Sometimes you have a scene with large decorrelated areas

In these cases, SNAPHU can take ~forever

Setting the correlation threshold higher won’t help (why?)

We can use interpolation to speed the computation and improve the results
We can think of the correlated pixels as a sparse dataset.

Nearest neighbor interpolation preserves the topology of any loops containing residues.

This means the unwrapped, masked result should be the same, whether or not we interpolate first.

Reference:
x, y bounds

minx = `grdinfo -C $in.grd |cut -f 2` 
maxx = `grdinfo -C $in.grd |cut -f 3` 
nx = `grdinfo -C $in.grd |cut -f 10` 
boundsx = "$minx $maxx"

miny = `grdinfo -C $in.grd |cut -f 4` 
maxy = `grdinfo -C $in.grd |cut -f 5` 
yy = `grdinfo -C $in.grd |cut -f 11` 
- for some reason we have to reverse these two 
boundsy = "$maxy $miny"

# first convert to ascii
xyz $in.grd -S -V > $in.gmt

# run gdal, then convert back to grd
rdgrid -of GTiff -txe $boundsx -tye $boundsy -outsize $nx $ny -l $in -a nearest $in.gmt $out.tiff
trans -of GMT -ot Float32 $out.tiff $out.grd

# fix the grd header metadata
grdedit $out.grd -T 
#(note: must be pixel node registration for snaphu)
grdedit $out.grd -R $minx/$maxx/$miny/$maxy
Mask, then interpolate

Masked

Filled
Case study: Imperial Valley, CA

Standard unwrapping

Running time: 3h 43m 33sec

ENVI_D084_2005268_2006078
Case study: Imperial Valley, CA

Interpolation unwrapping

Running time: 4m 9sec (54x speedup!)