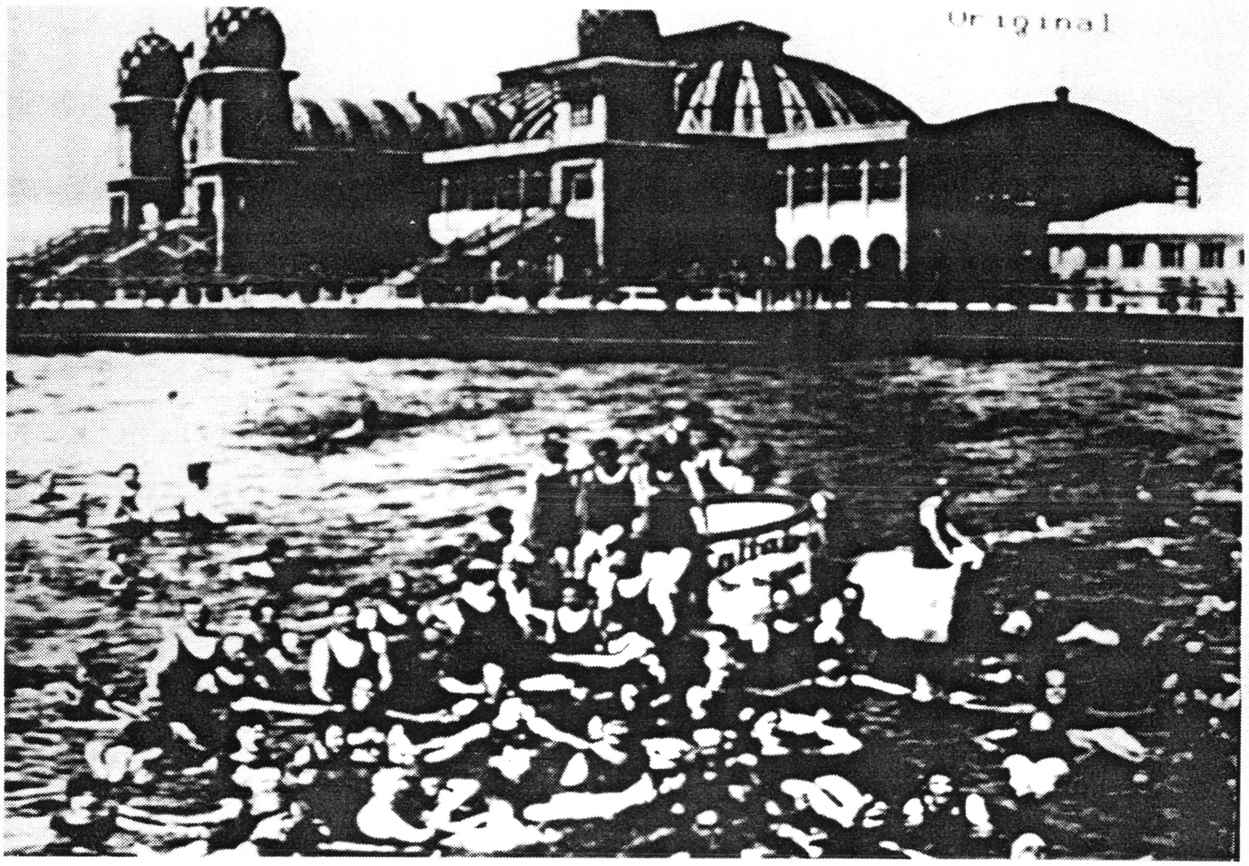

Figure 3: Original image, the Saltair resort, circa 1930.
(Photo from the Utah State Historical Society.)

