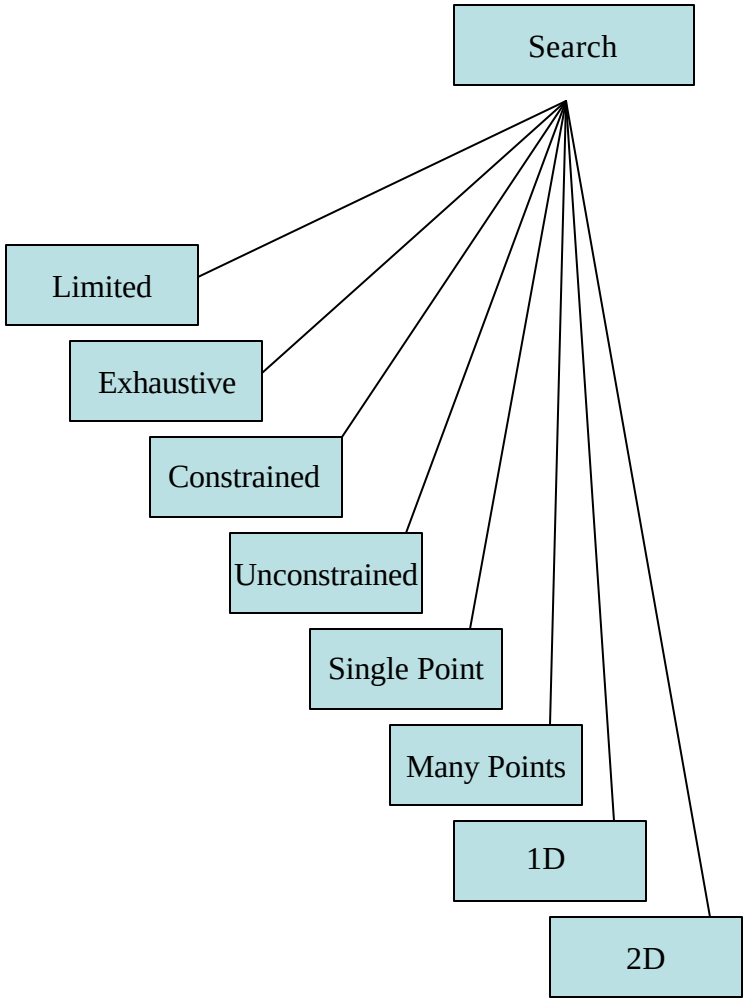
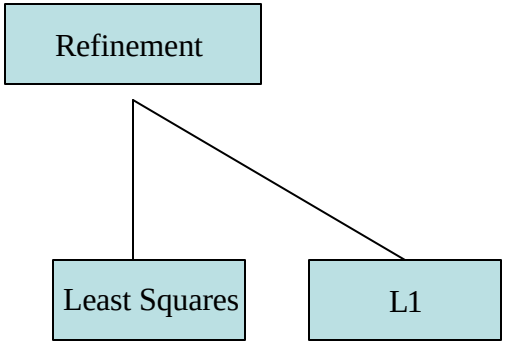
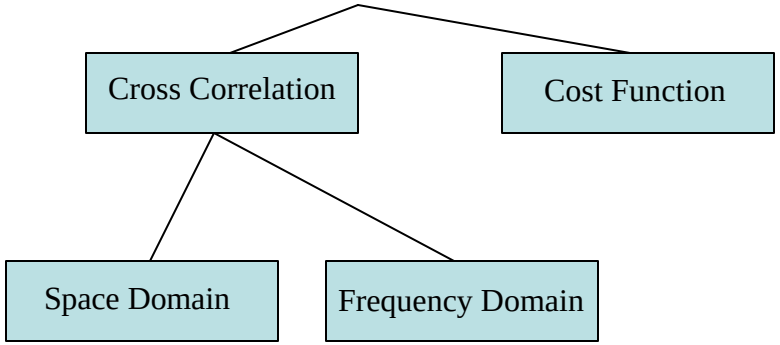
Matching: Signal & Feature





Similarity Metric



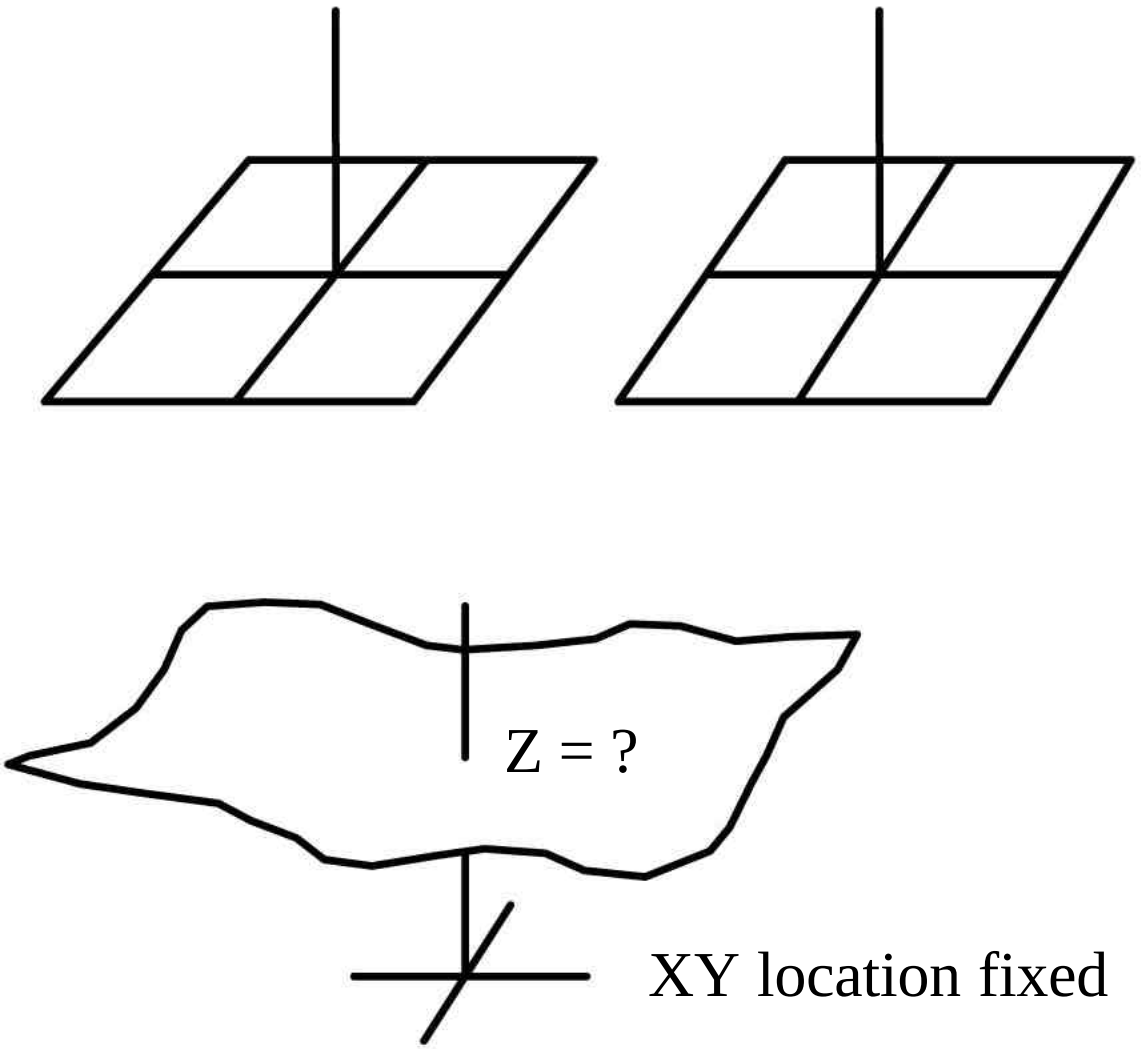
One Possible Means of Constraining (via knowledge gained from photogrammetry) a Search:

