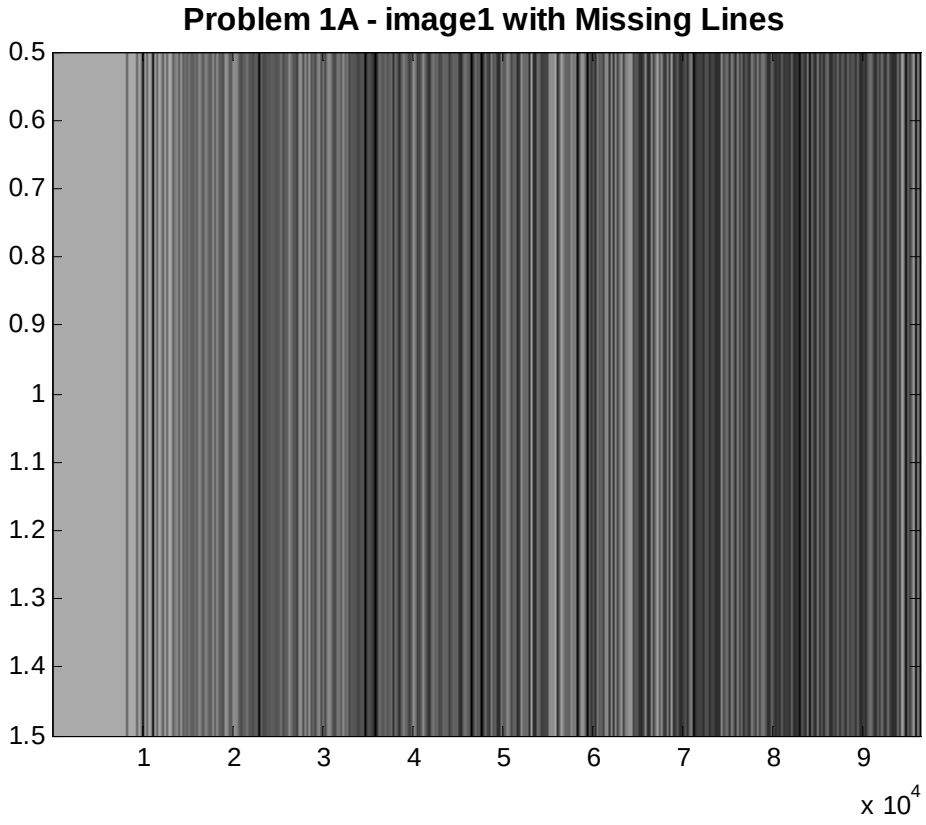
Christopher Tsai
December 25, 2007
EE 262: 2D Imaging

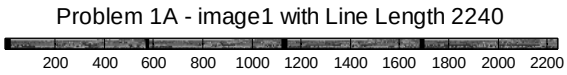
Problem Set I – The Image Plane

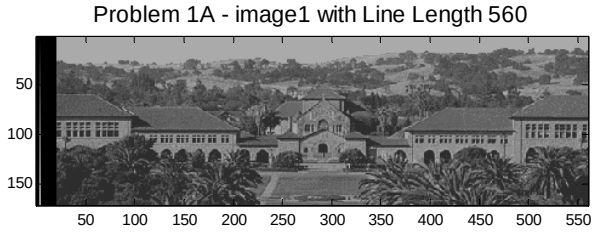
Problem #1 – Reading an Image Header

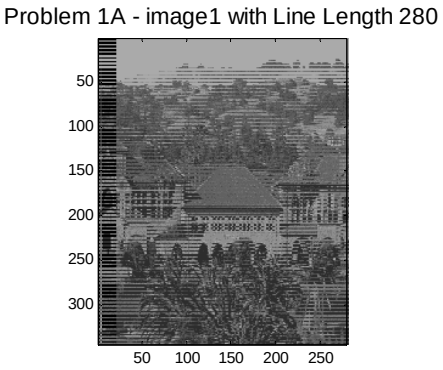
When we first plot the image in its raw form, we suspect misalignment of rows and columns:



The uniformly gray region at the left intimates proper isolation of the header bytes, but the remainder of the image makes no visual sense, so we attempt to rectify the image with proper reorganization of rows and columns. To determine the correct number of lines ($m \times n$), we can prime-factorize the number of samples and combine the prime factors into various different products, repeating this trial until our image resembles some identifiable object:







The prime factor combination (172 rows × 560 columns) yields the most sensible picture, with the next longer line length (86 rows × 1120 columns) stretching and replicating the image and the next shorter line length (344 rows × 280 columns) tainting the pure header columns. Thus, the proper line length is **560 samples.**

We differentiate the header bytes from the remainder of the image file by their lack of visual content. When numerically listed, the header bytes are almost all uniformly zero or linearly increasing, as if manmade; they exhibit no natural variation. For example, the first ten header columns bear noticeable pattern:

```
0     0     0     1     0     0     0     0     0     0
0     0     0     2     0     0     0     0     0     0
0     0     0     3     0     0     0     0     0     0
0     0     0     4     0     0     0     0     0     0
0     0     0     5     0     0     0     0     0     0
0     0     0     6     0     0     0     0     0     0
0     0     0     7     0     0     0     0     0     0
0     0     0     8     0     0     0     0     0     0
0     0     0     9     0     0     0     0     0     0
0     0     0    10     0     0     0     0     0     0
0     0     0    11     0     0     0     0     0     0
0     0     0    12     0     0     0     0     0     0
0     0     0    13     0     0     0     0     0     0
0     0     0    14     0     0     0     0     0     0
0     0     0    15     0     0     0     0     0     0
0     0     0    16     0     0     0     0     0     0
0     0     0    17     0     0     0     0     0     0
0     0     0    18     0     0     0     0     0     0
```

