Part 1 - Reading original Landsat 8 imagery files

1) Create functions to read imagery files and store it into matrices

Mathcad's READ_IMAGE sucks. It transforms the 16-bit images into 8-bits. Create another Mathcad sheet to read all the files listed by a bat file that reports is (see spikedefenestrator code) and create a batch that convert automatically all the GEOTIFF into ASC file, using GDAL code. Than, modify this part of the code so the reading will be directly over the ASC, which preserves all the 16-bits information.

DONE!!!

```
CWD := "C:\PhD\2015\00 Papers\Paper 2 - Amazon River\00 - Landsat 8\"

B2(im) := | Im ← READFILE(concat(im, "_B2.asc"), "delimited")
          | submatrix(Im, 5, rows(Im) − 1, 0, cols(Im) − 1)

B3(im) := | Im ← READFILE(concat(im, "_B3.asc"), "delimited")
          | submatrix(Im, 5, rows(Im) − 1, 0, cols(Im) − 1)

B4(im) := | Im ← READFILE(concat(im, "_B4.asc"), "delimited")
          | submatrix(Im, 5, rows(Im) − 1, 0, cols(Im) − 1)

B6(im) := | Im ← READFILE(concat(im, "_B6.asc"), "delimited")
          | submatrix(Im, 5, rows(Im) − 1, 0, cols(Im) − 1)

B1(im) := | Im ← READFILE(concat(im, "_B1.asc"), "delimited")
          | submatrix(Im, 5, rows(Im) − 1, 0, cols(Im) − 1)

B5(im) := | Im ← READFILE(concat(im, "_B5.asc"), "delimited")
          | submatrix(Im, 5, rows(Im) − 1, 0, cols(Im) − 1)
```

the submatrix is to remove the header of ASCII files.

2) Get the UTM extreme coords based on the metadata file

```
Limits(file) := | test ← READTEXT(concat(file, "_MTL.txt"))
                | for i ∈ 0 .. rows(test) − 1
                |   for j ∈ 0 .. cols(test) − 1
                |     if test[i,j] = "CORNER_LL_PROJECTION_X_PRODUCT"
                |       Xinf ← test[i,j+2]
                |     else
                |       if test[i,j] = "CORNER_UR_PROJECTION_X_PRODUCT"
                |         Xsup ← test[i,j+2]
                |       else
                |         if test[i,j] = "CORNER_LL_PROJECTION_Y_PRODUCT"
                |           Yinf ← test[i,j+2]
                |         else
                |           if test[i,j] = "CORNER_UR_PROJECTION_Y_PRODUCT"
                |             Ysup ← test[i,j+2]
                |           else
                |             if test[i,j] = "GEOMETRIC_RMSE_MODEL_Y"
                |               rmseY ← test[i,j+2]
                |             else
                |               if test[i,j] = "GEOMETRIC_RMSE_MODEL_X"
                |                 rmseX ← test[i,j+2]
                |               else
                |                 if test[i,j] = "RADIANCE_MULT_BAND_2"
                |                   ML2 ← test[i,j+2]
                |                 else
                |                   if test[i,j] = "RADIANCE_MULT_BAND_3"
                |                     ML3 ← test[i,j+2]
                |                   else
                |                     if test[i,j] = "RADIANCE_MULT_BAND_4"
                |                       ML4 ← test
```

$\mathrm{Xsup}(\mathrm{file}) := \mathrm{Limits}(\mathrm{file})_{0,0}$ $\quad \mathrm{Xinf}(\mathrm{file}) := \mathrm{Limits}(\mathrm{file})_{1,0}$ $\quad \mathrm{Ysup}(\mathrm{file}) := \mathrm{Limits}(\mathrm{file})_{0,1}$ $\quad \mathrm{Yinf}(\mathrm{file}) := \mathrm{Limits}(\mathrm{file})_{1,1}$

$$\mathrm{PixelX}(\mathrm{utmX}, \mathrm{file}) := \frac{\mathrm{utmX} - \mathrm{Xinf}(\mathrm{file})}{30} \qquad \mathrm{PixelY}(\mathrm{utmY}, \mathrm{file}) := \frac{\mathrm{Ysup}(\mathrm{file}) - \mathrm{utmY}}{30}$$

$$\mathrm{UtmX}(\mathrm{pixelX}, \mathrm{file}) := \mathrm{pixelX} \cdot 30 + \mathrm{Xinf}(\mathrm{file}) \qquad \mathrm{UtmY}(\mathrm{pixelY}, \mathrm{file}) := \mathrm{Ysup}(\mathrm{file}) - \mathrm{pixelY} \cdot 30$$

```
else
  if test[i,j] = "RADIANCE_ADD_BAND_2"
    AL2 ← test[i,j+2]
  else
    if test[i,j] = "RADIANCE_ADD_BAND_3"
      AL3 ← test[i,j+2]
    else
      if test[i,j] = "RADIANCE_ADD_BAND_4"
        AL4 ← test[i,j+2]
      else
        continue
```

$$\begin{bmatrix} Xsup & Ysup & ML2 & AL2 \\ Xinf & Yinf & ML3 & AL3 \\ rmseX & rmseY & ML4 & AL4 \end{bmatrix}$$

Filtering

Gaussian kernel to filter image.

$$kernel := \frac{1}{16} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$border := (rows(kernel) - 1) \cdot 0.5 = 1$$

$$\frac{1}{273} \cdot \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix} \quad \frac{1}{256} \cdot \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

[1]

$$\frac{1}{331} \cdot \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 20 & 33 & 20 & 4 \\ 7 & 33 & 55 & 33 & 7 \\ 4 & 20 & 33 & 20 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

```
Flt(Image, WindowSize) := | A ← Image
                          | for i ∈ 0 + border .. rows(A) − 1 − border
                          |   for j ∈ 0 + border .. cols(A) − 1 − border
                          |     K ← kernel
                          |     for m ∈ 0 .. rows(kernel) − 1
                          |       for n ∈ 0 .. cols(kernel) − 1
                          |         W[m,n] ← A[i − border + m, j − border + n]
                          |     B[i,j] ← Σ(m=0..rows(W)−1) Σ(n=0..rows(W)−1) K[m,n] · W[m,n]
                          | submatrix(B, border, rows(B) − 1, border, cols(B) − 1)
```

$$G(\sigma, x, y) := \frac{1}{\sqrt{2 \cdot \pi} \cdot \sigma} \cdot e^{-\left(\frac{(x-2)^2 + (y-2)^2}{2 \cdot \sigma^2}\right)}$$

$$G(1,0,0) = 0.007 \qquad \frac{G(1,2,2)}{G(1,0,0)} = 54.598$$

$$G(1,2,2) = 0.399$$

$$\frac{1}{25} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$\frac{1}{16} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad \frac{1}{9} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{49} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$
\sigma filt(im) := \left\|
\begin{array}{l}
A \leftarrow im \\
\text{for } i \in 0+border..rows(A)-1-border \\
\quad \left\| \begin{array}{l}
\text{for } j \in 0+border..cols(A)-1-border \\
\quad \left\| \begin{array}{l}
K \leftarrow kernel \\
\text{for } m \in 0..rows(kernel)-1 \\
\quad \left\| \begin{array}{l}
\text{for } n \in 0..cols(kernel)-1 \\
\quad \left\| W_{m,n} \leftarrow A_{i-border+m,\,j-border+n} \right.
\end{array} \right. \\
B_{i,j} \leftarrow \sqrt{(5\%)^2 \cdot \left(\sum_{m=0}^{rows(W)-1} \sum_{n=0}^{rows(W)-1} \left(\left(K_{m,n}\right)^2 \cdot \left(W_{m,n}\right)^2 \right) \right)}
\end{array} \right.
\end{array} \right. \\
submatrix(B, border, rows(B)-1, border, cols(B)-1)
\end{array} \right.
$$

$$
\sigma im(im) := \left\|
\begin{array}{l}
A \leftarrow im \\
\text{for } i \in 0+border..rows(A)-1-border \\
\quad \left\| \begin{array}{l}
\text{for } j \in 0+border..cols(A)-1-border \\
\quad \left\| B_{i,j} \leftarrow A_{i,j} \cdot 5\% \right.
\end{array} \right. \\
submatrix(B, 1, rows(B)-1, 1, cols(B)-1)
\end{array} \right.
$$

Part 2 - Creating a UTM mask and cutting original images

Define the pixel limits in one image. Then, calculate its UTM coordinates and use it to find the pixel coordinates of the other images relative to the mask defined in the first image.

$x1 := 2136$ $y1 := 1272$ $x2 := x1 + 2999$ $y2 := y1 + 2999$ Reference: "LC80800162015172LGN00"

$PrintFactor := \frac{1}{256}$

$UtmXinf := UtmX(x1, \text{"LC82250602013270LGN00"})$ $UtmYinf := UtmY(y2, \text{"LC82250602013270LGN00"})$

$UtmXsup := UtmX(x2, \text{"LC82250602013270LGN00"})$ $UtmYsup := UtmY(y1, \text{"LC82250602013270LGN00"})$

$UtmYsup = 77640$ $UtmXinf = 488280$

$UtmYinf = -12330$ $UtmXsup = 578250$

$PixelXsup(file) := PixelX(UtmXsup, file)$ $PixelYsup(file) := PixelY(UtmYsup, file)$

$PixelXinf(file) := PixelX(UtmXinf, file)$ $PixelYinf(file) := PixelY(UtmYinf, file)$

$MB1(file) := submatrix(B1(file), PixelYsup(file) - border, PixelYinf(file) + border, PixelXinf(file) - border, PixelXsup(file) + border)$

$MB2(file) := submatrix(B2(file), PixelYsup(file) - border, PixelYinf(file) + border, PixelXinf(file) - border, PixelXsup(file) + border)$

$MB3(file) := submatrix(B3(file), PixelYsup(file) - border, PixelYinf(file) + border, PixelXinf(file) - border, PixelXsup(file) + border)$

$MB4(file) := submatrix(B4(file), PixelYsup(file) - border, PixelYinf(file) + border, PixelXinf(file) - border, PixelXsup(file) + border)$

$MB6(file) := submatrix(B6(file), PixelYsup(file) - border, PixelYinf(file) + border, PixelXinf(file) - border, PixelXsup(file) + border)$

$MB5(file) := submatrix(B5(file), PixelYsup(file) - border, PixelYinf(file) + border, PixelXinf(file) - border, PixelXsup(file) + border)$

M stands for Mask, MB2 is the mask applied to channel Blue B2. The +1 and -1 increases in size at image matrix are due to filtering, where the extra pixel are added just for coherence during filtering process and then removed.

address := "C:\PhD\2015\00 Papers\Paper 2 - Amazon River\"

$$PrintMask(file) := \left\| \begin{array}{l} B6 \leftarrow MB6(file) \\ WRITEBMP(concat(address, file, \text{"\_B6\_mask.tif"}), scale(B6, 0, 255)) \end{array} \right.$$

$$Header := \left\| \begin{array}{l} Out_0 \leftarrow concat(\text{"ncols "}, num2str(1000)) \\ Out_1 \leftarrow concat(\text{"nrows "}, num2str(1000)) \\ Out_2 \leftarrow concat(\text{"xllcorner "}, num2str(UtmXinf)) \\ Out_3 \leftarrow concat(\text{"yllcorner "}, num2str(UtmYinf)) \\ Out_4 \leftarrow \text{"cellsize 30"} \\ Out_5 \leftarrow \text{"nodata\_value 2410"} \\ Out \end{array} \right.$$

A Comparative Study of Removal Noise from Remote Sensing Image, 2010

Reading imagery files

CWD := "C:\PhD\2015\00 Papers\Paper 2 - Amazon River"

$$Files := \left\| \begin{array}{l} Log0 \leftarrow READTEXT(\text{"arquivos.log"}) \\ \text{for } i \in 0..rows(Log0) - 1 \\ \quad \left\| \begin{array}{l} txt \leftarrow substr\left(Log0_i, 0, strlen\left(Log0_i\right) - 8\right) \\ Log_{i,0} \leftarrow txt \\ Log_{i,1} \leftarrow str2num(substr(txt, 9, 7)) \end{array} \right. \\ csort(Log, 1)^{\langle 0 \rangle} \end{array} \right.$$

$$Files = \begin{bmatrix} \text{"LT52250601986196XXX03"} \\ \text{"LT52250601987183CUB00"} \\ \text{"LT52250601996240CUB00"} \\ \text{"LT52250602000251CUB00"} \\ \text{"LT52250602006299CUB00"} \\ \text{"LT52250602008145CUB00"} \\ \text{"LC82250602013270LGN00"} \\ \text{"LC82250602014257LGN00"} \\ \text{"LE72250602015236CUB00"} \end{bmatrix}$$

index := 8

tempo1 := time(index)

File := $Files_{index}$

Table of Original Imagery Channels, Masked

Blue Channel (B2)	Green Channel (B3)	Red Channel (B4)	NIR Channel (B6)
i3B2 := MB2(File)	i3B3 := MB3(File)	i3B4 := MB4(File)	i3B6 := MB6(File)

Table of Imagery Channels, Masked and Filtered, Reading

Blue Channel (B2)	Green Channel (B3)	Red Channel (B4)	NIR Channel (B6)
im3B2 := Flt(i3B2, 3)	im3B3 := Flt(i3B3, 3)	im3B4 := Flt(i3B4, 3)	im3B6 := Flt(i3B6, 3)

$$\mathrm{WRITEBMP}\left(\text{“B4B6.tif”}, \frac{-(i3B6 - i3B4)}{255}\right) \qquad \mathrm{WRITEBMP}\left(\text{“B2B6.tif”}, \frac{-(i3B6 - i3B2)}{255}\right)$$

$$\mathrm{WRITERGB}\left(\text{“RGB.tif”}, \frac{\mathrm{augment}(i3B4, i3B3, i3B2)}{255}\right)$$

```
NDWI := | for i ∈ 0..rows(im3B6) − 1
        |   for j ∈ 0..cols(im3B6) − 1
        |     if (im3B3_{i,j} ≠ 0) ∧ (im3B6_{i,j} ≠ 0)
        |       Out_{i,j} ← (im3B4_{i,j} − im3B6_{i,j}) / (im3B4_{i,j} + im3B6_{i,j})
        |       if Out_{i,j} < 0
        |         Out_{i,j} ← 0
        |     else
        |       Out_{i,j} ← 0
        | maxi ← max(Out)
        | for i ∈ 0..rows(Out) − 1
        |   for j ∈ 0..cols(Out) − 1
        |     Out2_{i,j} ← round(Out_{i,j} · 255/maxi, 0)
        | address2 ← “C:\PhD\2015\00 Papers\Paper 2 - Amazon River\04 - Outputs Mathcad\”
        | WRITEBMP(concat(address2, File, “_NDWI.tif”), Out2)
        | Out2
```

```
Mask(im) := | for i ∈ 0..rows(im) − 1
            |   for j ∈ 0..cols(im) − 1
            |     if 0 < NDWI_{i,j}
            |       Im_{i,j} ← im_{i,j}
            |     else
            |       Im_{i,j} ← −9999
            | Im
```

i3B6 –

i3B6 –

```
Escala := | maxi ← max(im3B6)        = 1
          | if maxi < 256
          |   | Escala ← 1
          | else
          |   | Escala ← 1/256
```

Amp := 1

i3B6 –

address = "C:\PhD\2015\00 Papers\Paper 2 - Amazon River\"

Table of Imagery Channels, Masked, Filtered, without Land, Reading.

Blue Channel (B2)	Green Channel (B3)	Red Channel (B4)	NIR Channel (B6)
im3B2W := Mask(im3B2)	im3B3W := Mask(im3B3)	im3B4W := Mask(im3B4)	im3B6W := Mask(im3B6)

address2 := "C:\PhD\2015\00 Papers\Paper 2 - Amazon River\04 - Outputs Mathcad\"

```
WRITEBMP(concat(address2, File, "_WaterSep.tif"), zoom(im3B6W · Escala, Amp, Amp)) = 0
WRITEBMP(concat(address2, File, "_MaxSD.tif"), zoom(MaxSDIm(im3B6) · Escala, Amp, Amp))
WRITEBMP(concat(address2, "im3B6_filter_part.tif"), zoom(im3B6_Laplace · Escala, Amp, Amp))
WRITERGB(concat(address2, "im3B6_uncert_part.tif"), zoom(ImUncert(im3B6, minT3, maxT3) · Escala, Amp, Amp))
WRITERGB(concat(address2, File, "_color.bmp"), zoom(augment(im3B4, im3B3, im3B2) · Escala, Amp, Amp))
```

```
teste2(im1, im2) := | Im1 ← im1
                    | Im2 ← im2
                    | for i ∈ 0 .. rows(Im1) − 1
                    |   | for j ∈ 0 .. rows(Im1) − 1
                    |   |   | if Im1_{i,j} > 0 ∧ Im2_{i,j} > 0
                    |   |   |   | a_{i,j} ← ln(Im1_{i,j} / Im2_{i,j})
                    |   |   | else
                    |   |   |   | a_{i,j} ← −9999
                    | a
```

```
| LogRatio ← teste2(im3B3W, im3B4W)        = −2.648
| for i ∈ 0 .. rows(LogRatio) − 1
|   | for j ∈ 0 .. rows(LogRatio) − 1
|   |   | if LogRatio_{i,j} > −9999
|   |   |   | a_{cont} ← LogRatio_{i,j}
|   |   |   | cont ← cont + 1
| mean(a)
| stdev(a)
| max(a)
| min(a)
```

index = 8

$$CortePixel := \begin{bmatrix} 120 \\ 110 \\ 110 \\ 105 \\ 110 \\ 110 \\ 115 \\ 120 \\ 115 \end{bmatrix}$$

```
CortePixel ← CortePixel_{index}                   = [ −0.364  0.275  −0.025  0.048 ]
LogRatio ← teste2(im3B3W, im3B4W)
for i ∈ 0 .. rows(LogRatio) − 1
  | for j ∈ 0 .. rows(LogRatio) − 1
  |   | if LogRatio_{i,j} > −9999
  |   |   | a_{cont} ← LogRatio_{i,j}
  |   |   | cont ← cont + 1
mini ← min(a)
maxi ← max(a)
μ ← mean(a)
σ ← Stdev(a)
for i ∈ 0 .. rows(a) − 1
  | if μ + 3 · σ > a_i > μ − 3 · σ
  |   | b_{cont} ← a_i
  |   | cont ← cont + 1
a ← b
mini ← min(a)
```

CWD = "C:\PhD\2015\00 Papers\Paper 2 - Amazon River"

```
mini ← min(a)
maxi ← max(a)
μ ← mean(a)
σ ← Stdev(a)

for i ∈ 0..rows(LogRatio) − 1
  for j ∈ 0..rows(LogRatio) − 1
    if LogRatio[i,j] > −9999
      if trunc(((maxi − LogRatio[i,j])·255)/(maxi − mini)) < CortePixel
        Out[i,j] ← trunc(((maxi − LogRatio[i,j])·255)/(maxi − mini))
      else
        Out[i,j] ← 100
    else
      Out[i,j] ← 0
WRITEBMP(concat(address2, File, "_LogRatioB3B4.bmp"), zoom(Out, Amp, Amp))
[mini  maxi  μ  σ]
```

CWD = "C:\PhD\2015\00 Papers\Paper 2 - Amazon River"

$$HeaderTFW := WRITEEXCEL\left(concat(\mathbf{CWD}, \text{"\textbackslash 04 - Outputs Mathcad\textbackslash HeaderTFW.xlsx"}), \begin{bmatrix} 30 \\ 0 \\ 0 \\ -30 \\ UtmXinf \\ UtmYsup \end{bmatrix}\right)$$

$$HeaderTFW = \begin{bmatrix} 30 \\ 0 \\ 0 \\ -30 \\ 488280 \\ 77640 \end{bmatrix}$$

tempo2 := time(index)

$\Delta t := tempo2 - tempo1 = 290.371$

address2 = "C:\PhD\2015\00 Papers\Paper 2 - Amazon River\04 - Outputs Mathcad\"

```
teste2(im1, im2) := | Im1 ← im1
                    | Im2 ← im2
                    | for i ∈ 0..rows(Im1) − 1
                    |   for j ∈ 0..rows(Im1) − 1
                    |     if Im1[i,j] > 0 ∧ Im2[i,j] > 0
                    |       a[i,j] ← ln(Im1[i,j] / Im2[i,j])
                    |       b[cont] ← a[i,j]
                    |       cont ← cont + 1
                    |     else
                    |       a[i,j] ← −9999
                    | minb ← min(b)
                    | maxb ← max(b)
                    | for i ∈ 0..rows(Im1) − 1
                    |   for j ∈ 0..rows(Im1) − 1
                    |     if Im1[i,j] > 0 ∧ Im2[i,j] > 0
                    |       a[i,j] ← (a[i,j] − minb) · 255/(maxb − minb)
                    |     else
                    |       a[i,j] ← 0
                    | a
```

address3 := "C:\PhD\2015\00 Papers\Paper 2 - Amazon River\04 - Outputs Mathcad\Log Ratio full\"

WRITEBMP(concat(address3, File, "_Full_LogRatioB3B4.bmp"), zoom(teste2(im3B3W, im3B4W), Amp, Amp)) = 0