Rather than selecting a point in one image and asking the question, where is the corresponding or conjugate point in the second image?, ask the following question: Given a *planimetric location* in object space (on the ground), what is the elevation that gives the best match between the pair of sub-image patches, obtained by projecting the candidate points into the images?

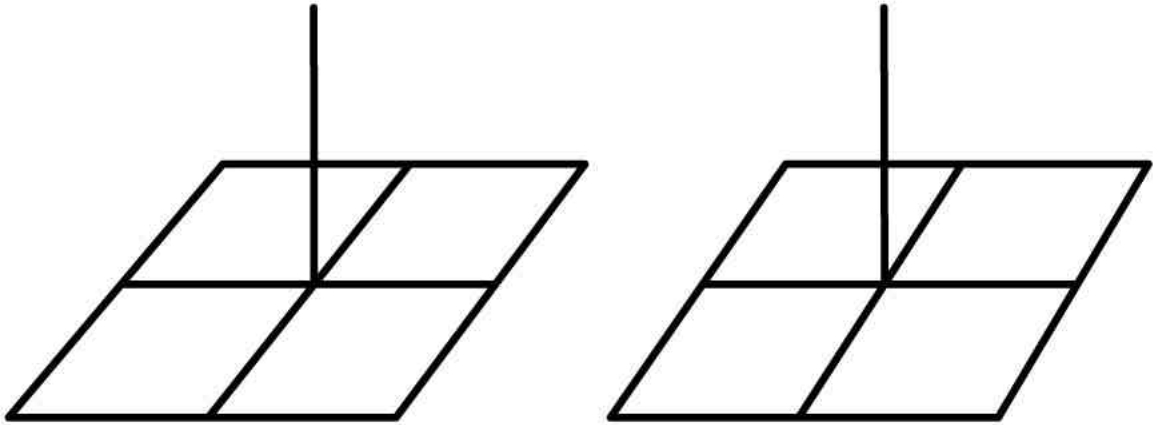
This is referred to as the Vertical Line Locus or VLL method. First suggested by Maurice Geyer ~ early 1980's

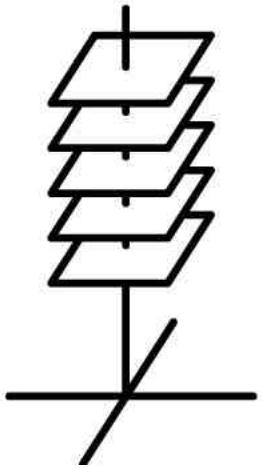
Key advantages of this method: (1) it is constrained by knowledge from photogrammetry (2D -> 1D), (2) it works with *any* sensor model – not just frame. (there are many variations on this method), and (3) you can produce a *regular* grid (i.e. national database)

VLL-1



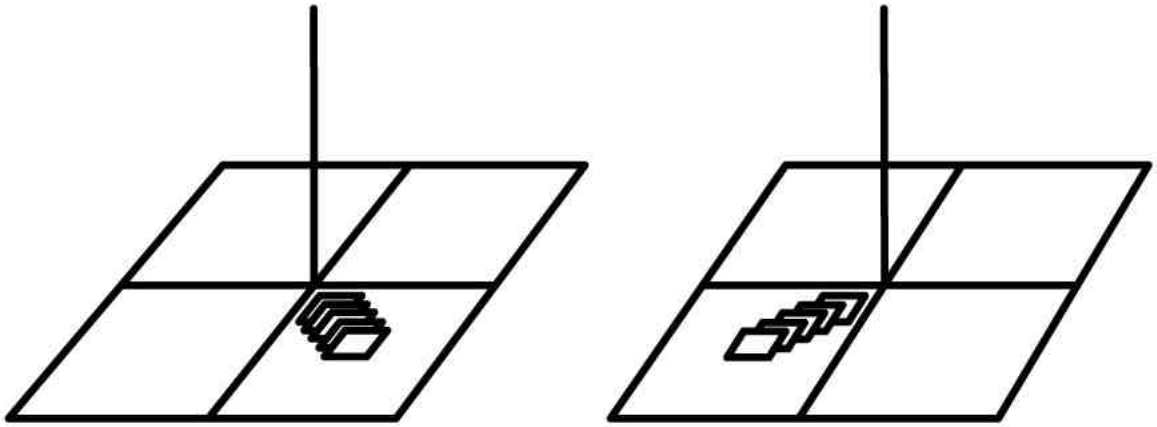
VLL-1



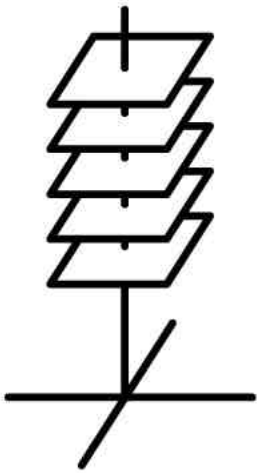


Bracket the estimated terrain elevation (above and below) by a sequence of “object patches” all at the same XY location, but at different elevations

VLL - 3



This method can function as either a search or refinement method depending on the spacing and extend of the object patches, and on the pyramid level of the imagery



Project each of these object patches into each of the two images – evaluate the similarity of each pair, select that pair with the best match – that elevation corresponds to the terrain elevation at that point

Most Common Similarity Measure for Signal Matching: Normalized Cross Correlation (Discrete Version)

$$C_{uv} = \frac{\sum (u_i - \bar{u})(v_i - \bar{v})}{\left[\sum (u_i - \bar{u})^2 \sum (v_i - \bar{v})^2\right]^{1/2}}$$

$$\bar{u} = \frac{\sum u_i}{N}$$

$$\bar{v} = \frac{\sum v_i}{N}$$

$$\mathrm{N} : \text{number of elements in u \& v}$$

Key insight to make the method stable and efficient: do it in a *hierarchical* approach. That is start with downsampled imagery (top of pyramid) and a coarse point density and a coarse elevation spacing of the object planes. Then progressively move down the image pyramid, densify the point grid, densify the elevation search, and reduce the extent of the elevation search.

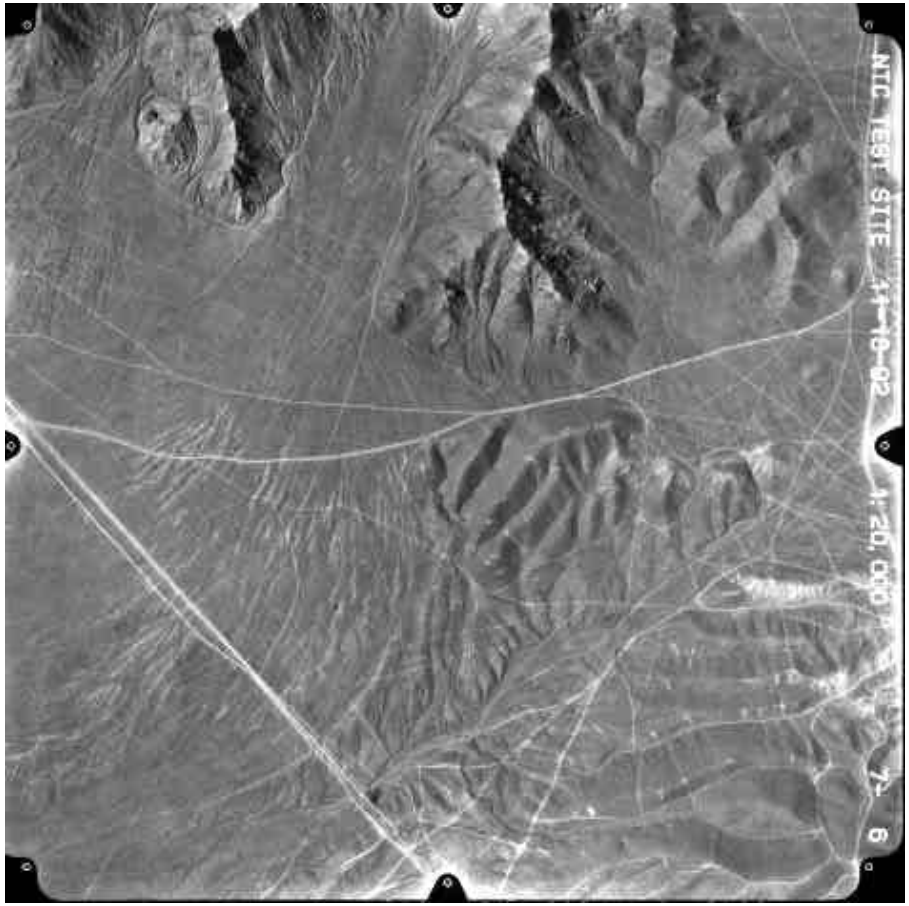
Each subsequent step starts with a coarse terrain model from the prior step and refines in both in density and in accuracy.

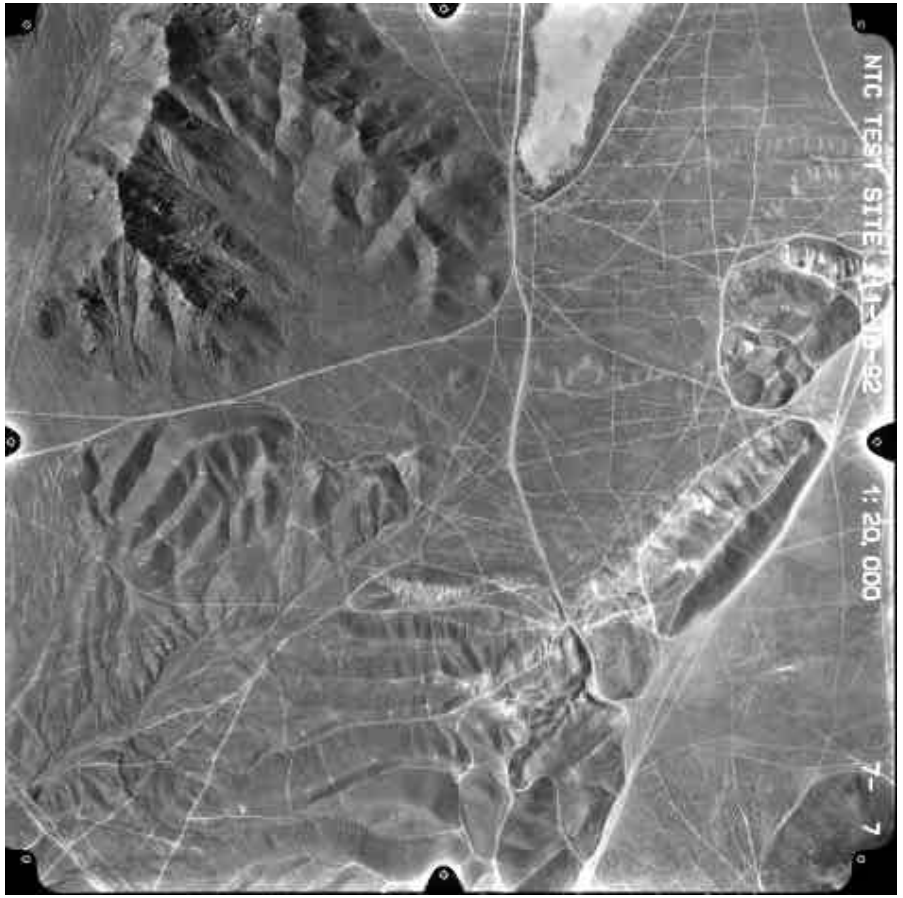
Image Pyramid – Successive 2x Downsampling – for Hierarchical Processing

Start at the “top” get coarse results from low resolution imagery quickly – then drop down a level and use prior level results as a starting point and limit search to refine – continue to the bottom making last refinement step using the highest spatial resolution imagery.

1:20000 NTC Model – benign terrain for DEM generation

Small/medium scale, minimal vegetation

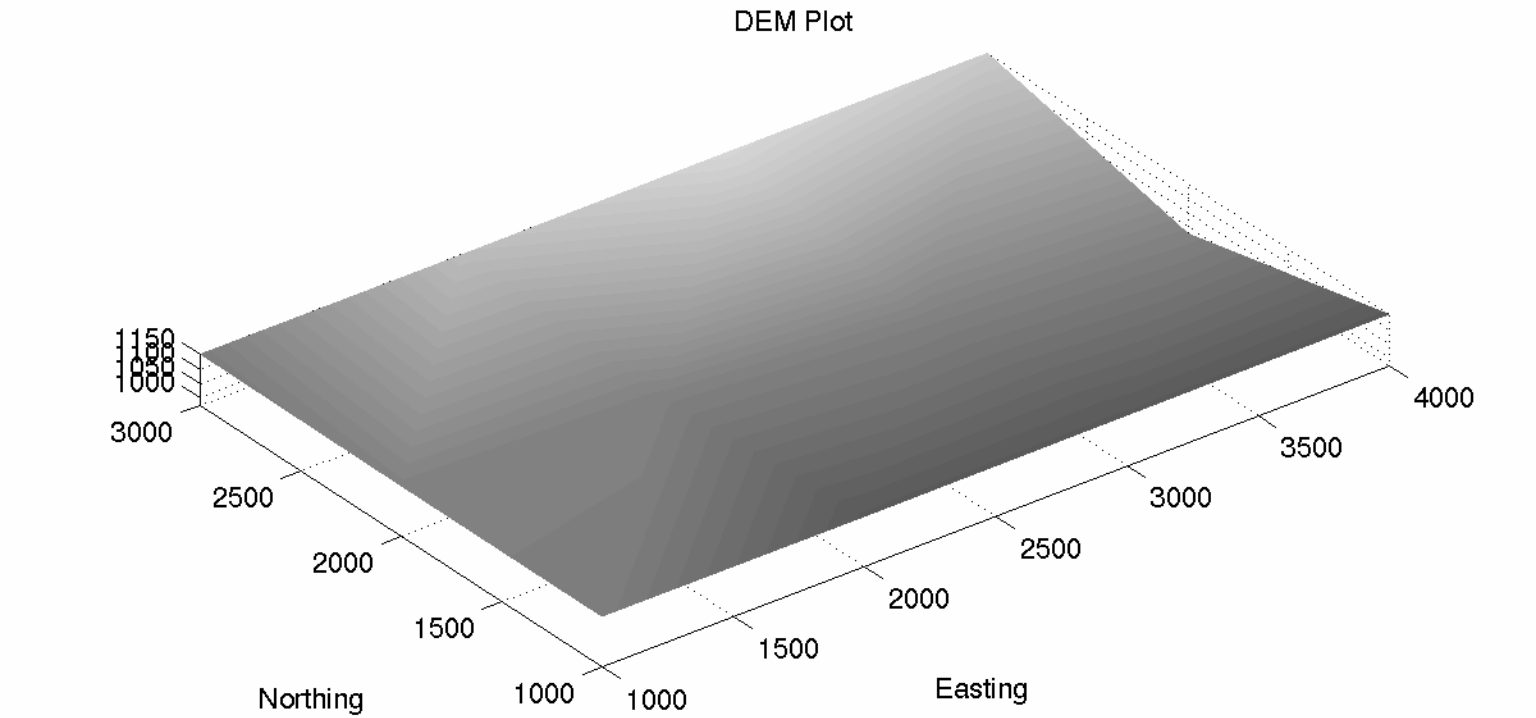




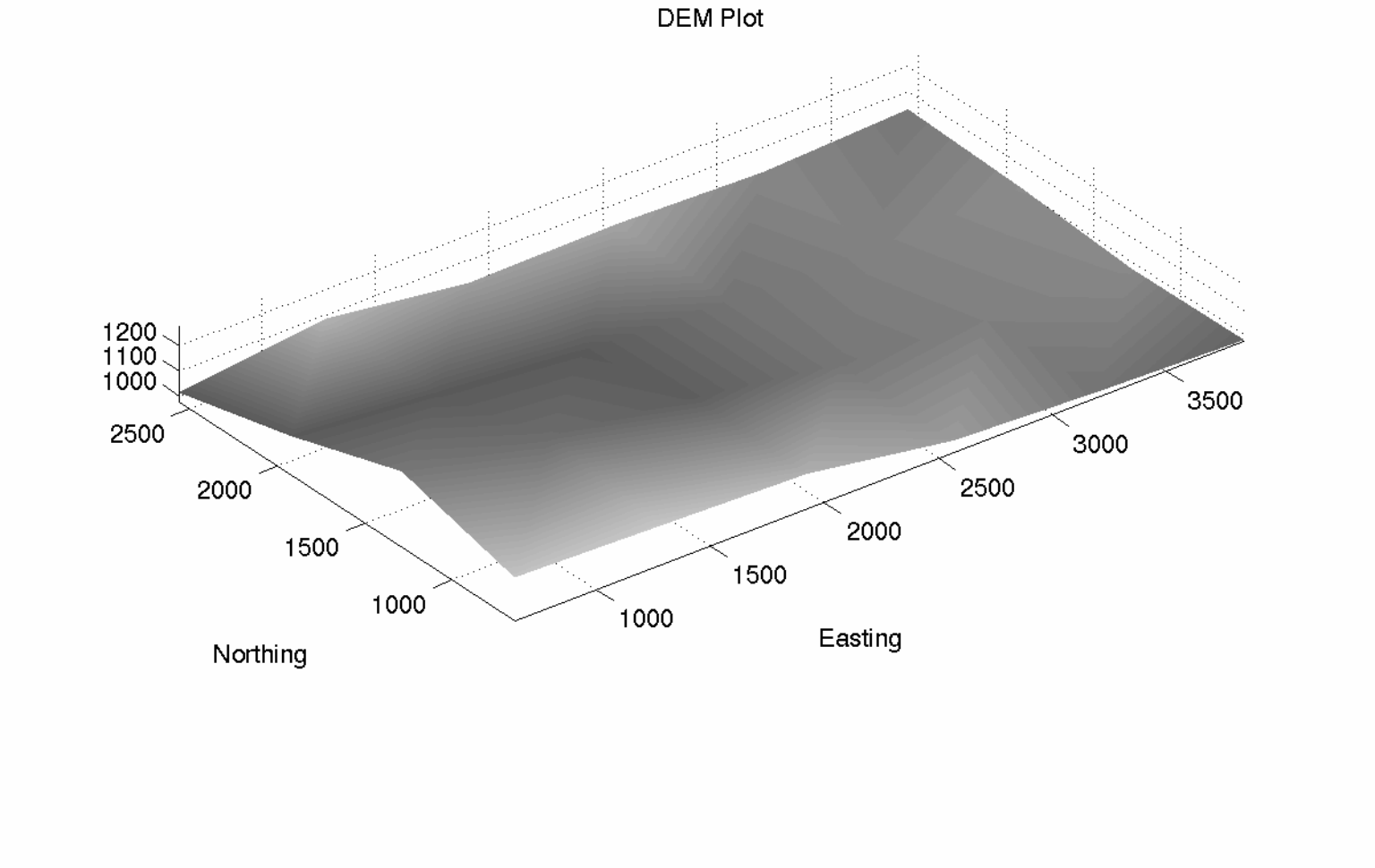
Following are a series of matching results from a hierarchical strategy, starting at the coarse (downsampled) end of the image pyramid, with correspondingly coarse post spacing, vertical interval, vertical range, etc. We then progress systematically to each higher resolution level of the pyramid, decreasing the post spacing, the vertical interval, the vertical range, etc. by a factor of 2, at each new level. The elevation results at each level become the starting point or the initial approximation for each point at the next level. Thus, other than the first level, where we may start with the mean terrain height everywhere, the processing at each subsequent level is not so much searching as refining. We just progressively refine the coarse estimates at the early levels to a finer and finer estimate at the last (higher resolution) levels.

The following series of matching results start at the 64x downsampled level, and proceed through the 32x, 16x, 8x, 4x, 2x, and finally the 1x (full resolution) imagery. You can see the progressive emergence of fine terrain detail as we reveal the higher levels of image detail.

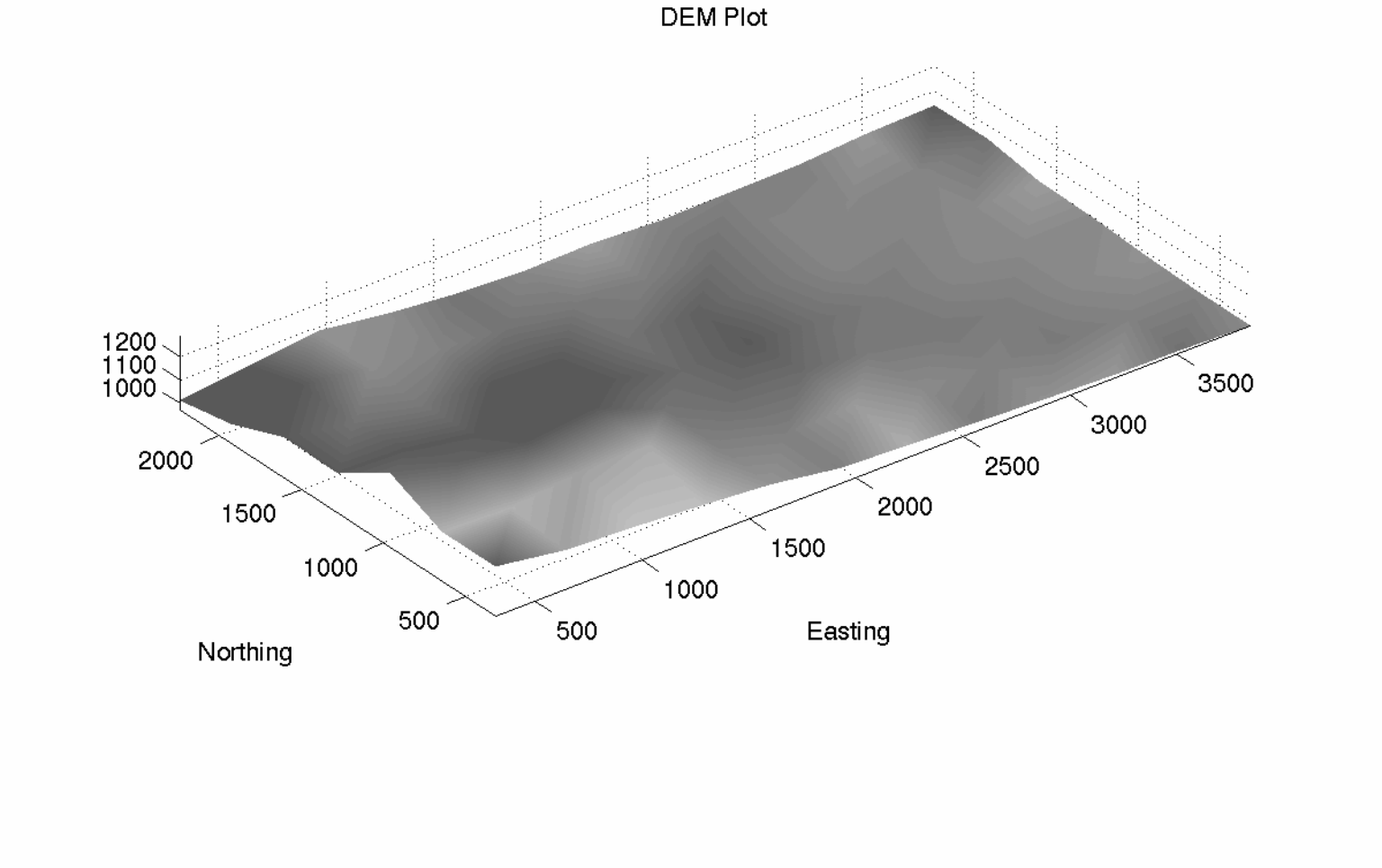
64x


DEM Plot
1150
3000
2500
2000
1500
1000
Northing
1000
1500
2000
2500
3000
3500
4000
Easting

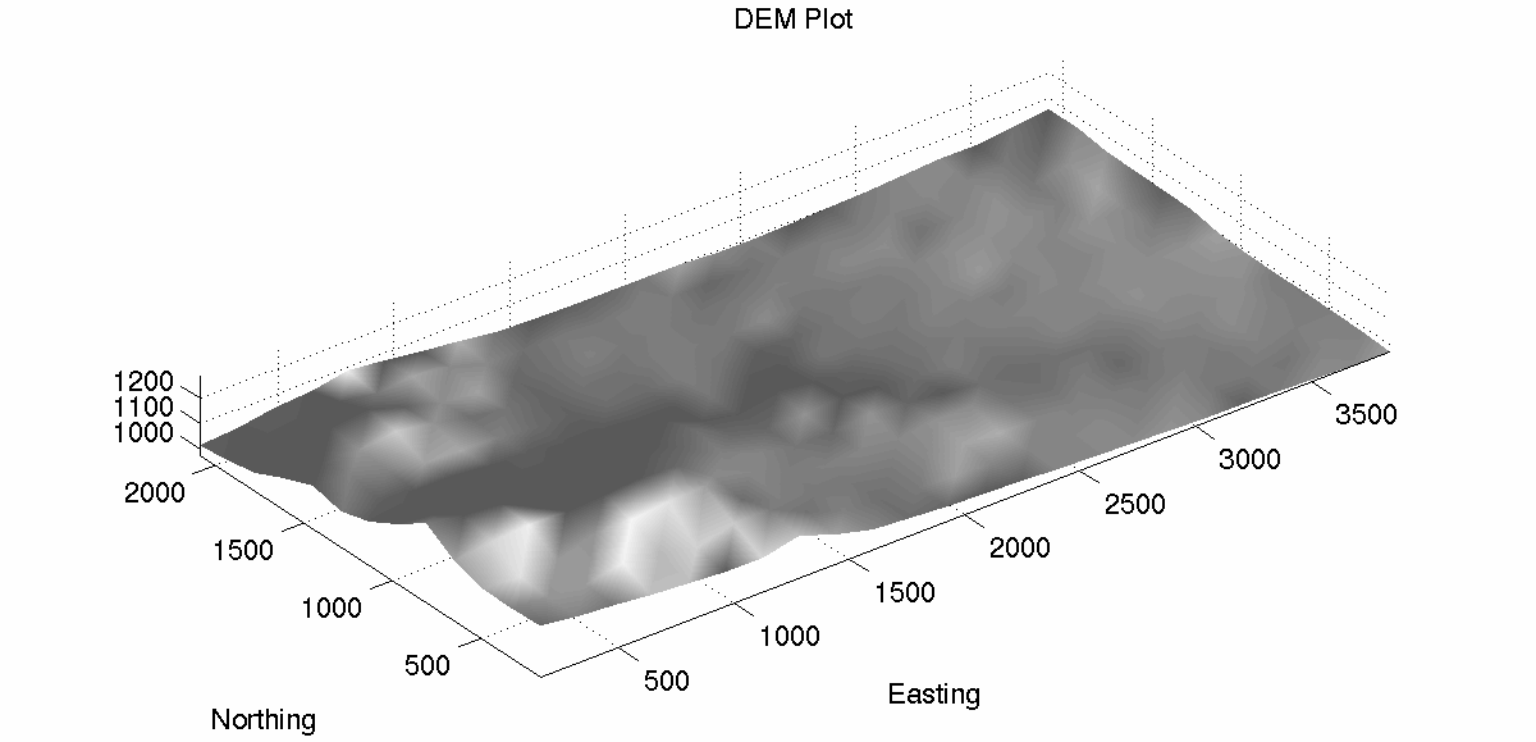
32x


DEM Plot
1200
1100
1000
2500
2000
1500
1000
Northing
1000
1500
2000
2500
3000
3500
Easting

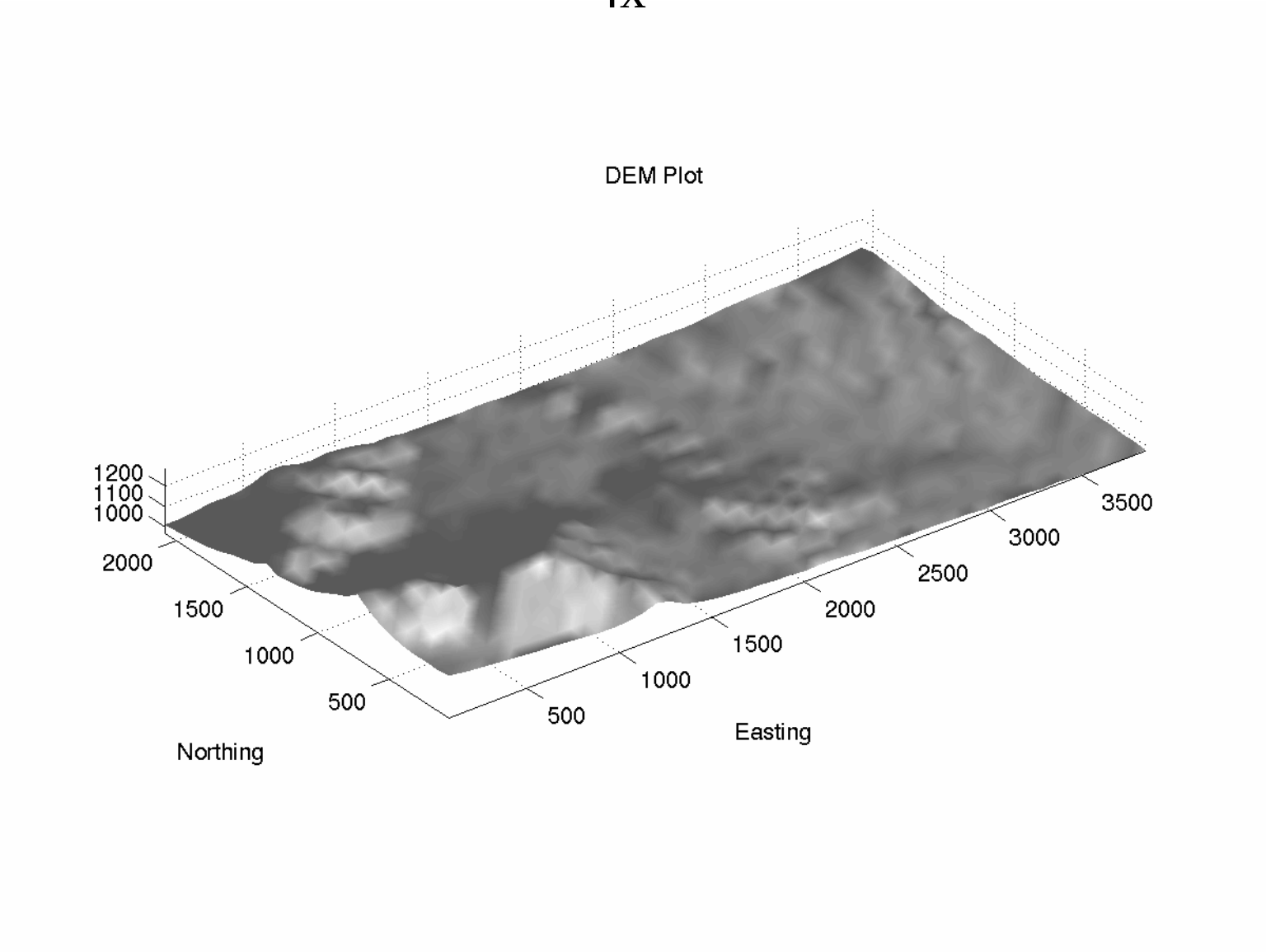
16x


DEM Plot
1200
1100
1000
2000
1500
1000
500
Northing
500
1000
1500
2000
2500
3000
3500
Easting

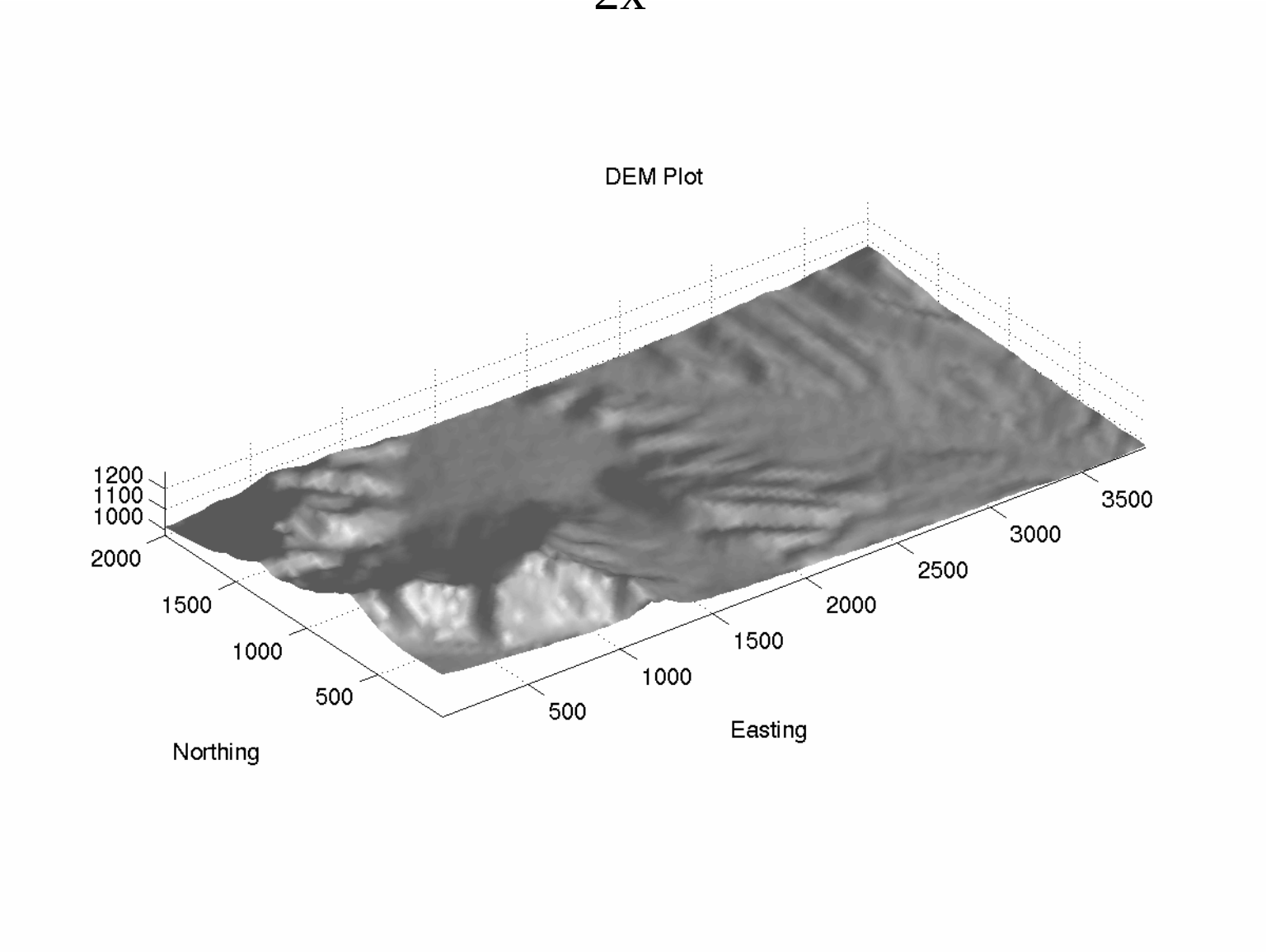
8x


DEM Plot
1200
1100
1000
2000
1500
1000
500
Northing
500
1000
1500
2000
2500
3000
3500
Easting

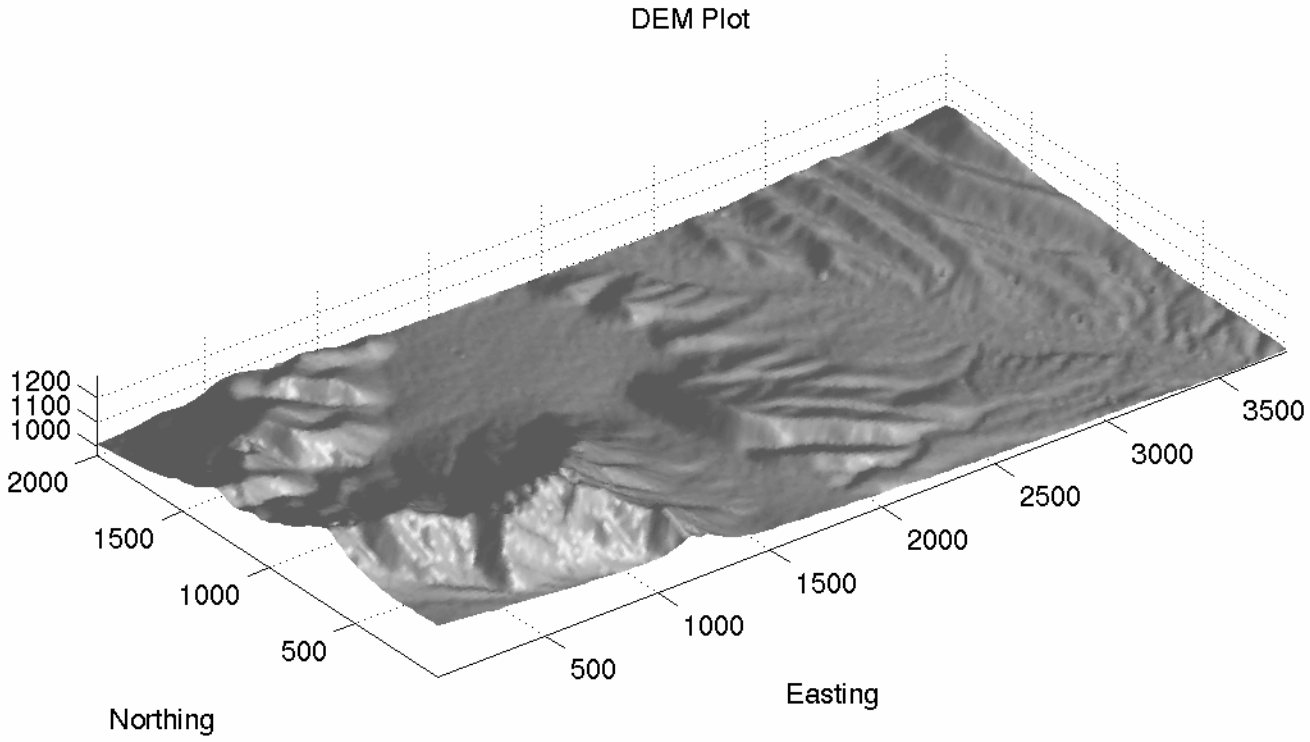
4x


DEM Plot
1200
1100
1000
2000
1500
1000
500
Northing
500
1000
1500
2000
2500
3000
3500
Easting

2x

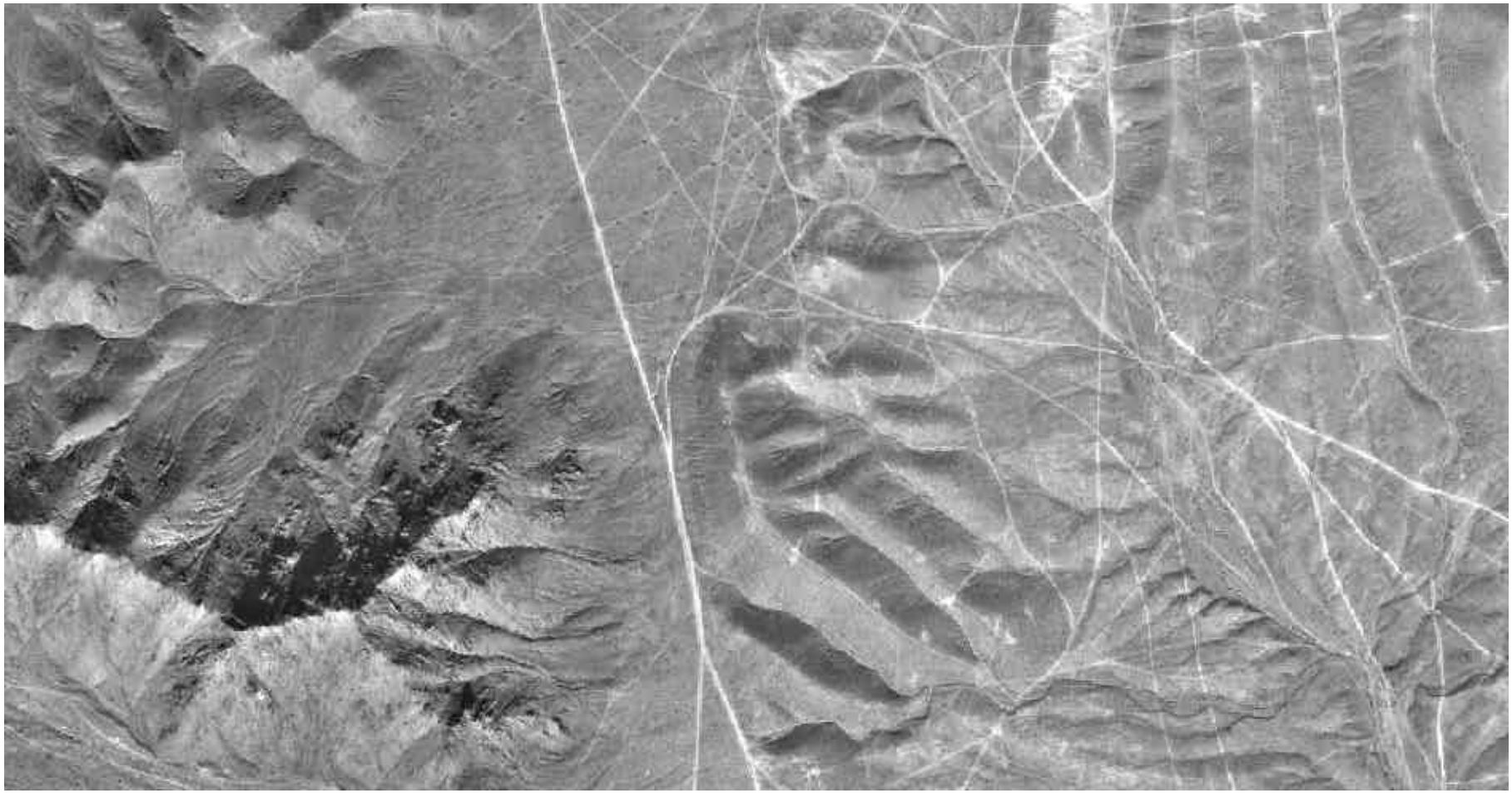

DEM Plot
1200
1100
1000
2000
1500
1000
500
Northing
500
1000
1500
2000
2500
3000
3500
Easting

1x


DEM Plot
1200
1100
1000
2000
1500
1000
500
Northing
500
1000
1500
2000
2500
3000
3500
Easting

Application of Generated DEM: Orthorectification

Each pixel is now in its planimetrically correct location – add ESRI world file and you are georeferenced – also easy to merge with DEM for visualization – VRML, etc.



```
#VRML V2.0 utf8
Group {
  children [
  # Viewpoints
    Viewpoint {
      description "view 1"
      position 1900.0 4012.9 1000.0
      orientation 1.0 0.0 0.0 -1.57
      },
    Viewpoint {
      description "view 2"
      position 3317.9 3182.3 2417.9
      orientation 0.6786 -0.6786 -0.2811 -1.0961
      },
  # Navigation
    NavigationInfo {
      type "EXAMINE"
      speed 1.0
      headlight TRUE
      avatarSize [1.0,1.0,1.0]
      },
  # Lighting
    DirectionalLight {
      on TRUE
      intensity 1.0
      ambientIntensity 1.0
      color 1.0 1.0 1.0
      direction 0.0 -1.0 0.0
      },
```

VRML – Virtual Reality Modeling Language

```
Shape {
  appearance Appearance {
    material Material { }
    texture ImageTexture { url "o.jpg" }
  }
  geometry ElevationGrid {
    xDimension    190
    zDimension    100
    xSpacing      20.0
    zSpacing      20.0
    solid         FALSE
    creaseAngle   0.785
    height [
976.5, 974.5, 972.3, 972.5, 972.2, 971.9, 970.2, 970.0, 969.7, 969.5,
969.5, 969.9, 970.9, 968.9, 969.9, 970.4, 971.4, 972.4, 972.9, 972.4,
972.9, 973.4, 974.6, 975.8, 976.5, 978.2, 978.7, 980.2, 981.2, 982.8,
984.4, 985.5, 985.5, 985.6, 987.2, 987.3, 987.4, 987.5, 988.8, 988.2,
988.1, 989.4, 988.8, 989.2, 988.5, 988.9, 989.6, 989.8, 990.5, 991.2,
992.7, 992.7, 993.2, 993.2, 994.0, 994.4, 994.7, 995.5, 996.3, 996.7,
998.0, 998.4, 998.9, 999.5, 1000.0, 1000.6, 1000.9, 1001.2, 1002.1, 1002.9,
1003.8, 1005.2, 1005.7, 1006.6, 1007.3, 1007.6, 1008.8, 1009.0, 1010.2, 1010.4,
1012.1, 1013.2, 1014.9, 1017.6, 1019.3, 1020.1, 1020.8, 1021.0, 1021.2, 1020.5,
```

Continuation of VRML code for the NTC DEM & ortho image