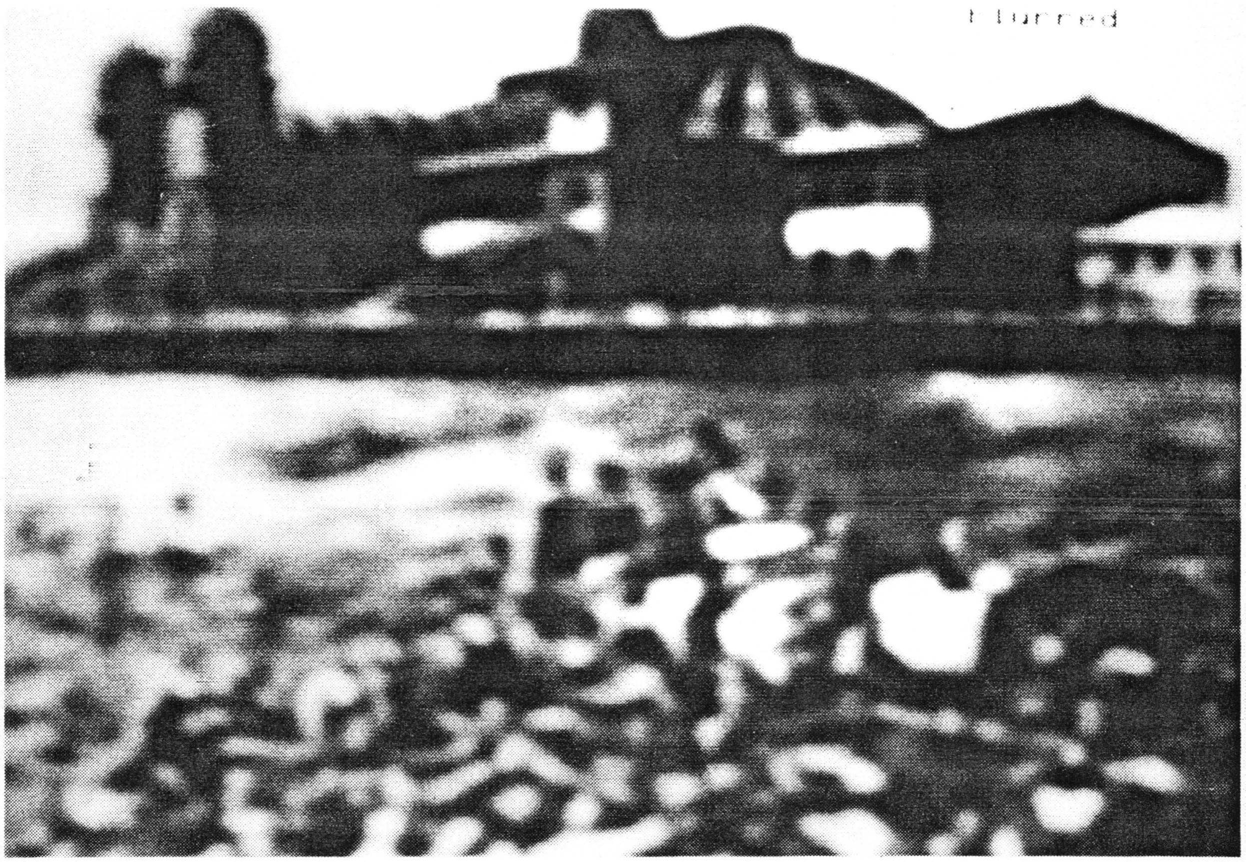

Figure 4: Blurred version of the original image in Figure 3.

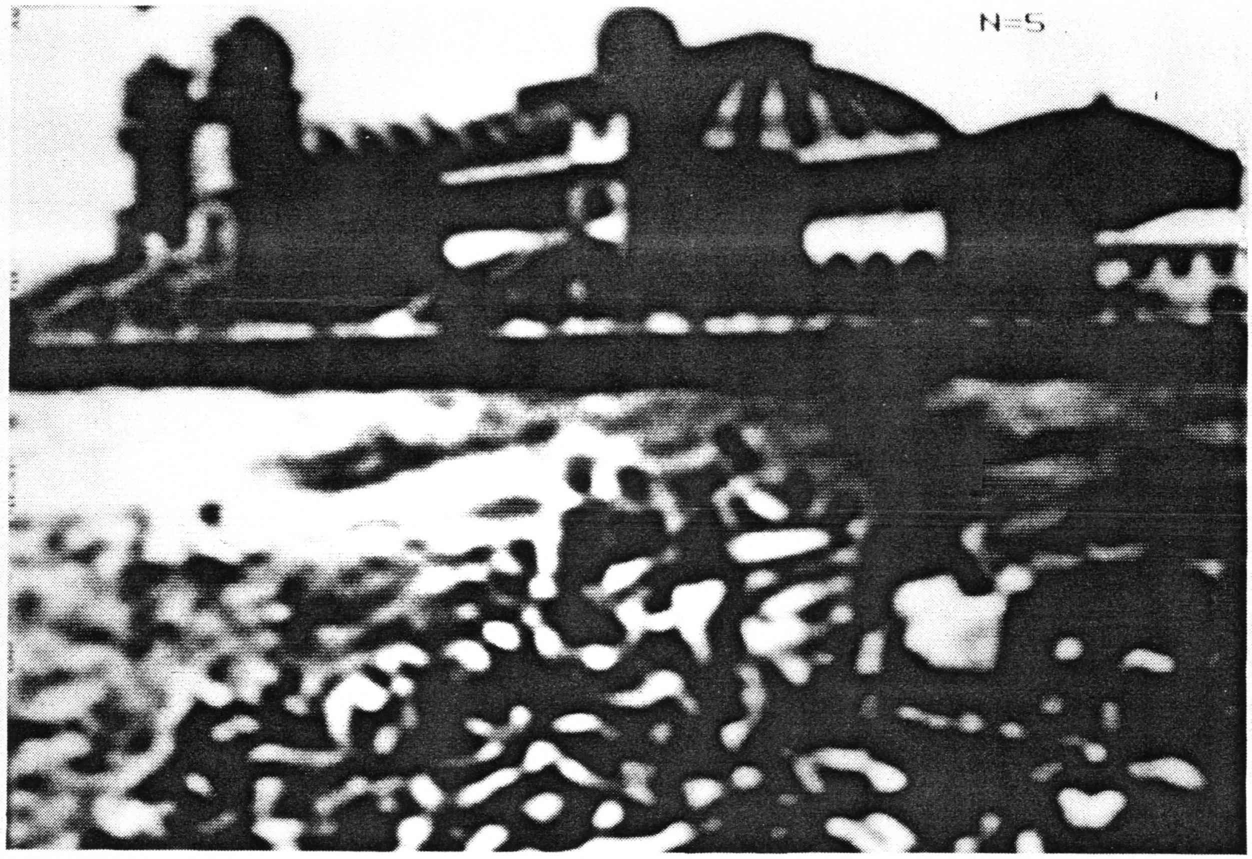

Figure 5: Deblurring kernel K_5^* applied to the blurred image in Figure 4.

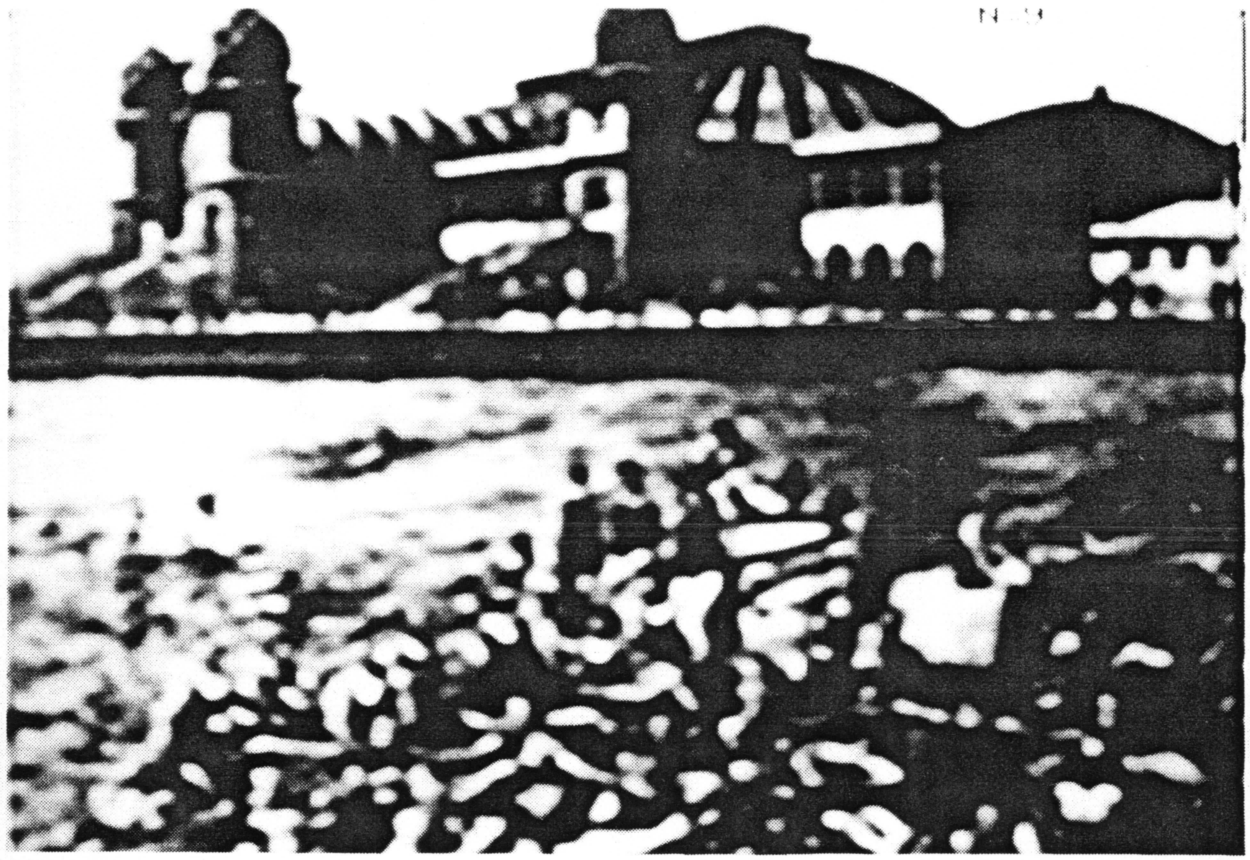

Figure 6: Deblurring kernel K_9^* applied to the blurred image in Figure 4.

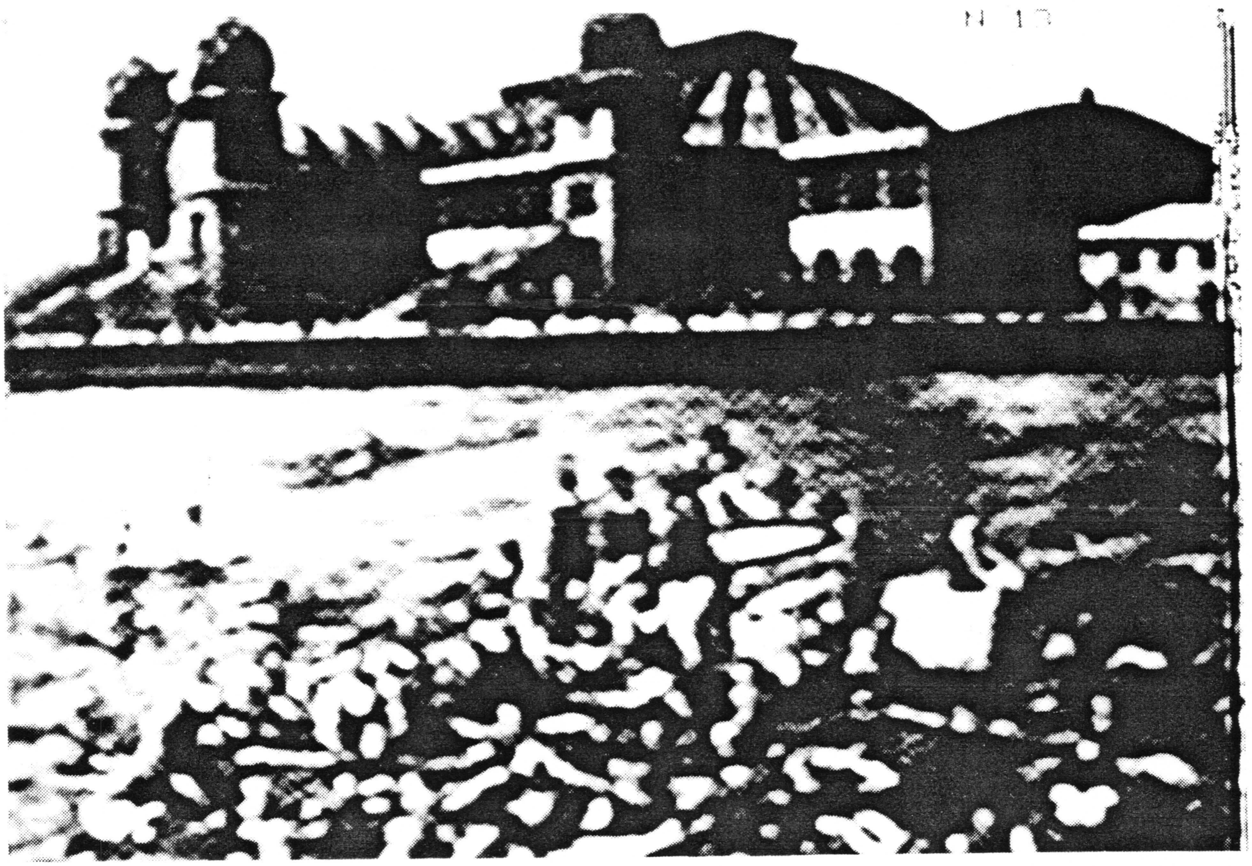

Figure 7: Deblurring kernel K_{13}^* applied to the blurred image in Figure 4.