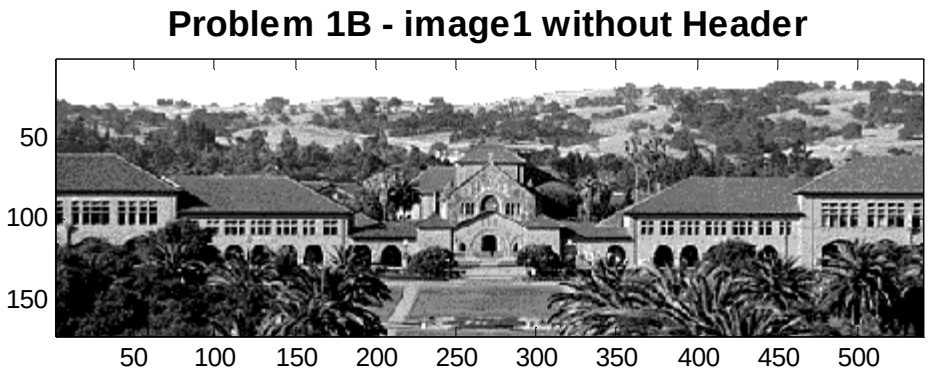
```
0     0     0    19     0     0     0     0     0     0
0     0     0    20     0     0     0     0     0     0
0     0     0    21     0     0     0     0     0     0
0     0     0    22     0     0     0     0     0     0
0     0     0    23     0     0     0     0     0     0
0     0     0    24     0     0     0     0     0     0
0     0     0    25     0     0     0     0     0     0
0     0     0    26     0     0     0     0     0     0
0     0     0    27     0     0     0     0     0     0
0     0     0    28     0     0     0     0     0     0
0     0     0    29     0     0     0     0     0     0
0     0     0    30     0     0     0     0     0     0
0     0     0    31     0     0     0     0     0     0
0     0     0    32     0     0     0     0     0     0
0     0     0    33     0     0     0     0     0     0
0     0     0    34     0     0     0     0     0     0
0     0     0    35     0     0     0     0     0     0
0     0     0    36     0     0     0     0     0     0
0     0     0    37     0     0     0     0     0     0
0     0     0    38     0     0     0     0     0     0
0     0     0    39     0     0     0     0     0     0
0     0     0    40     0     0     0     0     0     0
0     0     0    41     0     0     0     0     0     0
0     0     0    42     0     0     0     0     0     0
0     0     0    43     0     0     0     0     0     0
0     0     0    44     0     0     0     0     0     0
0     0     0    45     0     0     0     0     0     0
0     0     0    46     0     0     0     0     0     0
0     0     0    47     0     0     0     0     0     0
0     0     0    48     0     0     0     0     0     0
0     0     0    49     0     0     0     0     0     0
0     0     0    50     0     0     0     0     0     0
0     0     0    51     0     0     0     0     0     0
0     0     0    52     0     0     0     0     0     0
0     0     0    53     0     0     0     0     0     0
0     0     0    54     0     0     0     0     0     0
0     0     0    55     0     0     0     0     0     0
0     0     0    56     0     0     0     0     0     0
0     0     0    57     0     0     0     0     0     0
0     0     0    58     0     0     0     0     0     0
0     0     0    59     0     0     0     0     0     0
0     0     0    60     0     0     0     0     0     0
0     0     0    61     0     0     0     0     0     0
0     0     0    62     0     0     0     0     0     0
0     0     0    63     0     0     0     0     0     0
0     0     0    64     0     0     0     0     0     0
0     0     0    65     0     0     0     0     0     0
0     0     0    66     0     0     0     0     0     0
0     0     0    67     0     0     0     0     0     0
0     0     0    68     0     0     0     0     0     0
0     0     0    69     0     0     0     0     0     0
0     0     0    70     0     0     0     0     0     0
0     0     0    71     0     0     0     0     0     0
0     0     0    72     0     0     0     0     0     0
0     0     0    73     0     0     0     0     0     0
0     0     0    74     0     0     0     0     0     0
0     0     0    75     0     0     0     0     0     0
0     0     0    76     0     0     0     0     0     0
0     0     0    77     0     0     0     0     0     0
0     0     0    78     0     0     0     0     0     0
0     0     0    79     0     0     0     0     0     0
0     0     0    80     0     0     0     0     0     0
0     0     0    81     0     0     0     0     0     0
0     0     0    82     0     0     0     0     0     0
0     0     0    83     0     0     0     0     0     0
0     0     0    84     0     0     0     0     0     0
0     0     0    85     0     0     0     0     0     0
0     0     0    86     0     0     0     0     0     0
0     0     0    87     0     0     0     0     0     0
0     0     0    88     0     0     0     0     0     0
0     0     0    89     0     0     0     0     0     0
0     0     0    90     0     0     0     0     0     0
```

```
0     0     0    91     0     0     0     0     0     0
0     0     0    92     0     0     0     0     0     0
0     0     0    93     0     0     0     0     0     0
0     0     0    94     0     0     0     0     0     0
0     0     0    95     0     0     0     0     0     0
0     0     0    96     0     0     0     0     0     0
0     0     0    97     0     0     0     0     0     0
0     0     0    98     0     0     0     0     0     0
0     0     0    99     0     0     0     0     0     0
0     0     0   100     0     0     0     0     0     0
0     0     0   101     0     0     0     0     0     0
0     0     0   102     0     0     0     0     0     0
0     0     0   103     0     0     0     0     0     0
0     0     0   104     0     0     0     0     0     0
0     0     0   105     0     0     0     0     0     0
0     0     0   106     0     0     0     0     0     0
0     0     0   107     0     0     0     0     0     0
0     0     0   108     0     0     0     0     0     0
0     0     0   109     0     0     0     0     0     0
0     0     0   110     0     0     0     0     0     0
0     0     0   111     0     0     0     0     0     0
0     0     0   112     0     0     0     0     0     0
0     0     0   113     0     0     0     0     0     0
0     0     0   115     0     0     0     0     0     0
0     0     0   116     0     0     0     0     0     0
0     0     0   117     0     0     0     0     0     0
0     0     0   118     0     0     0     0     0     0
0     0     0   119     0     0     0     0     0     0
0     0     0   120     0     0     0     0     0     0
0     0     0   121     0     0     0     0     0     0
0     0     0   122     0     0     0     0     0     0
0     0     0   123     0     0     0     0     0     0
0     0     0   124     0     0     0     0     0     0
0     0     0   125     0     0     0     0     0     0
0     0     0   126     0     0     0     0     0     0
0     0     0   127     0     0     0     0     0     0
0     0     0   128     0     0     0     0     0     0
0     0     0   129     0     0     0     0     0     0
0     0     0   130     0     0     0     0     0     0
0     0     0   131     0     0     0     0     0     0
0     0     0   132     0     0     0     0     0     0
0     0     0   133     0     0     0     0     0     0
0     0     0   134     0     0     0     0     0     0
0     0     0   135     0     0     0     0     0     0
0     0     0   136     0     0     0     0     0     0
0     0     0   137     0     0     0     0     0     0
0     0     0   138     0     0     0     0     0     0
0     0     0   139     0     0     0     0     0     0
0     0     0   140     0     0     0     0     0     0
0     0     0   141     0     0     0     0     0     0
0     0     0   142     0     0     0     0     0     0
0     0     0   143     0     0     0     0     0     0
0     0     0   144     0     0     0     0     0     0
0     0     0   145     0     0     0     0     0     0
0     0     0   146     0     0     0     0     0     0
0     0     0   147     0     0     0     0     0     0
0     0     0   148     0     0     0     0     0     0
0     0     0   149     0     0     0     0     0     0
0     0     0   150     0     0     0     0     0     0
0     0     0   151     0     0     0     0     0     0
0     0     0   152     0     0     0     0     0     0
0     0     0   153     0     0     0     0     0     0
0     0     0   154     0     0     0     0     0     0
0     0     0   155     0     0     0     0     0     0
0     0     0   156     0     0     0     0     0     0
0     0     0   157     0     0     0     0     0     0
0     0     0   158     0     0     0     0     0     0
0     0     0   159     0     0     0     0     0     0
0     0     0   160     0     0     0     0     0     0
0     0     0   161     0     0     0     0     0     0
0     0     0   162     0     0     0     0     0     0
0     0     0   163     0     0     0     0     0     0
```

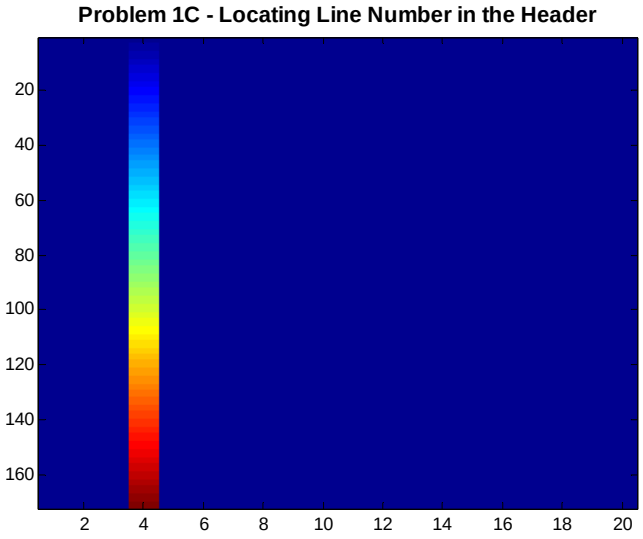
```
0     0     0   164     0     0     0     0     0     0
0     0     0   165     0     0     0     0     0     0
0     0     0   166     0     0     0     0     0     0
0     0     0   167     0     0     0     0     0     0
0     0     0   168     0     0     0     0     0     0
0     0     0   169     0     0     0     0     0     0
0     0     0   170     0     0     0     0     0     0
0     0     0   171     0     0     0     0     0     0
0     0     0   172     0     0     0     0     0     0
0     0     0   173     0     0     0     0     0     0
```

The image data contain twenty such header columns, so the header length is **20 bytes.**

When imaged graphically, the header bytes constitute the uniform black (zero) band to the left side of the image. After removing these contrast-sagging header bytes, the scaled image improves:



