Available online at www.sciencedirect.com
ScienceDirect
Journal of Statistical Planning and Inference 138 (2008) 2662–2680
journal of statistical planning and inference
www.elsevier.com/locate/jspi

Representing a function of two variables as a superposition over angles of ridge functions with a given shape via linear programming

Morgan Evans[a], Robert Perlmutter[a], Larry Shepp[b,*]

[a] *Varian Semiconductor Equipment Associates, Gloucester, MA 01930, USA*
[b] *Statistics Department, Rutgers University, Piscataway, NJ 08854, USA*

Available online 19 March 2008

Abstract

A ridge function with shape function g in the horizontal direction is a function of the form $g(x)h(y, 0)$. Along each horizontal line it has the shape $g(x)$, multiplied by a function $h(y, 0)$ which depends on the y-value of the horizontal line. Similarly a ridge function with shape function g in the vertical direction has the form $g(y)h(x, \pi/2)$. For a given shape function g it may or may not be possible to represent an *arbitrary* function $f(x, y)$ as a *superposition over all angles* of a ridge function with shape g in each direction, where $h = h_f = h_{f,g}$ depends on the functions f and g and also on the direction, $\theta : h = h_{f,g}(\cdot, \theta)$. We show that if g is Gaussian centered at zero then this is always possible and we give the function $h_{f,g}$ for a given $f(x, y)$. For highpass or for odd shapes g, we show it is impossible to represent an arbitrary $f(x, y)$, i.e. in general there is no $h_{f,g}$. Note that our problem is similar to tomography, where the problem is to invert the Radon transform, except that the use of the word inversion is here somewhat "inverted": in tomography $f(x, y)$ is unknown and we find it by inverting the projections of f; here, $f(x, y)$ is known, $g(z)$ is known, and $h_f(\cdot, \theta) = h_{f,g}(\cdot, \theta)$ is the unknown.

Keywords: Tomography; Radon transform; Ridge functions; Wafer doping; Ion implantation

1. Posing the problem

We ask for which given "shape functions", $g = g(x)$, $-\infty < x < \infty$, is it possible to represent *an arbitrary function*, $f(x, y)$, as a superposition over angles or directions, of ridge functions with shape g? More precisely, when is there a function $h(z, \theta)$ such that we can write say every function f which is compactly supported and smooth in some sense as $f = h \circ g$, i.e.,

$$f(x, y) = \int_0^{2\pi} g(x\cos\theta + y\sin\theta)h(-x\sin\theta + y\cos\theta, \theta)\,\mathrm{d}\theta. \tag{1}$$

Call such a shape function g "good" if it allows an arbitrary f to be so represented as $f = h \circ g$. If g happens to be even, $g(-x) = g(x)$, then it is no loss of generality to assume that h is reflective in the sense that $h(z, \theta + \pi) = h(-z, \theta)$. On the other hand, if g happens to be odd and integrable then we *cannot* represent an arbitrary integrable f as $f = h \circ g$, with h integrable. This is because if $f = h \circ g$, then f must integrate to zero as is immediately clear from (1). It seems

* Corresponding author.
E-mail address: shepp@stat.rutgers.edu (L. Shepp).

0378-3758/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.jspi.2008.03.028

likely that if g is good, then the even part of g, $g(x) + g(-x)/2$, is also good, but we have no proof of this. We will focus attention in this paper on even g's, though this is not the general case and there may well be cases of interest where g is not even. In our application to ion implantation g is, at least ideally, even. Whether or not g is even, if we restrict h to have the reflection property $h(z, \theta + \pi) = h(-z, \theta)$, then we may as well replace g by its even part, which is, of course, then even.

The problem arises in the design of a device to lay down a dopant distribution on a wafer if one can, at each angle θ, sweep out a dopant profile $g(x\cos\theta + y\sin\theta)$, multiplied by an arbitrary constant, $h(z, \theta)$, positive or negative, which depends on the direction and on the distance, $-x\sin\theta + y\cos\theta = z$, along the line. We then superpose the dopants that are laid down in all the directions. In the practical problem $h(z, \theta) \geqslant 0$, since there is no way to lay down a negative amount of dopant, but *relative to an overall constant* the dopant profile can be considered to be determinable by negative as well as positive h, as we do.

In tomography (Shepp and Logan, 1974; Shepp and Kruskal, 1978) one represents an essentially arbitrary function $f(x, y)$ as a superposition of ridge functions via the algorithm of convolution-back-projection. This is related (as will be seen later) to the problem of this paper but, at first glance, our problem seems more directly to do with the Fourier inversion formula which can be stated for a quite general smooth function f of compact support (denote by $\mathscr{F}$, the set of all such f) as

$$f(x, y) = \frac{1}{8\pi^2}\int_{-\infty}^{\infty} \mathrm{d}\rho|\rho| \int_0^{2\pi} \mathrm{d}\alpha \hat{f}(\rho, \alpha)\mathrm{e}^{\mathrm{i}\rho(x\cos\alpha + y\sin\alpha)},$$

where $\hat{f}(r, \alpha)$ is the Fourier transform of f,

$$\hat{f}(\rho, \alpha) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathrm{d}x\,\mathrm{d}y f(x, y)\mathrm{e}^{-\mathrm{i}\rho(x\cos\alpha + y\sin\alpha)}.$$

This Fourier representation of an arbitrary f already shows that the shape function $g \equiv 1$ is good since we may take (for given $f \in \mathscr{F}$),

$$h_f(z, \alpha) = \frac{1}{8\pi^2}\int_{-\infty}^{\infty} \mathrm{e}^{\mathrm{i}\rho z}|\rho|\hat{f}\left(\rho, \alpha - \frac{\pi}{2}\right)\mathrm{d}\rho. \tag{2}$$

Remark. For $g \equiv 1$, $h_{f,g}(z, \theta)$ depends only on the values of $\hat{f}(r, \alpha)$ with $\alpha = \theta - \pi/2$. This is probably because $g \equiv 1$ is very special, but we do not know if this is true in general. We shall see later on that $g \equiv 1$ is an overly special instance.

To see that there are many g's that are not good, observe that if f has a ridge function representation with some function h, then we may write by substituting the ridge function representation for f,

$$\begin{aligned}&\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\mathrm{e}^{\mathrm{i}(xu+yv)}\,\mathrm{d}x\,\mathrm{d}y\\ &\quad= \int_0^{2\pi}\mathrm{d}\theta\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x\cos\theta + y\sin\theta)h(-x\sin\theta + y\cos\theta, \theta)\mathrm{e}^{\mathrm{i}(xu+yv)}\,\mathrm{d}x\,\mathrm{d}y\right].\end{aligned}$$

Now if we change variables, x, y, inside the integral to rotated coordinates, x', y',

$$x' = x\cos\theta + y\sin\theta, \quad y' = -x\sin\theta + y\cos\theta,$$

we have, with $u = \rho\cos\alpha$, $v = \rho\sin\alpha$, that $\hat{f}(\rho, \alpha)$ is given by

$$\begin{aligned}&\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)\mathrm{e}^{\mathrm{i}(xu+yv)}\,\mathrm{d}x\,\mathrm{d}y\\ &\quad= \int_0^{2\pi}\mathrm{d}\theta\int_{-\infty}^{\infty} g(x')\mathrm{e}^{\mathrm{i}x'(u\cos\theta + v\sin\theta)}\,\mathrm{d}x'\int_{-\infty}^{\infty} h(y', \theta)\mathrm{e}^{\mathrm{i}y'(-u\sin\theta + v\cos\theta)}\,\mathrm{d}y'.\end{aligned}$$

Now if the Fourier transform

$$\hat{g}(z) = \int_{-\infty}^{\infty} g(y)\mathrm{e}^{-\mathrm{i}yz}\,\mathrm{d}y$$

vanishes for $|z| < A$, then the last line shows that

$$\hat{f}(\rho, \alpha) \equiv 0$$

for $|\rho| < A$ because

$$|-u\sin\theta + v\cos\theta| \leqslant \sqrt{u^2+v^2} < A,$$

for every θ and so the last integral is identically zero. This means that a highpass function g cannot be a good shape since f must be highpass in two dimensions if g is highpass in one dimension. On the other hand, bandpass functions are a very this set and they may be approximated by function which are not bandpass, so it is not clear whether or not an *essentially arbitrary* function, g is good.

2. Gaussian shapes

Suppose next that g is a Gaussian shape so that

$$g(z) = \mathrm{e}^{-z^2/2\sigma^2}.$$

A neat trick enables a reduction to the case $\sigma = \infty$ or $g \equiv 1$, which we have already handled above. Namely, let $h(z, \theta) = k_\sigma(z, \theta)\mathrm{e}^{-z^2/2\sigma^2}$. Then from (1) we must have that

$$f(x, y) = \int_0^{2\pi} k_\sigma(x\cos\theta + y\sin\theta, \theta)\mathrm{e}^{-(x\cos\theta+y\sin\theta)^2/2\sigma^2} g(x\cos\theta + y\sin\theta)\,\mathrm{d}\theta.$$

But for the Gaussian choice of g we see that this equation is the same as

$$f(x, y) = \int_0^{2\pi} k_\sigma(-x\sin\theta + y\cos\theta, \theta)\mathrm{e}^{-(x^2+y^2)/2\sigma^2}\,\mathrm{d}\theta,$$

which is the same as (1) with $g \equiv 1$, if we replace $f(x, y)$ by

$$f_\sigma(x, y) = f(x, y)\mathrm{e}^{(x^2+y^2)/2\sigma^2},$$

and $h(z, \theta)$ by $k_\sigma(z, \theta)$. We have already solved this in (2), so we can find $h = h_f(z, \theta)$, namely

$$k_\sigma(z, \theta) = \frac{1}{8\pi^2}\int_{-\infty}^{\infty} \mathrm{e}^{\mathrm{i}\rho z}|\rho|\hat{f}_\sigma(\rho, \theta)\,\mathrm{d}\rho,$$

and, finally,

$$h(z, \theta) = k_\sigma(z, \theta)\mathrm{e}^{-z^2/2\sigma^2}.$$

It should be pointed out that if $\sigma \to \infty$, then g becomes the δ function at zero, noting that multiplying g by a large constant for normalization is allowed since the large constant can be absorbed in h. So if we formally allow g to be the δ function at zero, we can ask whether this g is good. The answer is no, because if we write $x = r\cos\theta$, $y = r\sin\theta$, then

$$\lim_{\sigma\to 0}\int_0^{2\pi} h(x\cos\alpha + y\sin\alpha, \alpha)\frac{\mathrm{e}^{-(-x\sin\alpha+y\cos\alpha)^2/2\sigma^2}}{\sqrt{2\pi}\sigma}\,\mathrm{d}\alpha$$
$$= \lim_{\sigma\to 0}\int_0^{2\pi} h(r\cos(\theta-\alpha), \alpha)\frac{\mathrm{e}^{-r^2\sin^2(\theta-\alpha)/2\sigma^2}}{\sqrt{2\pi}\sigma}\,\mathrm{d}\alpha,$$

and if we set $\alpha = \theta + \sigma\gamma$ the limit becomes

$$\lim_{\sigma\to 0}\int_{-\pi/\gamma}^{\pi/\gamma} h(r\cos(\sigma\gamma),\, \theta+\sigma\gamma)\frac{\mathrm{e}^{-r^2\sin^2(\sigma\gamma)/2\sigma^2}}{\sqrt{2\pi}}\,\mathrm{d}\gamma = \frac{h(0,\theta)}{r},$$

which is not a general function $f(x, y)$.

3. Relating the problem to tomography

Now we will show that one really should think about tomography, and not Fourier transforms, if one wants to solve the problem of this paper.

Remark. In the application, it is only necessary to match $f(x, y)$ with the superposition of ridge functions for $x^2 + y^2 < 1$, because we may assume, by choosing units, that the wafer is the unit disk, and outside the wafer we are allowed to paint dopant arbitrarily. If g is good then f may be specified arbitrarily outside the unit disk, which shows that many different h's will produce the same $f(x, y)$ on the unit disk. This is interesting in itself, but it also suggests that we can study the problem numerically which will be done in the next sections.

In thinking more deeply about the above Remark, it is clear that if g is good, then $h_f(z, \theta)$ can depend on $f(x, y)$ only for $\sqrt{x^2 + y^2} \leqslant |z|$. This is true for $|z| = 1$, because if we extend $f(x, y)$ arbitrarily in the region outside the unit disc to another function $\hat{f}(x, y)$, say, with

$$\hat{f}(x, y) \equiv f(x, y) \quad \text{for } x^2 + y^2 \leqslant 1,$$

subject to $\hat{f}$ still being of compact support and smooth, then the function $h_{\hat{f}}(z, \theta)$ when back-projected to recover $\hat{f}(x, y)$ still produces $f(x, y)$ for $x^2 + y^2 \leqslant 1$. This means that we must have $h_{\hat{f}}(x, y) \equiv h_f(x, y)$ for $x^2 + y^2 \leqslant 1$ so that $h_f(x, y)$ does not depend on the values of f outside the unit disc. Of course there is nothing special about the unit disc so the same must hold outside the disc of radius $|z|$ for any z. More precisely, it follows that $h_f(x, y)$ only depends on the values of f in the disc centered at $(0, 0)$ with radius $\sqrt{x^2 + y^2}$, i.e.,

$$h_f(z, \theta) = \int_{\{\sqrt{x^2+y^2}\leqslant |z|\}} K_g(z, \theta; x, y) f(x, y)\,\mathrm{d}x\,\mathrm{d}y,$$

where the integral kernel K_g depends only on the good function g.

We can give the kernel K_g for $g \equiv 1$ or, as a consequence, for any of the Gaussian g's as follows: Let $P_f(t, \theta)$ denote the Radon transform of the function $f(x, y)$ which is set equal to zero for $x^2 + y^2 > 1$, i.e.,

$$P_f(t, \theta) = \int_{-\sqrt{1-t^2}}^{+\sqrt{1-t^2}} f(t\cos\theta + s\sin\theta, -s\cos\theta + t\sin\theta)\,\mathrm{d}s,$$

as in Shepp and Logan (1974) and Shepp and Kruskal (1978). Then since the one-dimensional Fourier transform of P_f is the same as $\hat{f}(\rho, \theta)$, which is the key theorem underlying tomography, we may replace $\hat{f}$ in (2) by $\hat{P}_f$; see Shepp and Logan (1974) and Shepp and Kruskal (1978), where it is also shown that with an appropriate filter function, ϕ, which realizes the derivative of the Hilbert transform of P_f, we have

$$f(x, y) \doteq \frac{1}{2\pi}\int_0^{2\pi}\mathrm{d}\theta\int_{-\infty}^{\infty}\mathrm{d}t\,\phi(x\cos\theta + y\sin\theta - t)P_f(t, \theta).$$

Now replace $P_f(t, \theta)$ by the previous formula to get

$$f(x, y) \doteq \frac{1}{2\pi}\int_0^{2\pi}\mathrm{d}\theta\int_{-\infty}^{\infty}\mathrm{d}t\,\phi(x\cos\theta + y\sin\theta - t)\int_{-\infty}^{\infty}\mathrm{d}s\, f(t\cos\theta + s\sin\theta, t\sin\theta - s\cos\theta).$$

We see that

$$h_f(z,\theta)=\frac{1}{2\pi}\int_{-\infty}^{\infty}\mathrm{d}t\,\phi(z-t)\int_{-\infty}^{\infty}\mathrm{d}s f(t\cos\theta+s\sin\theta,\,t\sin\theta-s\cos\theta).$$

Changing variables from (s,t) to (x,y) by setting

$$x=t\cos\theta+s\sin\theta,\quad y=t\sin\theta-s\cos\theta$$

shows that the kernel $K_g(z,\theta;x,y)$ is just for $g\equiv 1$ is

$$K_1(z,\theta;x,y)=\frac{1}{2\pi}\phi(z-x\cos\theta-y\sin\theta).$$

For ϕ, we may use the Shepp–Logan filter given in Shepp and Logan (1974) and Shepp and Kruskal (1978),

$$\phi(ka)=-\frac{4}{(\pi a^2(4k^2-1))},\quad k=0,\pm1,\pm,\ldots,$$

extended to be linear in between each ka and $(k+1)a$ where $a=2/(n+1)$ if the interval $[-1,1]$ is divided into n steps. Then we have

$$K(z,\theta;x,y)=\frac{a}{4\pi n}\phi(z-x\cos\theta-y\sin\theta).$$

This will give a good approximation to $f(x,y)$ when f is reasonably smooth; see Shepp and Logan (1974) and Shepp and Kruskal (1978). Similarly, we can find the kernel K_g for Gaussian

$$g(z)=g_\sigma(z)=\mathrm{e}^{-(1/2)\sigma^2z^2},$$

by replacing $f(x,y)$ by $f(x,y)\mathrm{e}^{(\sigma^2/2)(x^2+y^2)}$ as we have done and then using the above formula for convolution back-projection. The result is that with $K_\sigma(z,\theta;x,y)=K_{g_\sigma}(z,\theta;x,y)$,

$$K_\sigma(z,\theta,x,y)=\frac{1}{2\pi}\phi(z-x\cos\theta-y\sin\theta)\mathrm{e}^{(\sigma^2/2)(x^2+y^2-z^2)}.$$

The above method is not general. We discuss below *general* methods which unfortunately are not effective. Indeed obtaining an analytic formula for the case of general shape function seems impossible. Nevertheless, we shall solve the problem in the last section by an effective numerical approach, which is more practical anyway.

4. A Fourier series approach to the general problem

To reiterate, we are assuming that g is even which of course does not imply that g is good since there are bandpass functions which are even. Nevertheless, we shall expand our desired function $h=h_{f,g}(z,\beta)$ in a Fourier series to get

$$h(z,\beta)=\sum_{k=0}^{\infty}\gamma_k(z)\cos(k\beta)+\sum_{k=1}^{\infty}\sigma_k(z)\sin(k\beta),$$

where we seek the coefficients $\gamma_k(z)$, $k\geqslant 0$, $\sigma_k(z)$, $k\geqslant 1$. Since we may as well choose h reflexive, $h(z,\theta+\pi)=h(-z,\theta)$, we see that we must have, for all z,

$$\gamma_k(z)=(-1)^k\gamma_k(-z),\quad \sigma(z)=(-1)^k\sigma(-z),\quad k\geqslant 0.$$

Thus γ_k and σ_k are odd or even according as k is odd or even.

If we set $x=r\cos\theta$, $y=r\sin\theta$, from (1),

$$f(r\cos\theta,r\sin\theta)=\int_0^{2\pi}\mathrm{d}\beta h(r\cos(\theta-\beta),\beta)g(r\sin(\theta-\beta)).$$

Replace h by its Fourier series, multiply by $\cos(n\theta)$ (resp. by $\sin(n\theta)$), and integrate over $\theta \in (0, 2\pi)$ to get after cancelling odd terms which integrate to zero,

$$f_n^{(1)}(r) = \frac{1}{4\pi}\int_0^{2\pi} f(r\cos\theta, r\sin\theta)\cos(n\theta)\,\mathrm{d}\theta = \int_0^{\pi/2} \mathrm{d}\alpha\gamma_n(r\cos\alpha)g(r\sin\alpha)\cos(n\alpha),$$

$$f_n^{(2)}(r) = \frac{1}{4\pi}\int_0^{2\pi} f(r\cos\theta, r\sin\theta)\sin(n\theta)\,\mathrm{d}\theta = \int_0^{\pi/2} \mathrm{d}\alpha\sigma(r\cos\alpha)g(r\sin\alpha)\cos(n\alpha).$$

We can bring these equations to nearly convolution form if we substitute for α, $r\cos\alpha = u$, to obtain, with $T_n(z) = \cos(n\cos^{-1} z)$, the Chebysheff polynomial of degree n of the first kind,

$$f_n^{(1)}(r) = \int_0^r \mathrm{d}u\gamma_n(u)T_n\left(\frac{u}{r}\right)\frac{g(\sqrt{r^2-u^2})}{\sqrt{r^2-u^2}}, \quad n \geqslant 0,$$

$$f_n^{(2)}(r) = \int_0^r \mathrm{d}u\sigma_n(u)T_n\left(\frac{u}{r}\right)\frac{g(\sqrt{r^2-u^2})}{\sqrt{r^2-u^2}}, \quad n \geqslant 1.$$

We need to invert these equations to find γ_n and σ_n. It is fairly easy to find $\gamma_0(z)$ since the equation for γ_0 becomes after the substitutions, $r = \sqrt{s}$, $u = \sqrt{v}$,

$$2f_0^{(1)}(\sqrt{s}) = \int_0^s \frac{\gamma_0(\sqrt{v})}{\sqrt{v}}\frac{g(\sqrt{s-v})}{\sqrt{s-v}}\,\mathrm{d}v,$$

which is easily solved by deconvolution, at least in principle. Taking Laplace transforms gives the Laplace transform of $\gamma_0(z)$:

$$\int_0^\infty \mathrm{e}^{-\lambda s}\frac{\gamma_0(\sqrt{s})}{\sqrt{s}}\,\mathrm{d}s = \frac{2\int_0^\infty \mathrm{e}^{-\lambda s} f_0^{(1)}(\sqrt{s})\,\mathrm{d}s}{\int_0^\infty \mathrm{e}^{-\lambda s}(g(\sqrt{s})/\sqrt{s})\,\mathrm{d}s}.$$

Since we know $f_0^{(1)}(r)$ and $g(z)$, the right side is known in principle, and so the left side is also known and so we can obtain $\gamma_0(z)$ in principle by inverting a Laplace transform. The result is

$$\int_0^\infty \mathrm{e}^{-\lambda s}\gamma_1(\sqrt{s})\,\mathrm{d}s = \frac{2\int_0^\infty \mathrm{e}^{-\lambda s} f_1^{(1)}(\sqrt{s})\sqrt{s}\,\mathrm{d}s}{\int_0^\infty \mathrm{e}^{-\lambda s}(g(\sqrt{s})/\sqrt{s})\,\mathrm{d}s}.$$

We have to obtain all the other γ_n, as well as the σ_n, $n \geqslant 1$. Finding σ_n involves solving the same integral equation as γ_n, except the left side is $f_n^{(2)}(r)$ instead of $f^{(1)}(r)$. Thus we only need to study

$$f_n(r) = \int_0^r \gamma_n(u)T_n\left(\frac{u}{r}\right)\frac{g(\sqrt{r^2-u^2})}{\sqrt{r^2-u^2}}\,\mathrm{d}u.$$

The substitutions $r = \sqrt{s}$, $u = \sqrt{v}$ bring the equation to the form

$$2f_n(\sqrt{s}) = \int_0^s \frac{\gamma_n(\sqrt{v})}{\sqrt{v}}\frac{g(\sqrt{s-v})}{\sqrt{s-v}}T_n\left(\sqrt{\frac{v}{s}}\right)\mathrm{d}v. \tag{3}$$

For $n = 1$, (3) becomes, since $T_1(z) = z$,

$$2f_1(\sqrt{s})\sqrt{s} = \int_0^s \gamma_1(\sqrt{v})\frac{g(\sqrt{s-v})}{\sqrt{s-v}}\,\mathrm{d}v,$$

which is again a convolution, and we can solve it for $\gamma_1(z)$ in the same way as we solved for $\gamma_0(z)$ above. Similarly we can obtain $\sigma_1(r)$. When $n > 1$, (3) is no longer a convolution because $T_2(z) = 2z^2 - 1$. The equation becomes for

$n = 2$, after multiplying by s,

$$2f_2(\sqrt{s})s = \int_0^s \frac{\gamma_2(\sqrt{v})}{\sqrt{v}} \frac{g(\sqrt{s-v})}{\sqrt{s-v}}(v + v - s)\,\mathrm{d}v,$$

which is the sum of two convolutions. If we write

$$F(\lambda) = \int_0^\infty 2f_2(\sqrt{s})s\mathrm{e}^{-\lambda s}\,\mathrm{d}s,$$

$$G(\lambda) = \int_0^\infty \frac{\gamma_2(\sqrt{v})}{\sqrt{v}}\mathrm{e}^{-\lambda v}\,\mathrm{d}v,$$

$$\hat{g}(\lambda) = \int_0^\infty \frac{g(\sqrt{s})}{\sqrt{s}}\mathrm{e}^{-\lambda s}\,\mathrm{d}s,$$

then the above equation becomes after taking Laplace transforms,

$$F(\lambda) = -G'(\lambda)\hat{g}(\lambda) + G(\lambda)\hat{g}'(\lambda),$$

which is a differential equation for the unknown $G(\lambda)$. It is easily solved by multiplying by $1/\hat{g}(\lambda)$. The answer is

$$G(\lambda) = \hat{g}(\lambda)\int_\lambda^\infty \frac{F(u)}{\hat{g}^2(\lambda)}\,\mathrm{d}u,$$

from which, in principle, we can determine $\gamma_2(z)$. Similarly we can, in principle, determine $\sigma_2(z)$. The same simple differential equation works to obtain the Laplace transform of $\gamma_3(z)$ and $\sigma_3(z)$, but for $n = 4$, $T_4(z) = 8z^4 - 8z^2 + 1$, and to obtain $\gamma_4(z)$ involves solving a second order differential equation, which we do not pursue here.

