Image processing and Two-Dimensional FT

Jonathan M. Lees
University of North Carolina, Chapel Hill
Department of Geological Sciences
CB #3315, Mitchell Hall
Chapel Hill, NC 27599-3315
email: jonathan.lees@unc.edu
ph: (919) 962-0695

1 Simple Corrugated Surface

Next consider two examples of surfaces that consist of ripples or corrugations in the x and y-directions respectively. Note that when there are more ripples in one direction in the space domain the peaks in the FT domain are further apart. There is an inverse relation between the frequency (or wavenumber) and the wavelength.

Next rotate the corrugations:

The noise comes from the fact that the FT assumes the images is repeated, i.e. it is periodic in x-y directions. To get rid of some of the noise we can taper the signal and re-apply the FT.

```
d = dim(Arot)
dex = seq(from=-d[1]/2, by=1, length=d[1])
dwhy = seq(from=-d[2]/2, by=1, length=d[2])
M = meshgrid(dex, dwhy)
w2 = matrix(1, ncol=d[1], nrow=d[2])
dis2 = M$x^2+M$y^2
dis = sqrt(dis2)
sig = 35
w2 = exp( -dis2/sig^2)
w2[dis<sig ] = 1
image(w2)
```

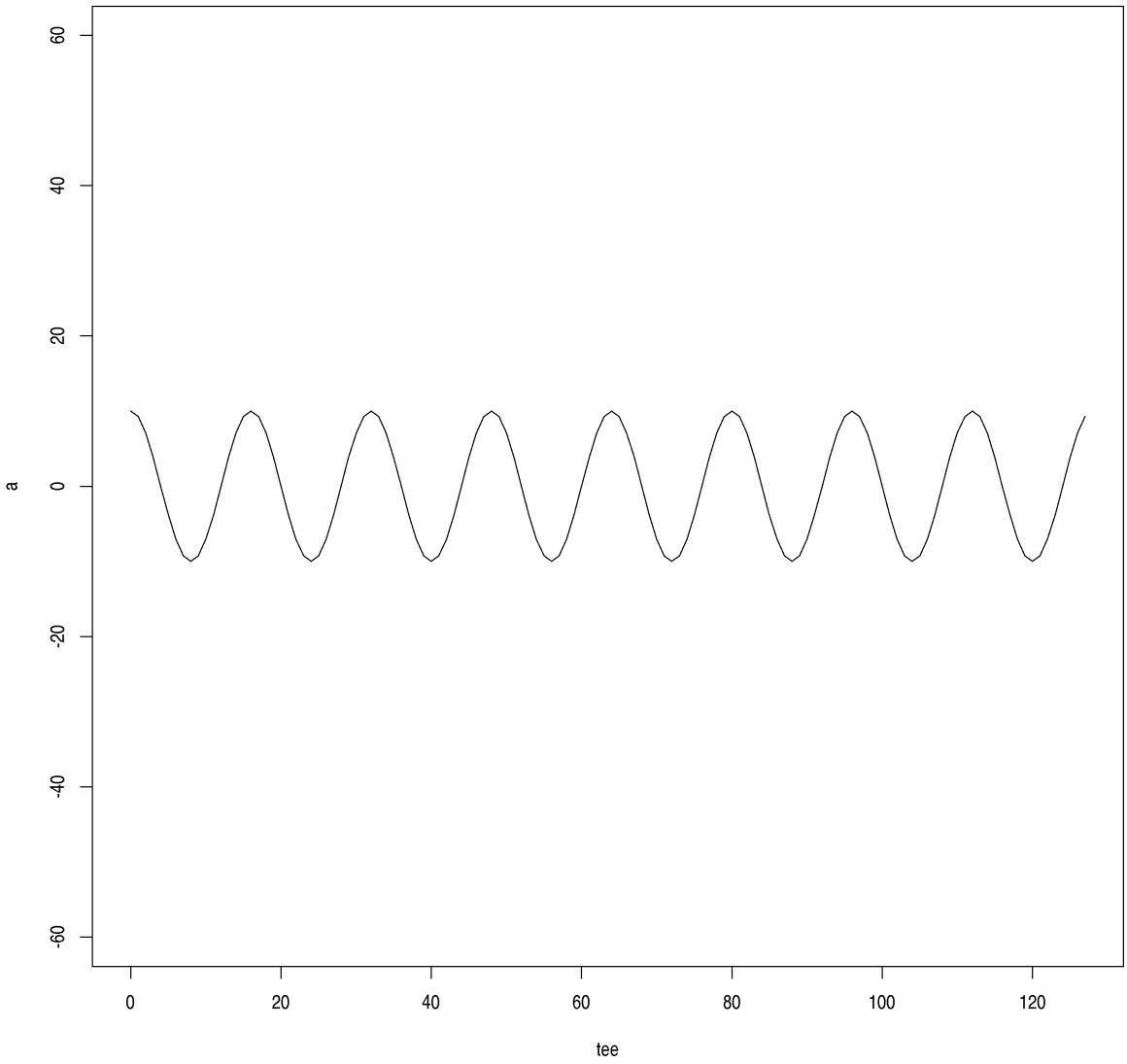

Figure 1: Simple sinusoid

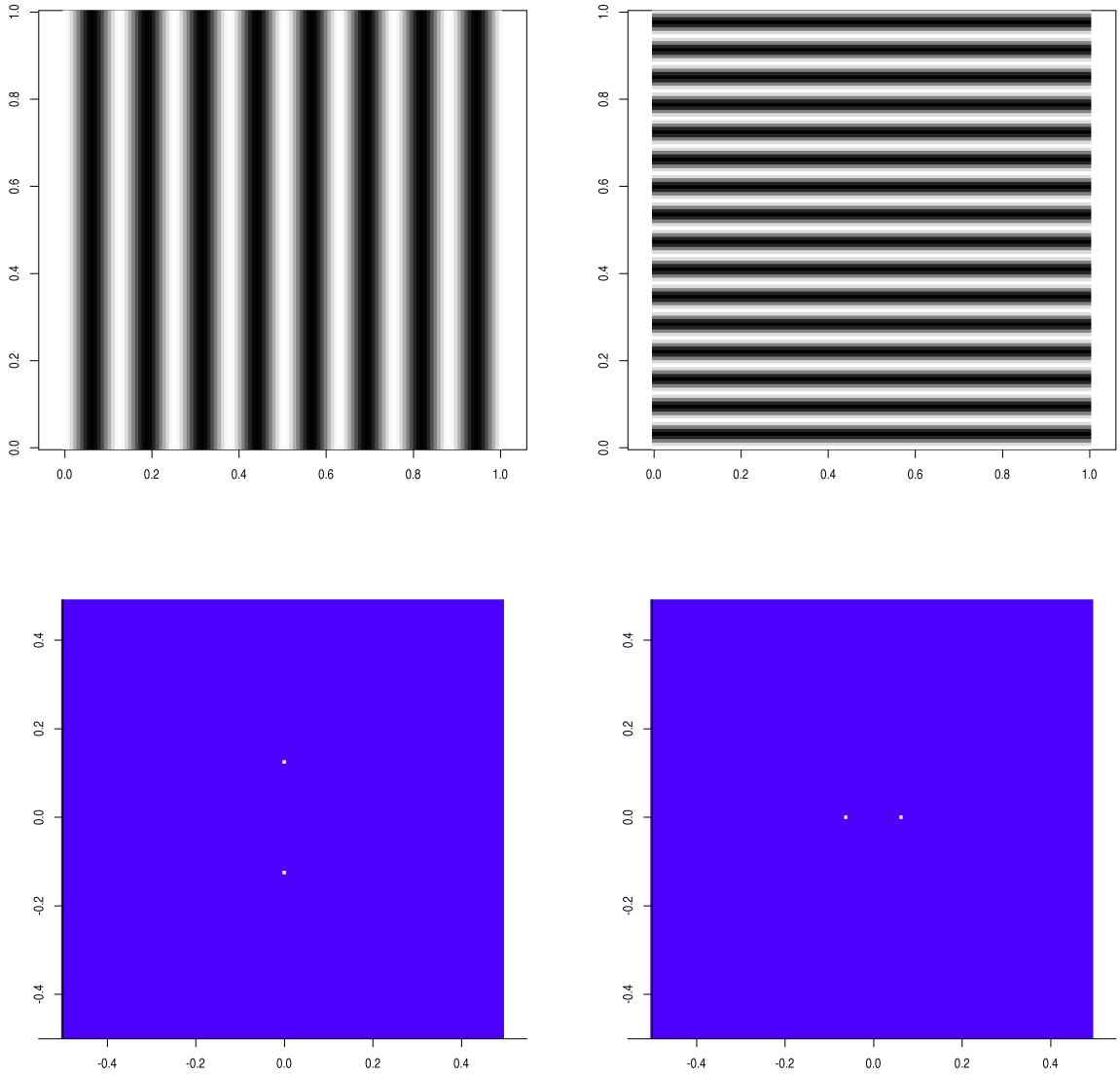

Figure 2: Corrugated Roofs and associated 2DFTs

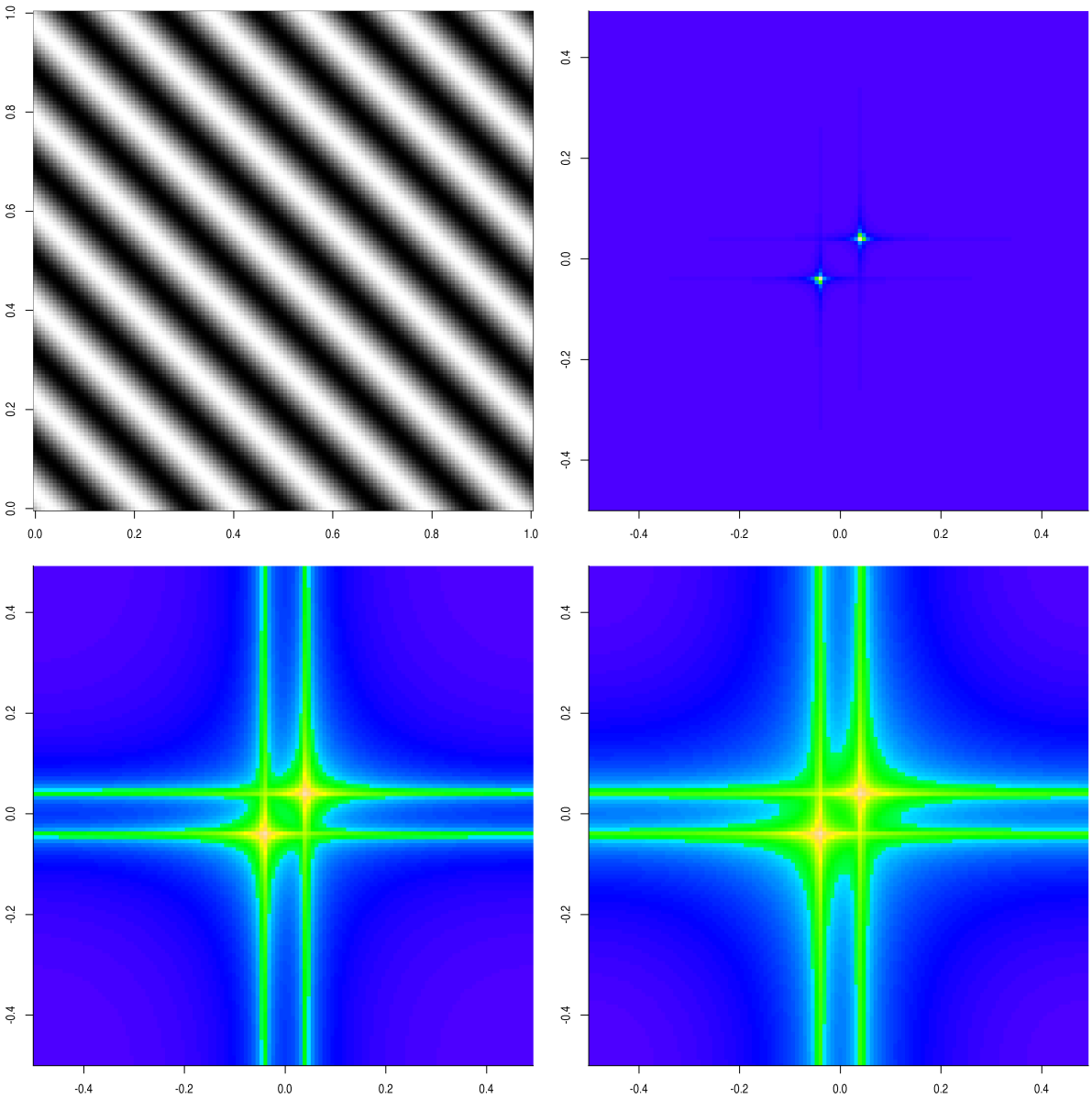

Figure 3: Rotated Corrugated Roofs and associated 2DFTs

```
w2 = costap(sqrt( dis2 )  , 45, 60)
Brot = Arot*w2
B2d = ddfft(Brot)
```

```
Brot = Arot*w2
B2d = ddfft(Brot)