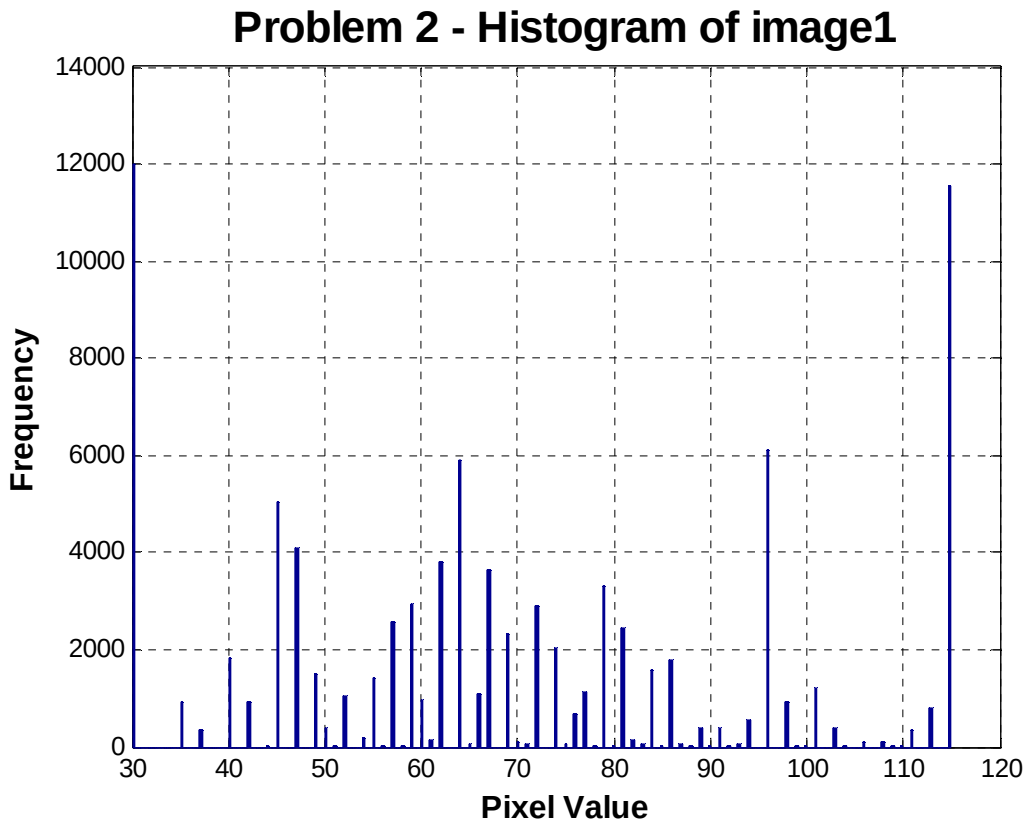
We can locate the line number in two ways. By displaying the matrix values on-screen, we have already seen one header column counting along a unit-increment tag. These incrementing integers represent the line number. We can also see the number in the plot:



Visual inspection, however, fails to discern the missing line. Only numerically can we perceive the skipped line number. To prevent manual examination from fatiguing the fallible human eye, we can catch abnormally large increments in the line number with a computer:

`missingLineNumbers = setdiff(1:173, lineNumbers)` = **line number 114**.

Problem #2 – Reading an Image Header



As recorded on the disk, the image statistics are $\mu \approx 69.05112$ and $\sigma \approx 26.615183$. As output from Matlab, we compute that…

```
The mean value of image1 is 69.051120.

The standard deviation of image1 is 26.615183.
```

If we seek to transform the mean to 128 and the standard deviation to 80, then we can perform the linear transformation:

$$\begin{cases} \mu_{desired} = \alpha\,\mu + \beta \\ \sigma_{desired} = \alpha\,\sigma \end{cases}$$

Multiplying each pixel by α increases the dynamic range of the pixels, stretching both the value of the mean and the separation represented by standard deviation. To achieve desired centrality, we can further shift the image value distribution by adding a constant; however, shifting the distribution

does not alter its variance, so the constant shift β does not disturb $\sigma_{desired}$. To solve this linear system, we compute $\begin{bmatrix}\alpha\\ \beta\end{bmatrix} = \begin{bmatrix}\mu & 1\\ \sigma & 0\end{bmatrix}^{-1}\begin{bmatrix}\mu_{desired}\\ \sigma_{desired}\end{bmatrix} = \begin{bmatrix}69.051 & 1\\ 26.615 & 0\end{bmatrix}^{-1}\begin{bmatrix}128\\ 80\end{bmatrix} \approx \begin{bmatrix}3.0058\\ -79.5541\end{bmatrix}$.

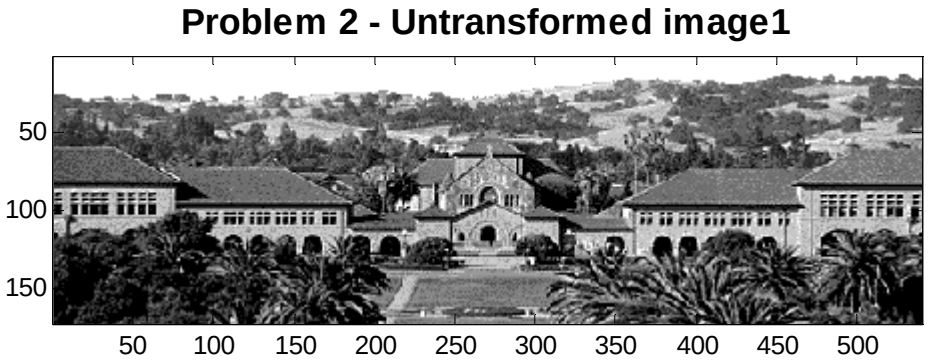
Therefore, in order to transform the image mean and standard deviation,

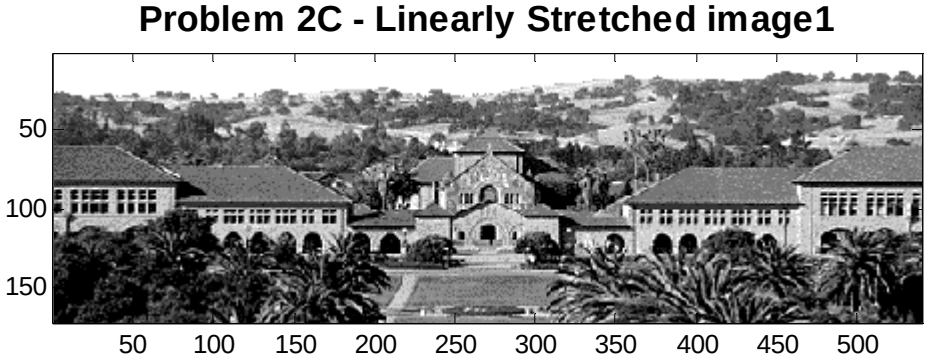
1.) Multiply each image pixel value by 3.005803,

2.) Add the constant -79.554072.

3.) Round negative values to zero and values above 255 to 255.