5. Expanding in Hermite polynomials

If we could find K_g for any good function g of the form

$$g(z) = \mathrm{e}^{-(z^2/2)}P(z),$$

where P is a non-zero polynomial in z, then since such functions are dense in all of L^2, by Hermite polynomial theory, this might completely settle the practical as well as the theoretical problem and show that the good functions are those that are not highpass. We would need to show that for any non-zero polynomial P, g is a good function. Indeed suppose P is given and non-zero we will show that if we choose

$$h(z, \theta) = \mathrm{e}^{-(z^2/2)}Q(z, \theta),$$

where Q is a polynomial in z whose coefficients depend on θ, then we can approximate any $f(x, y)$. It will then follow that the good functions are essentially those which are not highpass. Using this h will give the following $f = f_Q$:

$$f_Q(x, y) = \int_0^{2\pi} \mathrm{e}^{-(x^2+y^2)/2}Q(x\cos\theta + y\sin\theta, \theta)P(-x\sin\theta + y\cos\theta)\,\mathrm{d}\theta.$$

Let $\mathscr{F}$ denote the class of all $f \in L^2(D)$, with D the unit disk, which can be so represented with a choice of polynomial $Q(z, \theta)$ whose coefficients depend on θ. We claim that $\mathscr{F}$ is dense in $L^2(D)$ as $Q(x, \theta)$ varies. If not, there exists an $f \in L^2(D)$ which is orthogonal to all f_Q's. This means that in particular, for any m and any k, letting $Q(z, \theta) = z^m\mathrm{e}^{\mathrm{i}k\theta}$,

$$\int_D \mathrm{d}x\,\mathrm{d}y \int_0^{2\pi} \mathrm{d}\theta(x\cos\theta + y\sin\theta)^m\mathrm{e}^{\mathrm{i}k\theta}P(-x\sin\theta + y\cos\theta)f(x, y) = 0.$$

But if this holds, we can interchange the order of integration to obtain that identically in θ,

$$\int_D \mathrm{d}x\,\mathrm{d}y(x\cos\theta + y\sin\theta)^m P(-x\sin\theta + y\cos\theta)f(x, y) = 0.$$

But the left side is a trigonometric polynomial in θ for any m and so we must have that $f \equiv 0$, a.s. That is we can represent any $f \in L^2(D)$. Thus any g of the form $P(z)e^{-z^2/2}$ is good, and since this is a dense set of $g \in L^2(\mathscr{R})$, we see that the good functions are dense in $L^2(\mathscr{R})$. This is an existence proof that except for the highpass functions, all functions are good. But we need more than an existence proof: Can we get a formula for h_f in terms of f? This is the practical problem to solve.

6. A power series approach

Let us suppose that g is analytic in the unit disk, D, that is

$$g(z) = \sum_{n=0}^{\infty} g_n z^n,$$

and let us drop the assumption that g is even for a moment. We seek a function $h(z, \theta)$ which we also expand analytically as

$$h(z, \theta) = \sum_{n=0}^{\infty} \sum_{k=-\infty}^{\infty} h_{n,k} z^n e^{-ik\theta}.$$

As seen in the first section that when g is even we may as well assume that $h(z, \theta + \pi) = h(-z, \theta)$ which implies that

$$h_{n,k}((-1)^k - (-1)^n) \equiv 0,$$

so that unless $n - k$ is even, we have $h_{n,k} = 0$. Since we are not going to assume g is even, we have to find $h_{n,k}$ for all values of n, k.

Assume that $f(x, y)$ is an arbitrary analytic function in D, with power series

$$f(x, y) = \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} f_{j,k} x^j y^k, \quad x^2 + y^2 < 1.$$

Setting $x = r\cos\theta$, $y = r\sin\theta$, and replacing the earlier use of θ by α, we get the equation, for every choice of the coefficients, $f_{j,k}$,

$$\sum_{n=0}^{\infty} \sum_{j=0}^{n} f_{n-j,j} (r\cos\theta)^{n-j} (r\sin\theta)^j = \int_0^{2\pi} h(r\cos(\theta - \alpha), \alpha) g(r\sin(\theta - \alpha))\, d\alpha.$$

Expanding the right side analytically and equating powers of r on both sides,

$$\sum_{j=0}^{n} f_{n-j,j} (\cos\theta)^{n-j} (\sin\theta)^j = \int_0^{2\pi} \sum_{j=0}^{n} \sum_{k=-\infty}^{\infty} h_{n-j,k} (\cos(\theta - \alpha))^{n-j} g_j (\sin(\theta - \alpha))^j e^{-ik\alpha}\, d\alpha.$$

Setting $\alpha = \theta - \beta$, and interchanging sides, we get that

$$\int_0^{2\pi} \sum_{j=0}^{n} \sum_{k=-\infty}^{\infty} h_{n-j,k} (\cos\beta)^{n-j} g_j (\sin\beta)^j e^{-ik(\theta-\beta)}\, d\beta = \sum_{j=0}^{n} f_{n-j,j} (\cos\theta)^{n-j} (\sin\theta)^j.$$

The only way this can happen is if the right side expanded as a trigonometrical series in θ has the coefficients given on the left side, i.e., for all n, k, with $n \geqslant |k|$, $n - k$ even, we have

$$\sum_{j=0}^{n} (h_{n-j,k} g_j - f_{n-j,j}) A_{n,j,k} \equiv 0, \tag{4}$$

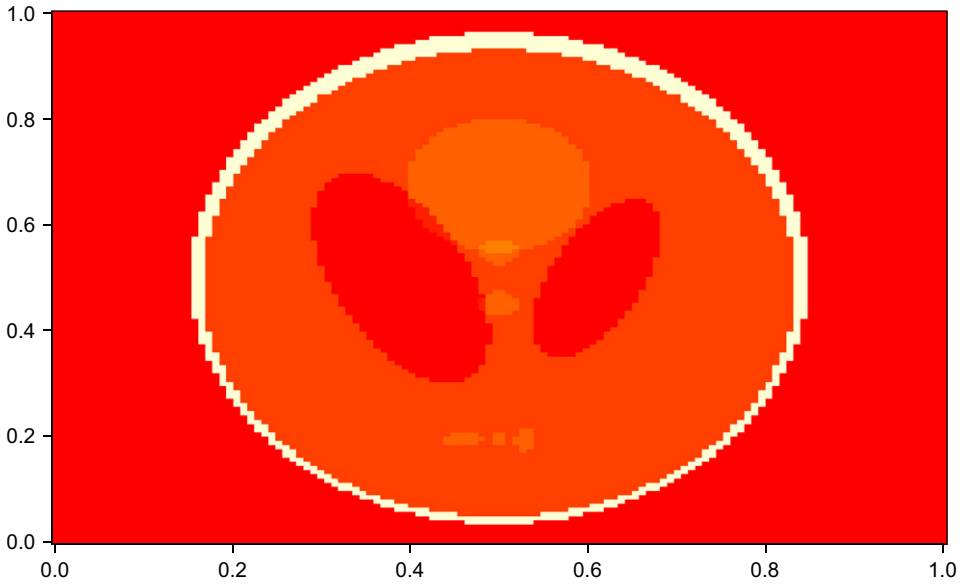

Fig. 1. This is a 128×128 plot of $f(x, y)$.

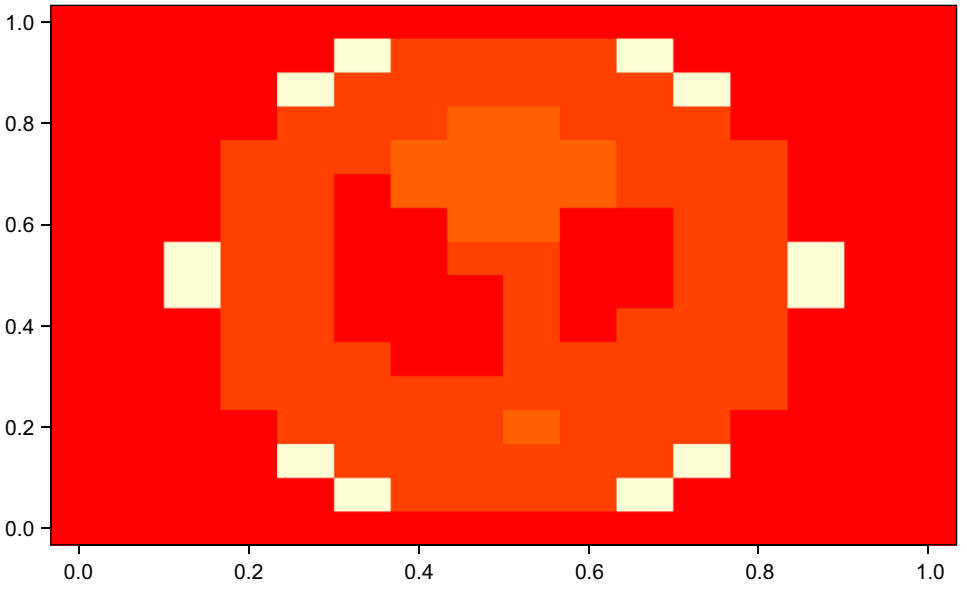

Fig. 2. This is a 16×16 plot of the original $f(x, y)$ phantom.

where

$$A_{n,j,k} = \int_0^{2\pi} (\cos\beta)^{n-j} (\sin\beta)^j \, \mathrm{e}^{\mathrm{i}k\beta} \, \mathrm{d}\beta,$$

noting that $A_{n,j,k} = 0$ if $n - k$ is odd. We can solve for all the non-zero values of $h_{n,k}$, $|k| \leqslant n$, $n - k$, even, in order, as follows: first set $n = 0$. There is only one non-zero value of $h_{0,k}$, that for $k = 0$, and we get from (4),

$$h_{0,0} g_0 = f_{0,0},$$

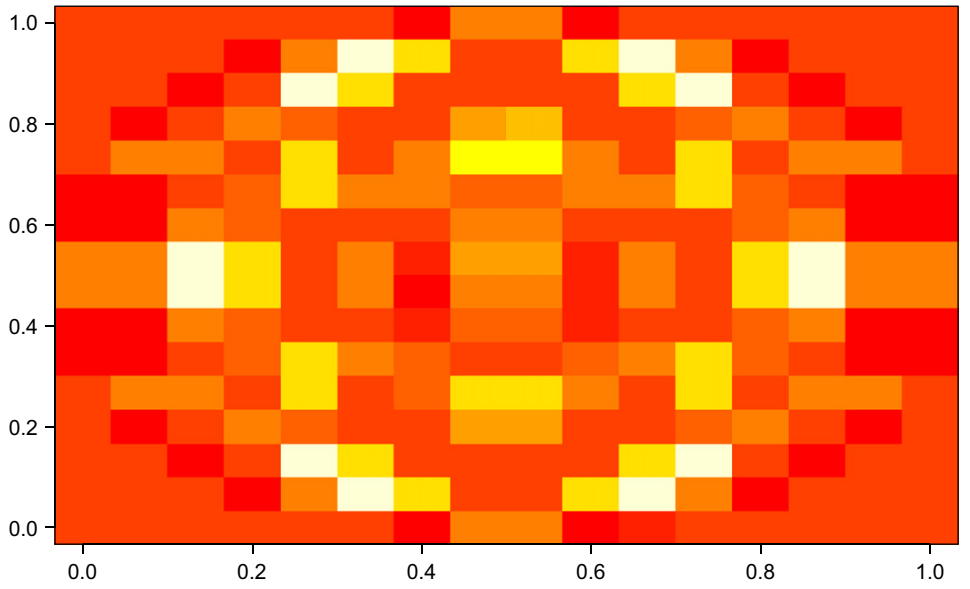

Fig. 3. A 16×16 plot of the sum of 10 ridge functions with shape $g(z) = \mathrm{e}^{-0.5z^2}$, optimally chosen to reconstruct $f(x, y)$ to within the smallest tolerance possible, which turns out to be $\varepsilon = 0.01978$. If instead we use 20 angles, we get $\varepsilon = 0.001886$, and there is no visual difference with Fig. 2.

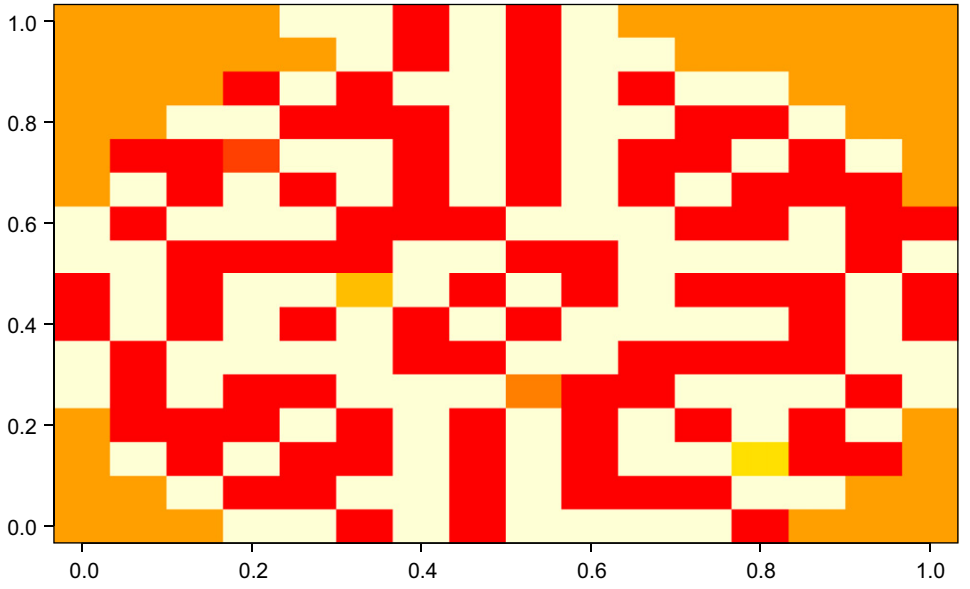

Fig. 4. The difference between the original $f(x, y)$ phantom on a 16×16 grid and the reconstruction of Fig. 3. Note that the largest difference is $\varepsilon = 0.001886$ and there are about as many positive as there are negative errors.

which we can solve as long as $g_0 \neq 0$. For $n = 1$, the non-zero values of $h_{1,k}$ are two, $h_{1,1}$, $h_{1,-1}$. From (4) these are

$$h_{1,1}g_0 = h_{1,-1}g_0 = f_{1,0}.$$

For $n = 2$, we need to find $h_{2,k}$, $k = -2, 0, 2$. From (4) we get for $n = 2$, $k = -2, 0, 2$, taking into account that

$$(h_{2,-2}g_0 - f_{2,0})A_{2,0,-2} + (h_{1,-2}g_1 - f_{1,1})A_{1,1,-2} = 0,$$

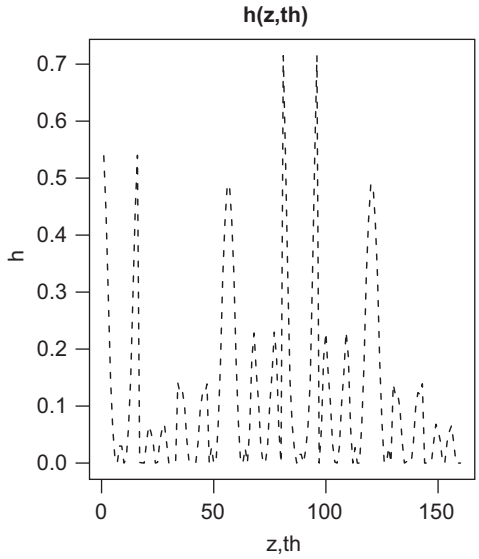

Fig. 5. A plot of the weights $h(z_i, \theta, k)$, $i = 1, \ldots, 16$; $k = 1, \ldots, 10$ in sequence for 10 angles. Note the weights are all positive, and the dynamic range seems quite reasonable.

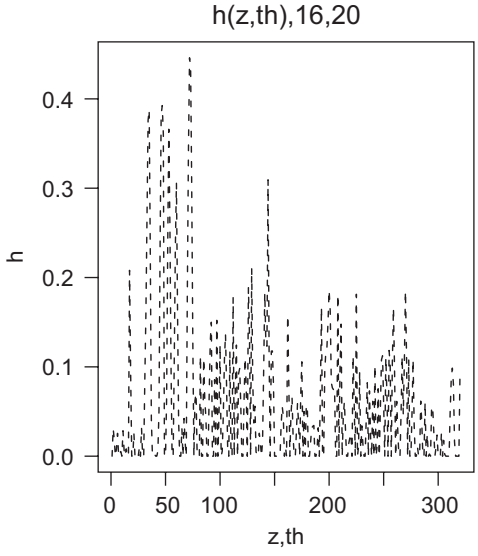

Fig. 6. A plot of the weights $h(z_i, \theta, k)$, $i = 1, \ldots, 16$; $k = 1, \ldots, 20$ in sequence for 20 angles. Note the weights are all positive, and the dynamic range seems even more reasonable than for 10 angles.

$$(h_{2,0}g_0 - f_{2,0})A_{2,0,0} + (h_{1,0}g_1 - f_{1,1})A_{1,1,0} = 0,$$

$$(h_{2,2}g_0 - f_{2,0})A_{2,0,2} + (h_{1,2}g_1 - f_{1,1})A_{1,1,2} = 0.$$

Noting that the terms $h_{n,k} = 0$ if $n - k$ is odd, we can solve for $h_{n,k}$ in terms of earlier known quantities because $A_{n,0,k} \neq 0$ whenever $|k| \leqslant n$ and $n - k$ is even. This algorithm allows a solution to the general problem at least for even analytic profile functions g with $g(0) \neq 0$. It remains to see whether or not this will give a truly practical solution.

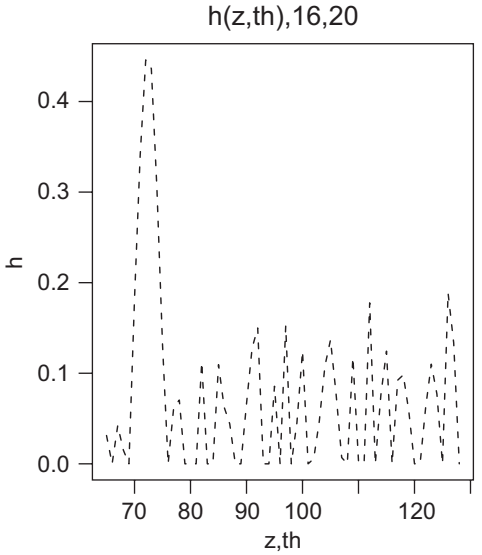

Fig. 7. A plot of the weights $h(z_i, \theta, k)$, $i = 1, \ldots, 16$; $k = 5, \ldots, 8$ in sequence for the case in Fig. 6. These are the values of h for four directions.

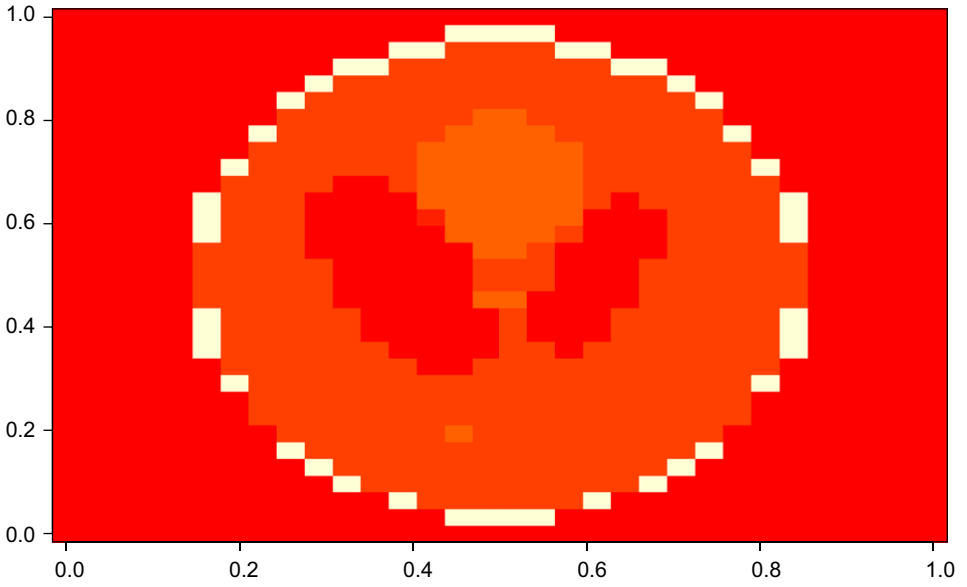

Fig. 8. This is a 32×32 plot of the original $f(x, y)$ phantom.

7. A general linear programming approach to the problem

In the early days of CT scanning a direct equation-solving approach, called ART, was widely used. Convolution-back-projection eventually became the method of choice, for several reasons, cf. Shepp and Kruskal (1978). In emission tomography, the method of EM and maximizing likelihood (Vardi et al., 1985) is regarded as at least competitive with convolution-back-projection, despite having longer execution times, because it deals with low count as the main physics based limiting factor. If one turns to numerical methods of equation-solving for our problem here, then which method

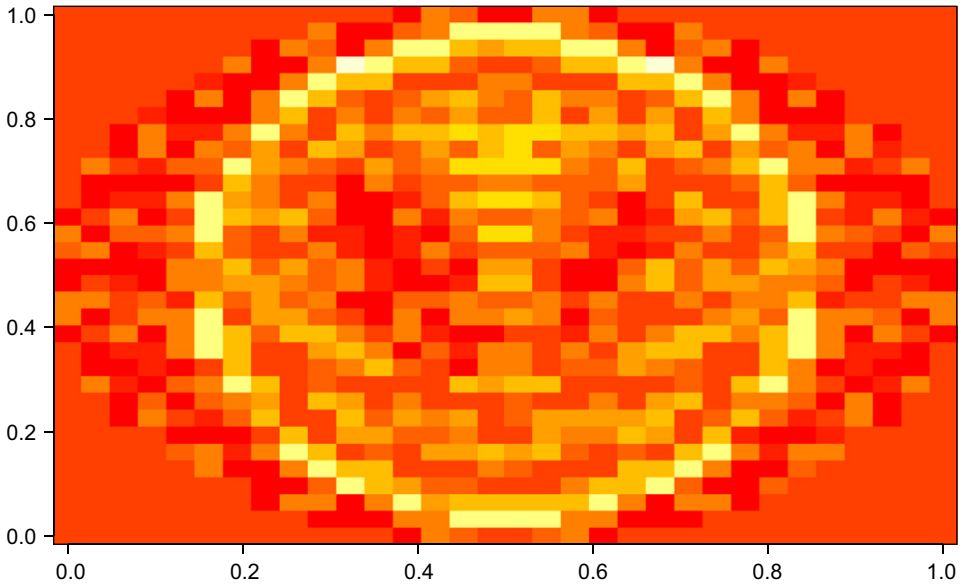

Fig. 9. A 32 × 32 plot of the sum of 16 ridge functions with shape $g(z) = e^{-0.5z^2}$, optimally chosen to reconstruct $f(x, y)$ to within the smallest tolerance possible, which turns out to be $\varepsilon = 0.020097$. If instead we will use 32 angles, we will probably have ε small enough that there is no visual difference with Fig. 8. Unfortunately the program does not terminate when $mc = mv = n = 32$. This needs a better linear programming package than was available to me.

should be used? The previous sections seem to indicate that there is no way to solve the problem with Fourier analysis or with any other closed form solution, so we seem to be forced into numerics. We could use the EM algorithm (Vardi et al., 1985) or some other ART type approach to find $h_f(z, \theta)$ numerically. We have not tried such alternatives, but, we argue that there are advantages to the linear programming approach.

For general g's we can formulate the practical problem as a *linear program* as follows: First suppose that $f = f(x, y)$ is given and that we seek values of unknowns $h(z, \theta)$, $0 \leqslant \theta \leqslant 2\pi$, $|z| \leqslant 1$ and ε to satisfy the constraints, for $x^2 + y^2 < 1$,

$$\int_0^{2\pi} g(x\cos\theta + y\sin\theta)h(-x\sin\theta + y\cos\theta, \theta)\,\mathrm{d}\theta < f(x, y) + \varepsilon,$$

$$\int_0^{2\pi} g(x\cos\theta + y\sin\theta)h(-x\sin\theta + y\cos\theta, \theta)\,\mathrm{d}\theta > f(x, y) - \varepsilon.$$

Remark. To help the reader think about ε, note that in practice, the dose error, ε, across the wafer, is described and specified statistically as the proportion of chips dosed-to-target within a specified dose range. If the target is N atoms per square centimeter, then a common semiconductor industry specification is that the standard deviation of the measured distribution of dose on chips across a wafer is less than 0.5% of the target level. Similar and concurrent statistical limits on dose variation exist for sets, or "batches", of wafers. These dose specifications on $f(x, y)$ provide limits on the confidence level of ε as it is derived from the measurement of the noisy beam current, $I(t)$, while it is profiled in x and y in measuring the beam shape, $g(x, y)$.

An example will help make this clear. For a given measure of $g(x, y)$, there is calculated a confidence level of g at each point (x, y). If the confidence level at any point, or the average of all points, is too high, then more measures are made, thus taking advantage of the uncorrelated nature of the time-noise of the ion beam. If the confidence level does not improve with added measures, then the beam is returned for lower and different noise. While the discussion heretofore focuses on the calculations using the best estimate of $g(x, y)$, in practice there exist at least two additional and parallel calculations at the upper and lower confidence level estimates. If any of these calculations yield an estimate of $f(x, y)$ outside the specified dose limits, then the wafers are not implanted.

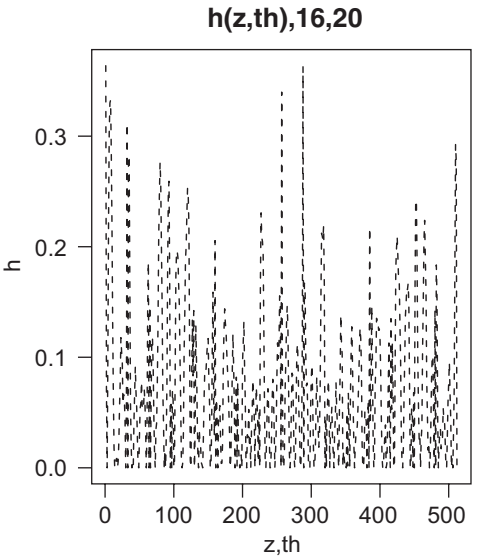

Fig. 10. A plot of the weights $h(z_i, \theta, k)$, $i = 1, \ldots, 32$; $k = 1, \ldots, 16$ in sequence for the case in Fig. 9.

We wish to minimize ε subject to the above constraints. In case g is good, then it follows that $\varepsilon = 0$ is the answer. In general this is a convex or infinite dimensional linear program whose solution will provide the needed $h_{f,g}(z, \theta)$. Even if g is not good, we can treat the problem as a linear program and obtain the solution for the minimum ε. Note that from the point of view to maximize the territory on the wafer for chip production it is desirable to use this so-called l_∞ approximation rather than l_2 or l_1. That is we want to match $f(x, y)$ within ε by choosing our $h_{f,g}$, uniformly, all across the wafer, otherwise we are giving up wafer territory for chip production. Linear programming provides a proof of optimality for the best available ε once the number, n, of angles has been set, and once the spacing or number m of lines has been set at which to match $f(x_i, y_j)$ within ε at an $m \times m$ array of points (x_i, y_j) on the wafer. Although minimizing other l_p norms on the error can be done for any value of $p < \infty$, the choice of l_∞ is simpler from a linear programming point of view, and also is better because of the "wafer territory" optimization point of view.

In the next section we describe a set of numerical experiments which shows the performance of the linear programming algorithm is flexible and good.

8. Wafer corrections via linear programming

We model, rather artificially, $f(x, y)$, the known and given distortion to the wafer, which is to be corrected by $h_{f,g}(z, \theta)$, for a given shape function, g. We chose the wafer to be the unit disc, $x^2 + y^2 < 1$, we choose $g(z) = e^{-z^2/2}$, and we choose $f(x, y)$ to be a variant of the phantom used in Shepp and Logan (1974) and Shepp and Kruskal (1978), except we make the gray levels of the phantom vary only between 1 and 1.1. See Fig. 1 (128×128 resolution) and Fig. 2 (16×16 resolution). Related results are given in Figs. 3–10. We are restricted to low resolution phantoms because our program (see the Appendix) is limited by the size of the matrix that the called linear programming package we are presently using, ddlprs (part of the package provided by the International Mathematical Statistical Language, IMSL), can handle. This limitation can be gotten around with more effort and a better linear programming package (see Fig. 11).

9. Conclusion

The experiments indicate that for the Gaussian shape g, a rather arbitrary f can be represented as a sum of ridge functions with shape g. Of course we already proved this in theory. The situation here is rather different than in tomography where Radon's theorem already provided a complete solution in theory, and this was borne out in practice

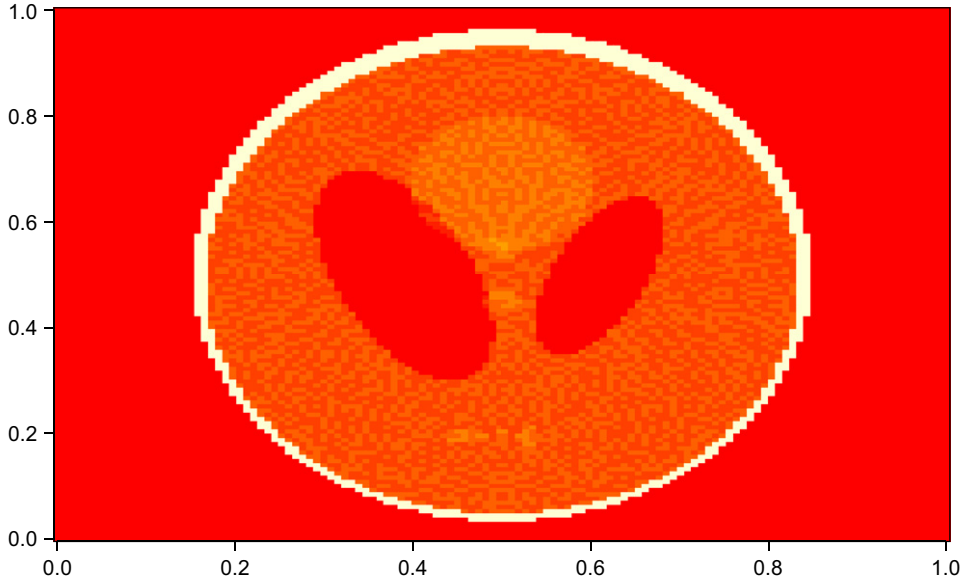

Fig. 11. A 128×128 plot of the sum of 128 ridge functions with shape $g(z) = e^{-0.5z^2}$, optimally chosen to reconstruct $f(x, y)$ to within the smallest tolerance possible, which turns out to be $\varepsilon = 0.00420$. This is quite comparable to the original 128×128 phantom in Fig. 1. We are grateful to David Applegate who ran the Cplex program which terminated in 5819.30 s using interior point method. The simplex program of Cplex did not terminate successfully.

as well. Here theory only allowed us to handle the Gaussian g. Further experiments should allow us to study the problem for more general g's.

The method we introduce here and the simulations parallel the method used in Shepp and Kruskal (1978), Shepp and Logan (1974) and Vardi et al. (1985), which gave much insight into the engineering problems of setting tolerances on various parameters; see Shepp and Stein (1976). In ion implantation, the method can be used to determine the necessary precision on knowing f, g, and on the precision of knowing the steps in z and in θ. The method should give insight into each step of the chip manufacturing process.

Appendix

The following is the listing of a Fortran program which calls the LP package `ddlprs(ncon,nvar,a,ncon,bl,bu,c,irtype,xlb,xub,obj,xsol,dsol)`. In the program, there is a matrix, a, dimensioned *ncon* by *nvar*, where *ncon* is the number of constraints and *nvar* is the number of variables. In general one seeks variables $x = x_j$, $j = 1, \ldots, nvar$, for which

$$\sum a_{i,j} x_j \leqslant b_i, \quad i = 1, \ldots, ncon.$$

Here the vector $b = bu = bl$ is specified, and one wants to minimize the inner product, $x \cdot c$ with c a given vector of length *ncon*. In our application we take $ncon = 2 \times m3$, where $m3$ is the number of points (x, y) at which we want to match a given "phantom" $f(x, y)$ and the factor of 2 is because we want to bound the difference between $f(x, y)$ and the sum of the ridge functions with shape g between $\pm\varepsilon$. We specify $f(x, y)$ in the program below. The unknowns, or variables, x_j, are the values of $h_{f,g}(z_i, \theta_k)$ for a discrete set $z = z_i$, $i = 1$, mv, and a discrete set of angles θ_k, $k = 1, \ldots, n$, where n is the number of angles. The angles need only go from 0 to π if $g = g(z)$ is even, which it is in these experiments, where $g(z) = e^{-0.5z^2\sigma^2}$, though g need not be even in general, and then θ_k would range from 0 to 2π.

The number of variables is $mv \times n + 2$, where the extra two variables are ε and Q. ε is the overall best l_∞ approximation to $f(x, y)$ by the sum of ridge functions with the shape g. Q is used to keep $h_{f,g}(z_i, \theta_k)$ non-negative, if needed. The vector c is zero except at the special unknowns ε and Q.

The array `irtype` is used in this package to specify the type of inequalities used. Other linear programming packages could replace this one and are widely available.

```
      program varian
      integer mc,mv,n,ncon,nvar,m3
      parameter(mc = 32, n = 10, mv = 44, nvar = mv*n+2)
      parameter(ncon = 1624)
      double precision a(ncon,nvar),bl(ncon),bu(ncon),c(nvar),xlb(nvar)
      double precision xub(nvar), xic(mc), xiv(mv), f(mc,mc), f1(mc,mc)
      double precision xsol(nvar),dsol(ncon), obj, sig,Tw,Tz,g
      double precision cs(n),sn(n),pi,sum,scr1,scr2,w,z,tot
      double precision x0(10),y0(10),th0(10),a0(10),b0(10),g0(10)
      integer irtype(ncon), k,i,i1,i2,lw,lz,iv,lump
      sig = 1.
      pi = 3.14159265
      do 71 i = 1,ncon
      irtype(i) = 1
71 continue
x0(1) = 0., y0(1) = 0., a0(1) = .69, b0(1) = .92, th0(1) = 0., g0(1) = 0.1
x0(2) = 0., y0(2) = −0.0184, a0(2) = .6624, b0(2) = .874, th0(2) = 0., g0(2) = −0.08
x0(3) = 0.22, y0(3) = 0., a0(3) = .11, b0(3) = .31, th0(3) = −.1 ∗ pi, g0(3) = −0.02
x0(4) = −0.22, y0(4) = 0.a0(4) = .16, b0(4) = .41, th0(4) = .1 ∗ pi, g0(4) = −0.02
x0(5) = 0., y0(5) = 0.35, a0(5) = .21, b0(5) = .25, th0(5) = 0., g0(5) = 0.01
x0(6) = 0., y0(6) = 0.1, a0(6) = .046, b0(6) = .046, th0(6) = 0., g0(6) = 0.01
x0(7) = 0., y0(7) = −0.1, a0(7) = .046, b0(7) = .046, th0(7) = 0., g0(7) = 0.01
x0(8) = −.08, y0(8) = −.605, a0(8) = .046, b0(8) = .023, th0(8) = 0., g0(8) = 0.01
x0(9) = 0., y0(9) = −0.605, a0(9) = .023, b0(9) = .023, th0(9) = 0., g0(9) = 0.01
x0(10) = 0.06, y0(10) = −.605, a0(10) = .023, b0(10) = .046, th0(10) = 0., g0(10) = 0.01
      do 1 k = 1,n
      cs(k) = cos(pi*(k−1.)/n)
      sn(k) = sin(pi*(k−1.)/n)
1 continue
      do 2 i = 1,mc
      xic(i) = −1. + (2.*i−1.)/mc
2 continue
      do 20 i = 1,mv
      xiv(i) = −1. + (2.*i−1.)/mv
20 continue
      ic = 0
      do 3 i1 = 1,mc
      do 4 i2 = 1,mc
      sum = 1.
      do 5 lump = 1,10
      scr1 = (xic(i1)−x0(lump))*cos(th0(lump))
      scr1 = scr1 + (xic(i2)−y0(lump))*sin(th0(lump))
      scr2 = −(xic(i1)−x0(lump))*sin(th0(lump) )
      scr2 = scr2 + (xic(i2)−y0(lump))*cos(th0(lump))
      tot = scr1*scr1/(a0(lump)*a0(lump) )
      tot = tot + scr2*scr2/(b0(lump)*b0(lump) )
      if(tot .gt. 1) go to 5
      sum = sum + g0(lump)
5 continue
      f(i1,i2) = sum
```

```
      write(6,*) f(i1,i2)
      if(xic(i1)*xic(i1) + xic(i2)*xic(i2) .ge. 1) go to 4
      ic = ic+1
      bl(ic) = sum
      bu(ic) = sum
4 continue
3 continue
      m3 = ic
      write(6,*) m3, ncon, nvar
      do 11 i = 1,m3
      bl(m3+i) = −bl(i)
      bu(m3+i) = −bu(i)
11 continue
      do 12 i1 = 1,ncon
      do 13 i2 = 1,nvar
      a(i1,i2) = 0.0d0
13 continue
12 continue
      ic = 0
      do 21 i1 = 1,mc
      do 22 i2 = 1,mc
      if(xic(i1)*xic(i1) + xic(i2)*xic(i2) .ge. 1) go to 22
      ic = ic+1
      a(ic,nvar−1) = −1
      a(ic+m3,nvar−1) = 1
      a(ic,nvar) = −1
      a(ic+m3,nvar) = −1
      do 24 k = 1,n
      w = −xic(i1)*sn(k) + xic(i2)*cs(k)
      ww = .5*( (w+1)*mv + 1)
      z = xic(i1)*cs(k) + xic(i2)*sn(k)
      zz = .5*( (z+1)*mv + 1)
      lw = ww
      lz = zz
      Tw = ww −lw
      Tz = zz −lz
      g = exp(−.5*z*z*sig)
      if( (lw .gt. 0) .and. (lw .lt. mv) ) then
      a(ic,lw+(k−1)*mv) = (1−Tw)*g
      a(m3+ic,lw+(k−1)*mv) = −(1−Tw)*g
      a(ic,lw+1+(k−1)*mv) = Tw*g
      a(m3+ic,lw+1+(k−1)*mv) = −Tw*g
      end if
      if(lw .eq. 0) then
      a(ic,lw+1+(k−1)*mv) = Tw*g
      a(m3+ic,lw+1+(k−1)*mv) = −Tw*g
      end if
      if(lw .eq. mv) then
      a(ic,lw+(k−1)*mv) = (1−Tw)*g
      a(m3+ic,lw+(k−1)*mv) = −(1−Tw)*g
      end if
24 continue
```

```
22 continue
21 continue
        do 30 i = 1,nvar
        xlb(i) = 0.0d0
        xub(i) = 100.d0
        c(i) = 0.
30 continue
        c(nvar−1) = 1.
        c(nvar) = 1.
        do 40 i1 = 1,ncon
        do 41 i2 = 1,nvar
41 continue
40 continue
        do 42 i = 1,nvar
42 continue
        call ddlprs(ncon,nvar,a,ncon,bl,bu,c,irtype,xlb,xub,obj,xsol,dsol)
        do 54 i = 1,ncon
54 continue
        do 55 i = 1,nvar
        write(6,*) i,xsol(i)
55 continue
        write(6,*) obj, xsol(nvar), xsol(nvar−1)
        do 61 i1 = 1,mc
        do 62 i2 = 1,mc
        if(xic(i1)*xic(i1) + xic(i2)*xic(i2) .ge. 1) then
           write(6,*) 1.
        go to 62
        end if
        sum = 0
        do 64 k = 1,n
        ww = .5*(1+ mv*(1−xic(i1)*sn(k) + xic(i2)*cs(k) ) )
        lw = ww
        Tw = ww −lw
        z = xic(i1)*cs(k)+xic(i2)*sn(k)
        g = exp(−.5*z*z*sig)
        if(lw .gt. 0. .and. lw .lt. mv) then
        sum = sum + xsol(lw+(k−1)*mv)*(1−Tw)*g
        sum = sum + xsol(lw+1+(k−1)*mv)*Tw*g
        end if
        if(lw .eq. 0) then
        sum = sum + xsol(lw+1+(k−1)*mv)*Tw*g
        end if
        if(lw .eq. mv) then
        sum = sum + xsol(lw+(k−1)*mv)*(1−Tw)*g
        end if
64 continue
        f1(i1,i2) = sum
        write(6,*) f1(i1,i2),f(i1,i2),f1(i1,i2)−f(i1,i2)
62 continue
61 continue
        do 81 i1 = 1,mc
        do 82 i2 = 1,mc
```

```
        if(xic(i1)*xic(i1) + xic(i2)*xic(i2) .ge. 1) then
        write(6,*) 1.
        go to 82
        end if
        write(6,*) f1(i1,i2)
82 continue
81 continue
        end
```

References

Shepp, L.A., Kruskal, J.B., 1978. Computerized tomography: the new medical X-ray technology. Amer. Math. Monthly 85, 420–439.

Shepp, L.A., Logan, B.F., 1974. The Fourier reconstruction of a head section. IEEE Trans. Nuclear Sci. 21, 21–43.

Shepp, L.A., Stein, J., 1976. Simulated artifacts in computerized tomography. In: Ter-Pogossian, M.M. (Ed.), Reconstructive Tomography in Diagnostics Radiology and Nuclear Medicine.

Vardi, Y., Shepp, L., Kaufman, L., 1985. A statistical model for positron emission tomography. J. Amer. Statist. Assoc. 80, 8–37.