JPNG(file="./FIGS/rooftaper.png" , width=12, height=12)
par(mfrow=c(2,2))
par(mai=c(.5, .4, 0.2, 0.1))
image(Brot, col=gr, asp=1)
show2dfft(B2d, scale=0, col=topo.colors(100)  )
title("Raw Periodogram")
show2dfft(B2d, scale=1, col=topo.colors(100)  )
title("Log(1+Mod)")
show2dfft(A2d, scale=1, col=topo.colors(100)  )
title("Non-tapered Log(1+Mod)")
dev.off()
```

2 Simple Patterns

2.1 Diagonal Line

Consider a simple case with a straight line running through the center of the image:

```
x = 0:255
y = 0:255
g = meshgrid(x, y)
z = matrix(0, ncol=256, nrow=256)
z[g$x==g$y] = 1
Fz  = ddfft(z)
gr = grey(0:100/100)
JPNG(file="./FIGS/Diagline.png" , width=8, height=4)
par(mfrow=c(1,2))
par(mai=c(.5, .4, 0.2, 0.1))
image(x=x, y=y, z, col=gr, asp=1, axes=FALSE, ann=FALSE)
```

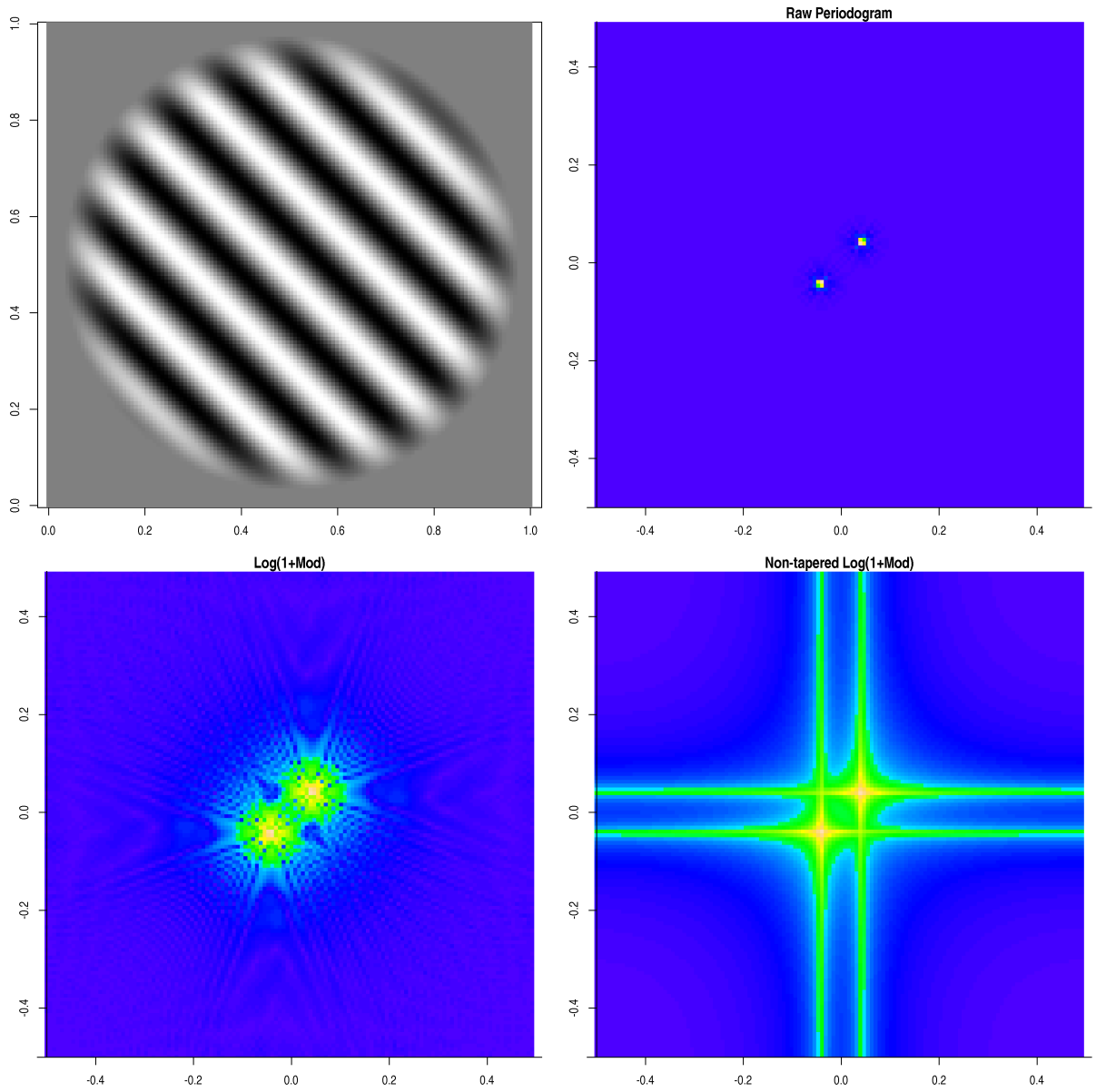

Figure 4: Tapered Rotated Corrugated Roofs and associated 2DFTs

```
zz  = show2dfft(Fz, scale=3)
dev.off()
```

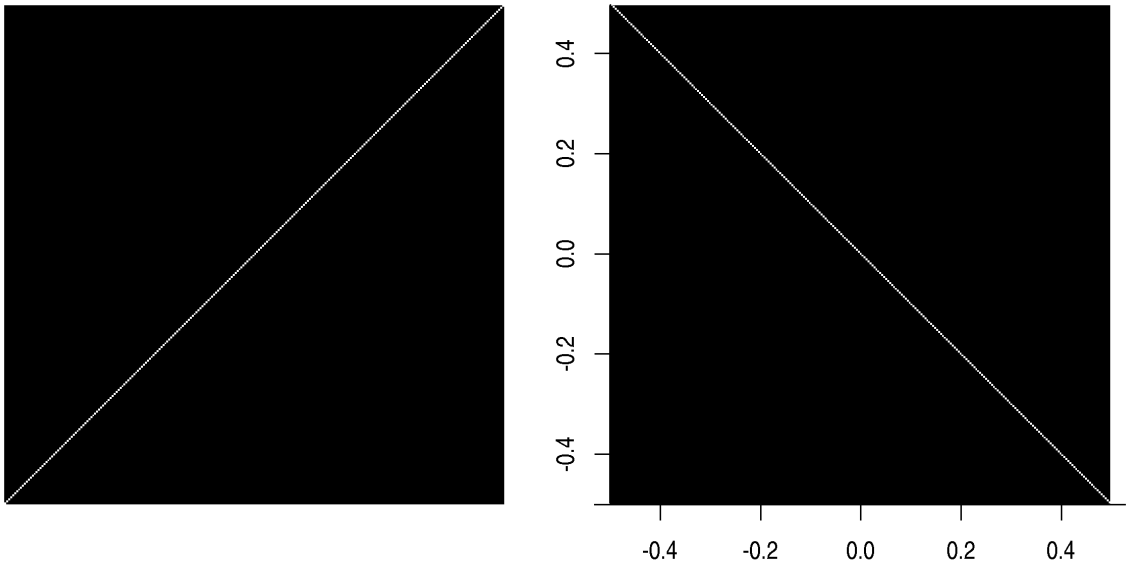

Figure 5: Diagonal line, and 2D F-transform

2.2 Square Filter

```
z = matrix(0, ncol=256, nrow=256)
a = round( 0.2*length(x) )
mid = floor(length(x)/2)
side = c(mid-a/2, mid+a/2    )
z[g$x>side[1] & g$x<side[2] &  g$y >side[1] & g$y<side[2]  ] = 1
Fz  = ddfft(z)
gr = grey(0:100/100)
JPNG(file="./FIGS/Square.png" , width=8, height=4)
par(mfrow=c(1,2))
par(mai=c(.5, .4, 0.2, 0.1))
image(x=x, y=y, z, col=gr, asp=1, axes=FALSE, ann=FALSE)
zz  = show2dfft(Fz, scale=3)
dev.off()
```