This linear transform increases the image's dynamic range, as shown below:





As the original histogram divulged, the original image pixel values cluster around extremely low near-zero values and extremely high (near 255) values, compromising the contrast of the image. By shifting the mean to the center of the possible values (120) and the standard deviation to be approximately $\frac{2}{3}$ of the range to either side of the mean (80), the image pixel values can roam the entire dynamic range of available values [0, 255], improving contrast in the display with a more

complete balance of bright whites and dark shades. Let us confirm that the transformation, despite edge truncation, has achieved the desired mean and standard deviation:

```
The mean value of the transformed image is 126.574829.
The standard deviation of the transformed image is 77.591208.
```

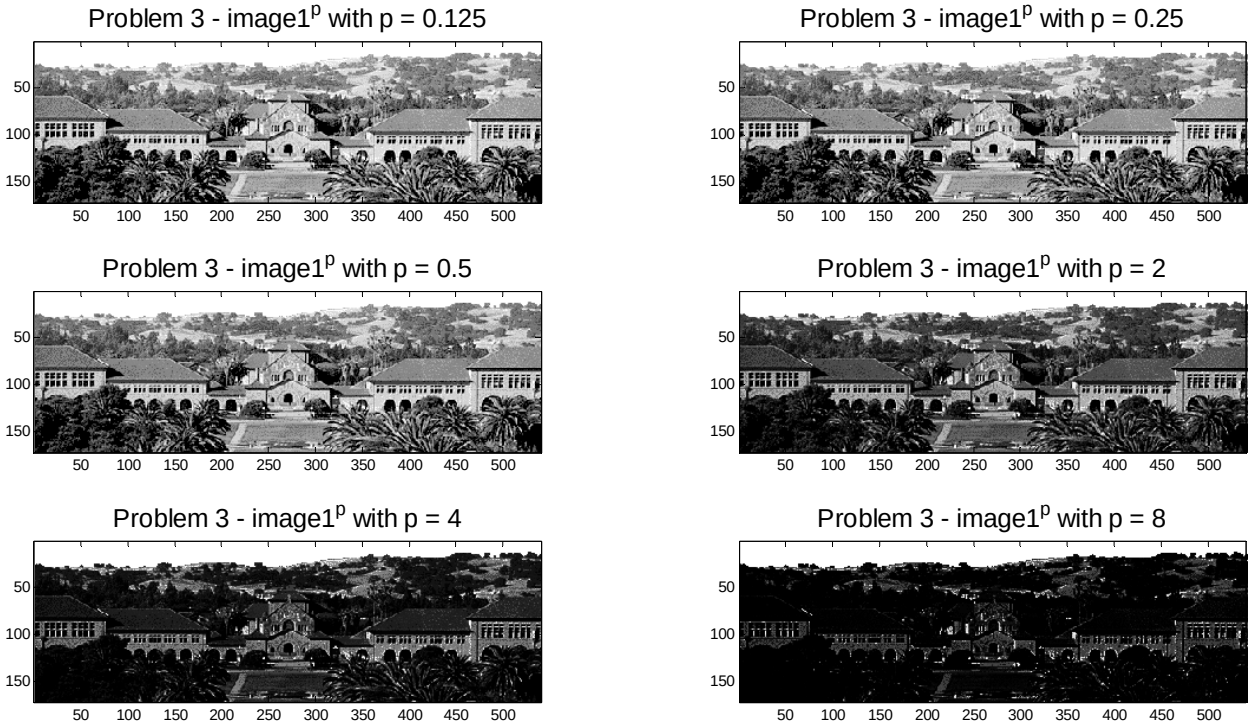
The mean is only slightly larger than desired (126.57 > 120), and the standard deviation is only 0.4 below the target of 80, which actually constitutes a conservative approximation since our pixel values will tend less to exceed the limits. In brief… success!

Problem #3 – Exponential Transform

We attempt to derive an exponential transform that exponentiates each input image pixel value:

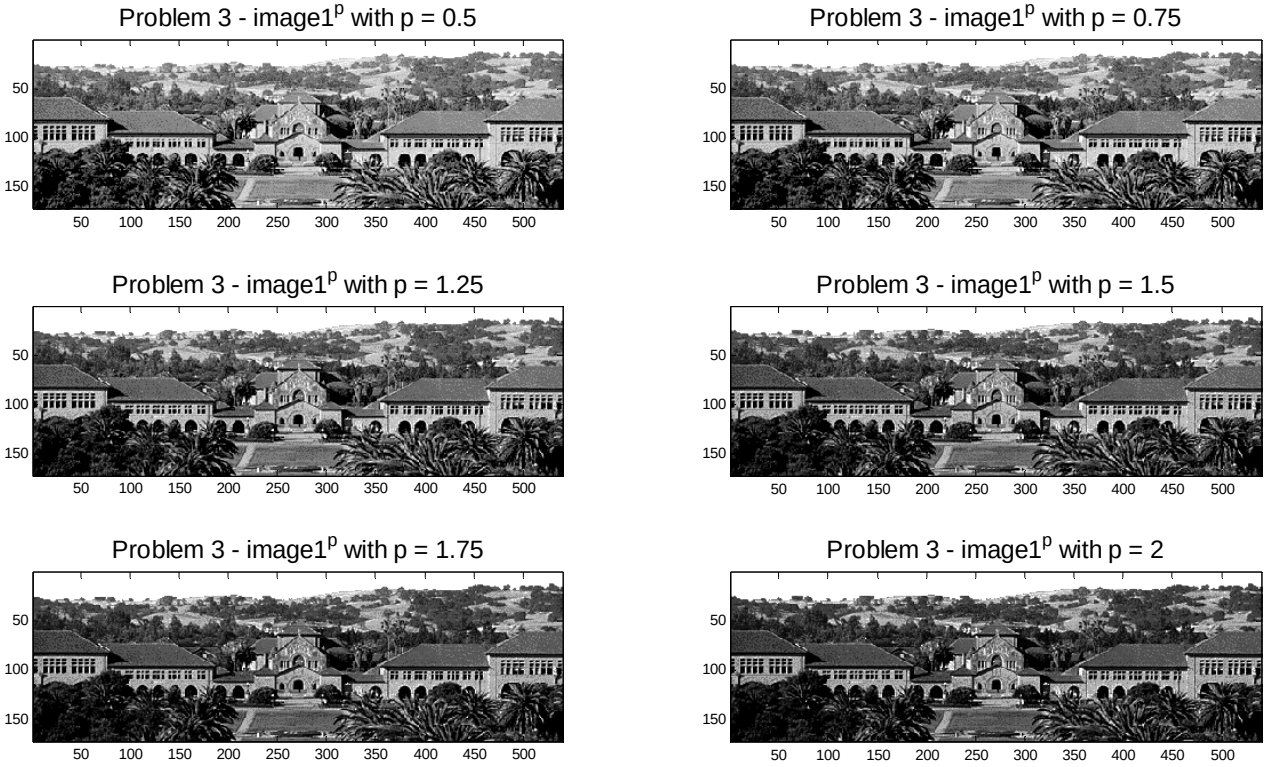
$$f_{out}(x,y) = f_{in}^{p}(x,y)$$

We heuristically substitute different values of p to scale the image most pleasingly. After scaling each pixel in the input image by p, we plot it and examine the image quality, varying p before appraising the image again:

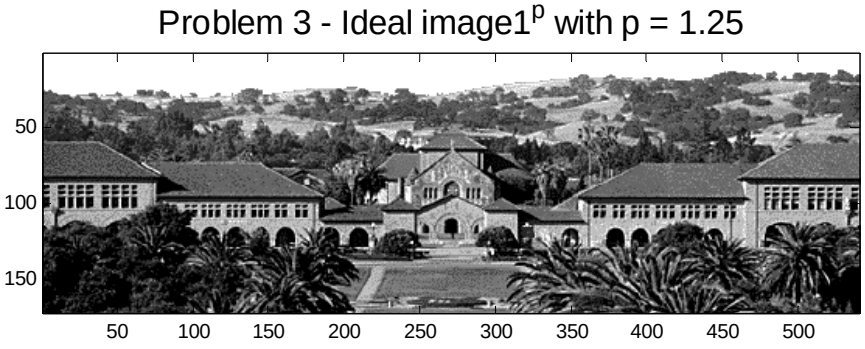


Although beauty lies in the eye of the beholder, the exponentiated images with $p < 0.5$ appear too light and generally overly white, whereas the images exponentiated to $p > 2$ display too many pitch dark shades with too few light tones to balance them out and illuminate fine details. If we focus on the medium between $p = 0.5$ and $p = 2$, we notice that the linearly scaled versions all appear similar, with minor details differentiating them. Whereas the $p = 1.5$ image seems to offer stronger darker shades with superior contrast, the $p = 1.25$ image seems to accentuate minute details best with a

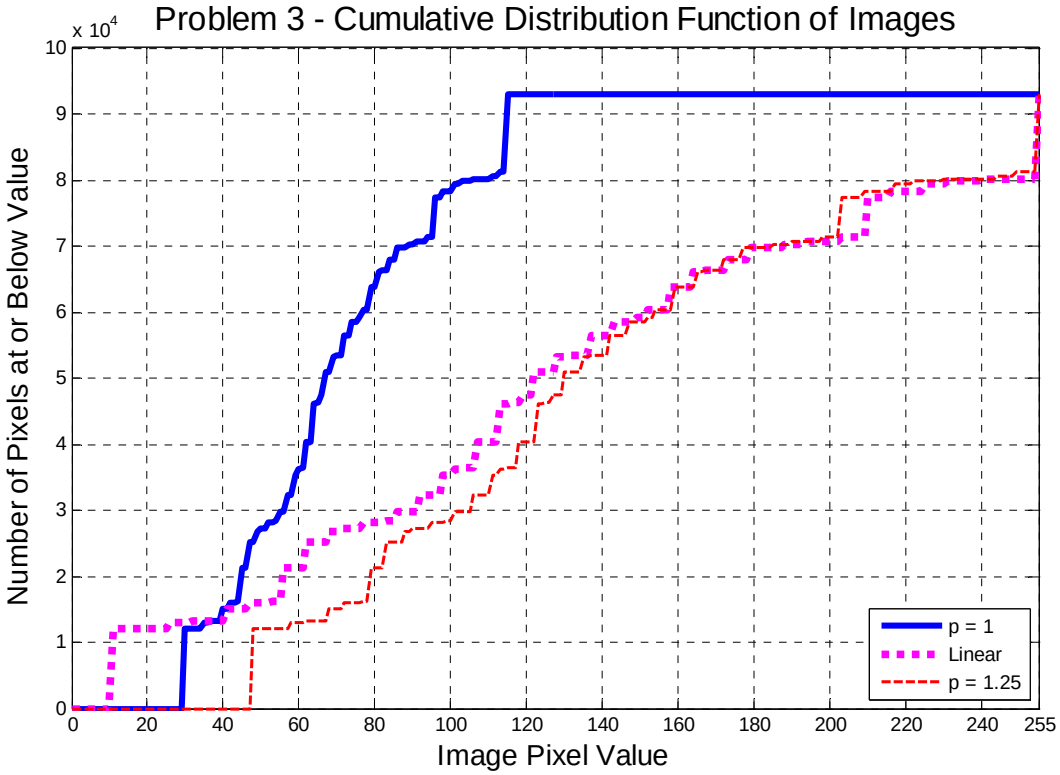
balance between dark features and lighter, grayer tones. We most easily notice these subtleties in the leaves of the palm trees on either side of the Oval and in the distant hills on the horizon. Furthermore, now that we have narrowed the feasible *p* to a shorter range, we can focus on achieving greater precision in honing in to the optimal value:



From this series of images, optimization becomes almost subjectively preferential. It appears that *p* = 1.25 achieves the best compromise between darks and lights, allowing the transformed image values to span the largest dynamic range possible. The ideal image preserves detail while balancing the brightness of the skies and the subtle grays of the grass with the dark leaves of the palm trees:



In order to juxtapose our most recent transformation with our linear endeavor, let us plot the cumulative distribution functions to display how the transformations – both linear and exponential – distribute the pixel values across the dynamic range:



As the cumulative distribution function manifests, the linear transformation and the exponential transform ($p = 1.25$) yield similar performance, with the exception of less action near the dark regions (due to the exponentiation of zero being zero).

Problem #4 – Pseudorandom Topography

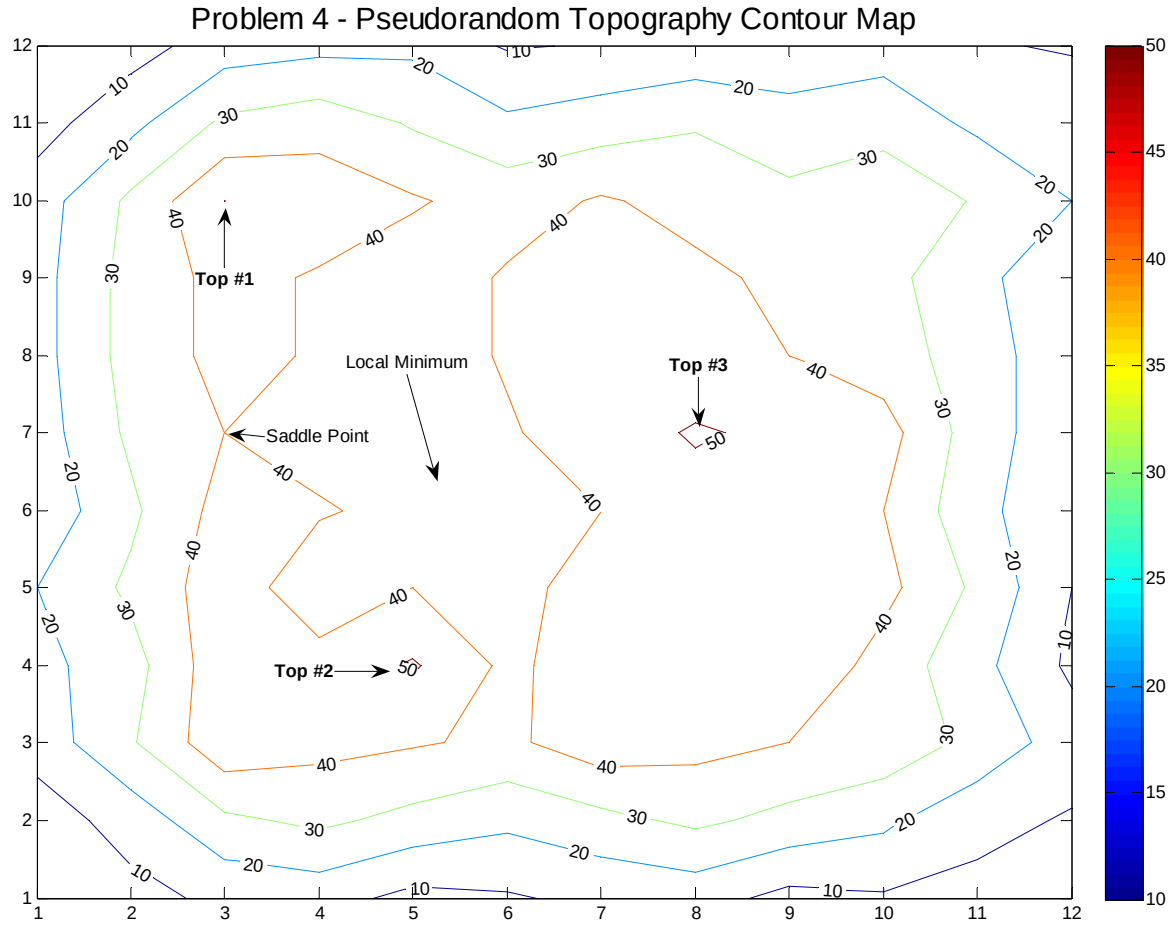
Using the expression $Integral\ Part\ \left\{\frac{1}{2} + 10\ Fractional\ Part\left[1000\ \boldsymbol{sin}\left(\frac{n}{10}\right)\right]\right\}$ modulo 10,

with integer *n* ranging from 0 to 99, we generate the pseudorandom array:

0	7	5	2	0	7	5	2	0	7
5	2	9	7	4	2	9	7	4	2
9	6	4	1	9	6	4	1	8	6
3	1	8	6	3	0	8	5	3	0
8	5	2	0	7	5	2	9	7	4
2	9	6	4	1	8	6	3	1	8
5	3	0	7	5	2	9	7	4	1
9	6	3	1	8	5	3	0	7	4
2	9	6	4	1	8	5	3	0	7
4	2	9	6	3	0	8	5	2	9

We smooth this map by replacing each digit with the sum of the numbers surrounding it (including itself), essentially performing convolution with a 3 × 3 matrix of ones, yielding the 12 × 12 map:

0	7	12	14	7	9	12	14	7	9	7	7
5	14	28	32	27	22	27	32	27	22	13	9
14	29	47	43	41	38	46	43	40	37	27	15
17	26	47	44	51	38	45	42	49	36	23	8
20	32	46	33	40	37	44	40	47	43	28	10
13	28	44	41	37	34	40	46	44	40	23	12
15	32	40	36	32	39	45	51	48	44	25	13
16	34	43	39	35	41	47	43	40	35	25	13
16	34	43	39	35	41	46	42	38	33	23	12
15	32	50	46	41	36	41	37	33	37	29	20
6	17	32	36	29	22	25	29	23	26	18	16
4	6	15	17	18	9	11	13	15	16	11	9



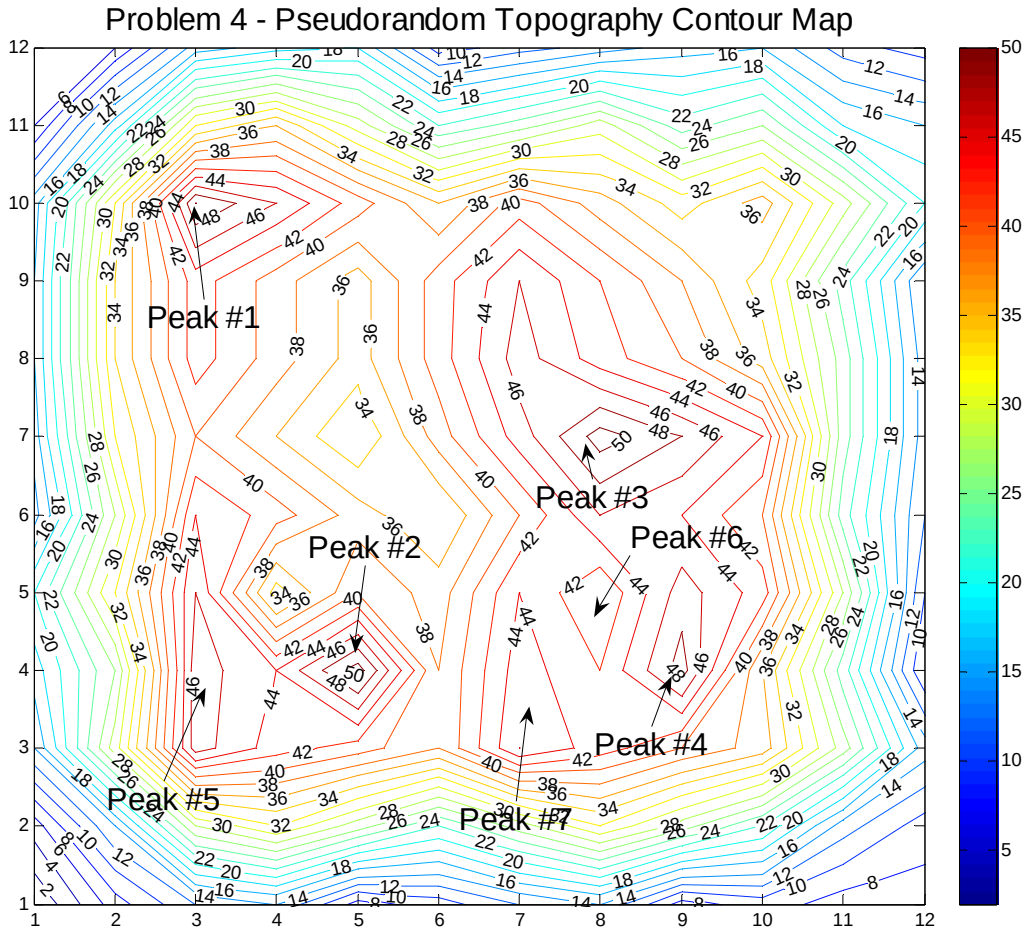
From the contours at 10, 20, 30, 40, and 50, we can discern only **three** peaks, at locations:

Peak #1: $(x, y) = (3, 10)$

Peak #2: $(x, y) = (5, 4)$

Peak #3: $(x, y) = (8, 7)$

Four other peaks exist, but require finer resolution in the contour map than mere 10-unit intervals:



Now, with finer detail, the remaining tops become visible:

Peak #1: $(x, y) = (3, 10)$ at height of 48

Peak #2: $(x, y) = (5, 4)$ at height of 50

Peak #3: $(x, y) = (8, 7)$ at height of 50

Peak #4: $(x, y) = (9, 4)$ at height of 48

Peak #5: $(x, y) = (3.25, 4)$ at height of 46

Peak #6: $(x, y) = (8, 4.5)$ at height of 42

Peak #7: $(x, y) = (7.25, 3.5)$ at height of 44

Three additional eccentric features protrude from the contour map, as confirmed in the more finely spaced contour map:

Saddle Point: $(x, y) = (3, 7)$ at height of 40

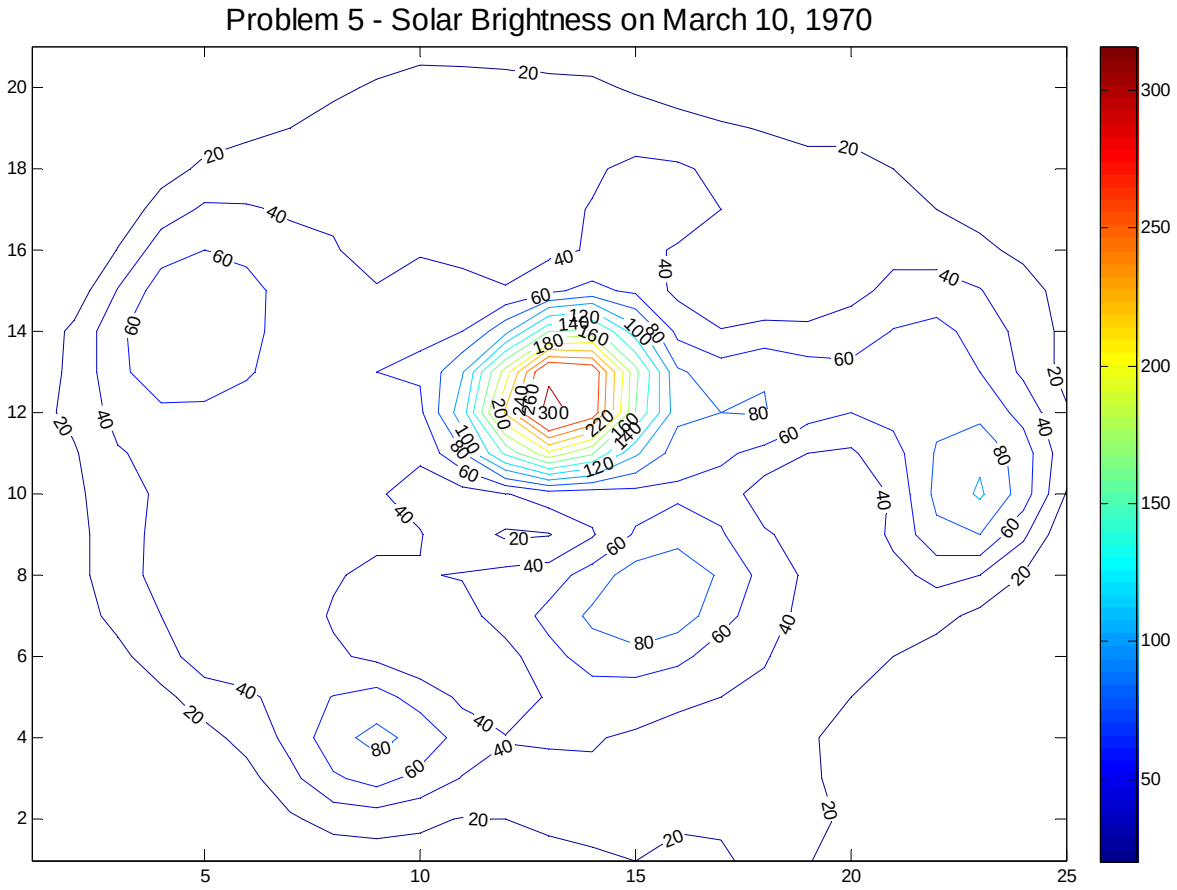
Local Minimum #1: $(x, y) = (5, 7)$ at height of 34

Local Minimum #2: $(x, y) = (4, 5)$ at height of 34

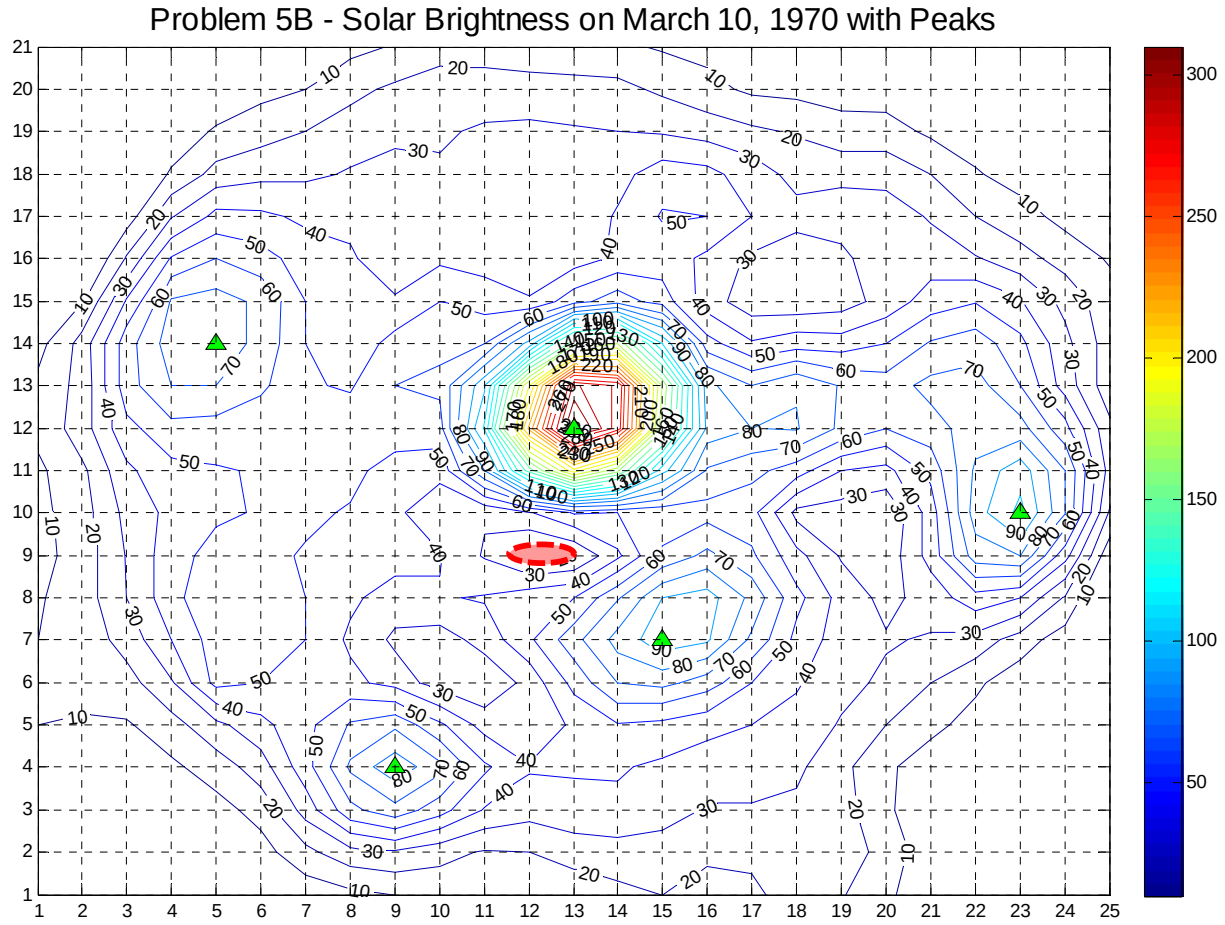
Interestingly, the height of the saddle point falls between the height of the local minima and the height of the peaks. The saddle point represents a crossing of contours, falling toward the minimum at $(x, y) = (5, 7)$ along the horizontal direction but rising toward the peaks at$(x, y) = (3, 10)$ and (3.25, 4) in the vertical directions. Thus, with directional derivatives of differing sign in differing directions, the point $(x, y) = (3, 7)$ truly is a saddle point.

Problem #5 – Topography

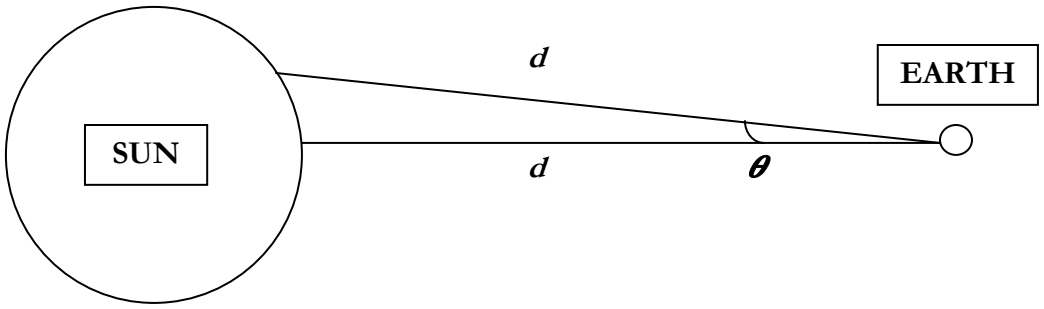
The following topographic map describes the microwave sun as it appeared on March 10, 1970, with temperature contours in units of 1000 Kelvins, at intervals of 20(,000 Kelvins):



We now mark the five distinct peaks with small, shaded green triangles, and the concave region with a shaded red ellipse. However, to discern these features, we must sharpen the resolution by decreasing contour spacing:



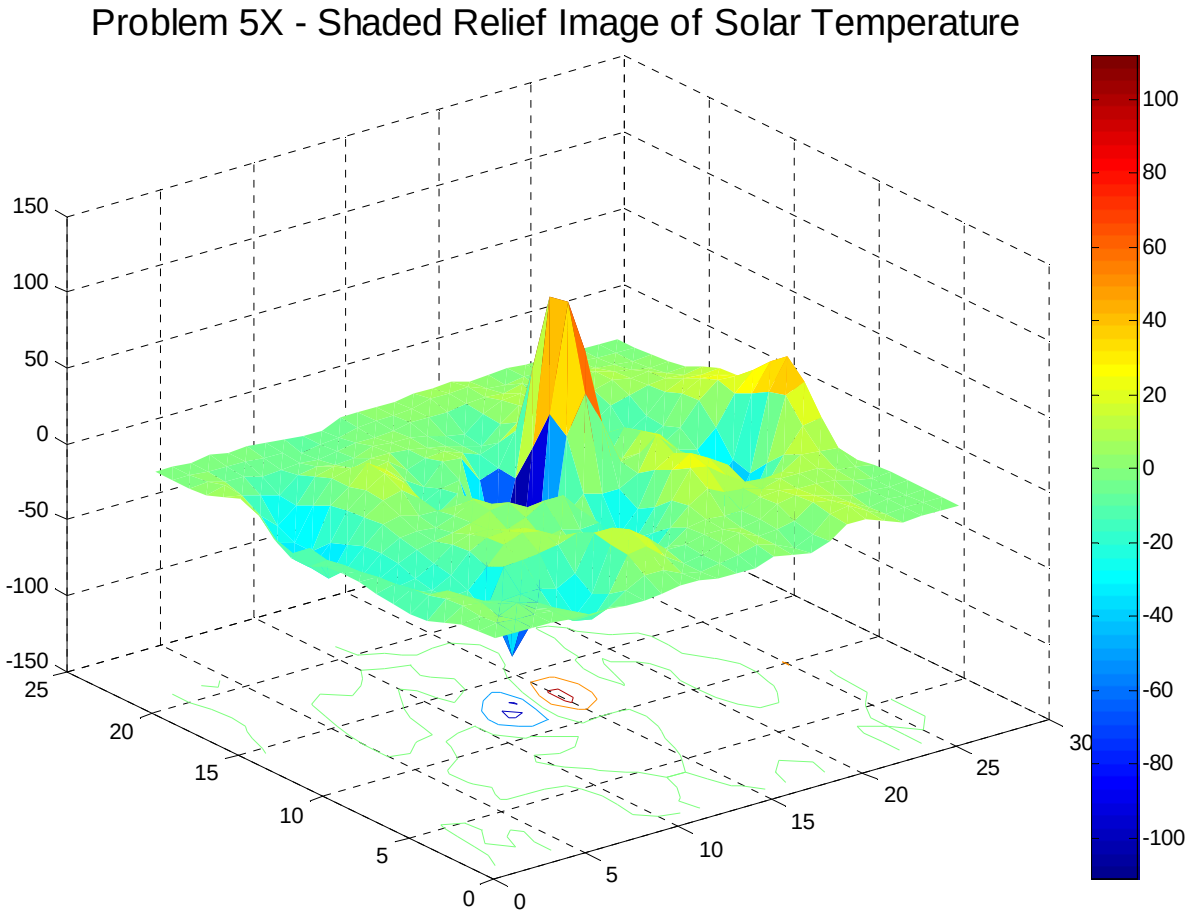
Knowing that the map represents the sun, we can estimate the element spacing using the Law of Cosines:



$$\theta = \boldsymbol{cos}^{-1}\left(\frac{d^2+d^2-\left(\frac{2\cdot R_{sun}}{n_{samples}}\right)^2}{2\cdot d\cdot d}\right) \approx \boldsymbol{cos}^{-1}\left(\frac{2\left(1.496\times10^{11}\ m\right)^2-\left(\frac{2\cdot 6.955\times10^{8}\ m}{25}\right)^2}{2\left(1.496\times10^{11}\ m\right)^2}\right) \approx 55640000 \text{ radians}$$

$\theta \approx 1.278602$ minutes of arc

EXTRA CREDIT: We display the temperature contour as a shaded relief map using Matlab:



Problem #6 – Laser Altimetry

The satellite period is

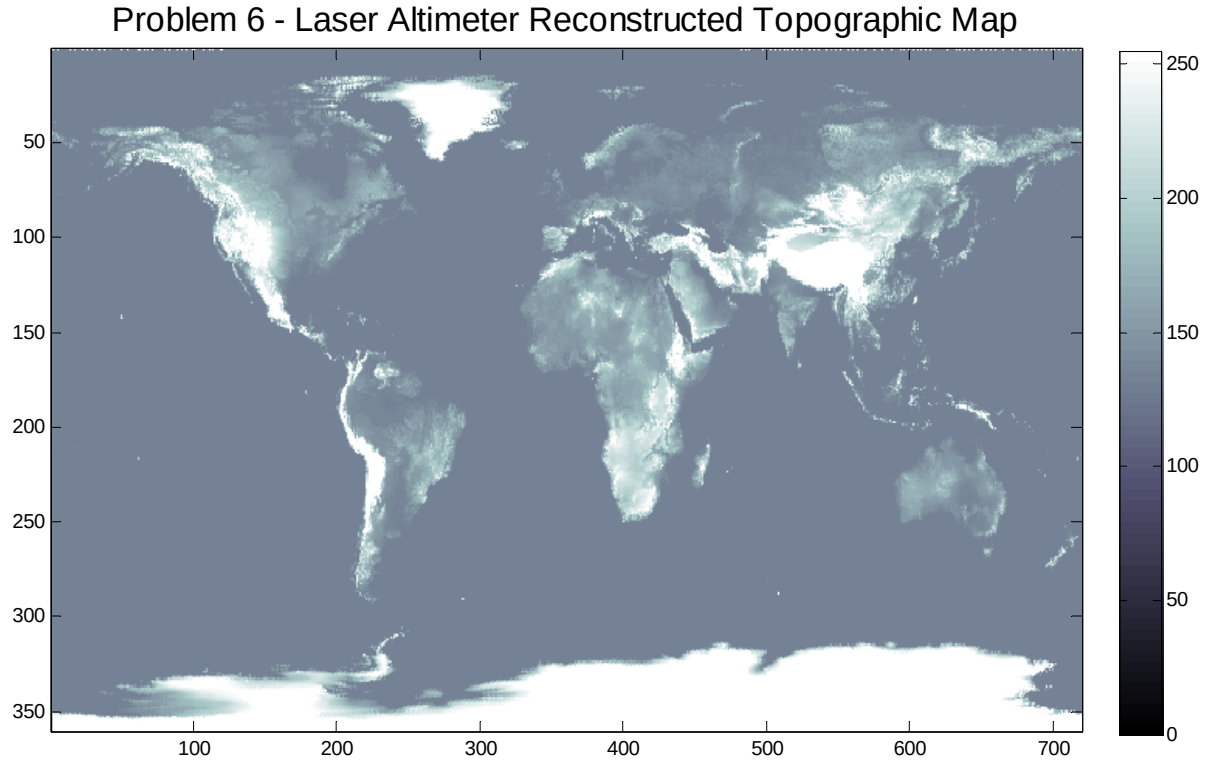
$$T = \frac{2\pi r^{\frac{3}{2}}}{\sqrt{GM_E}} \approx \frac{2\pi(42{,}470{,}426.73\ \text{m})^{\frac{3}{2}}}{\sqrt{\left(6.672\ \times 10^{-11}\ \frac{\text{m}^3}{\text{kg s}^2}\right)(5.9736\ \times 10^{24}\text{kg})}} \approx 86{,}494.547\ \text{sec} \approx 24.0262\ \text{hrs}$$

$T \approx$ 1 day, 1 minute, and 30.0108 seconds

The satellite velocity summarily follows:

$$v = \sqrt{\frac{GM_E}{r}} \approx \sqrt{\frac{\left(6.672\ \times 10^{-11}\ \frac{\text{m}^3}{\text{kg s}^2}\right)(5.9736\ \times 10^{24}\text{kg})}{(42{,}470{,}426.73\ \text{m})}} \approx 3070.632\ \frac{\text{m}}{\text{s}}$$

The topographic map resulting from our laser altimetry assumes the form:



The scale factor used to convert from data numbers to meters of elevation accounts for both the round-trip time of a laser pulse of light traveling at the speed of light *c* and the nanosecond delays:

$$\text{Conversion Factor} = -\frac{2.998 \times 10^{8} \ \frac{\text{meters}}{\cancel{\text{second}}}}{2} \cdot \frac{10^{-9} \ \cancel{\text{second}}}{1 \text{ nanosecond}} \approx 0.2998 \ \frac{\text{meter}}{\text{nanosecond}}$$

Consulting this topographic map, we estimate the following landmark (latitude, longitude) altitudes: The Tibetan Plateau (30°-35°, 80°-95°) has a mean elevation of approximately 4960.741715 meters. Madrid, Spain (40.45° lat, -3.72° long) has an elevation of approximately 633.917876 meters. Greenland (70° to 80°, (-60°) to (-30°)) has a mean elevation of approximately 1988.218497 meters. For the extended regions such as the Tibetan Plateau and Greenland, altitude varies considerably across the surface, so we average the computed altitudes of all these regions to estimate the altitude at the region's center.

Finally, we convert our topographic map into a shaded relief map, exercising the renowned two-dimensional convolution with the kernel $\begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$: