ELECTRONOTES
1016 Hanshaw Rd
Ithaca, NY 14850
607-266-8492

APPLICATION NOTE NO. 369

February 2007

TWO DIMENSIONAL UNIFORM SAMPLING

1. INTRODUCTION:

Many engineers, when confronted with the notion of sampling a two-dimensional (2D) function, will quite correctly resort to their familiarity with one-dimensional (1D) sampling to see if it can be simply extended. In a very common case (rectangular sampling), this gets us a long way. Figure 1a shows the 1D spectrum (the FFT) of a sampled signal, which we have set to rectangular just by typing in ones and zeros. This is X(k) for k=0 to k=63. In order to get a real signal corresponding to X(k), we have one fewer one for X(k) on the high side than on the low side. That is, X(0) is not repeated, X(1)=X(63), and so on. The signal is low-pass. Corresponding to X(k) is a time domain signal x(n), the inverse FFT of

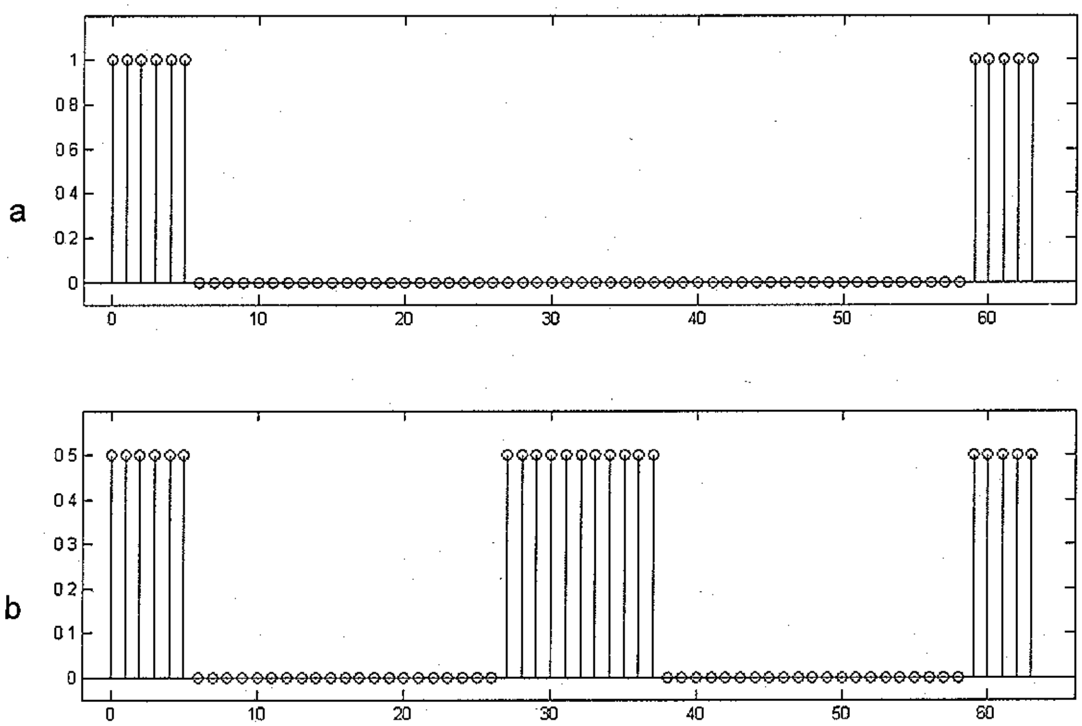


Fig. 1 Original spectrum (a) and spectrum of sampled signal (b)

X(k), which we know is a periodic sinc, but do not show. When we sample the length 64 x(n) by a factor of 2, we set every other sample to zero. If we now take the forward FFT of the sampled version of x(n), we get the result shown in Fig. 1b. That is, we get a sampling replica in the middle (and lose half the amplitude).

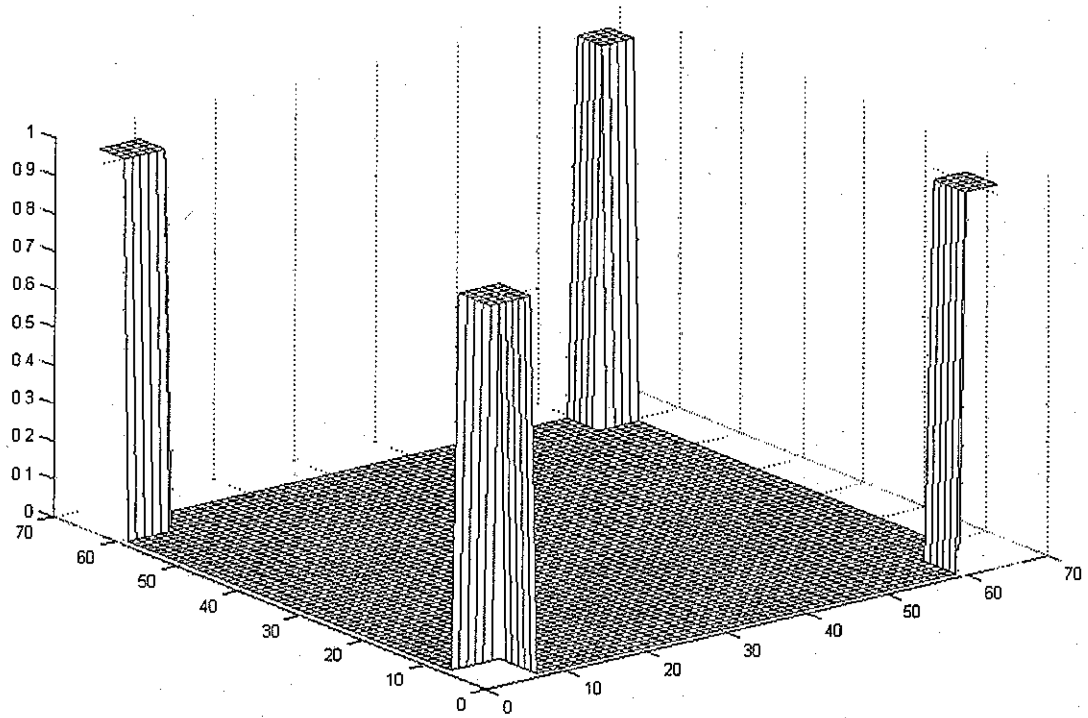


Fig. 2a An original 2D low-pass spectrum

We can repeat this experiment in 2D, using the 2D FFT's in Matlab. The code for figures 1 and 2 is given in the appendix. In the case of the 2D spectrum, we form our low-pass as a square in the lower left corner. This has natural replicas in the other three corners. (See the code for the easy way to do this). We can now take the inverse 2D FFT of this (a 2D periodic sync) which we need not look at. We want to sample this by 2, but it is hard to see how we would do this. We could keep every other line, or every other column. Later we will look at a lattice that does sample by 2, but for now, let's try sampling by 2 in both directions (sampling by 4 total). That is, every other line and every other row is zeroed. Fig. 2b shows the spectrum that results from this sampling. We see that we get now four replicas. That is, we get the original (four corners), two replicas each consisting of two pieces on the sides, and a full replica in the middle. Note that the heights are now 1/4 the original. The point is, hopefully, that this result is exactly what we likely expected based on our knowledge of 1D and our engineering intuition.

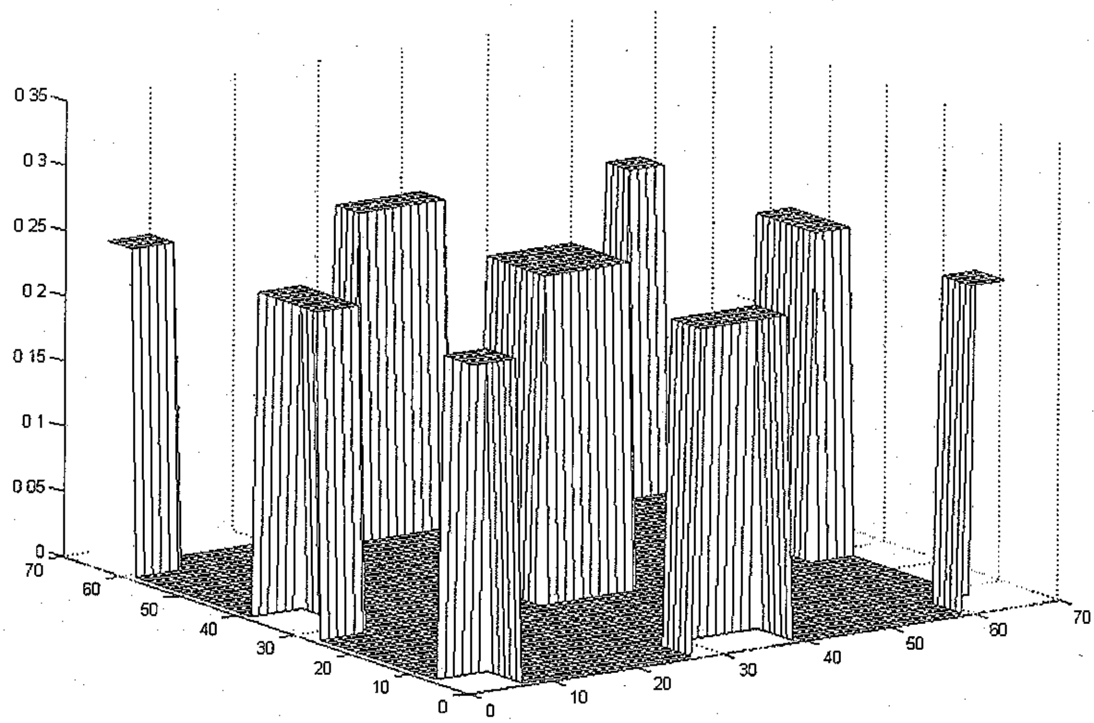


Fig. 2b The Spectrum of the sampled (by 2 in each direction) signal

This result is satisfying in that our "guess" as to what should happen seems to agree with out experiment. What is missing is a systematic procedure for finding how many replicas there are, and where they are.

2. SAMPLING LATTICES

In doing the experiments above, we have done our sampling by multiplying existing samples by a sampling function or "lattice" of 1's and 0's. That is, the original samples are assumed to exist on a grid consisting of all integer positions (in 1D or 2D), and when we sample, we keep some of these original samples, and set the rest to zero. This is sometimes properly thought of as "re-sampling." [We may also note at this point that we can also consider cases where the kept samples are rearranged so as to compress the total amount of data by removing the zeros – but not at the moment.]

For example, a 4x4 image "x" can be multiplied by a sampling function or lattice "s" to form "xs" as:

$$\begin{matrix} a & b & c & d \\ e & f & g & h \\ i & j & k & m \\ n & p & q & r \end{matrix} \quad X \quad \begin{matrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{matrix} \quad = \quad \begin{matrix} 0 & 0 & 0 & 0 \\ e & 0 & g & 0 \\ 0 & 0 & 0 & 0 \\ n & 0 & q & 0 \end{matrix} \qquad (1)$$

x   s (lattice)   xs

In a more general approach, we might choose s in an arbitrary way so that values need not be just 1's or 0's. However, we will look at the case of 1's and 0's first, and will look at "uniform" lattices. Note that the lattice can be represented in terms of its smallest possible unit or "sampling cell." For the example of equation (1), the sampling cell is not the 4x4 lattice shown but rather just the 2x2:

$$s = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \qquad (2)$$

In order to do the correct sampling, the cell is repeated until the sampling lattice is the same size as the image to be sampled. The sampling cell may well be larger than 2x2. For example, for hexagonal sampling (more below), it must be 4x4.

3. THE SAMPLING MATRIX

Here we will want to look at a variety of sampling lattices based on certain 2x2 matrices [1] which we will denote as M. Note that this M is not the sampling cell. About the only thing that is significant about M is that the column vectors of M are the basis vectors of the lattice. Let's look at an example.

$$M_R = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \qquad (3)$$

Here and beyond in this note we will find the square bracket Matlab notation to be easiest to write. Thus a 2x2 matrix can be written as:

$$M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} = [\, a \;\; b \; ; \; c \;\; d \,] \qquad (4)$$

This means that the basis vectors are the columns [a c] and [b d], or [2 0] and [0 2] in equation (3). Any point in the lattice associated with M is a linear combination of integer multiples of these two basic vectors. This includes the lattice point (0,0) of course and the point (2,2), the point(2,8) and an infinite number of other lattice points. This M defines the lattice. Note that the elements of the M matrix are all integers (positive, negative, or zeros).

The parallelogram spanned by the basic vectors, with the exception of the outer boundaries (the two boundaries not including the point [0 0]) are called the Fundamental Parallelepiped (FPD) of the matrix M. The lattice can be generated as the periodic repetition of the FPD. Fig. 3 shows the lattice associated with M_R=[2 0; 0 2]. The FPD is the 2x2 square cornered at zero, but does not include the line segments on n_1=2 and n_2=2. The integer points inside the FPD are (0,0), (1,0), (0,1), and (1,1), but not the outer boundary points (0,2), (1,2), (2,2), (2,1), or (2,0). In this case, the integer points inside the FPD are also the sampling cell (but this is not true in general).

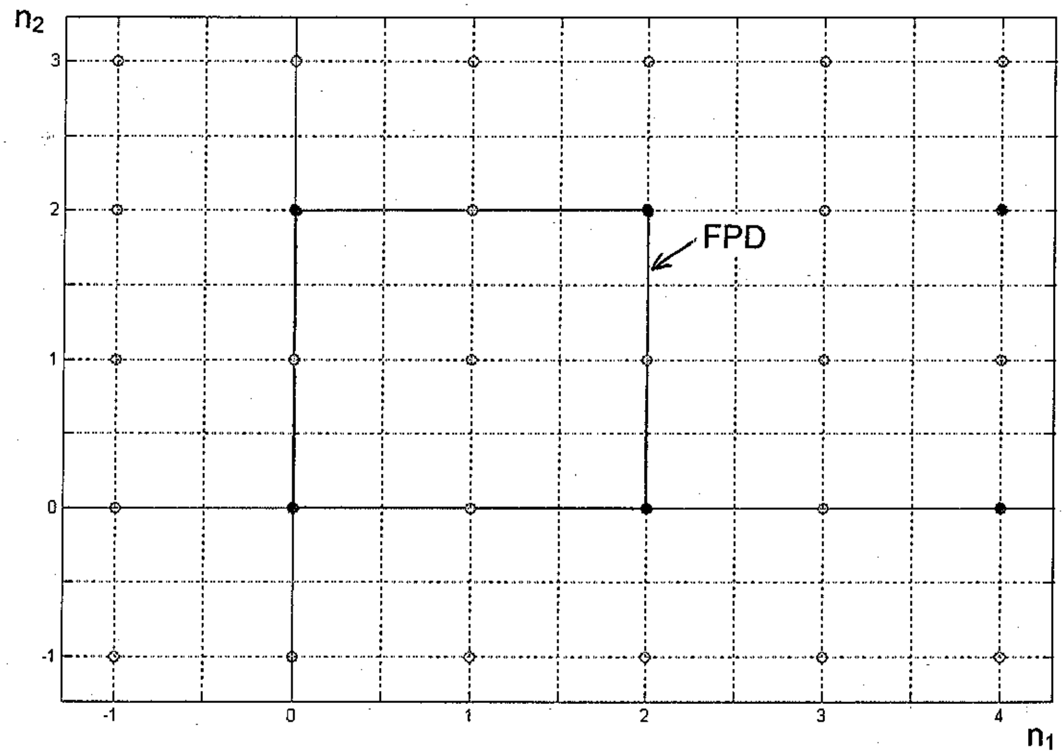


Fig. 3 The FPD and lattice (filled circles) for rectangular sampling by 4

In addition to rectangular lattices, two others are common: the "Quincunx" and the "Hexagonal." The Quincunx has a typical matrix M_Q = [1 1 ; 1 -1] while hexagonal is typically M_H = [1 1 ; 2 -2]. It is best to think of these in terms of their column vectors generating the lattice. Fig. 4 and Fig. 5 show the corresponding FPD's.

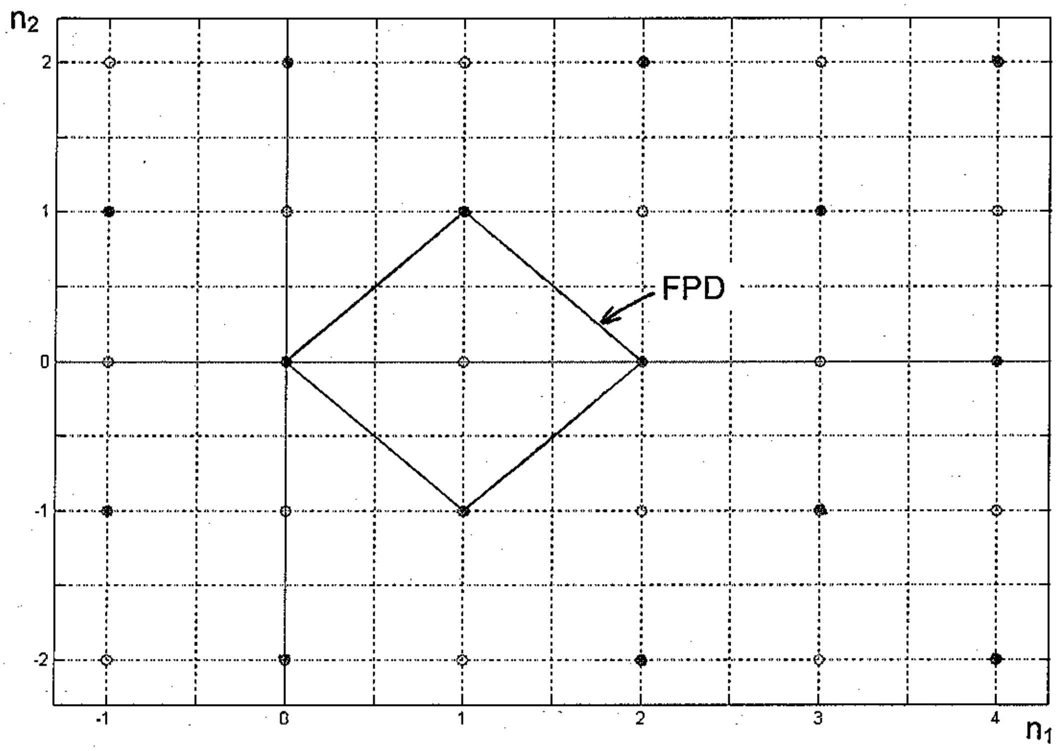


Fig. 4 The FPD and Lattice (Filled circles) of Quincunx sampling

We can see that the Quincunx sampling is composed of the rectangular sampling case plus additional samples offset by a vector [1, 1]. Rectangular sampling "by 4" means that every other line and every other row is zeroed. That is, it is "by 2" in both directions. Quincunx sampling involves every other sample of any row, the even samples of the even rows and the odd samples of the add rows. Thus the density is twice as great, and quincunx samples "by 2." The only other ways to sample by 2 would be to set every other row (or every other column) to zero. The term Quincunx refers to the pattern of five dots like the 5 face of a die. Note that the sampling factor is obtained as the absolute value of the determinant of M.

The FPD is the diamond shaped region shown. The integer vectors inside the FPD for this case include (0,0) and (1,0), but not those on the outside boundaries (1,1), (1,-1), or (2,0). The number of integer vectors inside the FPD is also given by the absolute value of the determinant. Note that in this case, the sampling cell is 2x2 and is given by:

$$s = [\,0 \;\; 1\,;\, 1 \;\; 0\,] \tag{5}$$

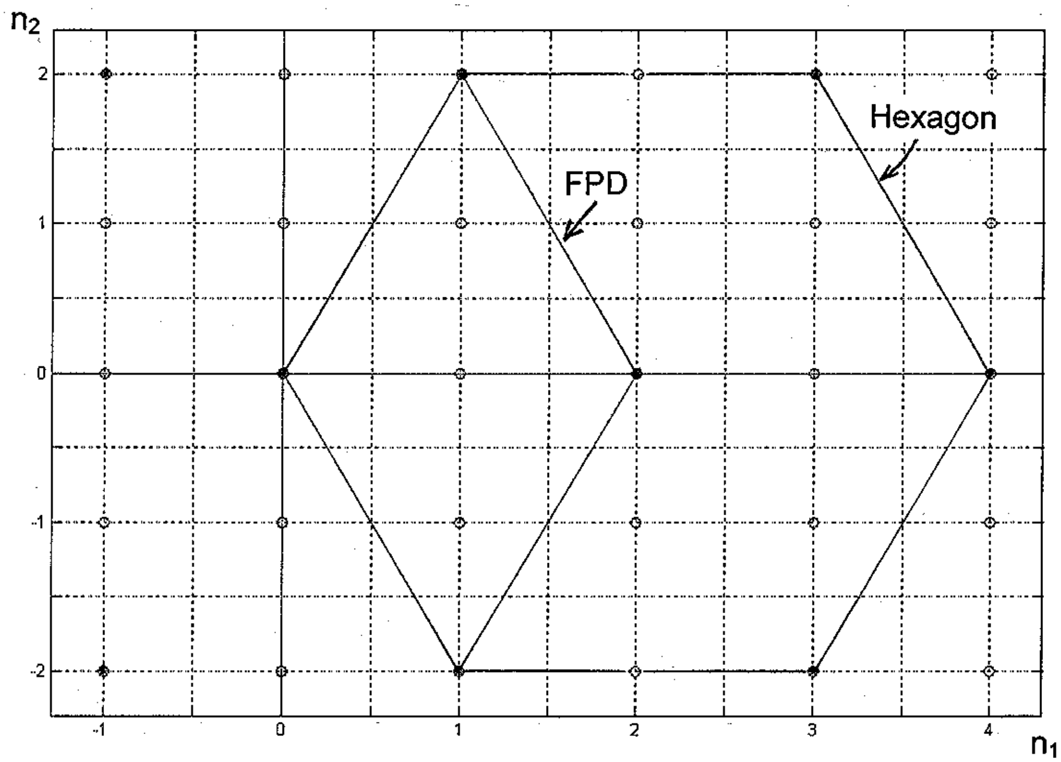


Fig. 5 The FPD and Lattice (filled circles) for Hexagonal Sampling

The lattice and FPD for the hexagonal lattice is seen in Fig. 5. Since the determinant of M_H has magnitude 4, this, like M_R, is sampling by 4. Like the Quincunx, we can interpret this as the superposition of two rectangular (by 8) lattices. Here the second one is offset by [1,2]. In an obvious way, there is an alternative offset by [2,1]. The actual (centered) "hexagon" is shown, and note that it is not a regular hexagon (the sides are 2 on the top and bottom, and $\sqrt{5}$ on the sides). We could also interpret quincunx as a hexagonal shape that is even less regular (the top and bottom are 2 while the sides are $\sqrt{2}$).

The FPD of M_H contains the points (0,0), (1,0), (1,1), (1,-1) but not the outside boundary points (1,2), (2,0), or (1,-2). Here the sampling cell is NOT 2x2, but rather 4x4.

$$s = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{bmatrix} \qquad (6)$$

4. FURTHER WORK WITH M

It is not likely that we would have paid much attention to the sampling matrix M if all we got out of it was the lattice. We might have just written down the lattice directly from chosen basis vectors, and directly calculated the sampling density. We have so far derived from the matrix M the following:

(1) The basis vectors

(2) The lattice

(3) The sampling density for the lattice

It is a trivial matter to obtain from M its transpose M^T (flip about diagonal), its inverse M^{-1}, and its "inverse transposed," M^{-T}. Here we will follow additional procedures with M to see that we can find the following additional things:

(4) The positions, in the frequency domain, of the sampling replicas associated with the lattice. (More than one matrix can give the same lattice.)

(5) A region, the SPD of $(1/2)M^{-T}$, usually called loosely "the SPD," that is a "bandlimiting" (region) for the matrix.

Just below we will describe the details of the computations we will make. This procedures is best illustrated by examples, so the reader should perhaps skim the "recipe," look at the examples, and then come back here for details if necessary. Here is the recipe:

(a) Compute the FPD of M^T (just as we did for M above). This we do in exactly the same way, finding the corners of the FPD by mapping (0,0), (0,1), (1,0), and (1,1). The outer boundaries are not part of the FPD. [Of course, we have cases (M_R and M_Q) where $M=M^T$.]

(b) Find the "k vectors." These are the integer vectors inside the FPD of M^T. These k vectors include points on the FPD boundary that are part of the two segments going through (0,0), but not those on the outer boundary. One of the k vectors is always the point (0,0).

(c) Compute $M^{-T}k$ for each of the k vectors. These give the positions in the f_1-f_2 plane where spectral replicas occur.

(d) Compute the SPD of $(1/2)M^{-T}$. The SPD means Symmetric Parallelepiped, and is computed in the same way that the FPD is except the corners are obtained by mapping (-1,-1), (-1,1), (1,-1), and (1,1). The two sides passing through the corner mapped from (1,1) are excluded (as with the FPD), but this fact does not matter much.

(e) The SPD, if replicated about all the spectral replication points will fill the entire frequency plane exactly. Thus any region of spectral support that is inside the SPD will not cause aliasing. This is analogous to low-pass 1D sampling.

(f) If the spectral support goes outside the SPD, there may or may not be aliasing. There are other regions of spectral support that completely fill the frequency plane. Among these are the SPD's corresponding to other matrices that have the same lattice.

5. VERIFYING A PREVIOUS EXAMPLE

Since we know a good deal about the rectangular sampling by 4 (Fig. 2b, Fig. 3), we want to apply the recipe to the matrix M_R first. As noted, $M_R^T=M_R$ so the FPD of M_R^T is the same as that for M_R in Fig. 3. We see that there are exactly four "k vectors" inside the FPD and these are (0,0), (1,0), (0,1), and (1,1). It is easy to find the inverse of a 2x2 matrix: we exchange the two diagonal elements, negate the two off diagonal element, and divide by the determinant. Thus M_R^{-1} is [1/2 0 ; 0 1/2] and this is also M_R^{-T}. This gives the spectral replicas as $M_R^{-T}k$ = (0,0), (1/2,0), (0,1/2), and (1/2,1/2) as shown in Fig. 6. This is in agreement with the "experimental" finding of Fig. 2b.

Next, taking $(1/2)M_R^{-T}$ we form the SPD by finding the corners generated by (-1,-1), (-1,1),(1,-1), and (1,1), which gives a square of side 1/2, centered at (0,0). This is shown in Fig. 7, along with the three replicas. These four (allowing for periodic extensions) completely fill the f_1-f_2 plane.

In the cases of Quincunx and Hexagonal sampling that we shall look at shortly, it is easy to see simple shapes other than the SPD that also fill the f_1-f_2 plane without aliasing. Here we will find a different shape by finding a different matrix that has the same lattice as M_R [step (f) of the recipe above.]. One such alternative (found by trial and error) is:

$$M_{R2} = [\,4\ \ 2\ ,\ 2\ \ 2\,] \qquad (7)$$

This case we will pursue to make the point about the spectral support, but also to give a full example of the recipe for an unfamiliar sampling matrix.

Fig. 6
Spectral replicas for rectangular sampling by 4 with M_R (periodic in units of 1 in both frequency directions).

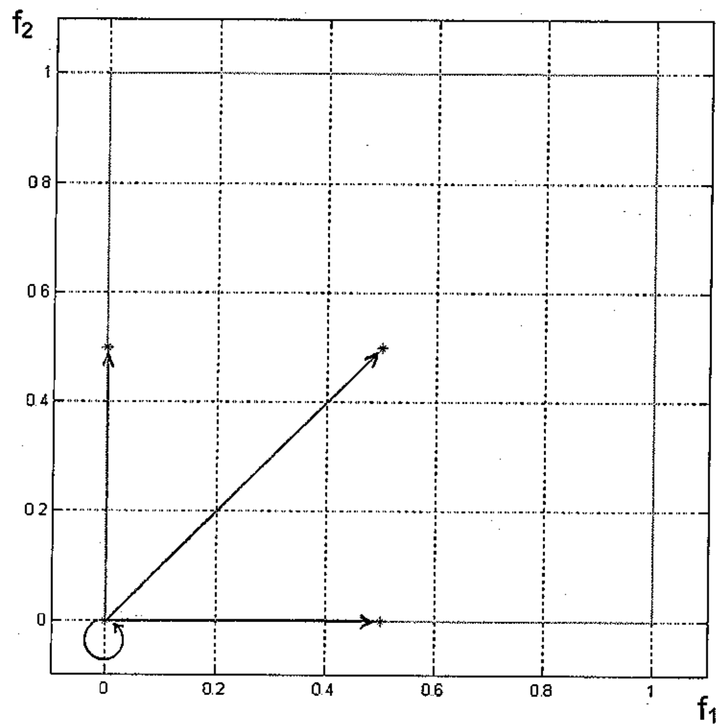
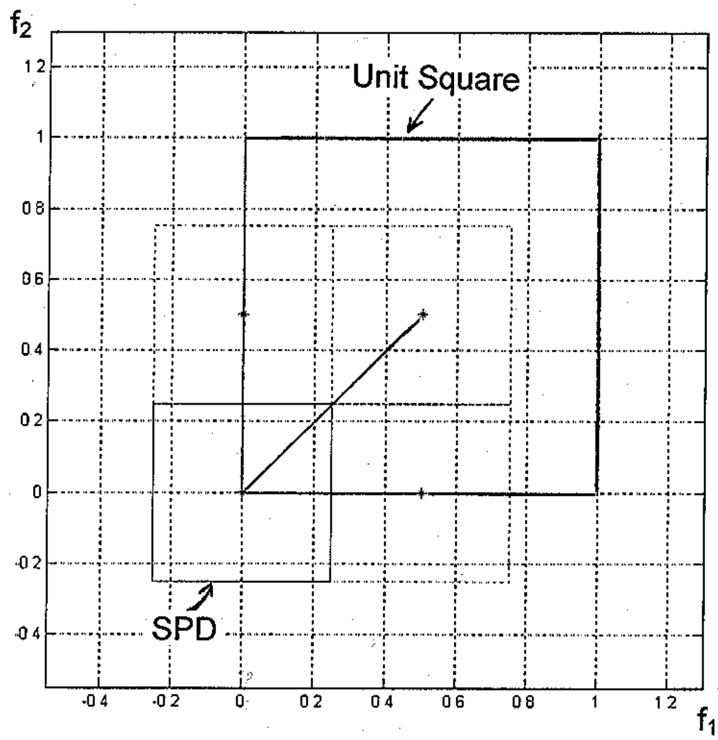


Fig. 7
SPD for M_R along with three replicas (periodically extended) fills unit square.



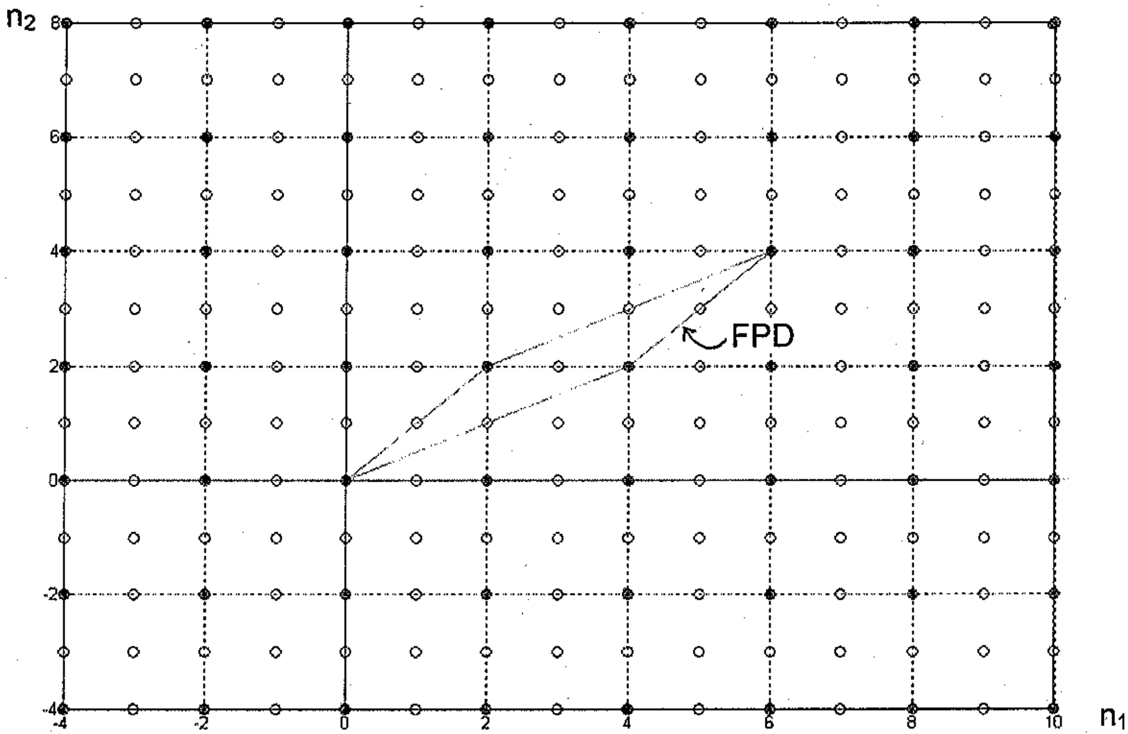


Fig. 8 M_{R2} gives the same lattice as M_R but with a different FPD and different k-vectors.

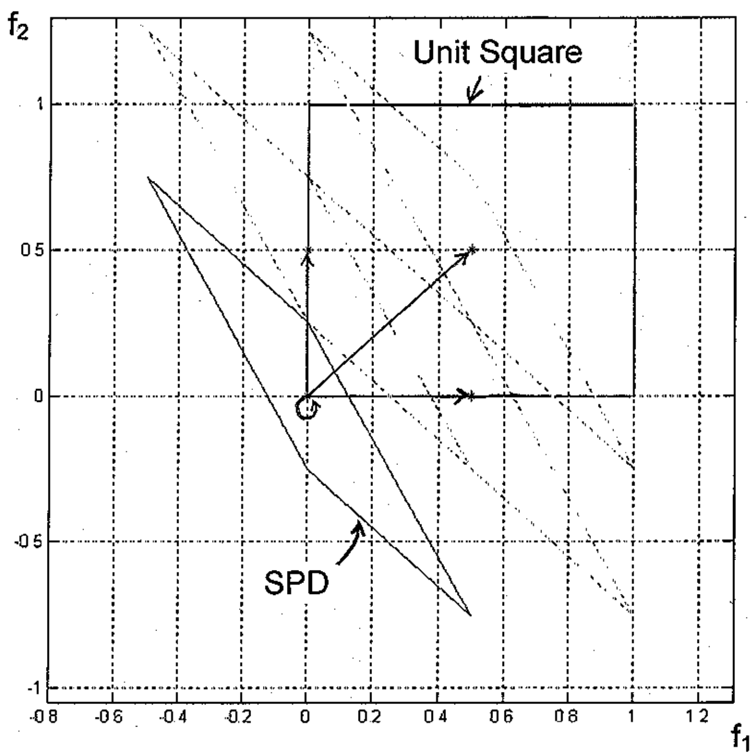


Fig. 9 SPD for M_{R2} along with three replicas. Note that the lattice of M_{R2} is the same as the lattice of M_R, and spectral replicas have the same offsets (as in Fig. 7) while the SPD's are different.

Fig. 8 shows the FPD and the lattice of M_{R2}^T (the same as the FPD of M_{R2}). We note that the lattice is the same as that of M_R (Fig.3). Here the integer vectors inside the FPD, the k-vectors, are (0,0), (1,1), (2,1), and (3,2), which are different from the M_R case. However, at the same time, we want to multiply these k vectors by M_{R2}^{-T}, not by M_R^{-T}. M_{R2}^{-T} is given by:

$$M_{R2}^{-T} = [1/2 \ \ -1/2 \ ; \ -1/2 \ \ \ 1]$$

which when multiplied by the k vectors here give spectral replicas at (0,0), (1/2 ,0), (0,1/2), and at (1/2,1/2), exactly what we had for M_R (same as Fig. 6).

Here for M_{R2}, the SPD is not the square of side 1/2 centered at (0,0), but rather the more complicated parallelogram shown in Fig. 9, which is centered at (0,0). Also seen in Fig. 9 are the three additional copies of this SPD. In order to see that these copies completely fill the f_1-f_2 plane completely, in Fig. 10 we have cut the pieces outside the unit square and pasted them inside. Fortunately, this is about as complicated at it is likely to ever get.

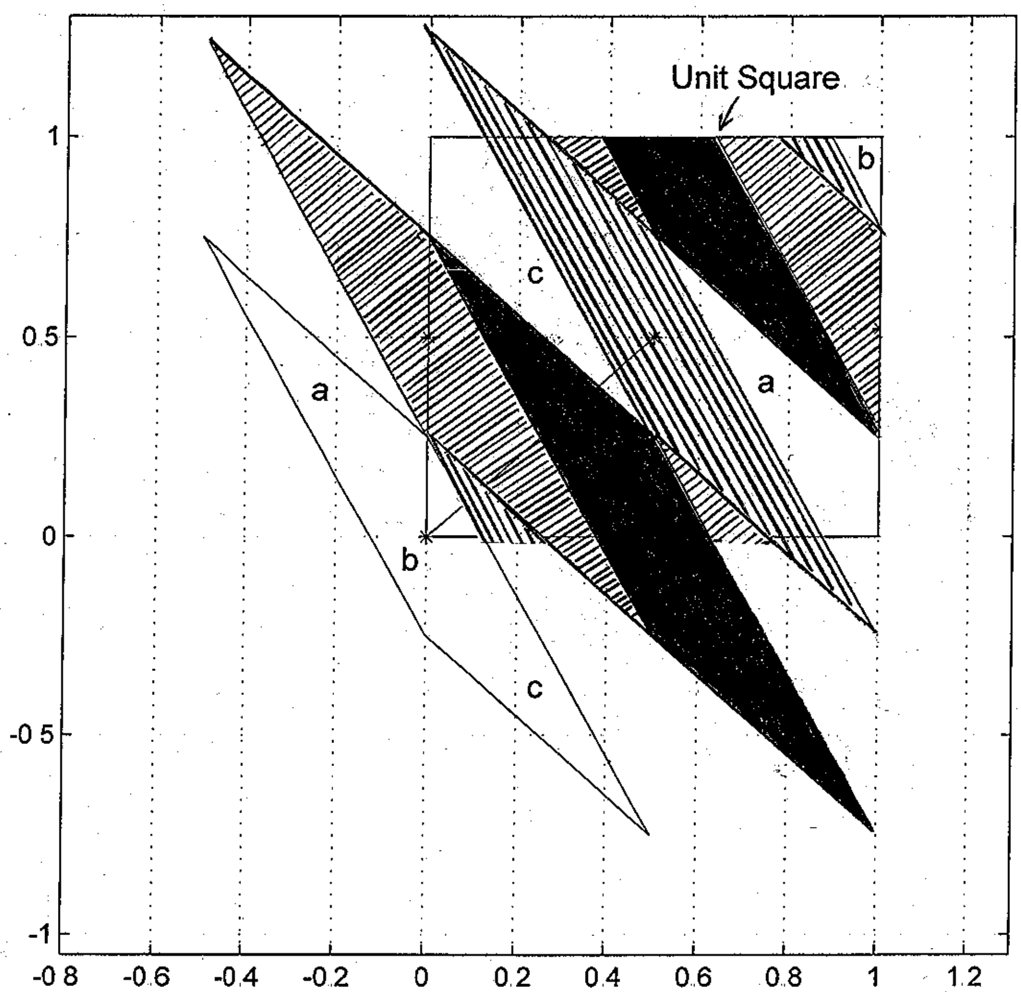


Fig. 10 Replicated SPD fills unit square. Replicas of the original SPD; segment a, b, and c are typical of the "cut and paste."

6. QUINCUNX SAMPLING

As noted above, Quincunx sampling offers the opportunity to achieve a sampling by 2 without discarding full rows or columns. Working from M_Q= [1 1 ; 1 -1] we can go through the full analysis.

Fig. 4 shows the FPD of M_Q, which is the same as the FPD of M_Q^T since M_Q^T=M_Q here. We note from this that there are two k-vectors, (0,0) and (1,0). M_Q^{-T} = [1/2 1/2;1/2 -1/2] so the spectral replicas occur at (0,0) and at (1/2,1/2) in the f_1-f_2 plane (Fig. 11).

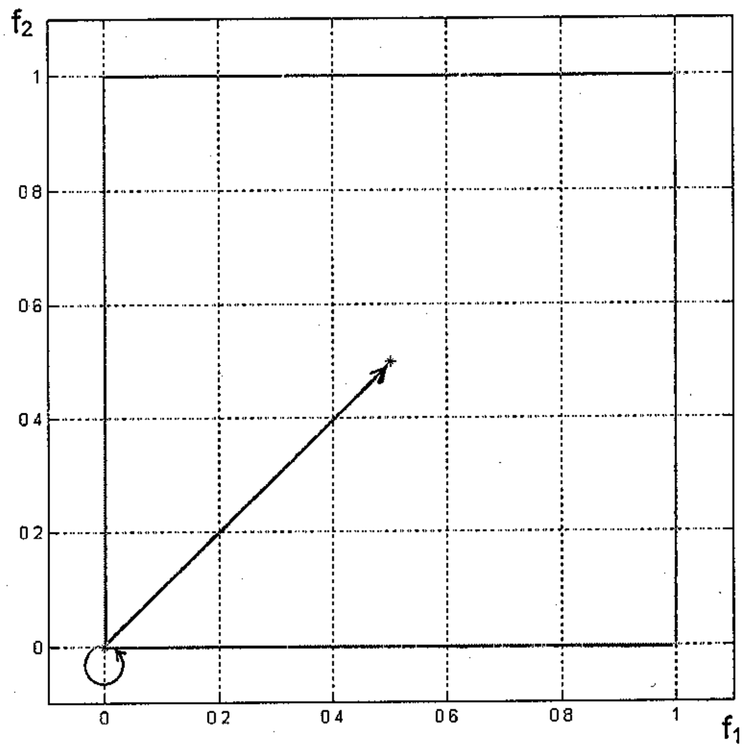


Fig. 11 Quincunx sampling results in an image at (1/2, 1/2)

The matrix (1/2)M_Q^{-T} is [1/4 1/4 ; 1/4 -1/4] so, multiplying the points (-1,-1), (-1,1), (1,-1), and (1,1) by this matrix, the SPD here is a diamond shape with corners at (0,1/2), ((1/2,0), (0,-1/2), and (-1/2,0) in the f_1-f_2 plane, as seen in Fig. 12. We note that this SPD when repeated periodically fills the unit square in the f_1-f_2 plane, in the same manner that we saw for a different case in Fig. 10, although this one is much simpler. The area of the SPD is of course 1/2 here. Further it is easy to see that the rectangular support shape also shown in Fig. 12 will fill the f_1-f_2 plane without overlap, and that this too has area 1/2. This rectangular shape falls outside the SPD. This situation is analogous to bandpass sampling in 1D.

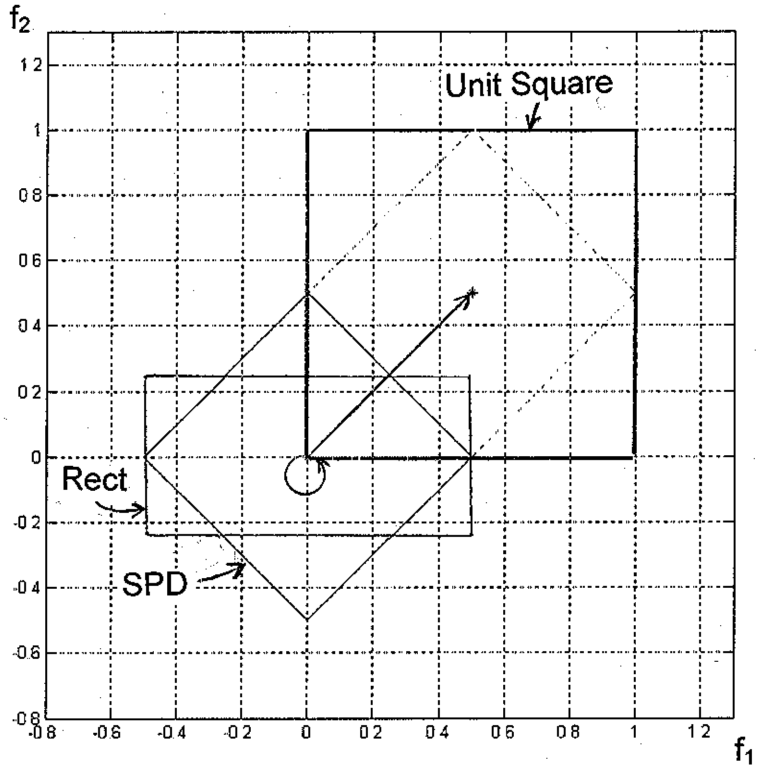


Fig. 12 The SPD for the Quincunx is the diamond shape.

7. HEXAGONAL SAMPLING

The case of Hexagonal sampling is given by $M_H = [\ 1\ \ 1\ ;\ 2\ \ -2]$. The lattice and the FPD of M_H were seen in Fig. 5. Here since $M_H{}^T$ is not the same as M_H, we do need to look specifically at the FPD of $M_H{}^T$, and this is seen in Fig. 13. We see that there are four k-vectors: (0,0), (1,0), (1,-1), and (2,-1)inside the FPD of $M_H{}^T$. The matrix $M_H{}^{-T}$ is given by [1/2 1/2 ; 1/4 -1/4] so spectral replicas occur at (0,0), (1/2,1/4), (0,1/2), and at (1/2,3/4) as shown in Fig. 14. The SPD in this case is a squashed diamond with corners at (0,1/4), (1/2,0), (0,-1/4) and at (-1/2,0), as seen in Fig. 15, along with the three spectral replicas. Note again that the unit square in the f_1-f_2 plane is filled. Also, there are other shapes that fill the unit square which extend outside the SPD. For example, the square with side 1/2 will work (Fig. 16).

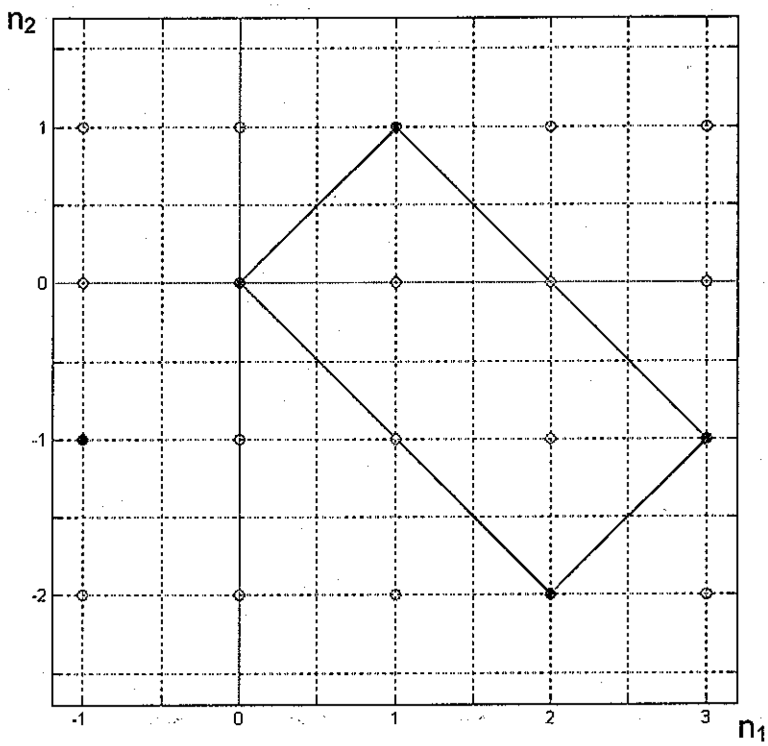


Fig. 13 For M_H, since M_H^T is not equal to M_H, we need the FPD of M_H^T as shown.

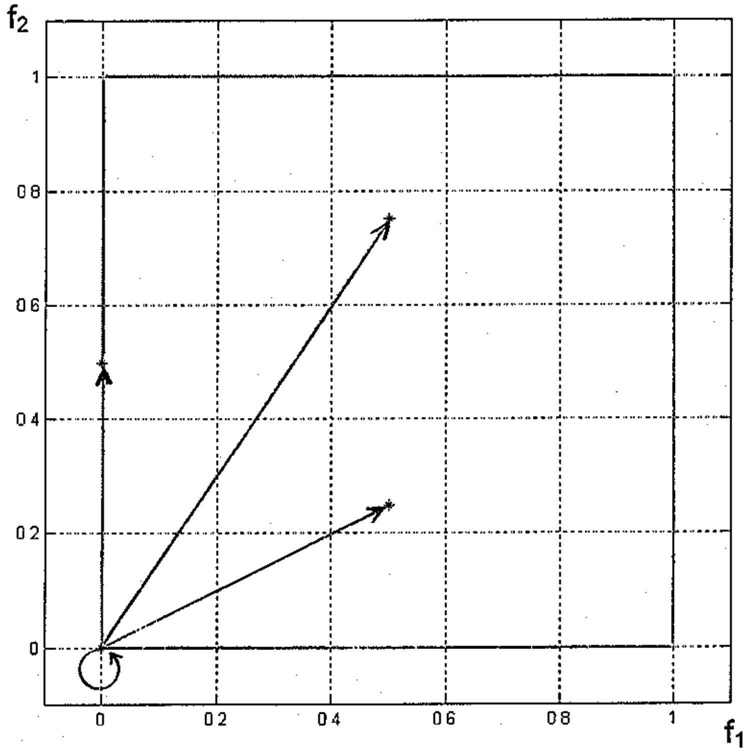


Fig. 14 Spectral replicas for M_H.

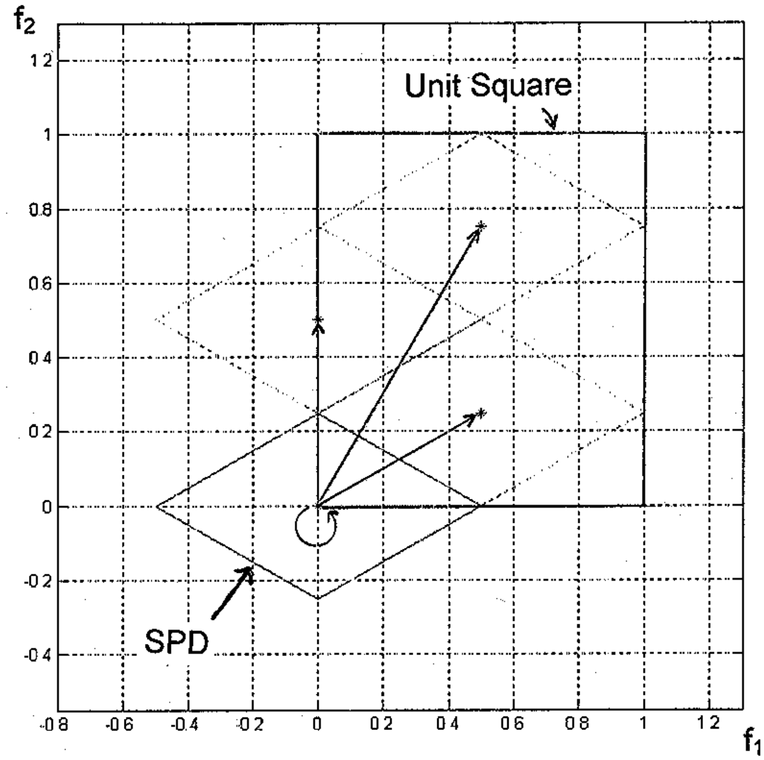


Fig. 15
SPD of M_H along with three replicas fills the unit square

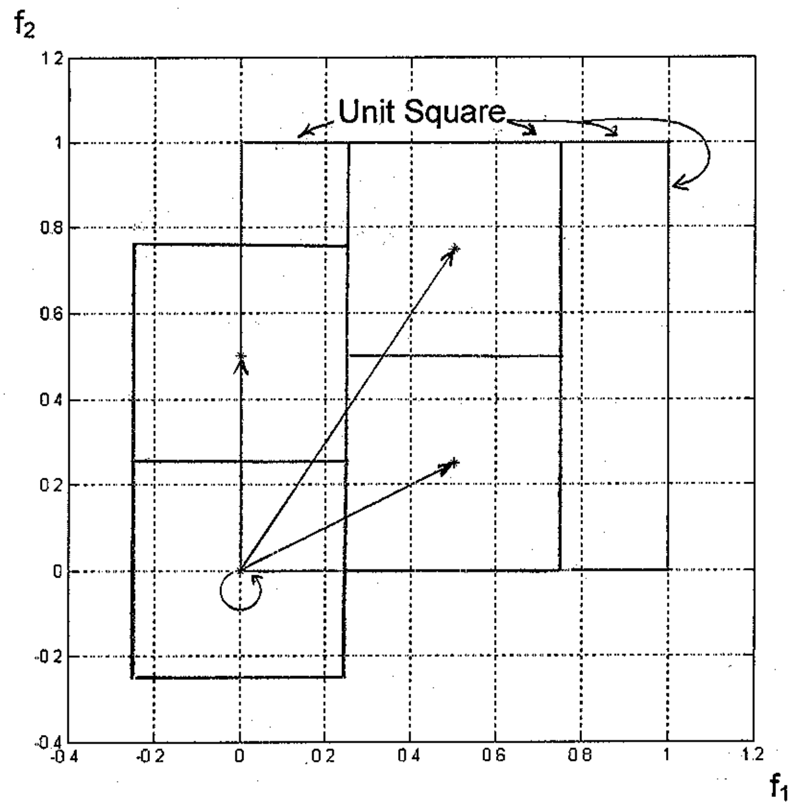


Fig. 16
A square region of support is also possible for M_H.

8. SAMPLING BY 3

Many times we may have in mind a general idea of the sampling factor we wish to achieve. For example, if we wanted to reduce the number of samples by a factor of 4, we could use our Rectangular of Hexagonal examples. If we only wanted sampling by 2, the Quincunx is the choice. Still there is a lot of room between 2 and 4, and we might like to consider sampling by 3 if this is possible. Clearly all that we need to do is find a matrix with a determinant of magnitude 3, so this is easy. However, we may also want to consider if the sapling pattern is relatively uniform. Consider the case of choosing a matrix M_3=[1 1 ; 1 -2]. Here $M_3^T=M_3$ so Fig. 17 shows the lattice of M_3 and the FPD of M_3^T. Note that if, for example we had a lattice that kept every third column, the points would be one unit apart in the vertical direction but three apart in the horizontal direction. In this case, the points are separated from others by distances of $\sqrt{2}$ or $\sqrt{5}$. Thus we think of the lattice for M_3 as being fairly well distributed.

Here M_3^{-T} = [2/3 1/3 ; 1/3 -1/3] while the k-vectors are (0,0), (1,0), and (1,-1), so spectral repetitions are a (0,0), (2/3,1/3), and at (1/3,2/3), Fig. 18, while the SPD and its two repetitions are shown in Fig. 19.

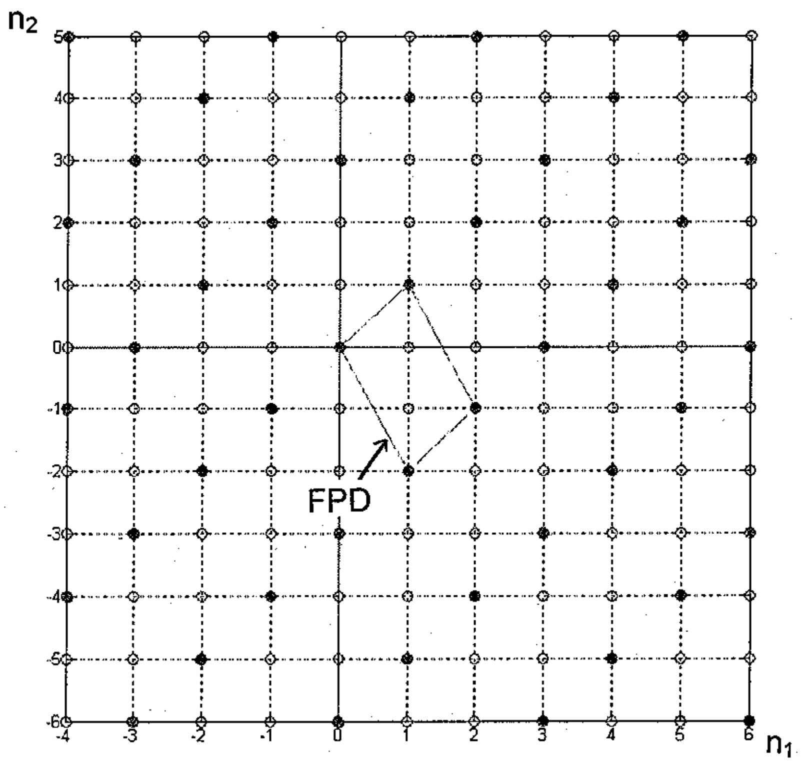


Fig. 17 The Lattice of M_3 and the FPD of M_3^T

Fig. 18
Spectral replicas for M_3.

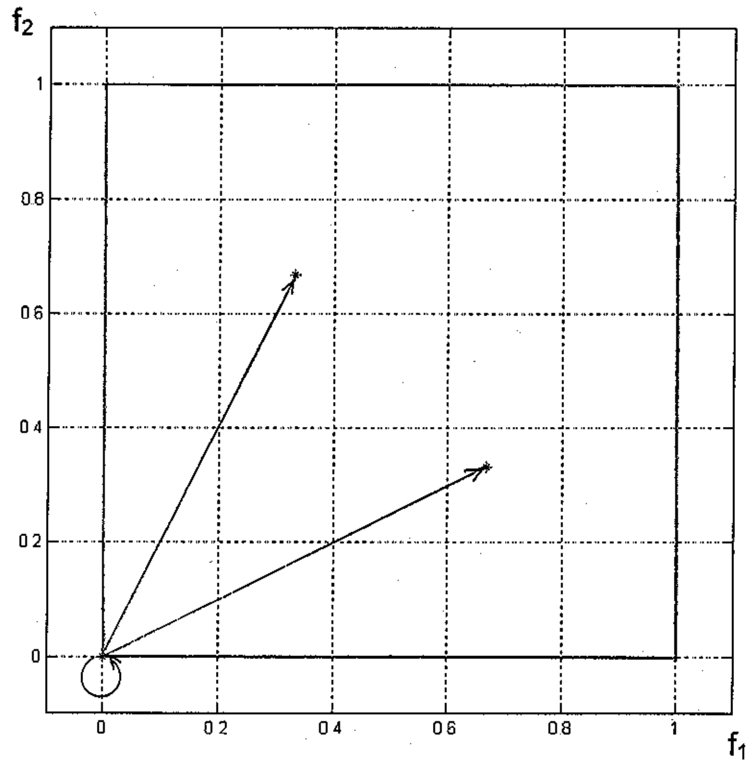
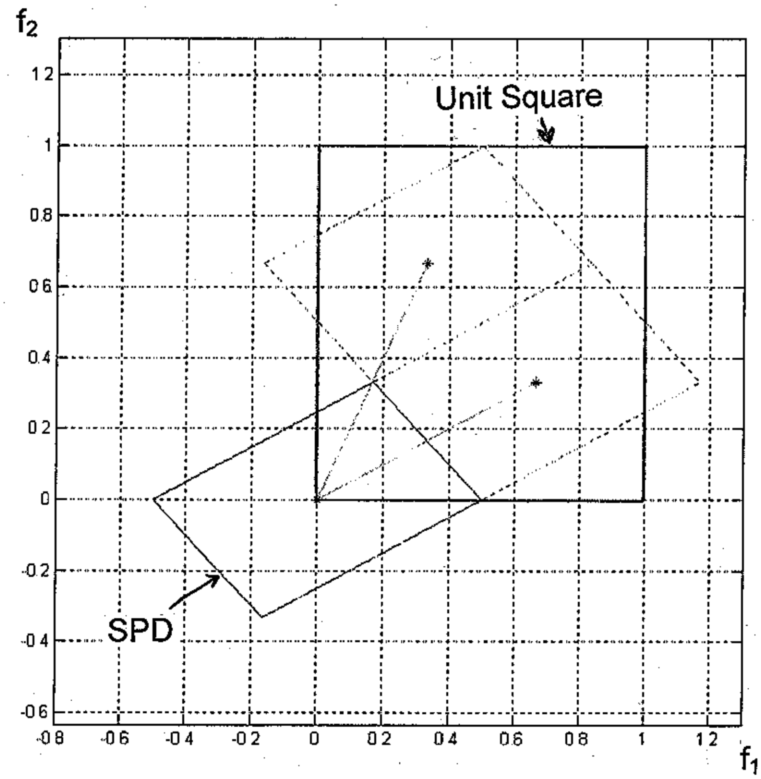


Fig. 19
SPD and two replicas for M_3.



9. CIRCULAR SUPPORT

Above we have shown that 2D sampling results in replicas of the original spectrum, and we have assumed certain shapes for the spectrum to see if the replicas overlap (result in aliasing). Various shapes, the actual SPD, a rectangle, a diamond, and so on can all be tried. But what is the spectrum of an actual image likely to be?

One approach is to take several images, take their FFT (2D), and plot it (likely removing the DC term which is likely to be very large for the usual 0-255 integer format of common images). Plotting these (perhaps with Matlab's fftshift2 to put low frequencies in the middle) we obtain the notions that different images can be very different, and that in a very general sense, they are circular. We might have thought they would be rectangular, but there are no preferred directions in a general image.

Accordingly we ask how large a circular image can be and not overlap. Our approach is largely geometrical here. Fig. 20 shows the case of rectangular sampling with circular support of radius (1/4). It is easy to see that inside the unit square there are four full circles [1 + 4 x (1/2) + 4 x (1/4)]. Thus the area used is $4\pi(1/4)^2 = \pi/4 = 0.7854$. The corresponding case for quincunx sampling is shown in Fig. 21, except here the radius of the circles is $\sqrt{2}/4$ – but there are only two of them. Perhaps surprisingly this too uses $\pi/4$ of the total area (the geometry is just rotated 45 degrees).

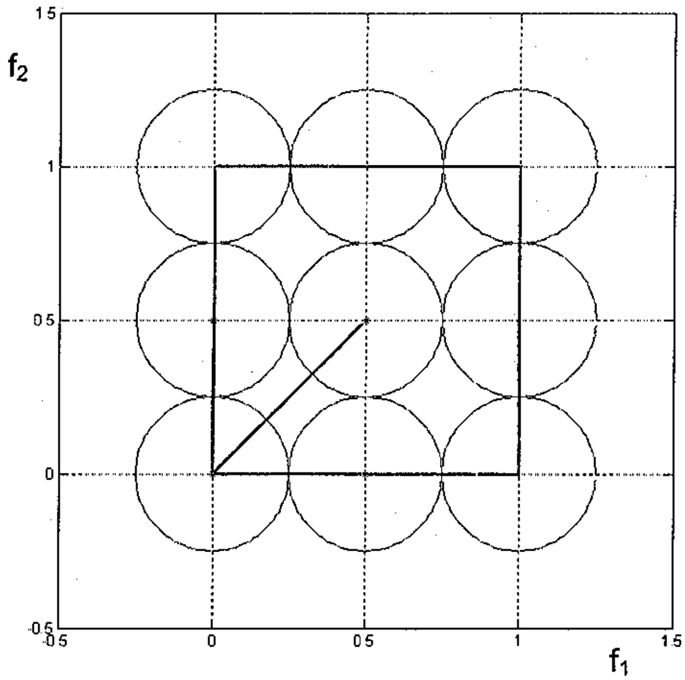


Fig. 20 Circular support of radius (1/4) with rectangular sampling

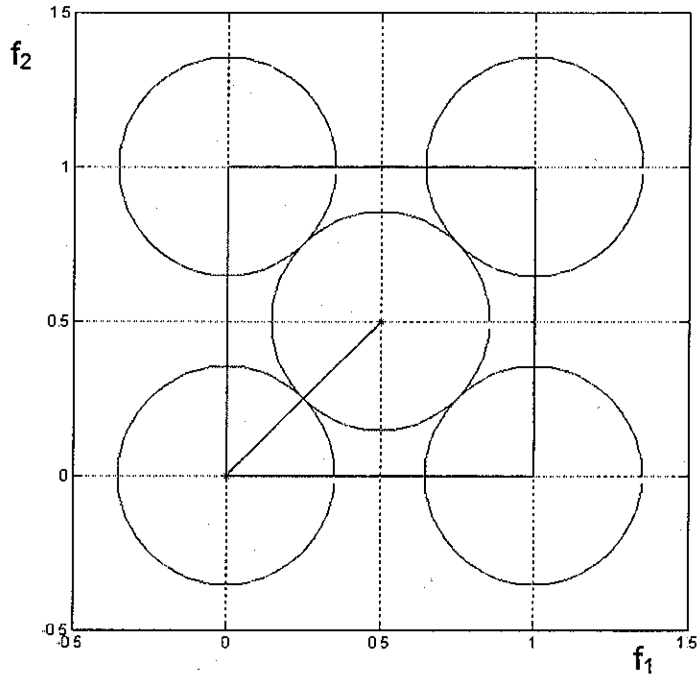


Fig. 21 Circular support of radius $\sqrt{2}/4$ with quincunx sampling

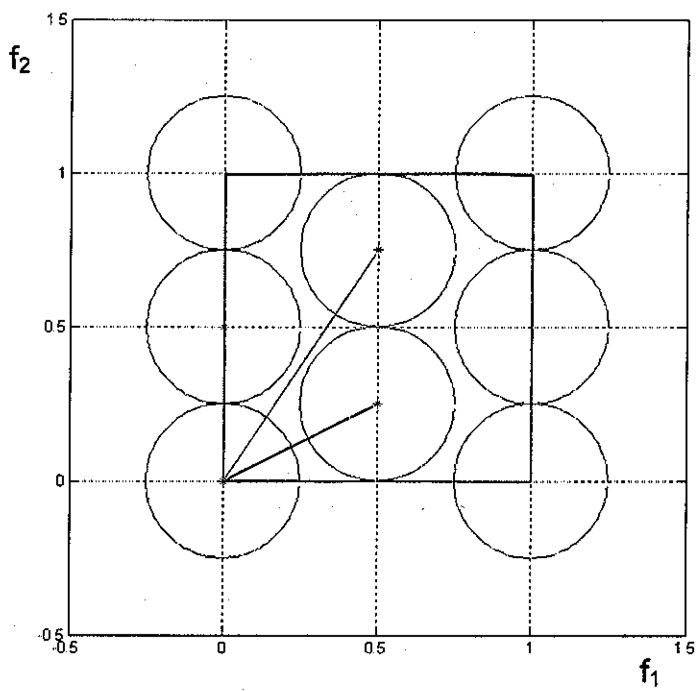


Fig. 22 Circular support or radius 1/4 with hexagonal sampling

Fig. 22 shows us the curious case where we show circular support of radius 1/4 using hexagonal sampling. Once again, we have four full circles inside the unit square, and the fractional usage is still, $\pi/4$! In fact, all that has happened is that the middle column of circles in Fig. 20 has slid down by 1/4. This has left open space between the columns. So while we might think the circles could be larger based on the f_1 axis, they are already touching in the f_2 direction. This is a consequence of the fact that we have taken our hexagonal lattice from a square lattice. This will be discussed more in Section 10.

For completeness, we have shown our sampling-by-three example in Fig. 23. Here we see that the circles touch along the diagonal when the radius is $\sqrt{2}/6$. There are three full circles inside the unit square, so the ratio is $\pi/6 = 0.5236$, which is not very efficient.

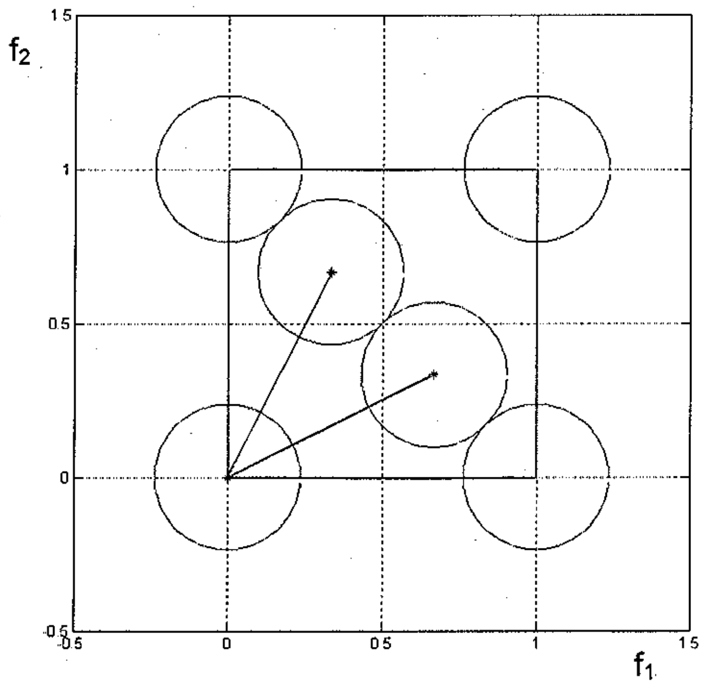


Fig. 23 Circular support of radius $\sqrt{2}/6$ while sampling by 3

10 A REGULAR HEXIGON?

As noted in Section 3, we chose a Hexagonal sampling pattern from among available rectangular lattice points, and this was not a "regular" hexagon (not all six sides are of equal length). Specifically, the four slanted sides in Fig. 5 are of length $\sqrt{5}$= 2.2359 while the two horizontal sides are 2. However, our notion that these are not equal is predicated on the idea that the spatial axes (n_1 and n_2) are integers. That is, it looked like n_1 and n_2 were both normalized spatial variables and were equal in spacing. A more general approach would be to say that the integers are the indices of the actual lattice points.

In this case, the actual lattice spacing could be L_1 in the n_1 direction and L_2 in the n_2 direction, where L_1 is not necessarily equal to L_2. This has a corresponding (and reciprocal) effect on the frequencies f_1 and f_2.

The required geometry for an equilateral hexagon is that circles centered on the nearest neighbor can be enlarged until the touch exactly (Fig. 24). With circular support, it is clearly not possible to get the circles any closer. How much of the total area does this arrangement use? In Fig. 24, the base of the equilateral triangle is 2 while its height is $\sqrt{3}$. Thus the area is $(1/2)\cdot 2\cdot\sqrt{3} = \sqrt{3}$. This triangle contains three 60° slices of the circles, or half a circle, for area $(1/2)\cdot\pi\cdot 1^2 = \pi/2$. The fraction of the available area that is actually used is thus $(\pi/2)/\sqrt{3} = 0.9069$. This we compare with rectangular, quincunx, and non-equilateral hexagonal which were $\pi/4 = 0.7854$, so this represents an improvement.

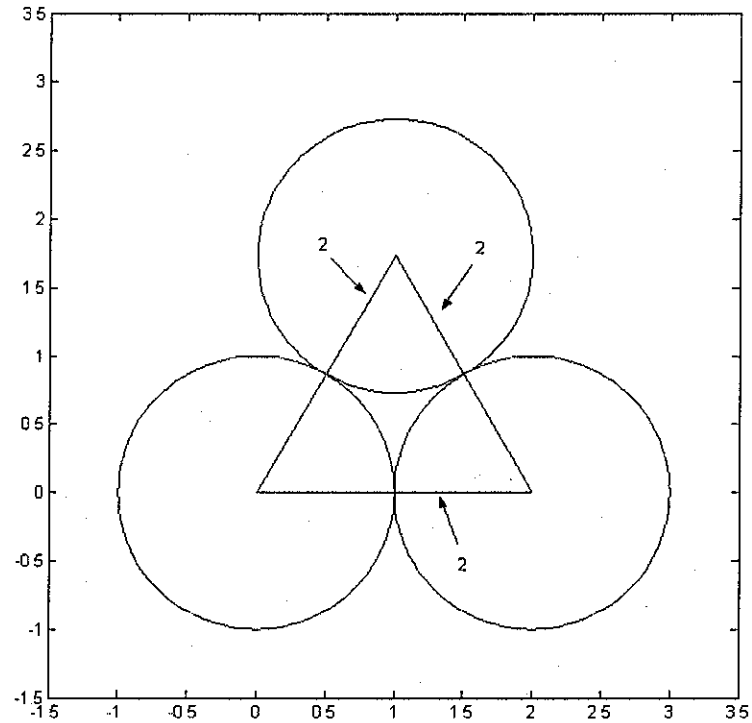


Fig. 24 Detail of Geometry of the Equilateral Hexagon

For the equilateral hexagon, the circles in the frequency domain would just touch, as would circles in the spatial domain if we were trying to maximize their area. However, fitting the spatial domain hexagon to the rectangular lattice increases L_2 to 1 from $\sqrt{3}/2$, and this decreases the vertical frequency by a factor of $\sqrt{3}/2$, and thus we see the circles

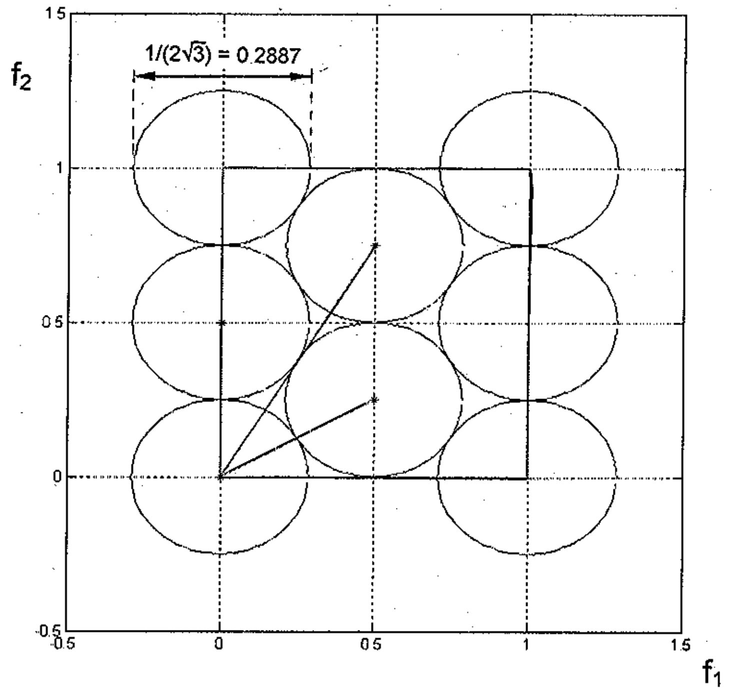


Fig. 25 Expanding the "circles" in the horizontal direction produces ellipses that touch (compare to Fig. 22.)

touching first in the vertical direction (Fig. 22). We can not make the circles larger, but we can increase their horizontal width by a factor of $2/\sqrt{3}$ = 1.1547 (making them ellipses) and now they will just touch (Fig. 25). Equivalently we could just reduce the horizontal scale.

Either interpretation convinces us that the frequency plane is fully used under equilateral hexagonal sampling. In this case, there is however slightly more resolution in frequency vertically than horizontally ("unit square" is $\sqrt{3}/2$=0.866 tall and 1 wide).

It might well be that we would prefer to have the increases resolution in the horizontal rather than in the vertical direction, and this would be easy to achieve. The basic vectors for the lattice would be (2,1) and (-2,1), and we would use M=[2 -2; 1 1], which is just the hexagonal lattice rotated 90°.

PROGRAMS

Generator for Fig. 1 and Fig. 2

```
X=[1 1 1 1 1 1 zeros(1,53) 1 1 1 1 1]
x=ifft(X)
s=[1 0 1 0];
s=[s s s s s s s s s s s s s s s s]
xs=x.*s;
XS=fft(xs)
XS=abs(XS);
figure(1)
subplot(211)
stem([0:63],X)
axis([-2 66 -.1 1.2])
subplot(212)
stem([0:63],XS)
axis([-2 66 -.05 .6])


X=zeros(65,65);
for k=1:6
    for m=1:6
         X(k,m)=1;
     end
end
X=X+fliplr(X);
X=X+flipud(X);
X=X(1:64,1:64);
figure(2)
mesh(X)

s=[0 0 0 0;1 0 1 0;0 0 0 0;1 0 1 0];
s=[s s;s s];
s=[s s;s s];
s=[s s;s s];
s=[s s;s s];

x=ifft2(X);
xs=s.*x;
XS = fft2(xs);
XS=abs(XS);
figure(3)
mesh(XS)
```

LAT.M

Main Processor of M Matrix

```
%   lat.m  find 2D sampling lattice, etc. corresponding
%   to a sampling matrix M
%   examples:  rectangular sampling M=[2 0; 0 2]
%              quincunx sampling    M=[1 1; 1 -1]
%              hexagonal sampling   M=[1 1; 2 -2]
%
%    B. Hutchins    Spring 2007

function d=lat(M)
d=det(M)

% find FPD of M by "corner mapping"
L1=M*[0 0]';
L2=M*[1 0]';
L3=M*[1 1]';
L4=M*[0 1]';

% find limits for plotting
top=max([L1(2),L2(2),L3(2),L4(2)]);
bot=min([L1(2),L2(2),L3(2),L4(2)]);
lef=min([L1(1),L2(1),L3(1),L4(1)]);
rig=max([L1(1),L2(1),L3(1),L4(1)]);
figure(1)
r=4;     % plot beyond FPD
% plot axes
plot([lef-r rig+r],[0 0],'r')
hold on
plot([0 0],[bot-r top+r],'r')

% plot integer vectors as circles
for k = bot-r:top+r
    for p = lef-r:rig+r
        plot(p,k,'o')
    end
end
% plot FPD
plot([L1(1) L2(1) L3(1) L4(1) L1(1)],[L1(2) L2(2) L3(2) L4(2) L1(2)],'g')

% try integer vectors to see if on lattice - overplot  o with *
for n1=-10:10
    for n2=-10:10
        y=M*[n1,n2]';
        plot(y(1),y(2),'*r')
    end
end
```

```
hold off
grid
axis([lef-r rig+r bot-r top+r]);
axis('square')
title('FPD(M) in Green, * = LAT(M)');
%  figure(1) completed
% ------------------------------------------------------------------


% Now do FPD of M' - this will be figure(2)
%  same as for FPD above

MT=M';

F1=MT*[0 0]';
F2=MT*[1 0]';
F3=MT*[1 1]';
F4=MT*[0 1]';

top=max([F1(2),F2(2),F3(2),F4(2)]);
bot=min([F1(2),F2(2),F3(2),F4(2)]);
lef=min([F1(1),F2(1),F3(1),F4(1)]);
rig=max([F1(1),F2(1),F3(1),F4(1)]);
figure(2)
r=4;
plot([lef-r rig+r],[0 0],'r')
hold on
plot([0 0],[bot-r top+r],'r')

for k = bot-r:top+r
    for p = lef-r:rig+r
        plot(p,k,'o')
    end
end

plot([F1(1) F2(1) F3(1) F4(1) F1(1)],[F1(2) F2(2) F3(2) F4(2) F1(2)],'g')

for n1=-10:10
    for n2=-10:10
        y=MT*[n1,n2]';
        plot(y(1),y(2),'*r')
    end
end

hold off
grid
axis([lef-r rig+r bot-r top+r]);
axis('square')
title('FPD(M Transpose) in Green, * = LAT(M Transpose)');
%  figure(2) completed
% ------------------------------------------------------------
```

```
% find the k vectors - this will be figure(3)
% integer vectors inside FPD of MT

e=0.00000001;
% tolerance for roundoff problems with comparisons at outer edge

% map back to unit square
kn=1;
 for n1=(lef-1):(rig+1)
    for n2=(bot-1):(top+1)
       kt=inv(MT)*[n1,n2]';
       if kt(1) >= 0 & kt(1) < (1-e) & kt(2) >= 0 & kt(2) < (1-e)
         k(kn,1)=n1;
         k(kn,2)=n2;
         kn=kn+1;
       end
    end
end
kn=kn-1
k=k'
% k-vectors found

% find positions of replicas, M inverse transpose times k
ss=[0 0]';
for m=1:kn
    ss=(inv(M)')*[k(1,m) k(2,m)]'
    s(m,1)=ss(1);
    s(m,2)=ss(2);
end

% frequency domain square
figure(3)
plot([0 1 1 0 0],[0 0 1 1 0],'g')
hold on
% cyan vectors with red * finals
for m=1:kn
plot([0 s(m,1)],[0 s(m,2)],'c')
end
for m=1:kn
plot(s(m,1),s(m,2),'*r')
end

grid
hold off
axis([-.1 1.1 -.1 1.1]);
axis('square')
title('M Inv Trans * k vectors  Spectral Replica Centers')
figure(3)
% figure(3) done
```

```
% find the SPD and plot
% SPD matrix
SPDM = (1/2)*inv(M)';
% corner map
spd1=SPDM*[-1 -1]';
spd2=SPDM*[-1 1]';
spd3=SPDM*[1 1]';
spd4=SPDM*[1 -1]';
% frequency domain square
figure(4)
plot([0 1 1 0 0],[0 0 1 1 0],'g')
hold on
% displacement vectors as in figure(3)
for m=1:kn
plot([0 s(m,1)],[0 s(m,2)],'c')
end
for m=1:kn
plot(s(m,1),s(m,2),'*r')
end

s1=[spd1(1) spd2(1) spd3(1) spd4(1) spd1(1)];
s2=[spd1(2) spd2(2) spd3(2) spd4(2) spd1(2)];
% replicas of SPD about displacements s
for m=1:kn
s11=s1+s(m,1);
s22=s2+s(m,2);
plot(s11,s22,'b:')
end
% overplot original solid
plot(s1,s2,'b')

hold off
lef=min([s1 0])-.3;
rig=max([s1 1])+.3;
top=max([s2 1])+.3;
bot=min([s2 0])-.3;
grid
axis([lef rig bot top]);
axis('square')
title('SPD 1/2M Inv Transp + replicas')
figure(4)
% figure(4) completed
```