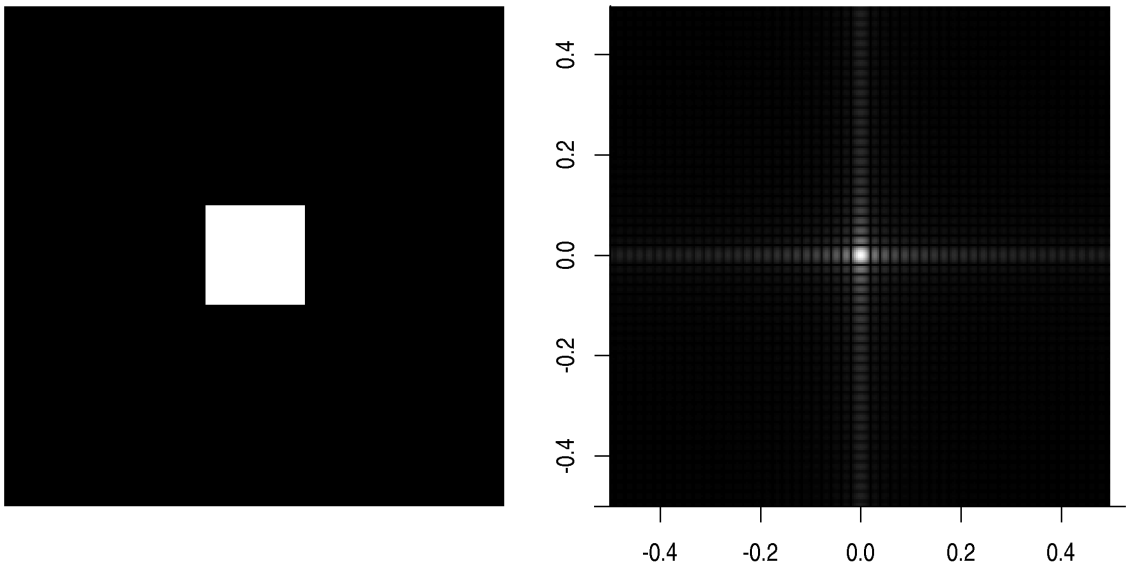

Figure 6: Square and 2D F-transform

2.3 Circular Filter

```
z = matrix(0, ncol=256, nrow=256)
a = round( 0.2*length(x) )
mid = floor(length(x)/2)
side = c(mid-a/2, mid+a/2    )
z[ ((g$x-mid)^2 + (g$y-mid)^2) < a^2  ] = 1
Fz  = ddfft(z)
gr = grey(0:100/100)
 JPNG(file="./FIGS/Circ.png" , width=8, height=5)
par(mfrow=c(1,2))
par(mai=c(.5, .4, 0.2, 0.1))
image(x=x, y=y, z, col=topo.colors(100) , asp=1, axes=FALSE, ann=FALSE)
zz  = show2dfft(Fz, scale=3, col=topo.colors(100))
dev.off()
```

2.4 Random Noise

```
z = matrix(runif(256*256) , ncol=256, nrow=256)
Fz  = ddfft(z)
```

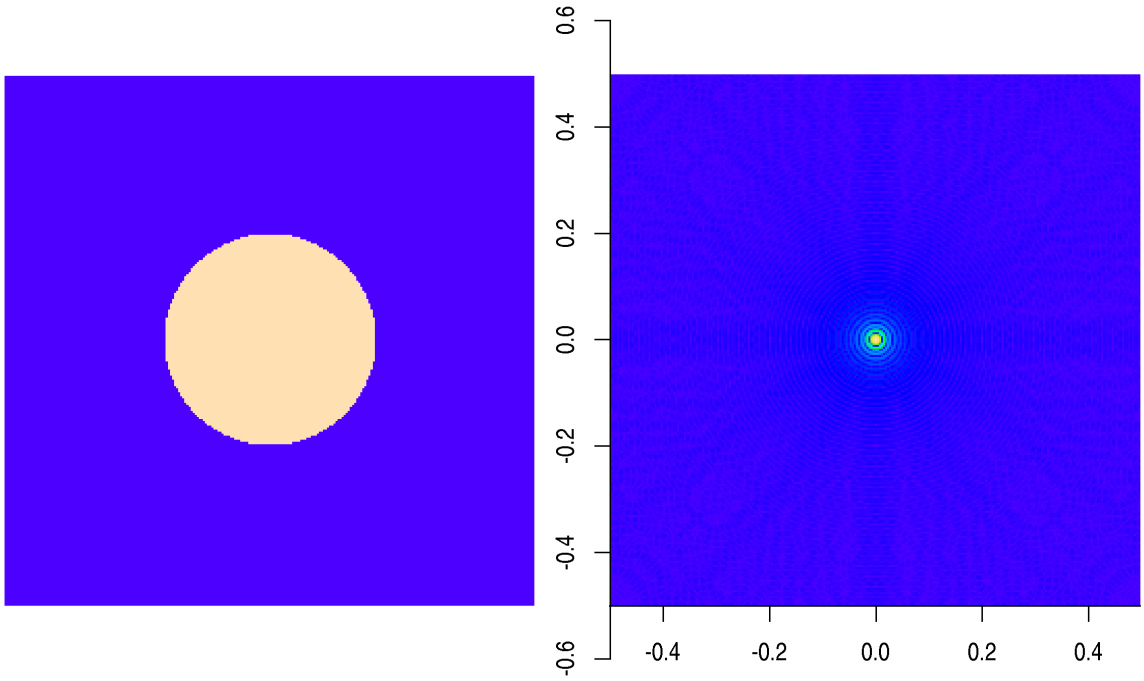

Figure 7: Circ and 2D F-transform

```
gr = grey(0:100/100)
 JPNG(file="./FIGS/unifNOISE.png" , width=8, height=4)
par(mfrow=c(1,2))
par(mai=c(.5, .4, 0.2, 0.1))
image(x=x, y=y, z, col=topo.colors(100) , asp=1, axes=FALSE, ann=FALSE)
zz  = show2dfft(Fz, scale=3, col=topo.colors(100))
dev.off()
```

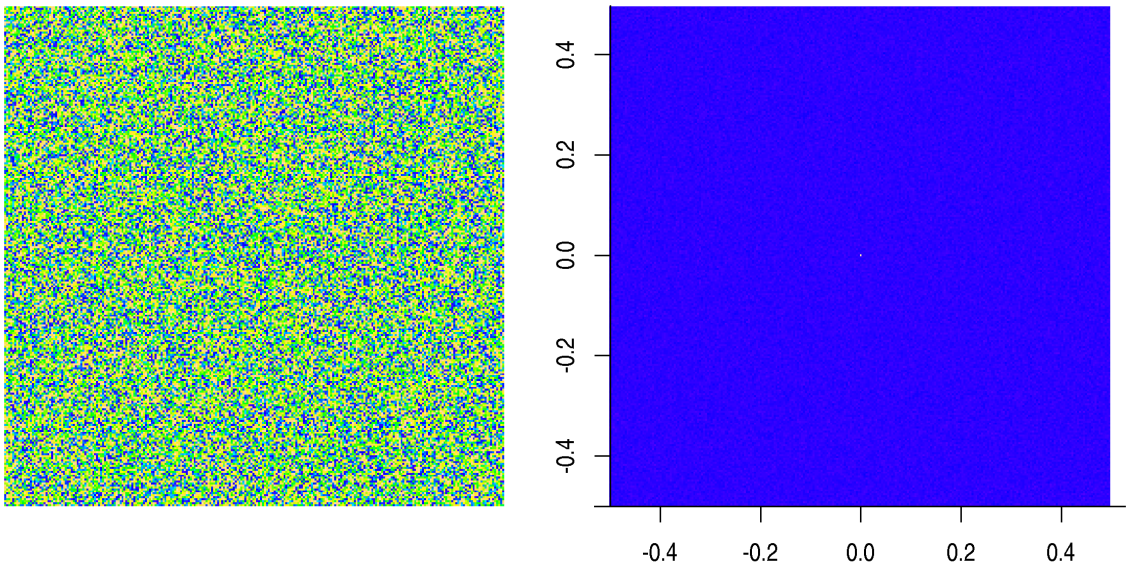

Figure 8: Uniform noise and 2D F-transform

3 Image Analysis

Read in the Coffee beans photo.

```
grpal = grey(0:100/100)
fn ='./DATA/2008925120125.pnm'
beanz = getimageGRAY(fn, PLOT=FALSE)
beanz = beanz-mean(beanz)
dbe = dim(beanz)
Fz = ddfft(beanz)
  JPNG(file="./FIGS/Beanz.png" , width=8, height=4)
par(mfrow=c(1,2))
image(x=0:(dbe[1]-1),y=0:(dbe[2]-1)   ,  beanz, col=grpal, asp=1, axes=FALSE,  ann=FALS
title(main="Beans")
########   use sqrt scaling to see details better
image(sqrt( Mod(Fz)), col=grpal, asp=1, axes=FALSE,  ann=FALSE)
title(main="2D FFT sqrt scaling")
dev.off()
```

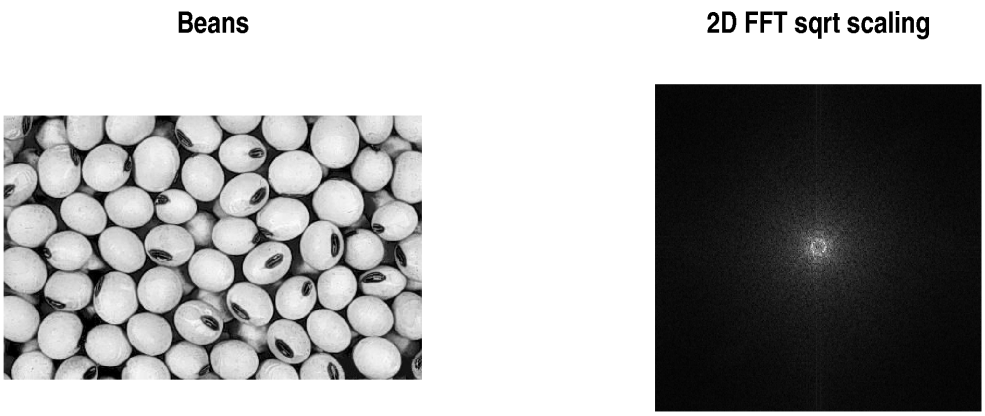

Figure 9: Beans Photo and 2dFt

4 Lena - Controversial Model

In the 1970's a famous image was used in many image processing studies. The face became a standard image used as an example. It has several interesting features useful for testing algorithms of image compression and filter studies. After the rise of feminism, it was discovered that the face came from a cropped photo of a nude model and some researchers objected. Lena was by then, however, very famous and invited to technical conferences, and even awarded prizes. She was proud of her contribution.

```
yum  <- read.pnm("./DATA/lena_std.pnm"  )
S1 = shade.col(100, c(94/255, 38/255, 18/255), bcol = c(1, 1, 1))
S2 = shade.col(100, c(112/255, 66/255, 20/255), bcol = c(1, 1, 1))
#############   grayscale version of Lena
y2 = as(yum, "pixmapGrey")
######################  flip the photo for use with image
z = attr(y2, 'grey')
zup = t(z[rev(1:512), ])
zup = zup-mean(zup)
 JPNG(file="./FIGS/Lena1.png" , width=15, height=15)
par(mfrow=c(2,2))
par(mai=c(.5, .4, 0.2, 0.1))
plot(yum, asp=1, axes=FALSE, ann=FALSE)
image(zup, col=gr, asp=1, axes=FALSE, ann=FALSE)
image(zup, col=S1, asp=1, axes=FALSE, ann=FALSE)
image(zup, col=S2, asp=1, axes=FALSE, ann=FALSE)
dev.off()
```

```
Flena = ddfft(zup)
 JPNG(file="./FIGS/LenaFt.png" , width=8, height=4)
par(mfrow=c(1,2))
par(mai=c(.5, .4, 0.2, 0.1))
image(zup, col=gr, asp=1, axes=FALSE, ann=FALSE)
show2dfft(Flena, scale=1, col=topo.colors(100) )
dev.off()
```

Figure 10: Lena, famous nerd model.

