Homework 5 Sat. Photogr.

Orthorectification assigned 1 Mar. 2017, due 24 Mar. (Fri)

Create function $\begin{bmatrix} l \\ s \end{bmatrix} = fg2i(\phi, \lambda, h, dp)$

This will call fi2g-pl_0 but we switch the roles of knowns and unknowns. Now ϕ, λ, h are known and l, s are to be computed. Do by iterative inversion.

- input ϕ, λ, h. start with initial approx. $(l^0, s^0) = (0, 0)$.
- call fi2g-pl_0_part to obtain $\partial F_\phi/\partial l$, $\partial F_\phi/\partial s$, F_ϕ; $\partial F_\lambda/\partial l$, $\partial F_\lambda/\partial s$, F_λ
- construct equations:

Newton iteration loop

$$\begin{bmatrix} \frac{\partial F_\phi}{\partial l} & \frac{\partial F_\phi}{\partial s} \\ \frac{\partial F_\lambda}{\partial l} & \frac{\partial F_\lambda}{\partial s} \end{bmatrix} \begin{bmatrix} \Delta l \\ \Delta s \end{bmatrix} = \begin{bmatrix} -F_\phi \\ -F_\lambda \end{bmatrix}$$

solve for $\Delta l, \Delta s$ by $\Delta = -J^{-1}F$

$J \quad \Delta = -F$

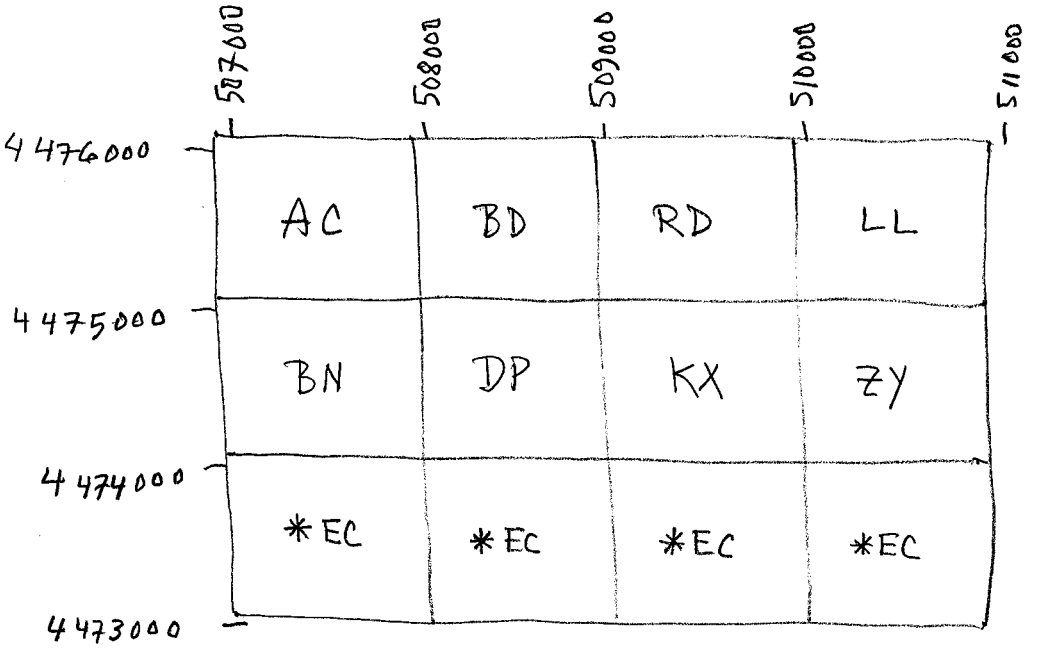
- update l^0, s^0

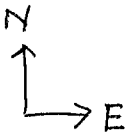
when it "works", verify numerically that it is inverse of fi2g-pl:

$l, s, h \xrightarrow{fi2g\text{-}pl} \phi, \lambda, (h)$

$\phi, \lambda, (h) \xrightarrow{fg2i} l, s, h$

Use the above created function fg2i to make an ortho rectification of your assigned tile at 0.5m GSD in coordinate system ISP West, meters





will be seamless mosaic no overlap between tiles.

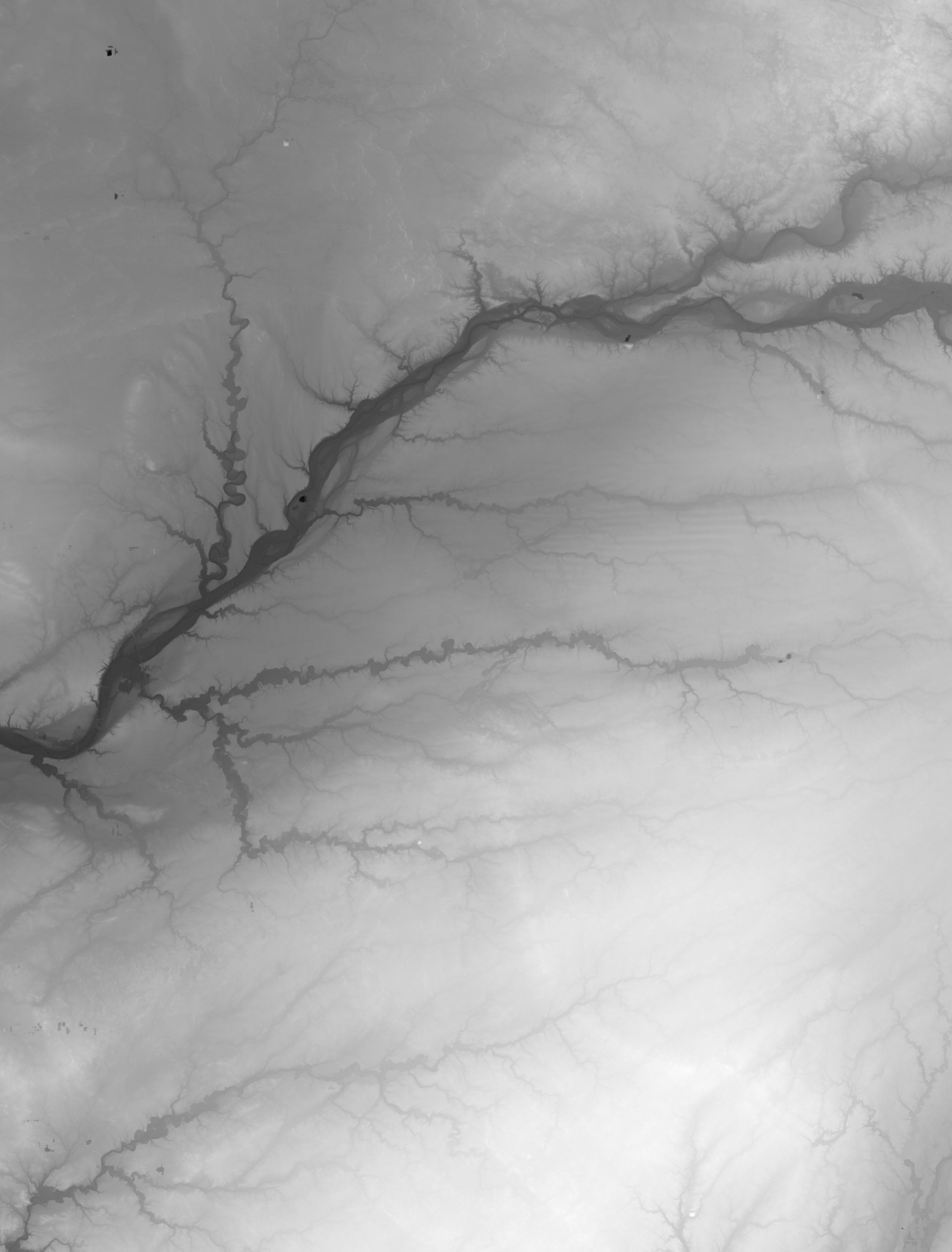
each tile is 1000 x 1000 m or 2000 x 2000 pixels

You may use ftmgeo_utmz16.m on ftp folder.

$$\begin{bmatrix} \phi \\ \lambda \end{bmatrix} = \text{ftmgeo\_utmz16}(E, N)$$

(radians) (meters)

More instructions coming on DEM, N, image structure, world file format, ESRI vector overlay, ...



reading in the DEM data:

```
fid = fopen('usgs.flt','r');
temp = fread(fid,[3612,3612],'single');
dem = temp';   % transpose
clear temp
```

Do this only once! at the beginning of program

---

interpolate height at ϕ, λ

$r_2 = fix\left[(\phi_T - \phi)/\Delta\phi\right] + 1$

$r_1 = r_2 + 1$

$c_1 = fix\left[(\lambda - \lambda_L)/\Delta\lambda\right] + 1$

$c_2 = c_1 + 1$

$\phi_1 = \phi_T - (r_1 - 1) * \Delta\phi$

$\phi_2 = \phi_1 + \Delta\phi$

$\lambda_1 = \lambda_L + (c_1 - 1) * \Delta\lambda$

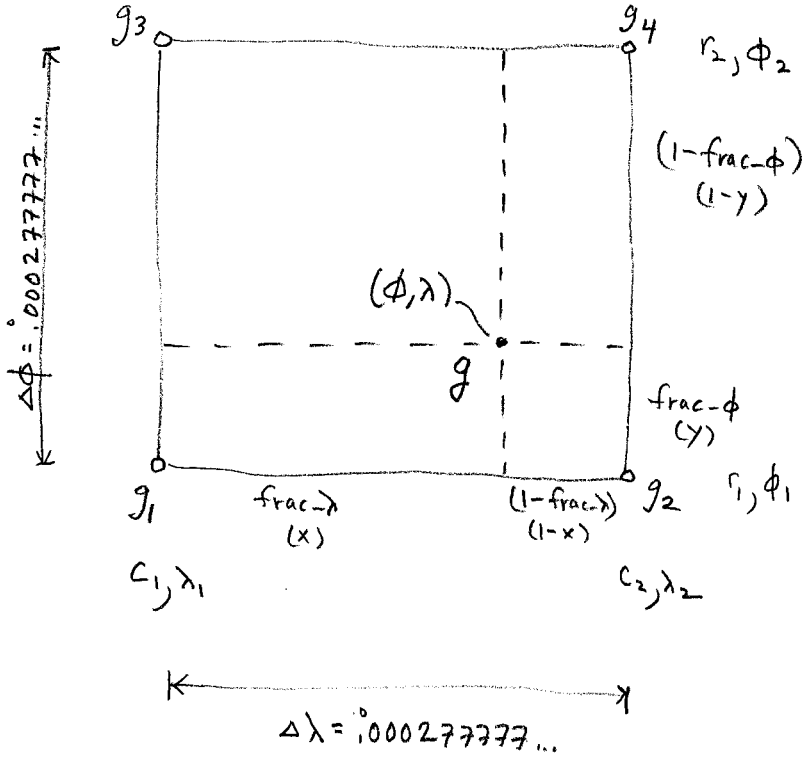
$\lambda_2 = \lambda_1 + \Delta\lambda$

$frac\_\phi = (\phi - \phi_1)/\Delta\phi$

$frac\_\lambda = (\lambda - \lambda_1)/\Delta\lambda$

$x = frac\_\lambda$

$y = frac\_\phi$



$g_1 = dem(r_1, c_1)$

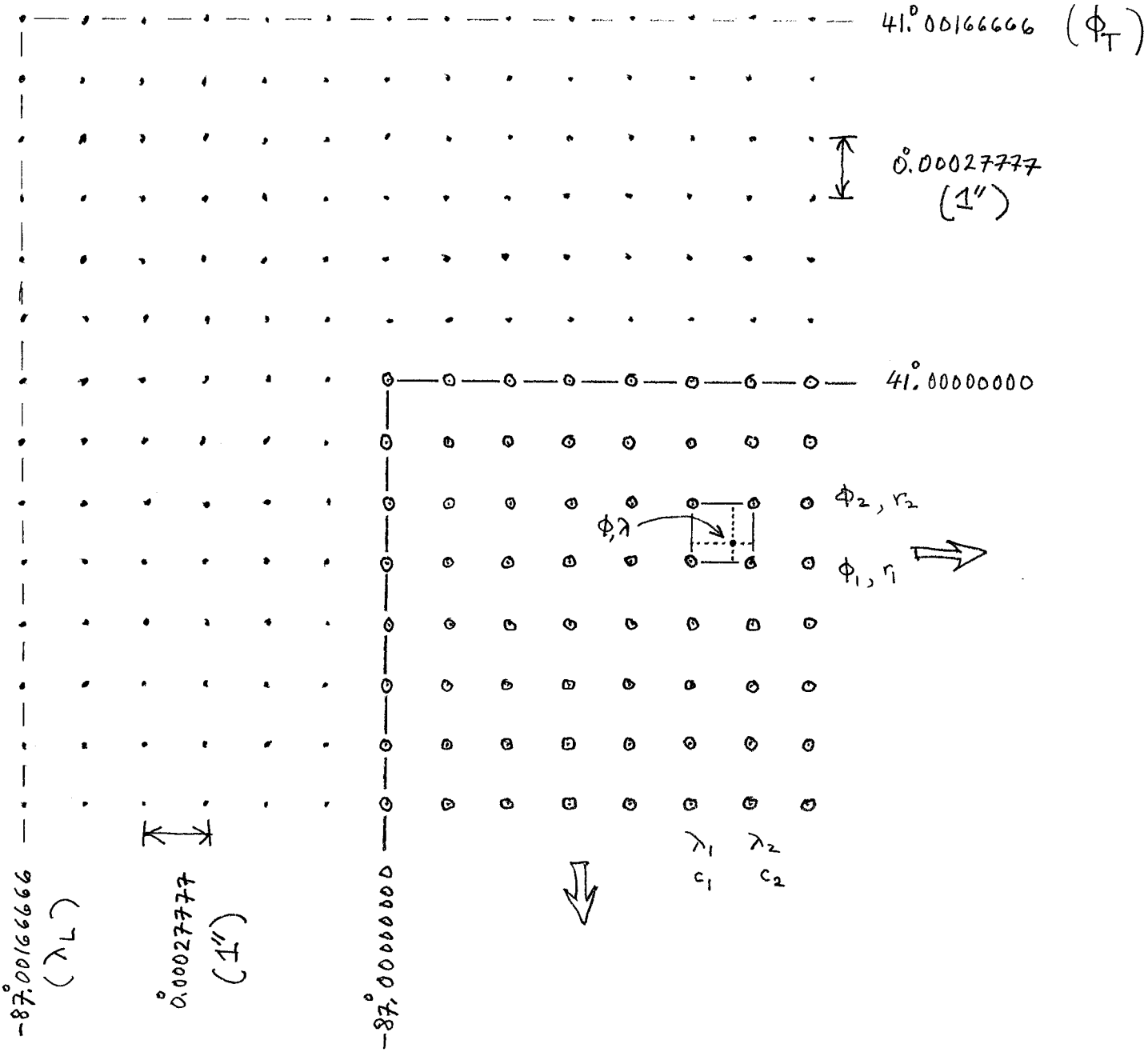
$g_2 = dem(r_1, c_2)$

$g_3 = dem(r_2, c_1)$

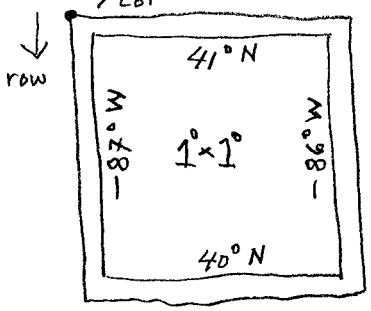
$g_4 = dem(r_2, c_2)$

$$g = (1 - x - y + xy) * g_1 + (x - xy) * g_2 + (y - xy) * g_3 + (xy) * g_4$$

g is the orthometric (sea level) height at location ϕ, λ usually denoted as H.



6 extra rows/columns on N, S, E, W edges



1st row is at the top (N)

elements in row proceed L→R (W→E)

need index arithmetic to interpolate height at φ, λ from the grid

USGS 1 arc second DEM "grid float" format

geoid12b_laf.txt

40 27 00	-33.68	-33.70	-33.72	-33.75	-33.78
40 26 00	-33.67	-33.69	-33.71	-33.74	-33.76
40 25 00	-33.66	-33.68	-33.70	-33.72	-33.74
40 24 00	-33.64	-33.66	-33.68	-33.71	-33.73

-86 56 00 | -86 55 00 | -86 54 00 | -86 53 00 | -86 52 00

load geoid12b_laf.txt

$\Delta\phi$ = 1 min = 1/60 deg = .01666...

$\Delta\lambda$ = 1 min = 1/60 deg = .01666...

$\phi_B = 40.4$, $\lambda_L = -86.9333...$

interpolate N at ϕ, λ =

$r_1 = fix\left[(\phi - \phi_B) / \Delta\phi\right] + 1$

$r_2 = r_1 + 1$

$c_1 = fix\left[(\lambda - \lambda_L) / \Delta\lambda\right] + 1$

$c_2 = c_1 + 1$

$\phi_1 = \phi_B + (r_1 - 1) * \Delta\phi$

$\lambda_1 = \lambda_L + (c_1 - 1) * \Delta\lambda$

$frac\_\phi = (\phi - \phi_1) / \Delta\phi$

$frac\_\lambda = (\lambda - \lambda_1) / \Delta\lambda$

$x = frac\_\lambda$

$y = frac\_\phi$

$g_1 = geoid12b\_laf(r_1, c_1)$

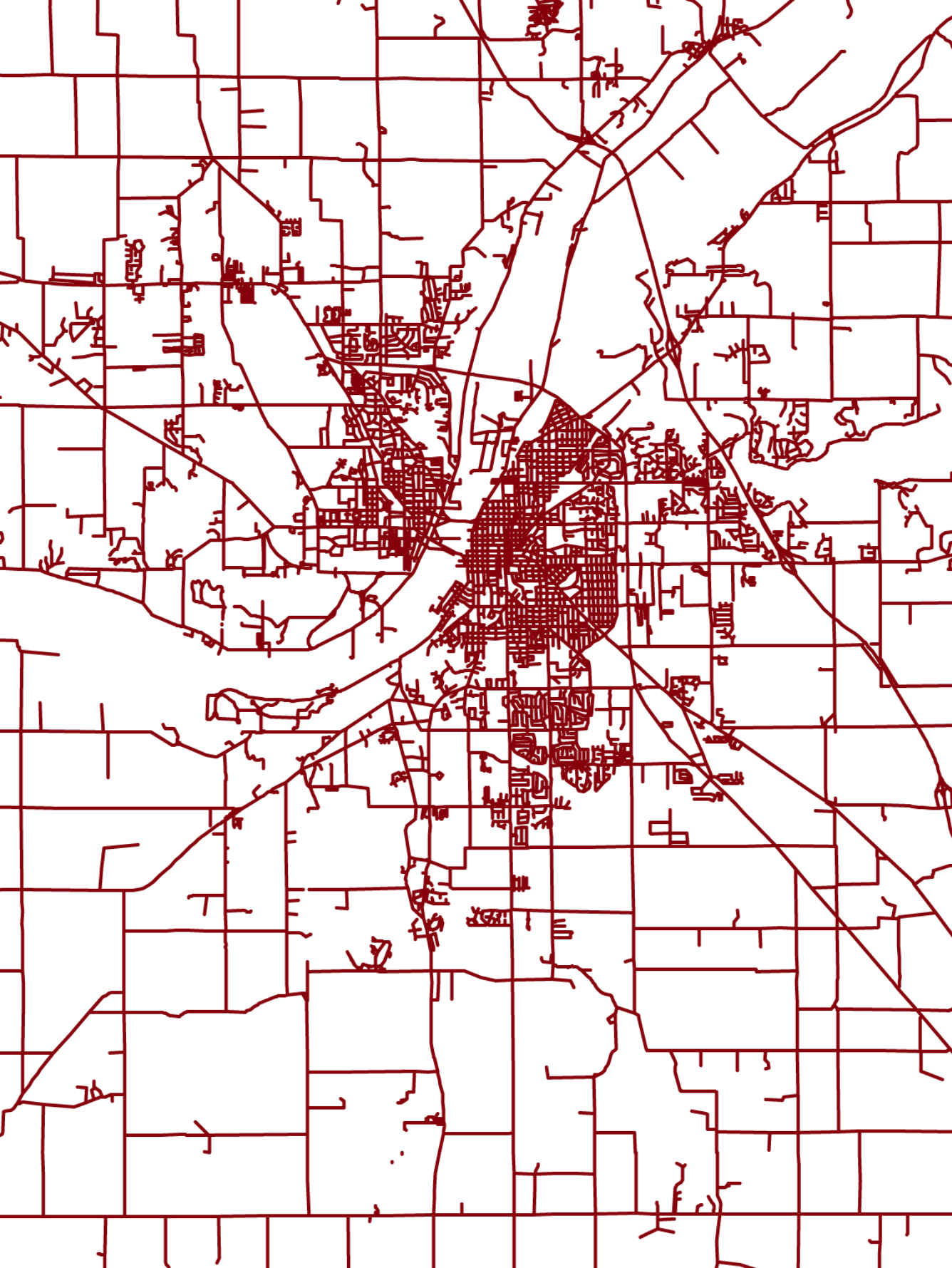
$g_2 = geoid12b\_laf(r_1, c_2)$

$g_3 = geoid12b\_laf(r_2, c_1)$

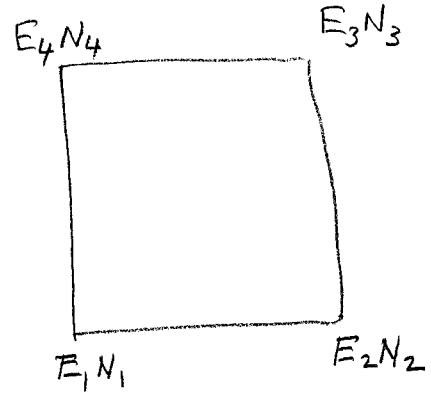
$g_4 = geoid12b\_laf(r_2, c_2)$

$g = (1 - x - y + x*y) * g_1 + (x - x*y) * g_2 + (y - x*y) * g_3 + (x*y) * g_4$

$N = g$



vector overlay to check geometry



create a clipping rectangle for your tile (in matlab):

```
Bdry.Geometry = 'Polygon';
Bdry.X = [E1 E2 E3 E4 E1 NaN];
Bdry.Y = [N1 N2 N3 N4 N1 NaN];
Bdry.Name = 'Lafayette Ortho';
Bdry.BoundingBox = [min(Bdry.X) min(Bdry.Y); max(Bdry.X) max(Bdry.Y)];
%   = [E1 N1; E2 N3]
Shapewrite(Bdry, 'boundary');
% This will write boundary.shp, .shx, .dbf
% you can use this to clip the vector data to only your image tile
```

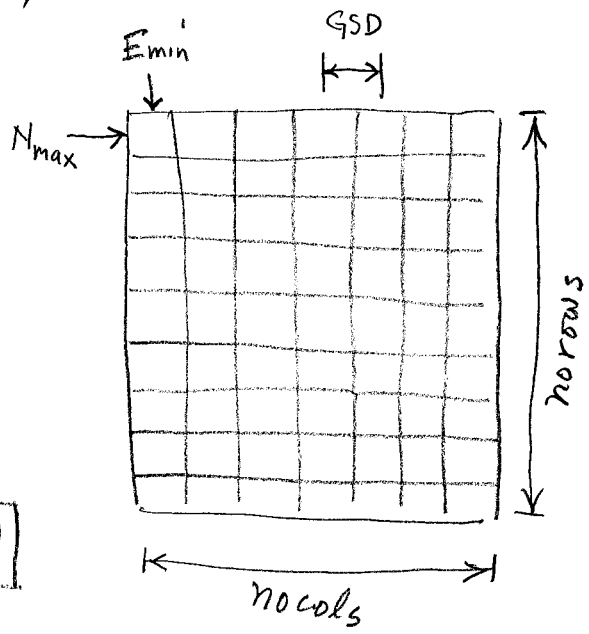
---

to clip (crop) vector data:

Start ArcMap
- blank map
- connect to folder (catalog tab)
- add data
  - road_2
  - boundary (from above!)
- geoprocessing
  - clip
    - clip dialogue
      - input feature
      - clip feature
      - output feature
      - xy tolerance 0.5
    - OK

template/ flowchart for orthorectification code

```
in_img = imread('laf01.tif', 'TIFF');
[nirows, nicols] = size(in_img);
out_img = zeros(norows, nocols, 'uint8');
for i = 1:norows
    for j = 1:nocols
        E = Emin + (j-1) * GSD;
        N = Nmax - (i-1) * GSD;
        [transform E,N → φ,λ]
        [interpolate H from DEM]
        [interpolate N from geoid grid]
        h = N+H;
        [l; s] = fg2i(φ,λ,h,dp)
        if l,s both in range (>2, < max-1)
            g = [interpolate intensity from in_img]   % 0-255
        else
            g = 128
        end
        out_img(i,j) = uint8(round(g))
    end % j-loop
end % i-loop
imwrite(out_img, 'outfile.jpg', 'JPEG');
```



Esri "world file"

□.tif → □.tfw

□.jpg → □.jgw

↓

text file with 6 numbers, create with notepad, etc.

GSD X
0
0
− GSD Y
X upper left
Y upper left

note: intensity from image is uint8, convert to double before doing math

gd = double(g); etc.

---

turn in:

1. code (hard copy & email, incl. all functions)
2. hardcopy maps, image + overlay, arrange brightness/contrast and color so visible
3. zoom in to show registration in a "few" places good, bad, consistent, hypotheses ?
4. □.jpg, □.jgw

± will merge all submissions into a mosaic