ELECTRONIC PUBLISHING, VOL. 2(1), 47–59 (APRIL 1989)

The importance of phase in the spectra of digital type

GUOZHEN DUAN* AND ROBERT A. MORRIS**

University of Massachusetts at Boston
Boston
MA 02125, USA

SUMMARY

The role of phase in the spectra of digital type is examined. Characters and text are found to have more information in the phase than in the magnitude, just as for natural images. For letterforms, this means that the position of features, not their size, has the greatest influence on their discrimination. An iterative reconstruction of characters from their phase and from a magnitude characteristic only of the font, not the individual characters, is examined.

KEY WORDS   Digital type   Signal reconstruction   Phase information

SPECTRAL INFORMATION IN DIGITAL TYPE

Digital type is a special case of a discrete image. As such, it is straightforward to use techniques of digital image processing to understand and to manipulate type. For example, one common usage is the production of grayscale fonts by the application of digital low-pass filters to binary fonts. In this paper we report some investigations of the spectral properties of type in the discrete frequency domain. These investigations suggest explanations for some already well-known (to typographers) properties of type. Through sampling theory, one can extend the results to analog type.

Introduction

We can consider type images as a (usually finite) sequence of intensity values $x(n_1, n_2)$ at each point (n_1, n_2) in the discrete plane. For black and white type, these values are 0 or 1, but for digital grayscale type they come from a fixed collection of numbers representing an intensity range between black and white.

The Fourier transform $X(\omega_1, \omega_2)$ of a sequence $x(n_1, n_2)$ is given by

$$X(\omega_1, \omega_2) = \sum_{(n_1, n_2)} x(n_1, n_2) \exp(-j(\omega_1 n_1 + \omega_2 n_2)) \tag{1}$$

Here $j = \sqrt{-1}$. The pair of real numbers (ω_1, ω_2) is a two-dimensional *spatial frequency*. Roughly, ω_1 and ω_2 measure the rate with respect to distance at which the image alternates between black and white in the x-direction and in the y-direction, respectively. Computationally, these are best approximated by the discrete Fourier

---

* Permanent address: Beijing Institute of Radio Measurement

**Supported in part by NSF Grants DCR-8503154 and IRI-8715960 at UMASS-Boston. Robert Morris is Professor of Mathematics and Computer Science at UMASS-Boston and also a member of the technical staff at Interleaf Inc., Cambridge, MA 02141, USA.

0894–3982/89/010047–13$06.50   *Received 20 March 1988*

Revised 30 January 1989

transform (DFT) which we discuss in detail later, but for now do not distinguish from the Fourier transform. Either of these transforms is called the *spectrum* of the image.

X is a complex valued function, so for each point (ω_1, ω_2) in the frequency domain we can write $X(\omega_1, \omega_2)$ as $|X(\omega_1, \omega_2)|e^{j\phi(\omega_1,\omega_2)}$ with $|X| = |X(\omega_1, \omega_2)|$ the *magnitude* and $\phi(X) = \phi(\omega_1, \omega_2)$ the *phase* at (ω_1, ω_2). (Later, we will describe how images can be reconstructed when one or the other of these functions is not completely known.)

When the sequence x consists of samples of a continuous image, that image can be perfectly reconstructed provided that it is *band limited.* This means that its spectrum has finite support, i.e., it is zero outside some fixed rectangle. Such a reconstruction also requires that the image be sampled at a high enough rate, i.e., that its resolution is high enough. However, an *image* which has finite support is never band limited, so the reconstruction necessarily becomes an approximation, suffering from *aliasing*, the misrepresentation of high frequency components by lower ones. The treatment of this is beyond the scope of this paper, but see [11,9,1] and especially [6] for some details.

There is some reason to believe that human vision is not subject to the kind of aliasing mentioned above [17]. In any case, we are dealing with signals which are already discrete—namely digital representations of type—and these can be completely reproduced from their discrete spectra.

In many classical signal processing applications, the spectrum of a signal is known incompletely. Often one, but not both of the phase or magnitude is accurately known, and it is desired to reproduce the original signal from this incomplete or distorted spectrum. Such reproduction from the deficient spectrum is referred to as “retrieval” of the remaining information. A large literature exists on both phase and magnitude retrieval [10,12,14,4,5,7]. In the following section we discuss, without formal argument, how these kinds of results give insight into type images. We give more mathematical detail in a subsequent section.

Experimental results

In this section we describe the experimental results we have obtained. However, first we want to put them in the context of the situation described above. The collection of phases in an image comprise largely position information. For example, shifting an image adds a linear term to the phases and does nothing to the amplitudes. On the other hand, amplitude is mostly a measure of the local contrast variation in an image. Thus, rapid changes from black to white to black are manifested as high intensities at high spatial frequencies in the amplitude spectrum. For example, the spectra in the figures show amplitude above and phase below. In figure 2 the first amplitude spectrum has more energy at great distance from the center (higher spatial frequencies) because the thinner strokes correspond to more rapid variation of the image intensity.

In the figures, we normalize the magnitude to DC, i.e., the $(0, 0)$ term in the discrete spectrum, and assign a gray level corresponding to the magnitude at each point in the discrete frequency plane. (It can easily be shown that DC always has the largest magnitude in a given spectrum.) Similarly, at each point we have assigned a gray level by quantizing the phases $-\pi \leq \omega < \pi$ and assigning a gray level to each interval. Higher spatial frequencies are further from the centers than are lower ones.

There are several things about letterforms which one might hope to quantify, or at least explain, in image processing terms. One of the outstanding features of type is that we can recognize and distinguish characters despite a very wide range of manipulations of the letterforms. These two tasks, recognition and discrimination, are visually somewhat different. They also correspond roughly to two issues for image retrieval problems, those of convergence and uniqueness of certain algorithms. (Note that we are not suggesting any model in which human character recognition proceeds by the sort of iterative algorithms described below.)

The central image processing result to which we will later appeal asserts that the information which contributes most to discrimination between images lies in their phases. In the figure 1, magnitude and phase spectra of the characters ‘b’ and ‘p’ are shown. The magnitude spectra are virtually indistinguishable from one another, but the phase spectra are quite distinct.

The distinction of phases—which is in fact typical—may account for some of the ease with which characters are distinguished, since it is known [16] that human vision is 5–10 times more sensitive to position than to magnitude differences. More precisely, humans can resolve distinct lines which are only as little as 1 minute of visual angle apart, but can detect *displacement* of about 12 seconds, or, with practice, 6 seconds. (This should not be surprising, since the image processing literature [8] suggests that natural images in the world have most information in their phase, not magnitude.) For the letterform designer, the importance of phase vs. magnitude is that the *position* of character features has more of an impact on character recognition and distinction than does the size of those features, something which is intuitively clear and also corresponds to conventional type design wisdom.

In figure 2, a sequence of characters derived from the Computer Modern Sans Serif ‘p’ is shown, together with their Fourier magnitude and phase spectra. The phase spectra of those characters are remarkably similar, but the magnitude spectra are not.

Among the technical requirements of the theorems which make these remarks precise is that the images should not have perfect horizontal and vertical symmetry around some point, and, in fact, very few characters have such symmetry. (Among roman letters, only the upper case ‘I’ will exhibit it. One might think that ‘o’ and ‘O’ also exhibit it, but careful examination of the ‘O’ in most typefaces will reveal that typeface designers themselves often eschew this bi-orthogonal symmetry and typically rotate the vertical symmetry axis slightly. Among Chinese characters, which we have examined briefly, few would be expected to have it. A real string of text as a whole will never be bi-orthogonally symmetric.)

The role of symmetry in distinguishability may play yet another role in the digital font domain. When grayscale “anti-aliased” fonts are produced by digitally filtering high resolution binary fonts, the filters used are always symmetric. Presumably this is to insure that left and right edges and top and bottom edges of characters are not smoothed differently.[1] Several filters are in common use for grayscale production [15], but no author has claimed a clear superiority of one over another (although varying filter *parameters*, such as support width, can have a dramatic effect). We propose that this is due to the following reason: symmetric filters necessarily do not change the phases in an image; so they all equally preserve the information which we claim really carries the letter shape.

[1] However, vision researcher Gordon Legge suggested to one of us that this is not an obvious visual requirement, since we mostly read in one direction, so that, say, readers of English, may need different information at the left than at the right of characters.

string: p

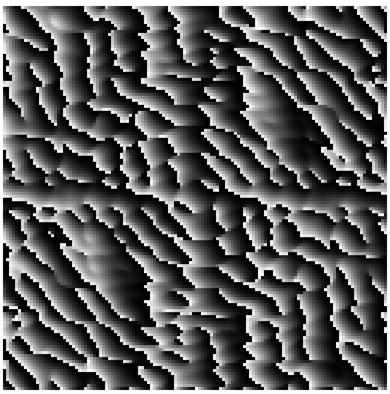
font: cmss10.1500pxl

sample: 128 x 128

string: b

font: cmss10.1500pxl

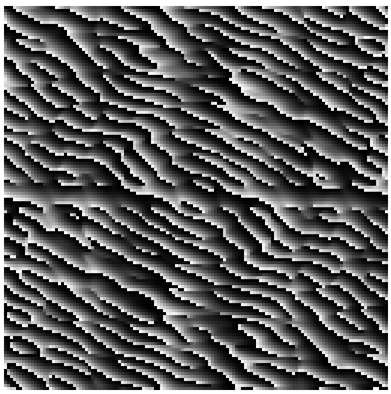
sample: 128 x 128



string: p

font: cmss10.1500pxl

sample: 128 x 128

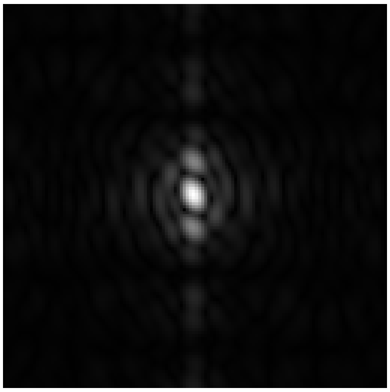


string: b

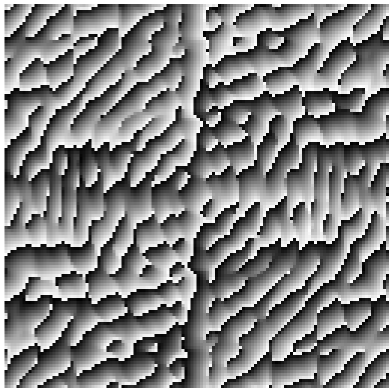
font: cmss10.1500pxl

sample: 128 x 128

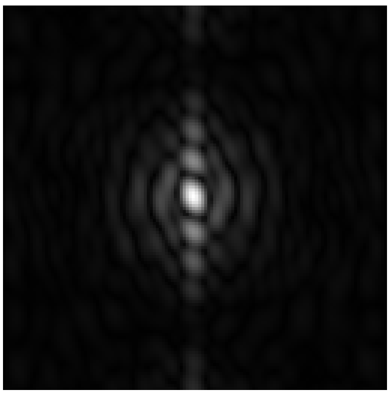
Figure 1. Comparison of amplitude and phase spectra



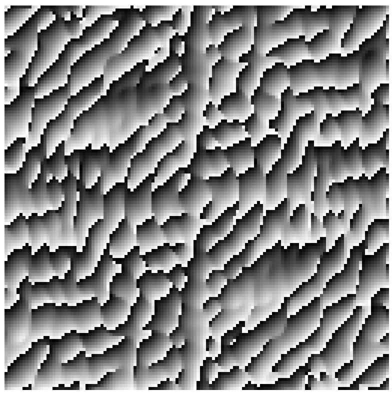
string: p
font: cmssdk10.1500pxl
sample: 128 x 128



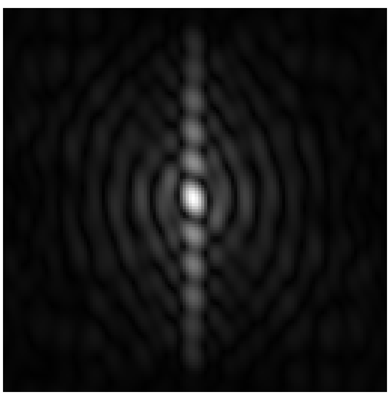
string: p
font: cmssdk10.1500pxl
sample: 128 x 128



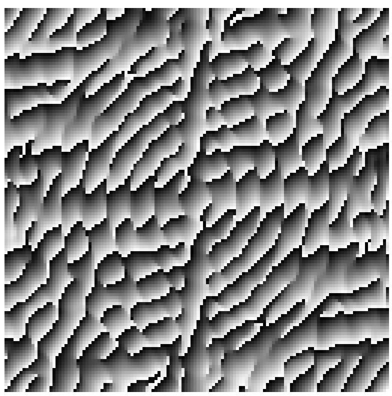
string: p
font: cmss10.1500pxl
sample: 128 x 128



string: p
font: cmss10.1500pxl
sample: 128 x 128



string: p
font: cmsslt10.1500pxl
sample: 128 x 128



string: p
font: cmsslt10.1500pxl
sample: 128 x 128

Figure 2. Effect of stroke width variation

The reconstruction algorithm

Distinguishing images from one another corresponds to the "uniqueness" part of the image reconstruction problem described below. The reconstruction algorithms we use follow the models described in [13,5,14]. They use fixed point theorems to guarantee their convergence, but more conditions are needed to insure that the limit images are the desired ones. The conditions are not difficult to impose; they represent destruction of conditions of extreme symmetry by adding at most one pixel to the image. Remembering that bi-orthogonal symmetry is not typically found in type, the conditions seem to obtain without further manipulation. In this section, we continue with informal discourse on the conditions which make these algorithms converge. Mathematical details are in the following section.

Reconstruction proceeds by imposing constraints at each point in either the spectral or signal domain, and throwing out that portion of the data which does not meet the constraints. These constraints are chosen from conditions known to be met by the desired image. For us, the most important reconstruction is that from "phase only", which we describe next. We begin with the Fourier transform, X, of a character or text string x. In this spectrum, we initially replace the magnitude with a "virtual magnitude" which is independent of the character. There are several natural choices for this, but the uniqueness results described below imply that the choice does not matter.

Several authors experiment with various choices in the case of natural images. Hayes *et al.* [5] use constant magnitude. Garcia and Calero [3] describe reconstructions with the magnitude of low pass filters for the initial estimate. We have had success starting with the magnitude of a four-point black square situated at the origin and also with constant magnitude. With either of these as the initial spectral magnitude estimate, the algorithms below converge (after thresholding) in as few as 10 iterations, as shown in the lower reconstruction in figure 3. The first two images in this figure are the first iteration with each of these virtual magnitudes. They suggest that algorithms based on combining two amplitude estimates might be successful; we can see that reconstruction using the constant magnitude estimate quickly gives the edges and the one using the four point estimate emphasizes the interior of the character.

The algorithm begins by making a $2N \times 2N$ DFT and replacing the magnitude by the virtual magnitude described above. We back-transform this initial "estimate" of the spectrum into the image domain and then impose additional constraints: the resulting image is forced to zero outside its original $N \times N$ region of support or at any point at which it is negative or non-real. Next, we again take a $2N \times 2N$ forward transform. (The doubling is a technical requirement arising from the theorem's guaranteeing uniqueness from phase). Then we replace the phase in the resulting spectrum by that of the original character. This guarantees that at every step of the algorithm, the current image and the original image have the same phase. Finally, this spectrum is back-transformed into a $2N \times 2N$ image which is subject to the image domain constraints described above. We continue in this fashion, enforcing the constraint of original phase in the spectral domain, and real, non-negative and support constraints in the signal domain. These constraints suffice to guarantee that this process converges to *something*. Since the phase always coincides with that of our original image, the convergence is to the original character, by the convergence and uniqueness results detailed below.

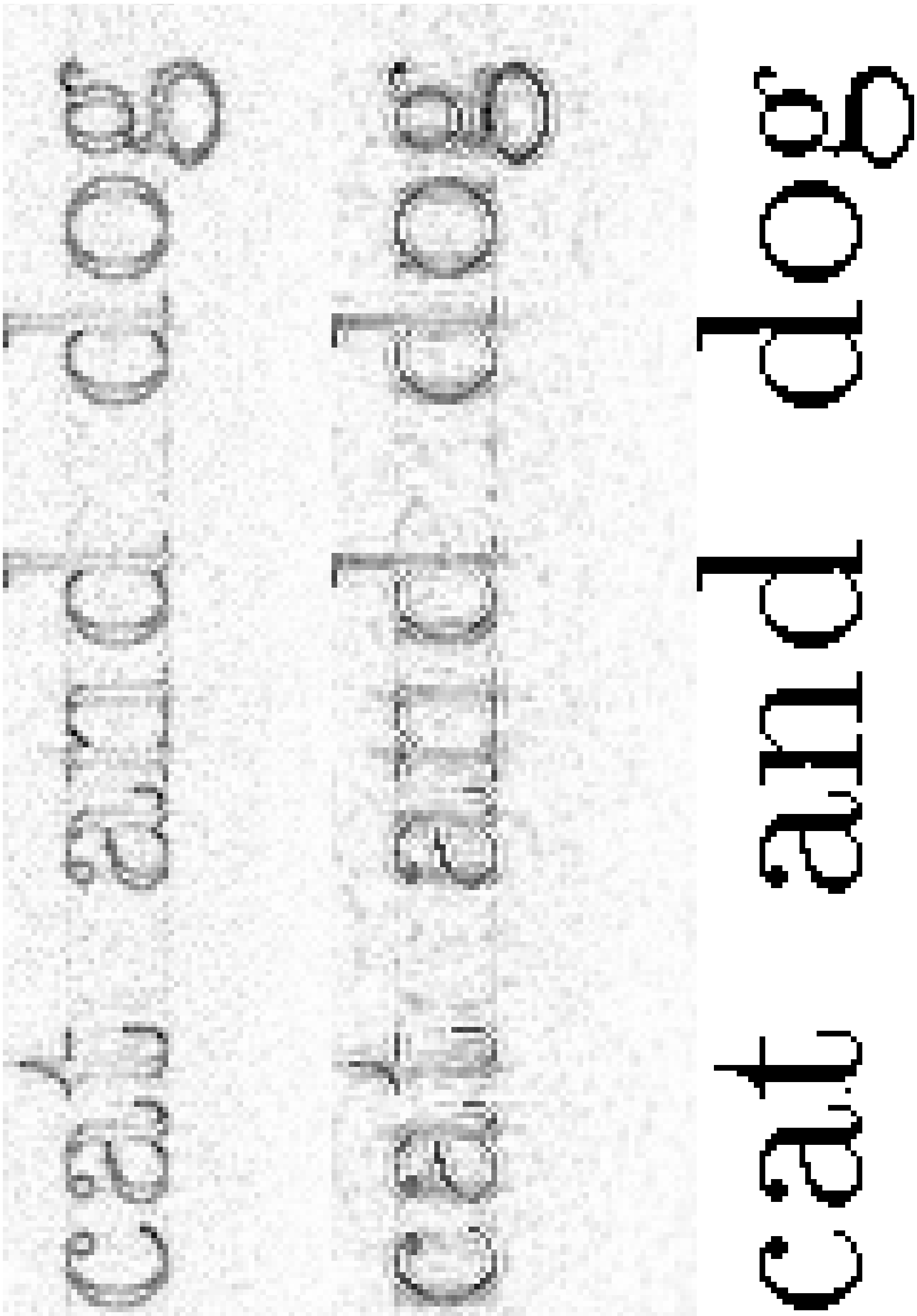


Figure 3. Reconstructions from phase only

However, our characters are black and white and, as we will show, we can recover the original character after finitely many steps if we apply a threshold operator. This means we force all pixels that are darker than the threshold to black and all others to white. It is not difficult to see that the choice of threshold level affects only the number of iterations required to recover the character completely.

We do not have a good formulation for the number of iterations needed to recover the character, but our experience with both Roman (figure 3) and Chinese characters (figure 4) suggests that it is as few as 10–20. This is in marked contrast to the reports in the reconstruction literature of numbers of iterations ranging from dozens to several hundred. We note, however, that the restriction to black and white images makes it easier to skip the “last” of the iterations.

When we attempt to recover the character from magnitude only, adding enough information to the image to guarantee uniqueness [2], we do not get rapid convergence. This again confirms the view that there is more information in the phase than in the magnitude.

string: 锴

font: song10.2250pxl

sample: 64 x 64

Figure 4. Reconstruction from phase only

CONVERGENCE AND UNIQUENESS

We now give the arguments as to why the algorithm described above converges to the original images after finitely many iterations.

Discrete Fourier transforms and Z-transforms

Following Hayes [4] we adopt standard notation for multi-dimensional sequences over the real and complex numbers. In what follows, we assume dimension $m = 2$ if not otherwise stated, but everything holds for greater dimensions. We will observe below that there is a distinction to be made between one-dimensional and higher dimensional signals. A multi-index $\boldsymbol{n} = (n_1, n_2, \ldots, n_m)$ will be denoted with bold face. The obvious notation holds for variables $\boldsymbol{z} = (z_1, z_2, \ldots, z_m)$ and their powers, $\boldsymbol{z}^{\boldsymbol{n}} = z_1^{n_1} z_2^{n_1} \ldots z_m^{n_m}$.

A discrete image is then a real valued sequence x with support a region R in the integer plane. That is, $x(i, j) = 0$ if (i, j) is not in R. Generally it will be convenient to let R be a rectangle in the positive quadrant, but the results are easily extended to arbitrary finite support. We will say that the support of x is $\boldsymbol{N} = (N_1, N_2)$ meaning N_1 and N_2 are the upper bounds of the rectangle. If $N_1 = N_2 = N$ we write $R(N)$ for this rectangle. Henceforth, by "image" we will always mean a discrete image with finite support, unless otherwise described. It is convenient to write $x(\boldsymbol{n})$ when $\boldsymbol{n} = (n_1, n_2, \ldots, n_m)$.

In this notation, the Fourier transform $X(\boldsymbol{\omega})$ of a sequence $x(\boldsymbol{n})$ is given by

$$X(\boldsymbol{\omega}) = \sum_{\boldsymbol{n}} x(\boldsymbol{n}) e^{-j\boldsymbol{n}\cdot\boldsymbol{\omega}} \tag{2}$$

The z-transform $X(\boldsymbol{z})$ is defined by

$$X(\boldsymbol{z}) = \sum_{\boldsymbol{n}} x(\boldsymbol{n}) \boldsymbol{z}^{-\boldsymbol{n}} \tag{3}$$

Note that the Fourier transform is just the z-transform evaluated on the unit polydisk $\boldsymbol{z} = \exp(j\boldsymbol{\omega})$.

For images with finite support $\boldsymbol{N}$ in the positive quadrant it is easy to see that the z-transform is a polynomial in $\boldsymbol{z}^{-1}$ of degree $\boldsymbol{N}$ (i.e., N_i in z_i^{-1}).

At each point (ω_1, ω_2) in the frequency domain, as before, we can write $X(\omega_1, \omega_2)$ as $|X(\omega_1, \omega_2)| e^{j\phi(\omega_1, \omega_2)}$. We define the $\boldsymbol{M}$-point discrete Fourier transform $X_{\boldsymbol{M}}$ by

$$X_{\boldsymbol{M}}(k_1, k_2) = \sum_{\boldsymbol{n} \epsilon \boldsymbol{M}} x(\boldsymbol{n}) e^{-2\pi j (n_1 k_1 / M_1 + n_2 k_2 / M_2)} \tag{4}$$

If $\boldsymbol{M}$ contains the support of x, then this expression coincides with evaluation of the continuous Fourier transform at roots of unity. More precisely,

$$X_{\boldsymbol{M}}(k_1, k_2) = X(\omega_1, \omega_2)|_{\omega_1 = 2\pi k_1 / M_1,\ \omega_2 = 2\pi k_2 / M_2} \tag{5}$$

Indeed, where the context is clear (or irrelevant), we will not distinguish between these notations, and will write $X_{\boldsymbol{M}}(k_1, k_2)$ or sometimes simply $X(\omega_1, \omega_2)$ for either of them. Similarly, we will write $\phi(x) = \phi_{\boldsymbol{M}}(x)$ for the phase, although clearly this depends on $\boldsymbol{M}$, which, in turn, determines the roots of unity in the above. We will disambiguate this by writing $\phi_x(\boldsymbol{k})_{\boldsymbol{M}}$ for the phase thus represented.

For sequences whose support is contained in a rectangle $R(\boldsymbol{M})$ (in the upper quadrant, but this is without loss of generality), we can recover the sequence from its *inverse DFT* given by

$$x(n_1, n_2) = \frac{1}{M_1 M_2} \sum_{n_1=0}^{M_1} \sum_{n_2=0}^{M_2} X(k_1, k_2) \exp(2\pi j (n_1 k_1 / M_1 + n_2 k_2 / M_2)) \tag{6}$$

See [1, p.64]. Just as band-limited signals can be recovered from samples of the signals, finite-support signals can be recovered from samples of their Fourier transforms. For this reason, the DFT is an appropriate object of study in the case of digital type images.

If $\boldsymbol{z} = \boldsymbol{x} * \boldsymbol{y}$ is the discrete convolution of two sequences of finite support, and if $\boldsymbol{M}$ contains the support of both, then

$$Z_{\boldsymbol{M}}(k_1, k_2) = X_{\boldsymbol{M}}(k_1, k_2) Y_{\boldsymbol{M}}(k_1, k_2) \tag{7}$$

More generally,

$$(X * Y)(z) = X(z)Y(z) \tag{8}$$

and hence

$$Z(\omega_1, \omega_2) = X(\omega_1, \omega_2) Y(\omega_1, \omega_2) \tag{9}$$

It is then easy to see that

$$|Z(\omega_1, \omega_2)| = |X(\omega_1, \omega_2)||Y(\omega_1, \omega_2)| \tag{10}$$

and

$$\phi(z) = \phi(x) + \phi(y) \tag{11}$$

Determination by phase

The factorization of the z-transform as a polynomial in $\boldsymbol{z}^{-1}$ is a matter of interest to us below. In one dimension, polynomials over the complex numbers are irreducible only if they are of degree 1, that is every polynomial $p(u)$ can be factored into a product of the form $\alpha(u - a_1)(u - a_2) \cdots (u - a_m)$ given by the roots a_i of the polynomial. In higher dimensions there are irreducible polynomials of every degree. We are interested in certain kinds of factors of the z-transform, whether irreducible or not.

The z-transform $X(\boldsymbol{z})$ is *symmetric* if there is a vector $\boldsymbol{k}$ of positive integers with

$$X(\boldsymbol{z}) = \pm \boldsymbol{z}^{-\boldsymbol{k}} X(\boldsymbol{z}^{-1}) \tag{12}$$

For a finite sequence $x(i, j)$, this is equivalent to a symmetry condition on the sequence of the form

$$x(i, j) = x(k_1 - i, k_2 - j) \tag{13}$$

It is not difficult to show from equation (11) above that if x has a symmetric z-transform, then its phase $\phi(x)$ is linear. Multiplying a z-transform $Y(\boldsymbol{z})$ by a symmetric factor thus adds a term of the form $(k_1\omega_1, k_2\omega_2)$ to $\phi(y)$; doing so simply corresponds to a shift in the image by (k_1, k_2).

For notational convenience we consider square regions of support. If N is a positive integer, $\boldsymbol{N} = R(N)$ and we will write ϕ_N for $\phi_{\boldsymbol{N}}$. The following theorem of Hayes shows that a sequence without a linear term in its phase is determined, up to scaling, by that phase.

Theorem ([4, Theorem 5.]), Let x and y be sequences with support $R(N)$ and suppose $M \geq 2N - 1$. If $X(z)$ has no symmetric factors and $\phi_x(\boldsymbol{k})_M = \phi_y(\boldsymbol{k})_M$ then $x = \beta y$ for some $\beta > 0$.

Both the asymmetry and the support conditions are necessary for this result. The support conditions can be relaxed if one considers continuous transforms and the phases agree at all points, not just on a discrete lattice (see [4, Theorem 3.]). One reviewer, having replicated our use of the algorithm, suggested that convergence may be independent of the support condition. In general, the Hayes result is needed not for convergence but for uniqueness—that is, to guarantee that the original image returns. It is an interesting question whether some other uninvestigated properties of type images guarantee this. Since symmetry is so rare in type images, we have not investigated its rôle in detail.

Image reconstruction from phase information

The set $I = I(N)$ of images supported on $R(N)$ forms a complete metric space, with the standard Euclidean distance d between two sequences given by:

$$d(x,y) = \Big(\sum_{i,j=0}^{N} |x(i,j) - y(i,j)|^2 \Big)^{1/2} \tag{14}$$

This is equivalent to describing a *norm* $|| \ ||$ on I given by $||x|| = d(x,0)$. Conversely, given a norm $|| \ ||, d(x,y) = ||x - y||$ defines a metric. Either of these can be used to describe convergence of a sequence $\{x_p\}$ of images to an image x in the obvious way.

Suppose a sequence $\{x_p\}$ can be formed by iterating a mapping C on the space of images, and that C is a *contraction mapping*, i.e., there is some α, $0 < \alpha < 1$, with $d(C(x), C(y)) < \alpha d(x,y)$ whenever $x \neq y$. Then $\{x_p\}$ converges, and its limit x is the unique fixed point of C, i.e., $C(x) = x$. This so-called *contraction mapping theorem* guarantees not only convergence of the iterative algorithm given by C, but determines its solution. One way of constructing such a mapping is as a series of enforced constraints on the images. That is, at each iteration one replaces the result with a related image satisfying some condition known about the original image. In practice, the constraints will not insure strict contraction, that is, the strict inequality above will be replaced with $\leq$. In this case, convergence to a fixed point is still guaranteed, but the operator C may have several fixed points. In such a *non-expansive* case, we will have to use Hayes's theorem above to guarantee uniqueness. The constraints described earlier (real, positive, fixed support) do yield a non-expansive operator [4].

It will be convenient below to consider the *sup* norm $||x|| = \sup_{i,j} |x(i,j)|$. It is straightforward to show that this norm and its associated metric are equivalent to the Euclidean ones, in the sense that convergence is the same in both metrics.

Before completing this section, we need an observation about applying binary thresholding to images (i.e., forcing to 0 everything below the threshold value and to 1 everything at or above it). Thresholding is not a nonexpansive operator because any non-expansive operator is continuous. Thresholding is far from continuous on the space of images, since two images can be arbitrarily close together in the metric above, but quite far apart after thresholding. To see this, simply take two constant images, one slightly above and one slightly below the threshold; the original images can be arbitrarily close, but the thresholded images will be quite distant.

Nevertheless, let T be the threshold operator and suppose $\{x_p\}$ converges to an image x which has no values equal to the threshold. Then, *after a finite number* of steps, $T(x_p) = T(x)$. This guarantees that iterative algorithms based on nonexpansive mappings can be terminated after a finite number of steps if we are only interested in thresholding. This will always be the case if our images are quantized, as in our case, where the final image is black and white.

The above remarks have the following formalization:

Proposition. Let $t > 0$ and $T(x)$ be thresholding at t, i.e.,

$$T(x)(i,j) = \begin{cases} 1 & \text{if } x(i,j) \geq t \\ 0 & \text{otherwise} \end{cases}$$

Let $\epsilon > 0$ and suppose that $|x(i,j) - t| > \epsilon$ for all i, j. If y is an image with $||x - y|| < \epsilon$ then $T(x) = T(y)$.

Proof. As remarked above, we can use the sup norm. In particular, for each (i,j) the value $y(i,j)$ must lie in the interval of size ϵ around $x(i,j)$. By hypothesis, this interval must be properly contained in the interval of radius t. It follows that if $x(i,j) < t$, then $y(i,j) < t$ and if $x(i,j) > t$, then $y(i,j) > t$. This exactly means $T(x) = T(y)$.

Corollary. Let $\{x_p\}$ converge to x and suppose $x(i,j) \neq t > 0$ for all i, j. Then there is a p_0 such that whenever $p > p_0$, we have $T(x_p) = T(x)$.

Proof. Choose $\epsilon > \sup |x(i,j) - t| = ||x - t||$. By convergence, we can find a p_0 such that $||x_p - x|| < \epsilon$ for all $p > p_0$, and the Proposition applies.

GRAYSCALE CHARACTERS

In this section we briefly describe the application of our results to grayscale fonts, and in particular to explain a phenomenon which is often observed by workers in the field, but for which no explanation has yet been offered.

Grayscale fonts represent an attempt to substitute intensity resolution for spatial resolution. To the visual system, these seem to be visually equivalent. Quantization error, which manifests itself as jagged staircase effects in curves and lines, is compensated for by applying low pass filters to high resolution binary characters to produce low resolution grayscale characters [15]. In effect, this blurring of the edges is perceived as smoothing the staircase. It represents an extremely cost effective way to improve the apparent resolution of video displays: memory requirements to support an n-bit/pixel display of a given size are the same as for a binary display with $2^{n/2}$ times as much resolution in each direction. The sweep frequencies of the latter, however, are also $2^{n/2}$ times higher, which generally contributes to higher manufacturing cost.

All the filters in common use have perfect symmetry in the sense of equations (12) and (13). Any perfectly symmetric filter has zero phase, so when applied will not change the phase. (Output phase is the sum of input and filter phase.) Because text is determined by phase, this kind of filtering has little effect.

Finally, note that phase determines grayscale characters just as it does binary ones, provided, at least, that they are produced from binary type by application of filters of finite support. For example, suppose that for two binary characters x and y the filtered characters are equal. That is, if L is the filter which produces the grayscale characters, suppose $L * x = L * y$. Since L, x and y all have finite support and their z-transforms are all polynomials in $\boldsymbol{z}^{-1}$, we have

$$\phi(L) + \phi(x) = \phi(L * x) = \phi(L * y) = \phi(L) + \phi(y) \tag{15}$$

The desired result follows from this applied to x and y after subtracting the filter phase $\phi(L)$ (which, as remarked above, is typically zero in any case).

REFERENCES

1. Dan E. Dudgeon and Russell M. Mersereau, *Multi-Dimensional Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1984.
2. M.A. Fiddy, B.J. Brames, J.C. Dainty, and Mark A. Richards, 'Enforcing irreducibility for phase retrieval in two dimensions', *Optics Letters*, **8**, 96–98 (1983).
3. Narciso Garcia and Alberto Calero, 'Faster phase only image reconstruction'. In *IEEE International Conference on ASSP*, pp. 12A.3.1–12A.3.4, 1984.
4. Monson H. Hayes, 'Reconstruction of a multidimensional sequence from the phase or magnitude of its Fourier transform', *IEEE Transactions on ASSP*, **39**(2), 140–154 (1982).
5. Monson H. Hayes, Jae S. Lim, and Alan V. Oppenheim, 'Signal reconstruction from phase or magnitude', *IEEE Transactions on ASSP*, **28**(6), 672–680 (1980).
6. J. Kajiya and M. Ullner, 'Filtering high quality text for display on raster scan devices', *Computer Graphics*, **15**(3), 7–15 (1981), Proc. SIGGRAPH 81.
7. Russell M. Merserou, Monson H. Hayes, and Ronald W. Schafer, 'A survey of methods for iterative signal reconstruction', In *IEEE International Conference on ASSP*, pp. 128–132, 1982.
8. Alan V. Oppenheim and Jae S. Lim, 'The importance of phase in signals', *Proceedings of the IEEE*, **69**(5), 529–541 (1981).
9. Alan V. Oppenheim and Ronald W. Schafer, *Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1975.
10. A.V. Oppenheim, J.S. Lim, and S.R. Curtis, 'Signal synthesis and reconstruction from partial Fourier-domain information', *Journal of the Optical Society of America*, **73**(11), 1413–1420 (1983).
11. William Pratt, *Digital Image Processing*. Wiley Interscience, New York, 1978.
12. Jorge L.C. Sanz and Thomas S. Huang, 'Unique reconstruction of a band-limited multi-dimensional signal from its phase or magnitude', *Journal of the Optical Society of America*, **73**(11), 1446–1450 (1983).
13. Ronald W. Schafer, Russell M. Mersereau, and Mark A. Richards, 'Constrained iterative restoration algorithms', *Proceedings of the IEEE*, **69**, 432–450 (1981).
14. Victor T. Tom, Thomas F. Quatieri, Monson H. Hayes, and James H. McClellan, 'Convergence of iterative nonexpansive signal reconstruction algorithms', *IEEE Transactions on ASSP*, **29**(5), 1052–1058 (1981).
15. J.E. Warnock, 'The display of characters using gray level sample arrays', *Computer Graphics*, **14**(3), 302–307 (1980), Proc. SIGGRAPH 81.
16. Gerald Westheimer, 'The spatial sense of the eye', *Investigative Ophthalmology and Visual Science*, **18**, 893–912 (1979).
17. John I. Yellott Jr., 'Spectral analysis of spatial sampling by photoreceptors: topological disorder prevents aliasing', *Vision Research*, **22**, 1205–1210 (1982).