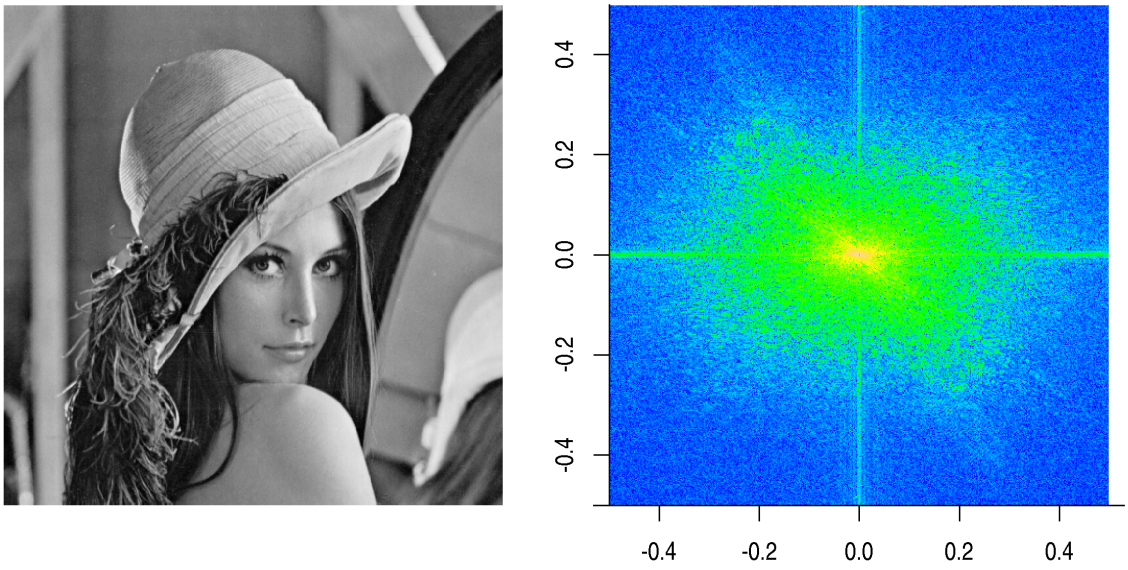

Figure 11: Lena, 2D Ft

5 Mandrake Replaces Lena

Later Lena was replaced by this photo of a Mandrake.

```
mand1  <- read.pnm("./DATA/4.2.03.pnm")
gmand1 = as(mand1, "pixmapGrey")
z = attr(gmand1, 'grey')
mzup = t(z[rev(1:512), ])
#### image(z , col=gr)
JPNG(file="./FIGS/MandFt.png" , width=8, height=4)
par(mfrow=c(1,2))
par(mai=c(.5, .4, 0.2, 0.1))
image(mzup, col=gr, asp=1, axes=FALSE, ann=FALSE)
Fmand = ddfft(mzup)
show2dfft(Fmand , scale=1, col=topo.colors(100) )
dev.off()
```

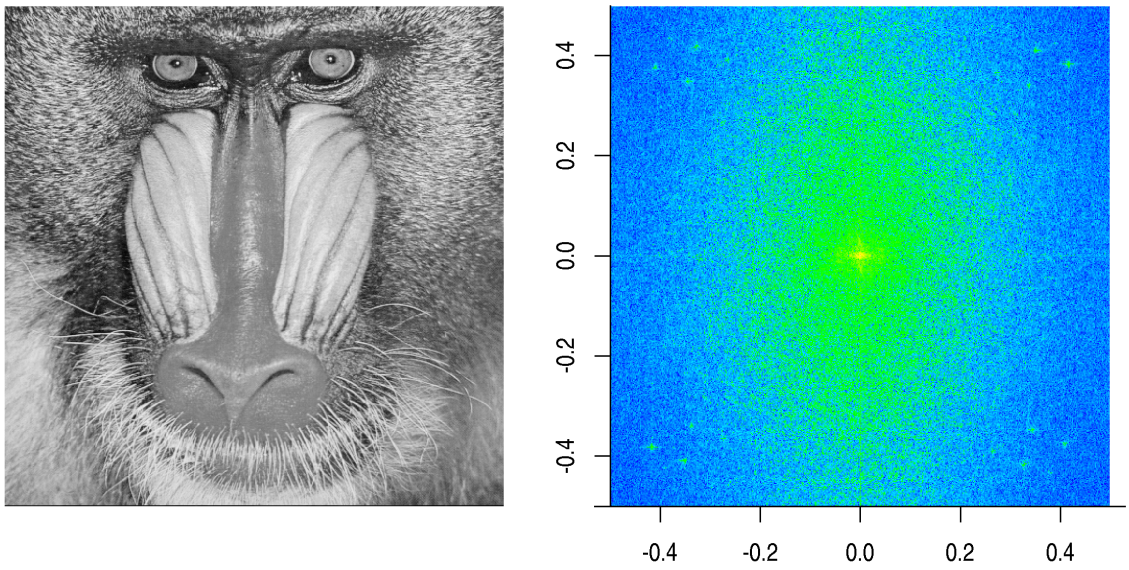

Figure 12: Mandrake, 2D Ft

6 Geological Example: Basin and Range

```
fn ='./DATA/nevadamapthumb.pnm'
###fn ='nevadamapthumb.pnm'
JPNG(file="./FIGS/NEVADA.png" , width=8.980428, height=11.784744)
##   par(pin=c(7.740428, 9.944744))

zz = getimageGRAY(fn, PLOT=FALSE)
 ## par(pin=c(7.740428, 9.944744))
image(x=0:399, y=0:606, zz,  asp=1, col=grpal, axes=FALSE,  ann=FALSE)
dev.off()
```

Lets guess what the pattern is and use the 2DFT to verify or modify our estiamte. We clicked 4 times on the figure. The first two clicks were meant to get the direction of the dominant pattern. The second 2 clicks are perpendicular and are meant to measure the dominant wavelength of the repeating pattern.

```
d=dim(zz)
L=list()
L$x=c(164.651971358,214.672476440,184.660173391,204.668375423)
L$y=c(323.927226022,408.135826308,357.353540639,339.354755845)
DIR = c(L$x[2]-L$x[1], L$y[2]-L$y[1])
dis  = sqrt(  (L$x[4]-L$x[3])^2+  (L$y[4]-L$y[3])^2 )
ang1 = 180*atan2(DIR[2], DIR[1])/pi
ang2 = ang1+90
phiy=dis*cos(pi*ang2/180)
phix=dis*sin(pi*ang2/180)
```

The estimates phiy and phyx are the corresponding wavenumbers in the x and y directions. The variable dis is the distance (in pixels) between ridges in the geology.

```
d=dim(zz)
g = meshgrid(1:d[1], 1:d[2])
JPNG(file="./FIGS/Nev2.png" , width=12, height=8)
par(mfrow=c(1,2))
image(x=1:d[1], y=1:d[2], zz, col=grpal, asp=1, axes=FALSE, ann=FALSE)
arrows(L$x[1], L$y[1], L$x[2], L$y[2], col='red', lwd=2)
arrows(L$x[3], L$y[3], L$x[4], L$y[4], col='blue', lwd=2)
```

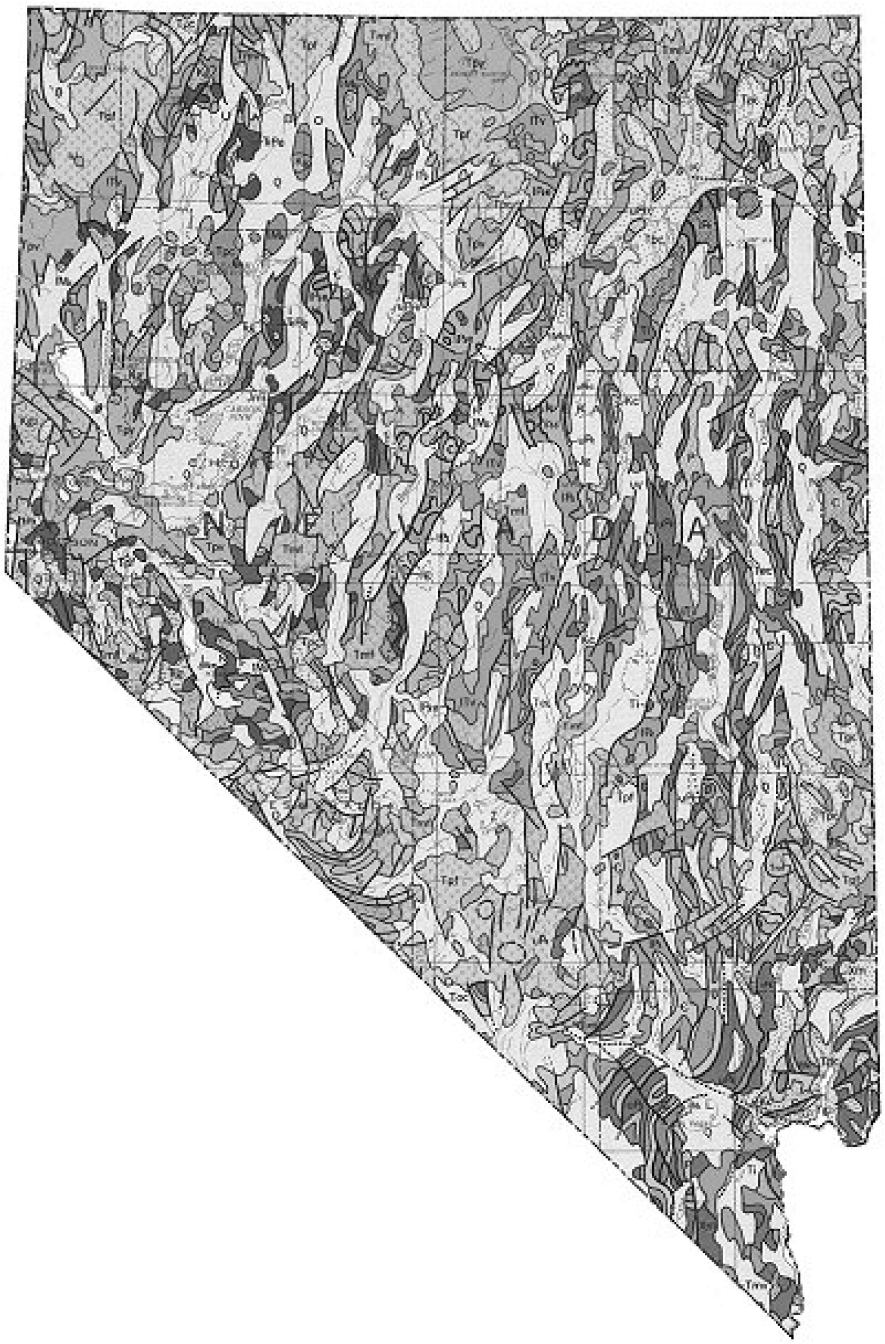

Figure 13: Nevada Geology, grey scale

```
kx = 1/(2*phix)
ky = 1/(2*phiy)
Arot =  1*(cos(2*pi*(g$y*ky) + 2*pi*(g$x*kx) ) )
image(x=1:d[1], y=1:d[2], t(Arot), col=grpal, asp=1, axes=FALSE, ann=FALSE)
arrows(L$x[1], L$y[1], L$x[2], L$y[2], col='red', lwd=2)
arrows(L$x[3], L$y[3], L$x[4], L$y[4], col='blue', lwd=2)
dev.off()
```

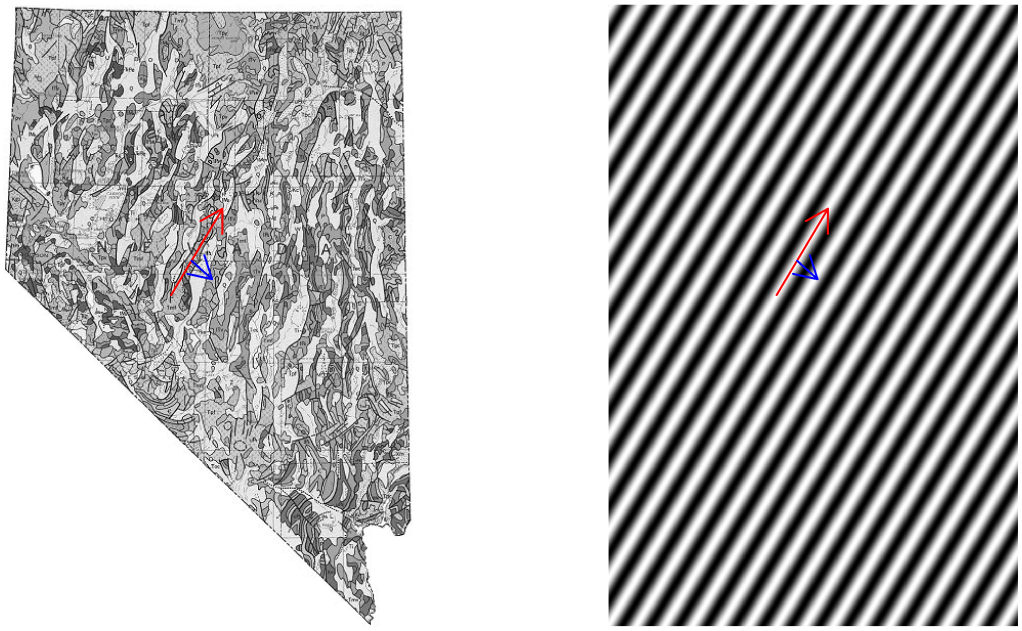

Figure 14: Nevada with clicks shown

to see how these match up, lets combine the images:

```
JPNG(file="./FIGS/NEV3.png" , width=8, height=12)
nz = zz+0.2*t(Arot)
image(x=1:d[1], y=1:d[2], nz, col=grpal, asp=1, axes=FALSE, ann=FALSE)
dev.off()
```

What do the respective FT's look like?

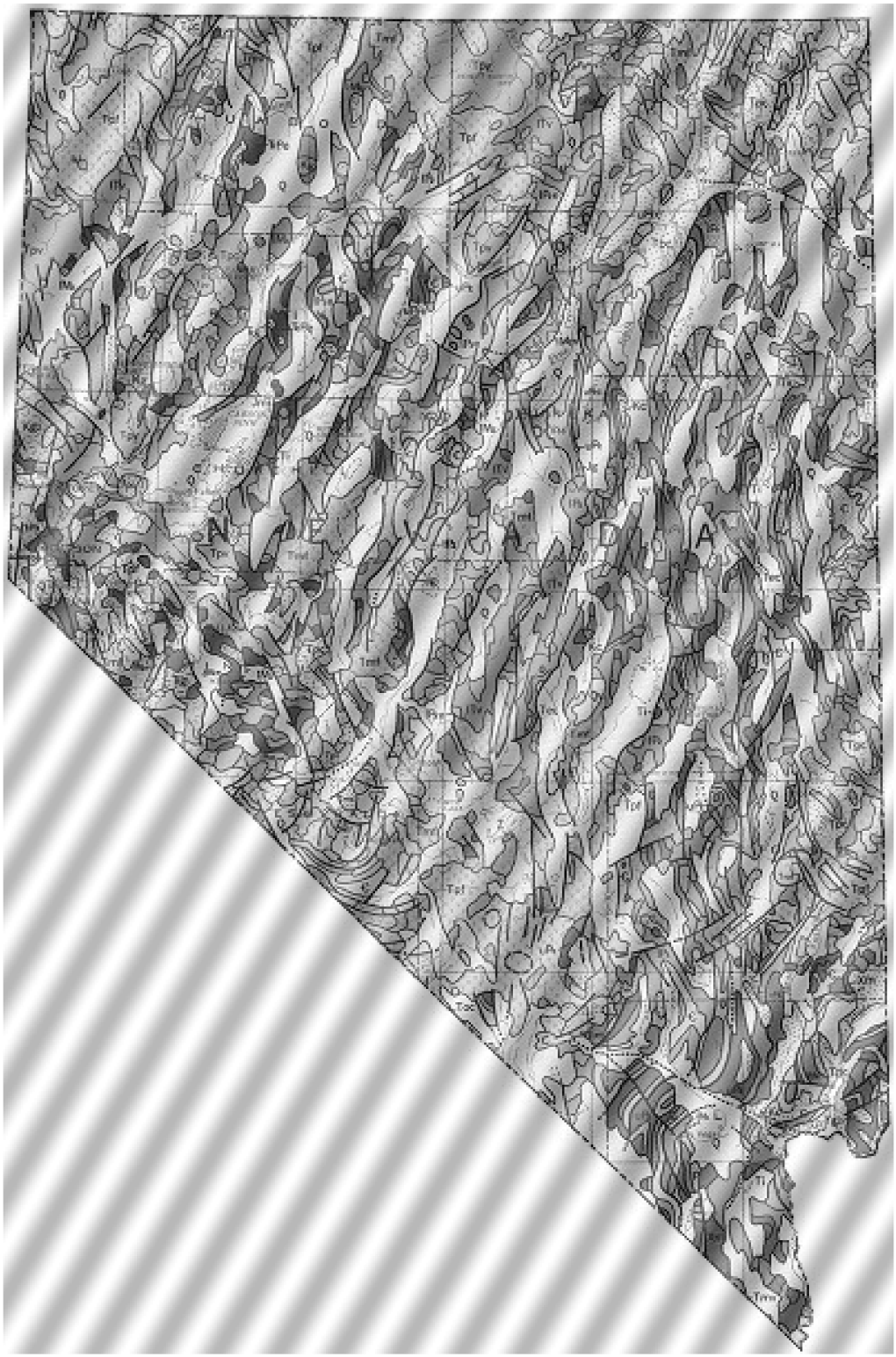

Figure 15: Combined corrugation and Geology

```
#####################  How do the respective FFT's look like?

JPNG(file="./FIGS/NEVft1.png" , width=12, height=8)
Fsym = ddfft( t(Arot) )
Fz = ddfft(zz )
par(mfrow=c(1,2))
image(x=1:d[1], y=1:d[2],zz, col=grpal, asp=1, axes=FALSE, ann=FALSE)
title("Nevada geology greyscale")
sfs =  show2dfft(Fz)
##   image(log(1+ Mod(Fz)), col=grpal, asp=1, axes=FALSE, ann=FALSE)
title("2Dfft log(1+mod) scale")
abline(v=0, col=rgb(1,.86, .86) )
abline(h=0, col=rgb(1,.86, .86))
dev.off()
```

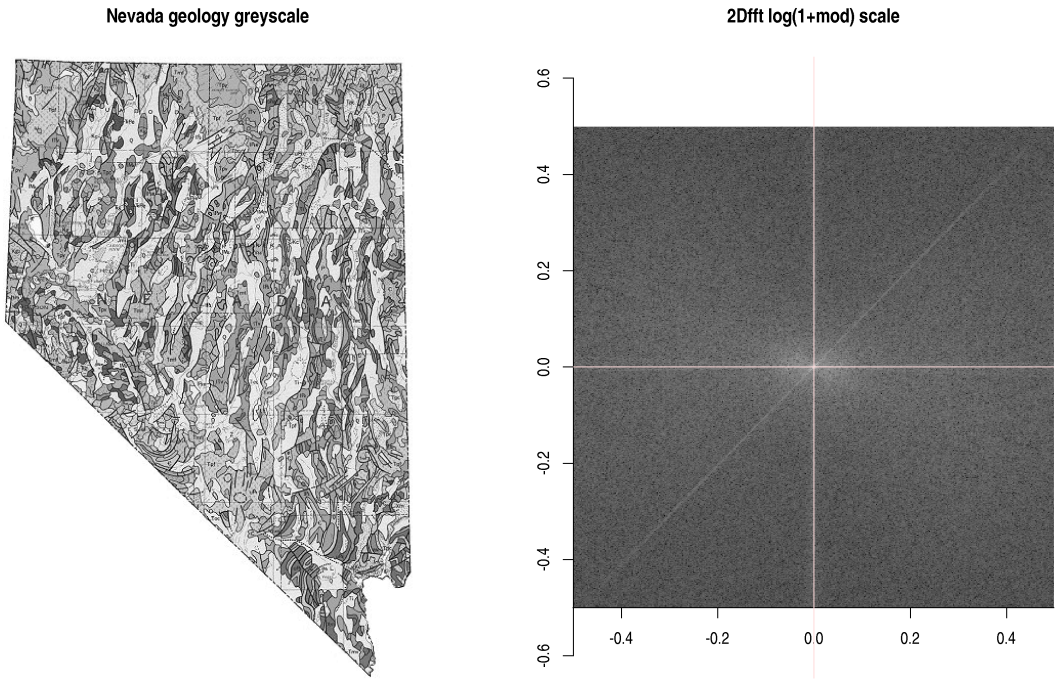

Figure 16: 2D-FT of Nevada Geology

The FT of the Nevada geology seems pretty noisy, as one might expect. We note that there are parts of the image (like the lower left) that have no information. the sharp line seperating these two further introduce bad noise in the FT - see the diagonal line running from bottom left to upper right - that is a result of the border between Nevada and California. So, lets take a sample of the Nevada image and work on just the subset.

```
JPNG(file="./FIGS/nevsamp1.png" , width=8, height=12)
LocSamp = list(x=88, y=266)
image(x=1:d[1], y=1:d[2],zz, col=grpal, asp=1, axes=FALSE, ann=FALSE)
rect(floor(LocSamp$x[1]), floor(LocSamp$y[1]), floor(LocSamp$x[1])+255, floor(LocSamp$y
dev.off()
```

```
JPNG(file="./FIGS/NEVADroof.png" , width=15, height=6)
par(mfrow=c(1,3))
sampX =seq(from=floor(LocSamp$x[1]), length=256, by=1)
sampY =  seq(from=floor(LocSamp$y[1]), length=256, by=1)
SAMP = zz[ sampX, sampY]
SAMP = SAMP-mean(SAMP)
image(x=sampX, y=sampY, SAMP, col=grpal, asp=1, axes=FALSE, ann=FALSE)
roof = t(Arot)
Samproof =  roof[ sampX, sampY]
image(x=sampX, y=sampY, Samproof, col=grpal, asp=1, axes=FALSE, ann=FALSE)
newroof = SAMP+0.2*Samproof
image(x=sampX, y=sampY, newroof, col=grpal, asp=1, axes=FALSE, ann=FALSE)
arrows(L$x[1], L$y[1], L$x[2], L$y[2], col='red', lwd=2)
arrows(L$x[3], L$y[3], L$x[4], L$y[4], col='blue', lwd=2)
dev.off()
```

```
JPNG(file="./FIGS/NEVADfft.png" , width=12, height=12)
Fsam  = ddfft(SAMP)
Froof  = ddfft(newroof)
par(mfrow=c(2,2))
image(SAMP, col=grpal, asp=1, axes=FALSE, ann=FALSE)
sf = show2dfft(Fsam)
title("Log(1+Mod)scaling")
sf =  show2dfft(Fsam, scale=3)
## image(sqrt( Mod(Fsam)), col=grpal, asp=1, axes=FALSE, ann=FALSE)
title("sqrt scaling")
sf =  show2dfft(Fsam, scale=0)
title("no scaling")
abline(h=0, v=0, col='blue')
dev.off()
```

```
JPNG(file="./FIGS/nevfft2.png" , width=12, height=8)
par(mfrow=c(1,2))
```

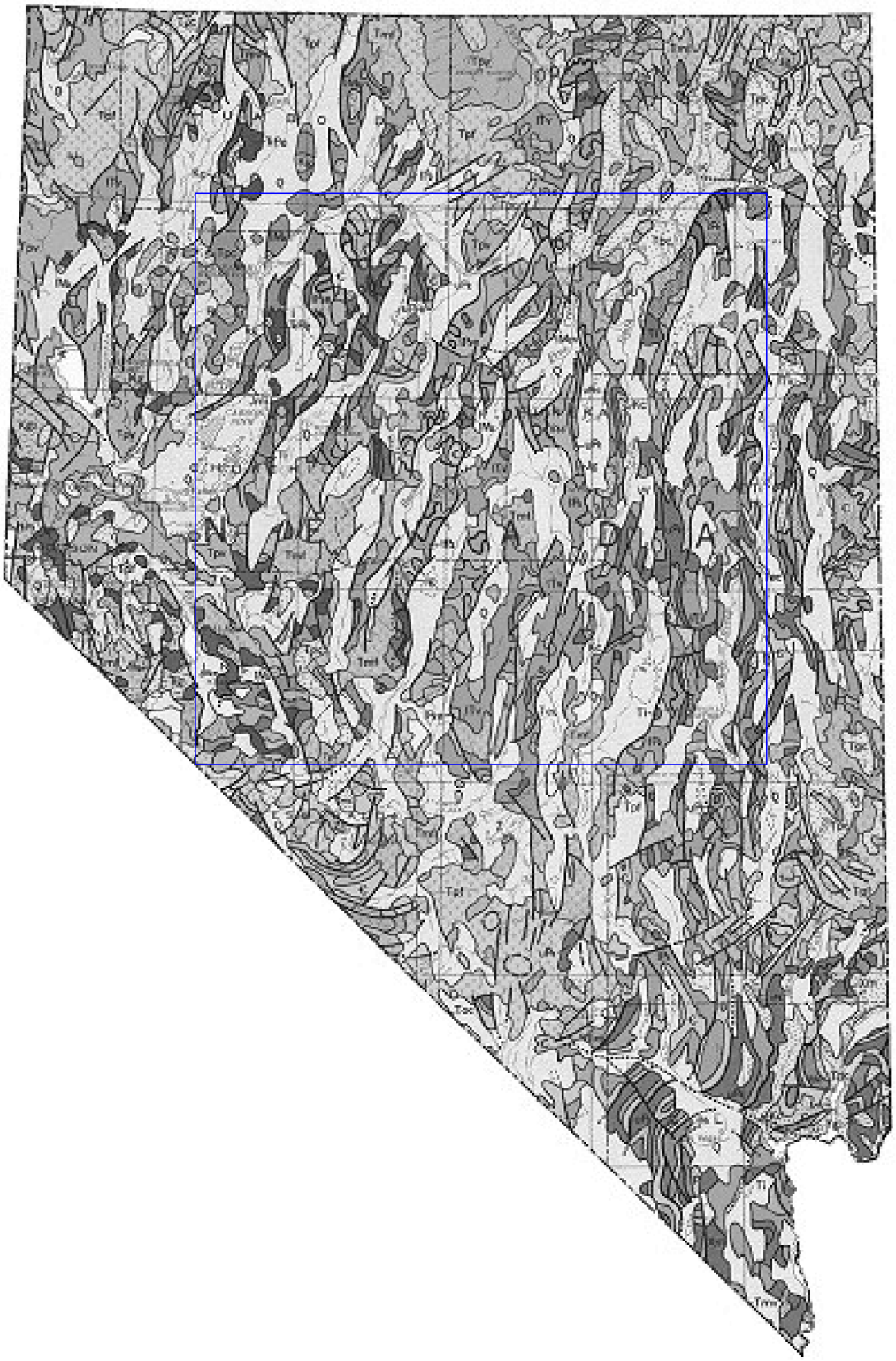

Figure 17: Sample of Nevada geology

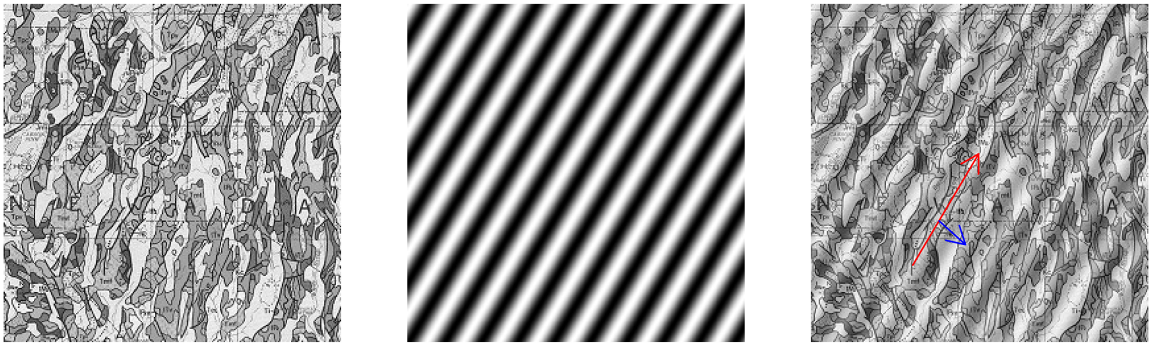

Figure 18: Nevada Corrugated roof

```
###############
sf = show2dfft(Froof, scale=0)
kgrid = meshgrid(sf$kx, sf$ky)
maxroof = which( Mod(Froof )>  0.90*max(Mod(Froof ))  )
points(t(kgrid$x)[maxroof], t(kgrid$y)[maxroof], col='gold')
###############
sf = show2dfft(Fsam, scale=3)
maxsamp = which( Mod(Fsam)>  0.90*max(Mod(Fsam ))  )
kgrid = meshgrid(sf$kx, sf$ky)
points(t(kgrid$x)[maxsamp], t(kgrid$y)[maxsamp], col='gold')
points(t(kgrid$x)[maxroof], t(kgrid$y)[maxroof], col='cyan')
dev.off()
```

close up of image fft

```
JPNG(file="./FIGS/ftzoom1.png" , width=12, height=8)
par(mfrow=c(1,2))
image(x=sampX, y=sampY, SAMP, col=grpal, asp=1, axes=FALSE, ann=FALSE)
xlim = c(-.2, .2)
image(x=sf$kx, y=sf$ky , log( 1+Mod(Fsam)), col=grpal, asp=1, xlim=xlim, ylim=xlim, axe
abline(v=0, h=0, col=rgb(1, .8, .8) )
title("Detail, 2D-fft")
maxsamp = which( Mod(Fsam)>  0.90*max(Mod(Fsam ))  )
points(t(kgrid$x)[maxsamp], t(kgrid$y)[maxsamp], col='red')
points(t(kgrid$x)[maxroof], t(kgrid$y)[maxroof], col='blue')
dev.off()
```

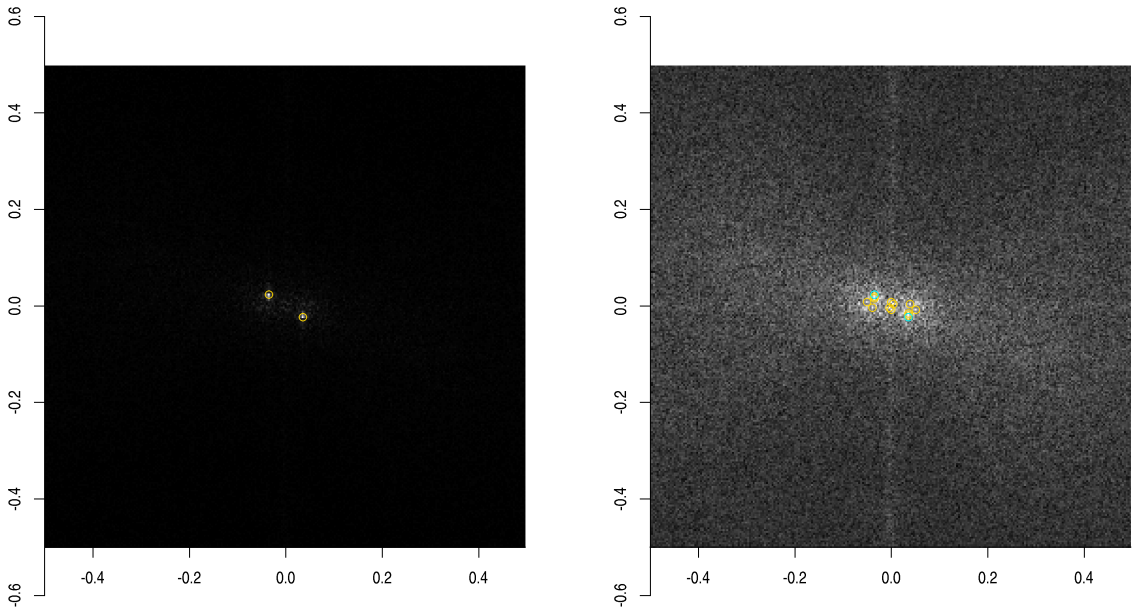

Figure 19: FT of close up region

Note in figure **??** the peak from the corrugated roof (blue dot) - estimated from eye-balling the original grey shade plot, is close to, but does not exactly overlap, the peaks of the true geological FT (blue dots). This suggests that we can improve the estmate by shifting the location of the wavenumber estimates slightly.

```
kloc=list()
kloc$x=c(0.0766325316100,0.0350366057743)
kloc$y=c(-0.0371227578646,-0.0157096306866)
ky = 0.5/ kloc$y[2]
kx = 0.5/ kloc$x[2]
kayx = round(0.5/ kloc$x[2])
kayy = round(0.5/kloc$y[2])
jx = 0.5/kayx
jy = 0.5/kayy
kx = 1/(2*kayx)
ky = 1/(2*kayy)
Arot2 =  1*(cos(2*pi*(g$y*ky) + 2*pi*(g$x*kx) ) )
nz2 = zz+0.2*t(Arot2)
```

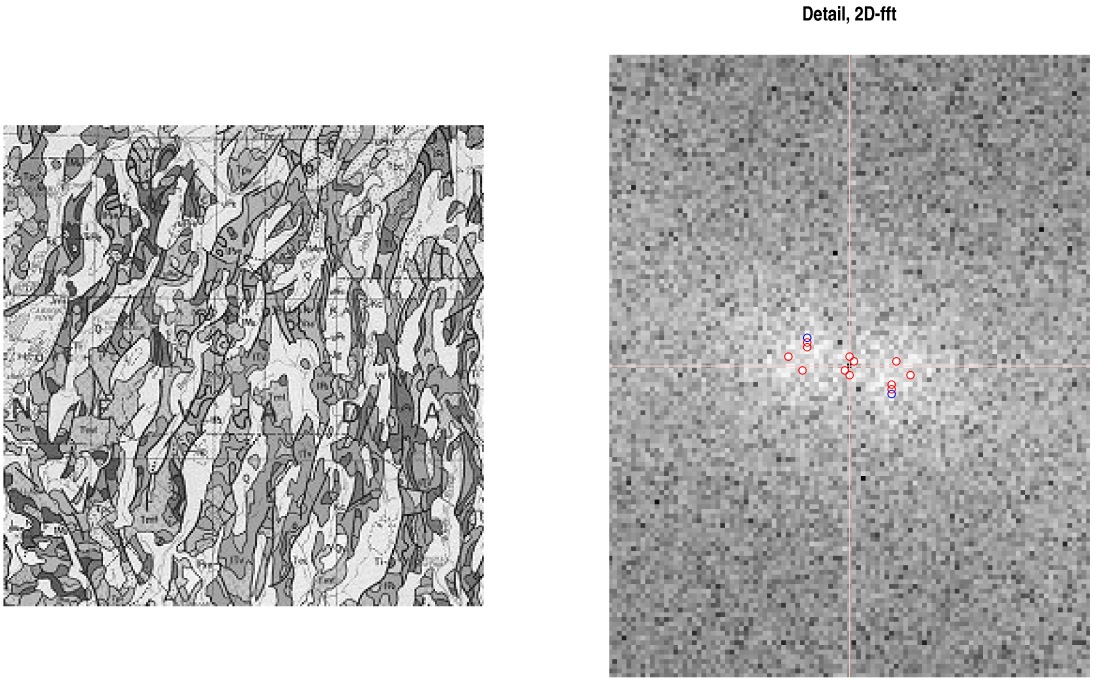

Figure 20: Detail of FT of Nevada Geology. Red locations are whare the sepctrum is high in the on the 2dFT of the geological grey map. The 2 blue dots show where the peaks of the corrugated roof lie. Note that they are close but do not overlap the red dots exactly.

```
JPNG(file="./FIGS/ftzoom2.png" , width=12, height=8)
par(mfrow=c(1,1))
xlim = c(-.1, .1)
image(x=sf$kx, y=sf$ky, log( 1+Mod(Fsam)), col=grpal, asp=1, xlim=xlim, ylim=xlim, axes
abline(v=0, h=0, col=rgb(1, .8, .8) )
title("Detail, 2D-fft")
maxsamp = which( Mod(Fsam)>  0.90*max(Mod(Fsam ))  )
points(t(kgrid$x)[maxsamp], t(kgrid$y)[maxsamp], col='red')
points(t(kgrid$x)[maxroof], t(kgrid$y)[maxroof], col='blue', pch=2)
arrows(kloc$x[1] ,kloc$y[1] , kloc$x[2] ,kloc$y[2], lwd=2)
points(kloc$x[2] ,kloc$y[2], col='purple', pch=8)
dev.off()
```

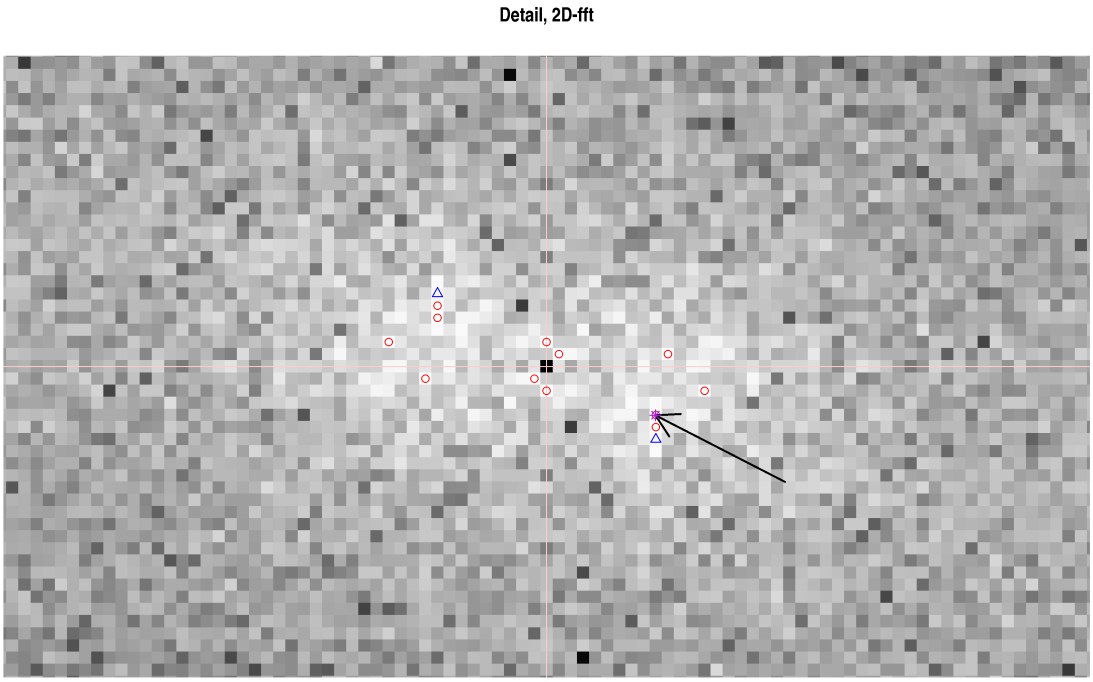

Figure 21: Zoom in of Fourier Transform of Nevada

7 EXAMPLE

In this example we present a case that may represent a real life situation. First we set up some of the required libraries and codes. This uses functions in the KFR package.

```
load("./DATA/z3.RDATA")
library(splancs)
library(RPMG)
source("/home/lees/R_PAX/KFR/R/getKXKY.R")
source("/home/lees/R_PAX/KFR/R/filtpoly2D.R")
source("/home/lees/R_PAX/KFR/R/MY2DFILT.R")
```

Plot the image.

```
gr = grey(0:100/100)
x = attr(z3, "x")
y = attr(z3, "y")
JPNG(file="./FIGS/RawIMG_A.png" , width=12, height=8)
image(x=x, y=y, z=z3)
dev.off()
```

The image is contaminated by noise that seems to have a single frequency. We can remove the noise by filtering in the wavenumber (frequency) domain. In this case we can use some functions alreadyt established iun package KFR.

```
FAB = fft(z3)
PARMS = getKXKY(FAB)
JPNG(file="./FIGS/FTRawIMG_A.png" , width=12, height=8)
par(mfrow=c(1,1))
 image(x=PARMS$kx, y =PARMS$ky,     Mod(FAB), col=rainbow(100), asp=1)
dev.off()
```

And then we can design a filter in the frequency domain by forming a polynomial and zeroing out the frequencies in those bands.

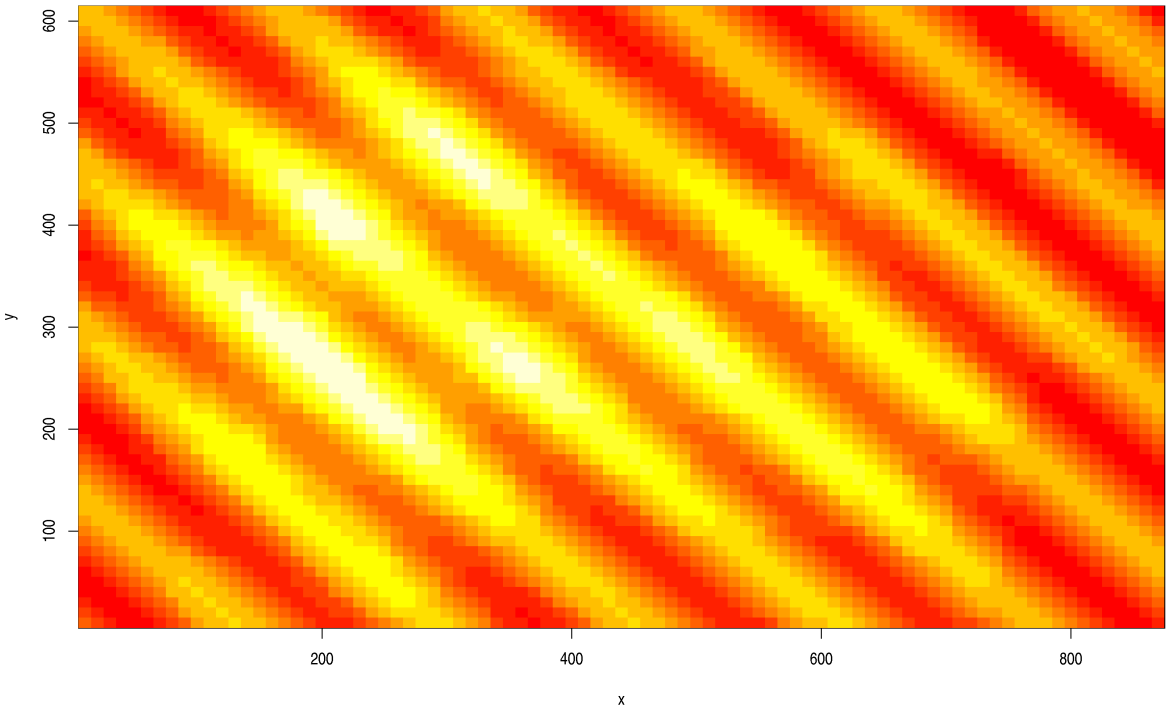

Figure 22: Image with regular single wavenumber noise

```
###    AFL1 = filtpoly2D(FAB, col=rainbow(100), scale=0)

AFL1 = list()
AFL1$FL=list()
AFL1$FL$x=c(-0.461698178264,-0.505493718281,-0.503060632724,-0.466564349377,-0.31084687
AFL1$FL$y=c(-0.464331402238,-0.420260418573,-0.310082959412,-0.209699052177,-0.21214744
JPNG(file="./FIGS/DesignFilter_A.png" , width=12, height=8)
AFL2 = filtpoly2D(FAB, col=rainbow(100), FL =AFL1$FL ,  scale=0)
points(AFL1$FL)
text(AFL1$FL, labels=1:length(AFL1$FL$x) , pos=3, xpd=TRUE)
## tLOC = locator()

tLOC=list()
tLOC$x=c(-0.2013580237220, 0.0443836174814)
tLOC$y=c(-0.251321647860,-0.177870008419)
arrows(tLOC$x[2] , tLOC$y[2] ,tLOC$x[1] ,tLOC$y[1] )
text(tLOC$x[2] ,tLOC$y[2], labels="Click to create a polygon for filter", pos=4, xpd=TR
dev.off()
```

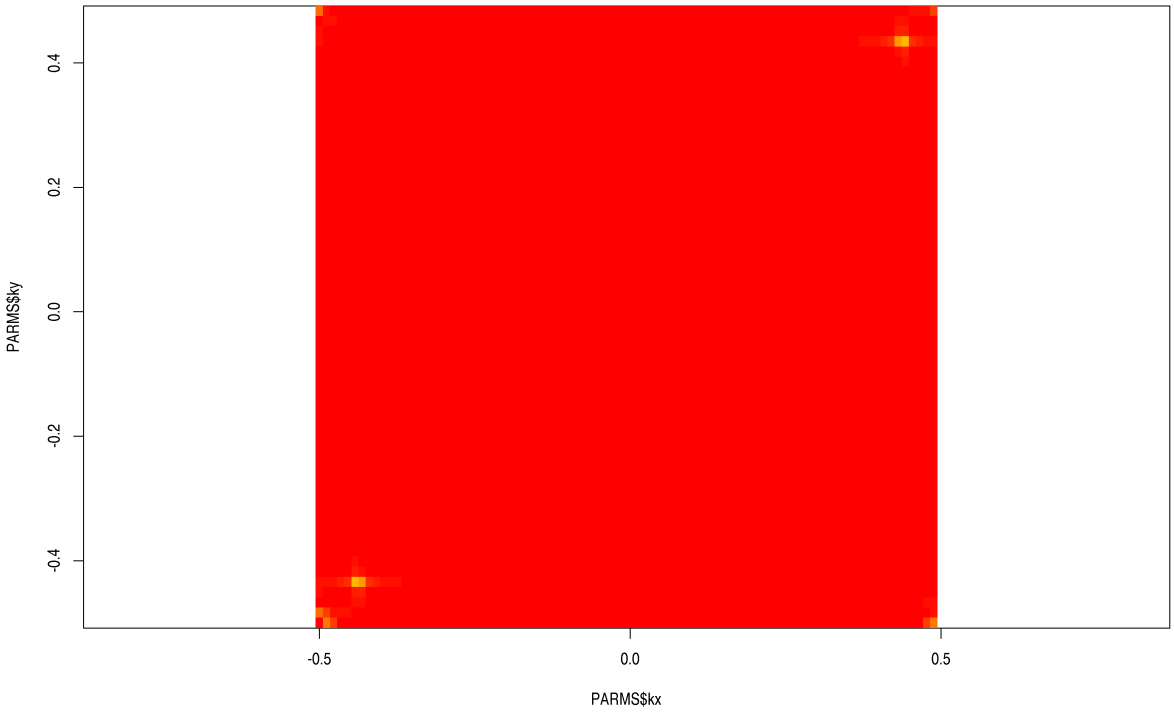

Figure 23: FFT of raw image

```
Fsmo = MY2DFILT(FAB  , PARMS$kx, PARMS$ky, AFL1$FL)
ZREC = Re(Fsmo)/(2*length(z3))
```

```
JPNG(file="./FIGS/filteredIMG_A.png" , width=12, height=8)
image(ZREC, col=gr, asp=1)
contour(ZREC,  add=TRUE)
title("filtered image")
dev.off()
```

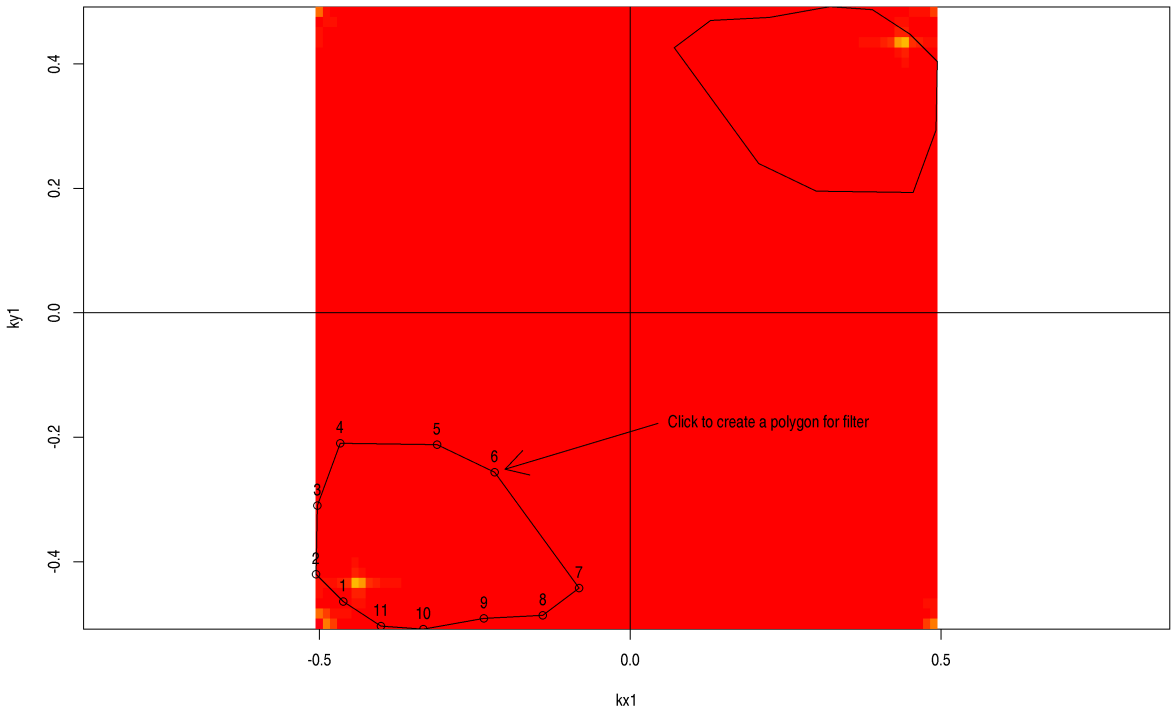

Figure 24: Design a filter in the wavenumber domain by clicking on the frequency representation and forming a polynomial that will be rejected.

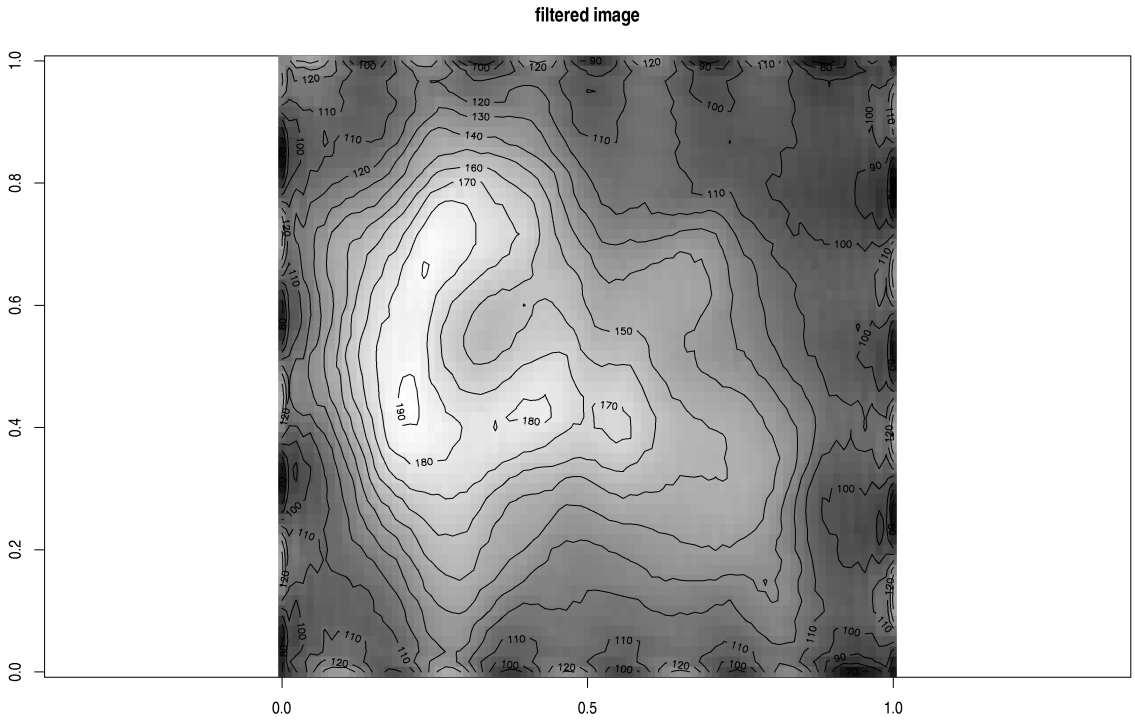

Figure 25: