GIST 8118 Lecture 04
Spatial Enhancement

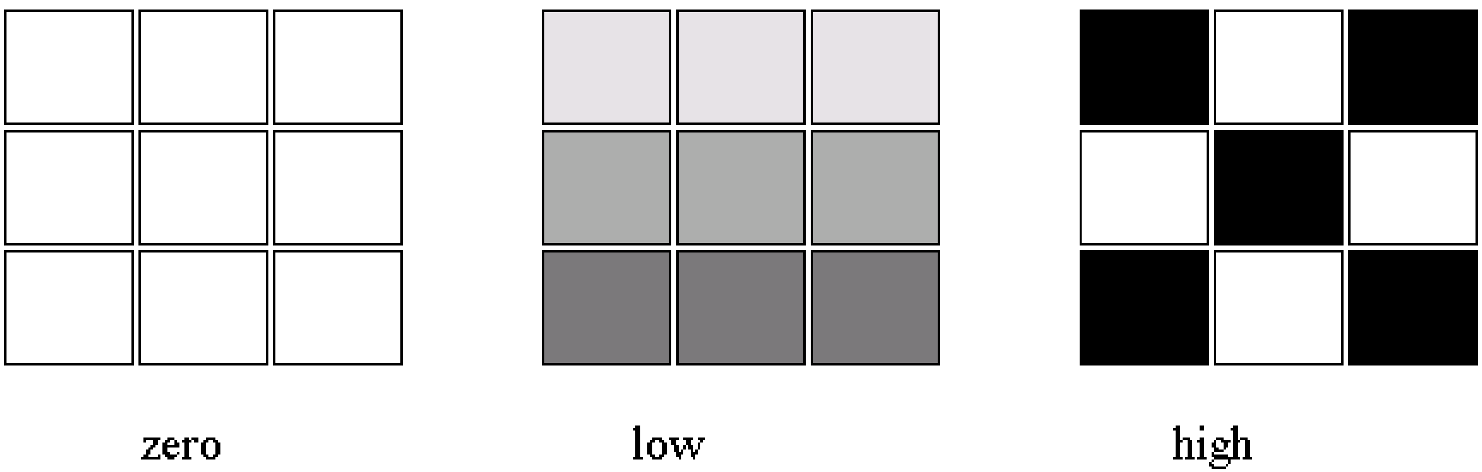
- Lecture
  - Spatial Frequency
  - Filtering
  - Edge detection
  - Edge Enhancement
  - Edge Mapping
- Lab
  - Module 4
    - Filtering
    - Create a line network

Why Filter

- Extract useful information
  - Find roads in an area
  - Identify areas of vegetation vs man made structures
- Create vector lines for
    - Shoreline
    - Boundaries
    - Roads
  - Look for Edges
- Identify areas of sameness

Spatial frequency

- number of changes in brightness value per unit distance in any part of an image
  - low frequency - tonally smooth, gradual changes
  - high frequency - tonally rough, abrupt changes



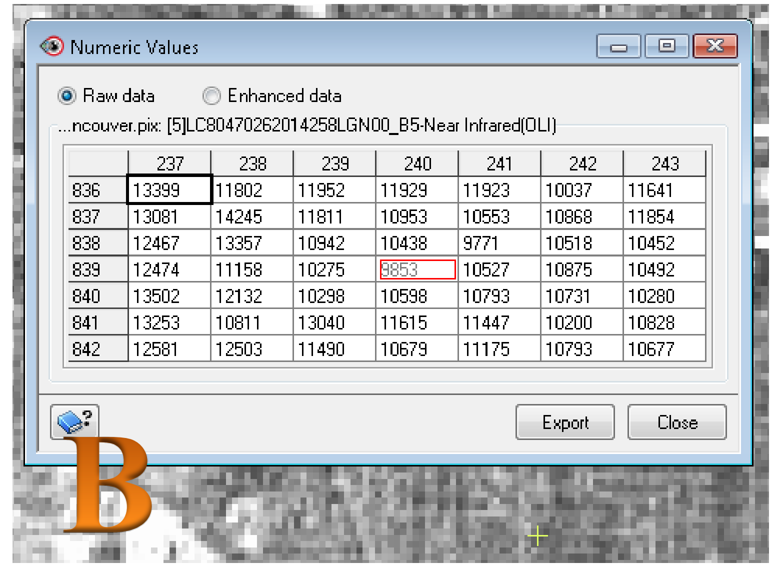
Vancouver Example

Identify the frequency of the three (A B C) images

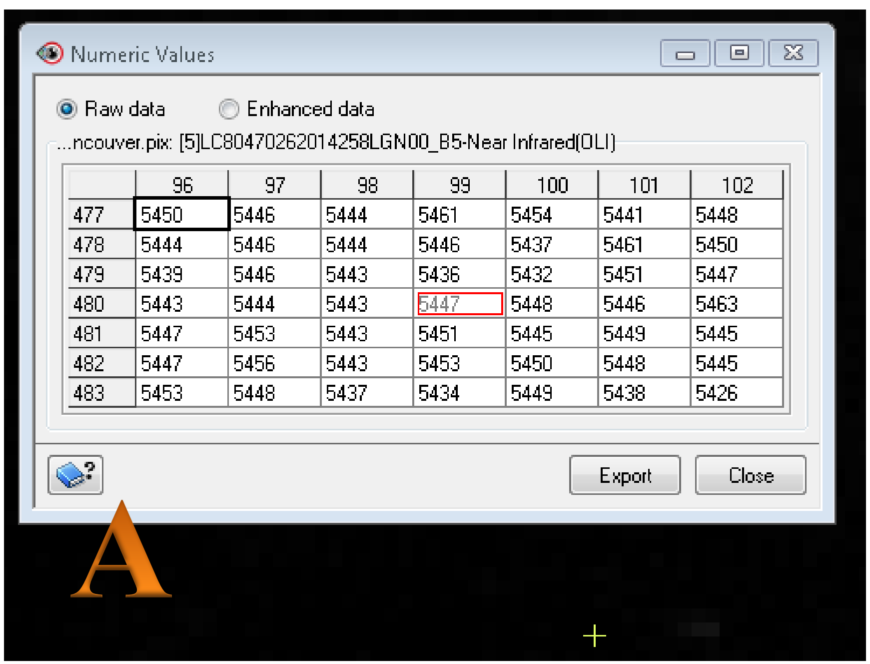
Low -

Medium -

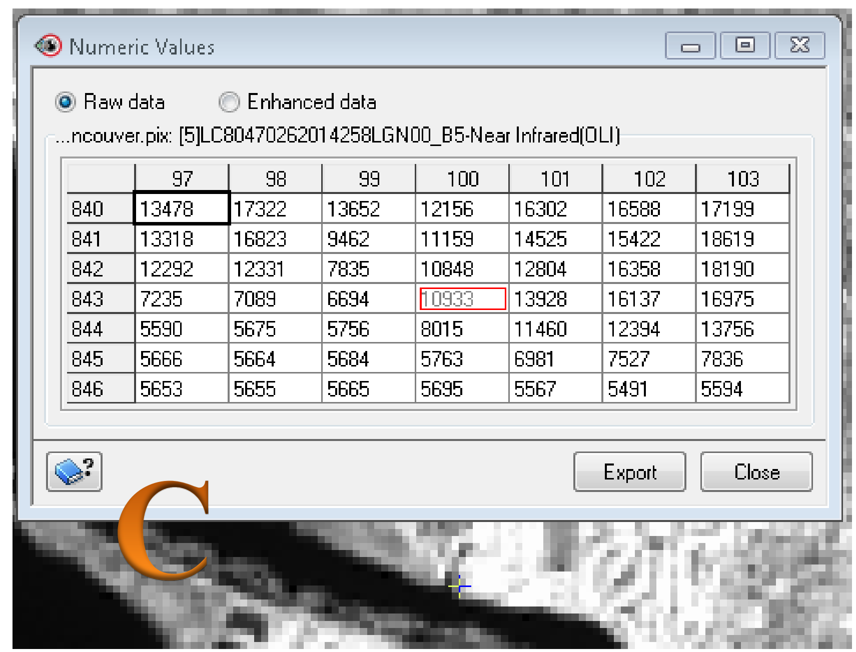
High -



	237	238	239	240	241	242	243
836	13399	11802	11952	11929	11923	10037	11641
837	13081	14245	11811	10953	10553	10868	11854
838	12467	13357	10942	10438	9771	10518	10452
839	12474	11158	10275	8853	10527	10875	10492
840	13502	12132	10298	10598	10793	10731	10280
841	13253	10811	13040	11615	11447	10200	10828
842	12581	12503	11490	10679	11175	10793	10677

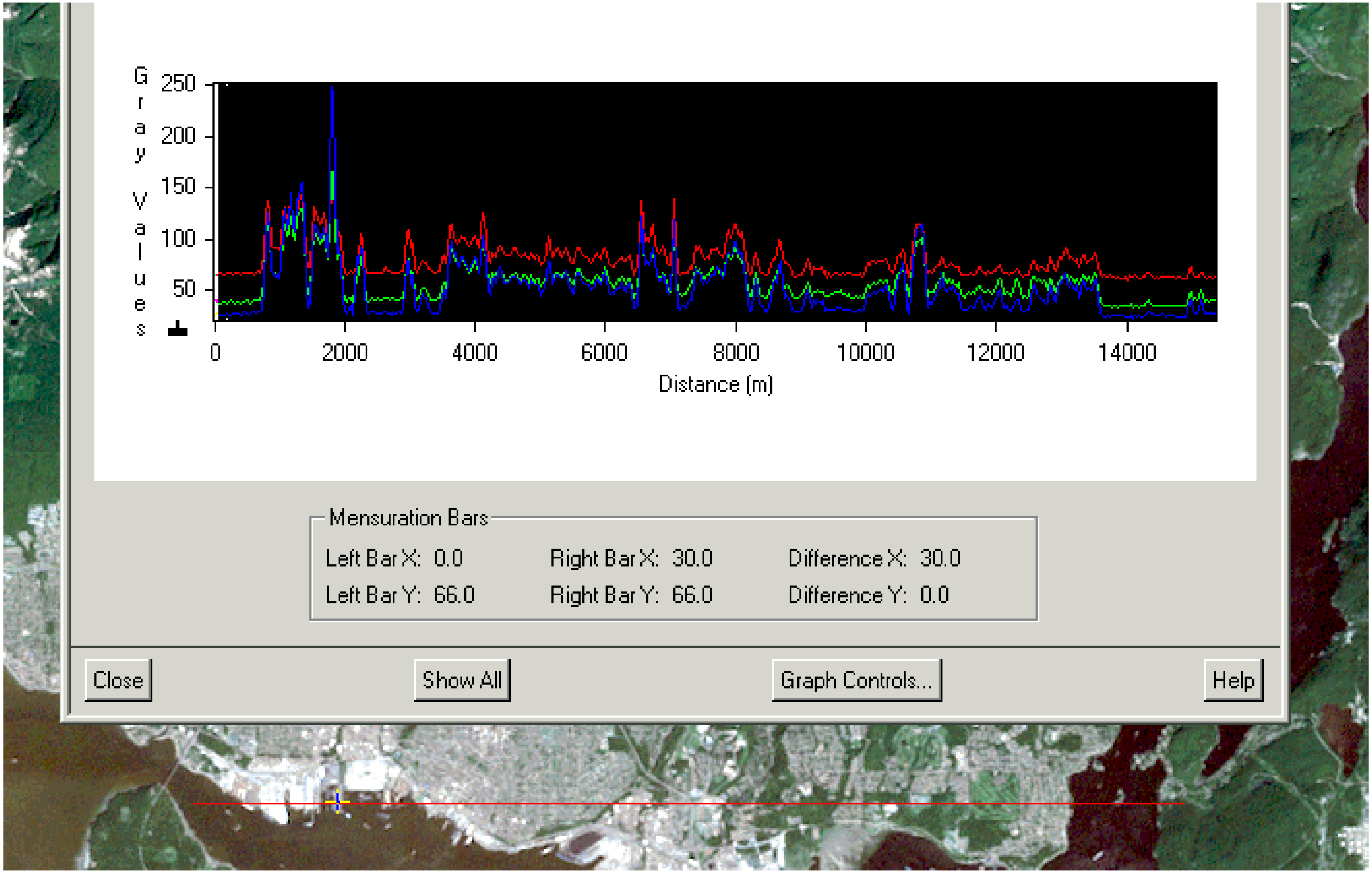


	96	97	98	99	100	101	102
477	5450	5446	5444	5461	5454	5441	5448
478	5444	5446	5444	5446	5437	5461	5450
479	5439	5446	5443	5436	5432	5451	5447
480	5443	5444	5443	5447	5448	5446	5463
481	5447	5453	5443	5451	5445	5449	5445
482	5447	5456	5443	5453	5450	5448	5445
483	5453	5448	5437	5434	5449	5438	5426



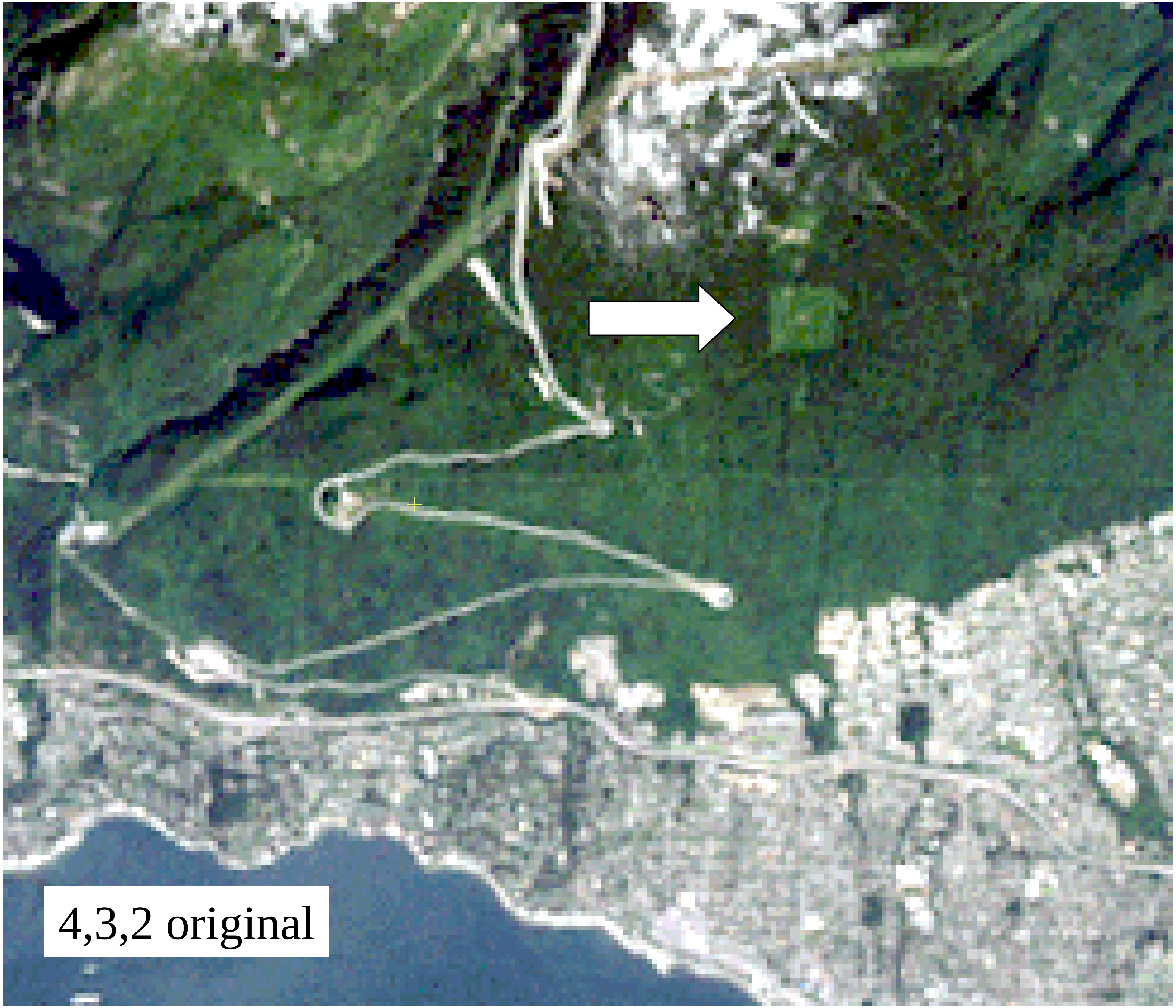
	97	98	99	100	101	102	103
840	13478	17322	13652	12156	16302	16588	17199
841	13318	16823	9462	11159	14525	15422	18619
842	12292	12331	7835	10848	12804	16358	18190
843	7235	7089	6694	10933	13928	16137	16975
844	5590	5675	5756	8015	11460	12394	13756
845	5666	5664	5684	5763	6981	7527	7836
846	5653	5655	5665	5695	5567	5491	5594

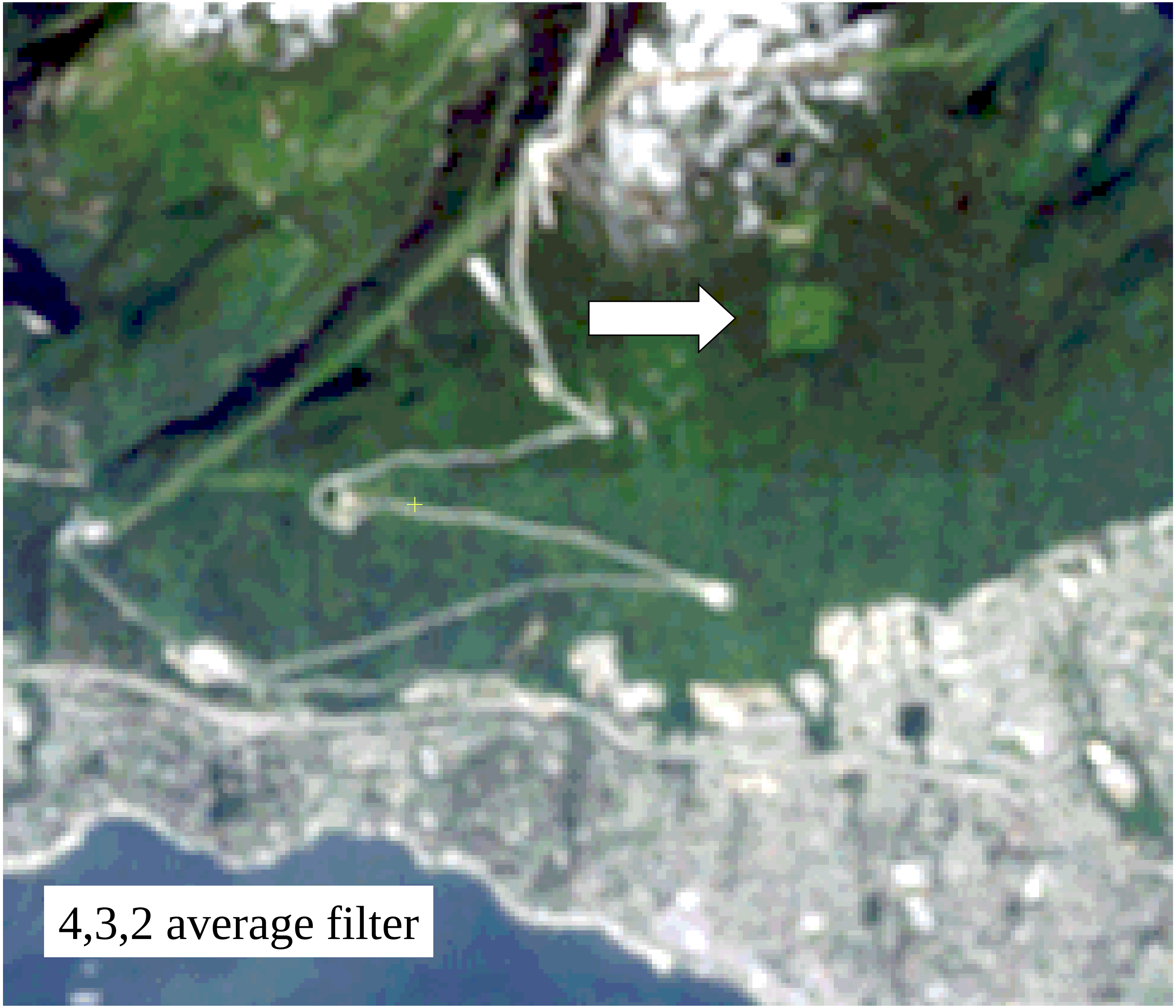
Profile of Vancouver

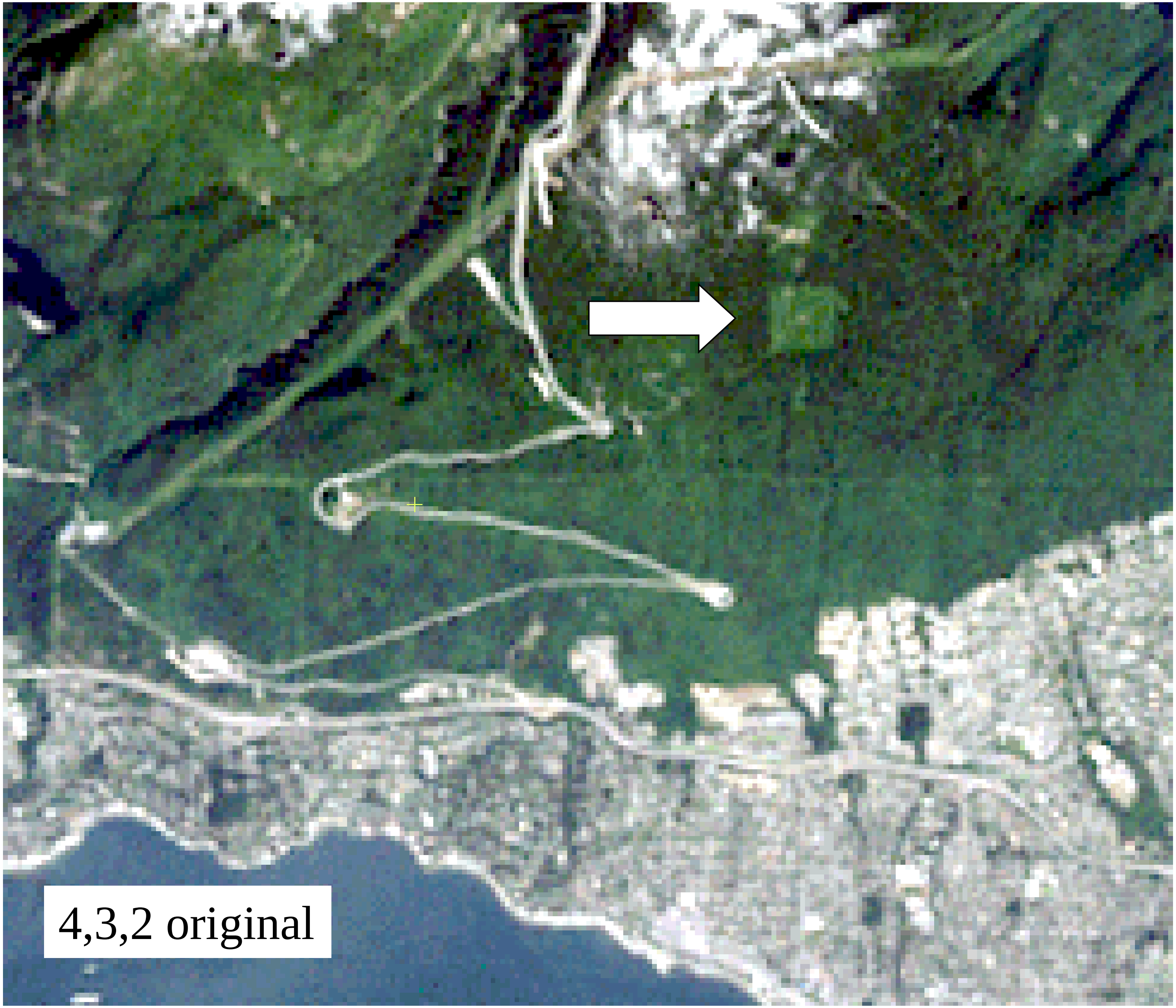


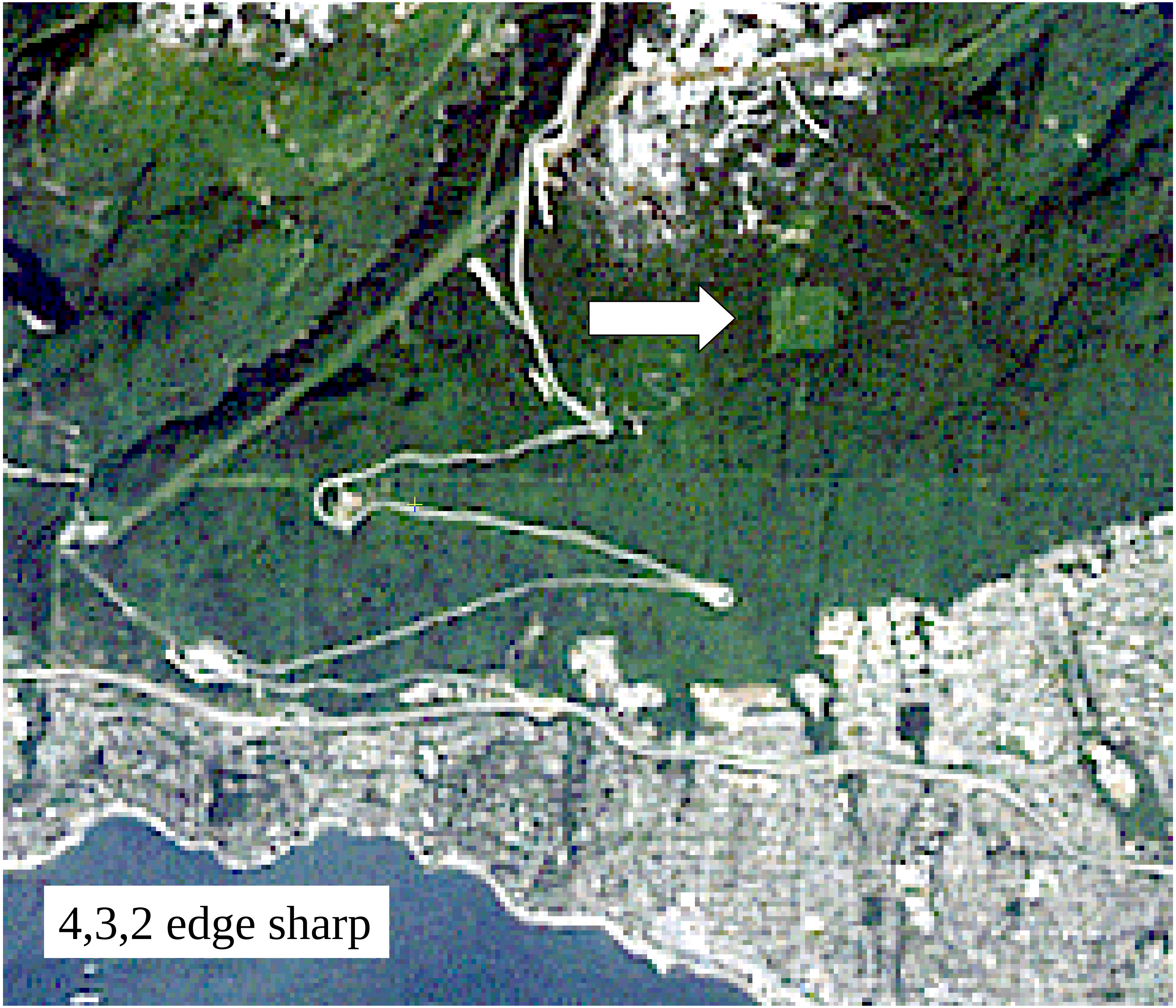
Spatial Filtering

- Can adjust frequency by looking at surrounding DN's and adjusting pixel DN
- Spatial filters suppress certain frequencies and pass or emphasize other frequencies
  - High pass filter
    - emphasize fine detail and highlight edges, pass high frequencies
      - aka high frequency enhancers
  - Low pass filters
    - pass low frequencies and emphasize gradual change smooth image remove noise
      - aka low frequency enhancers


4,3,2 original

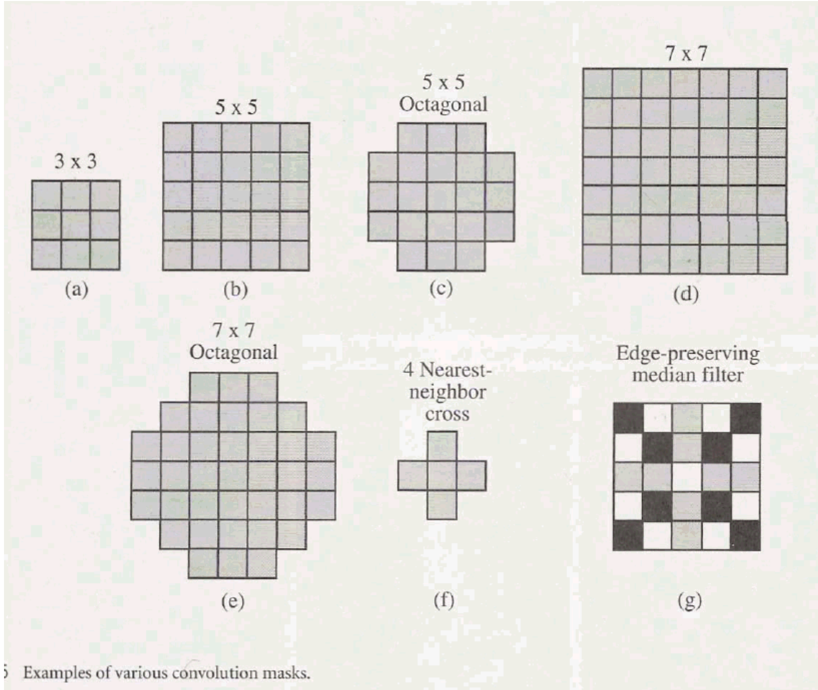

4,3,2 average filter


4,3,2 original


4,3,2 edge sharp

Spatial Filtering

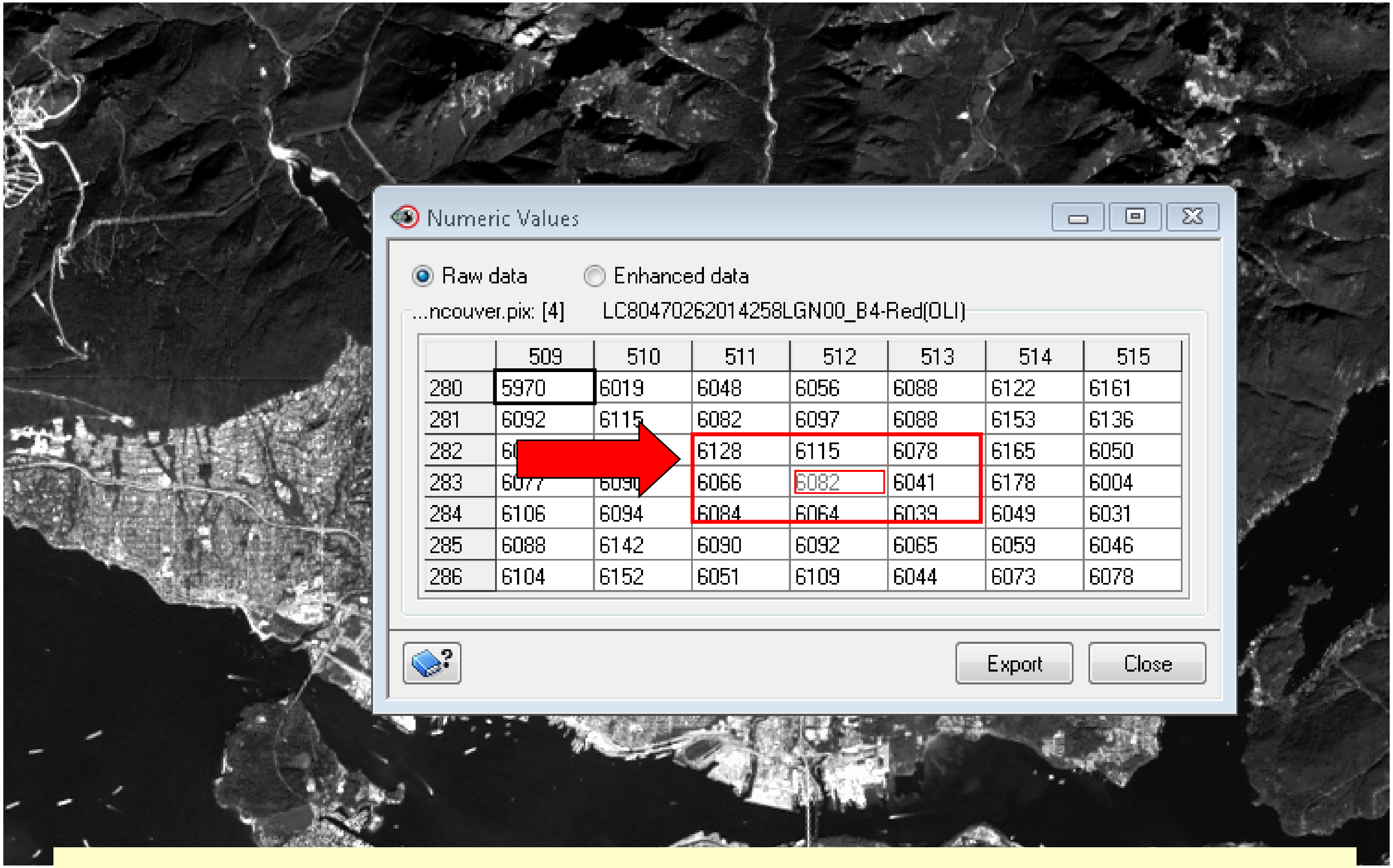
- Requires
  - filter
    - Kernel
    - Convolution mask
  - Image
- Results in
  - Filtered image
    - New channel with data in database



Example Average filter

1	1	1
1	1	1
1	1	1

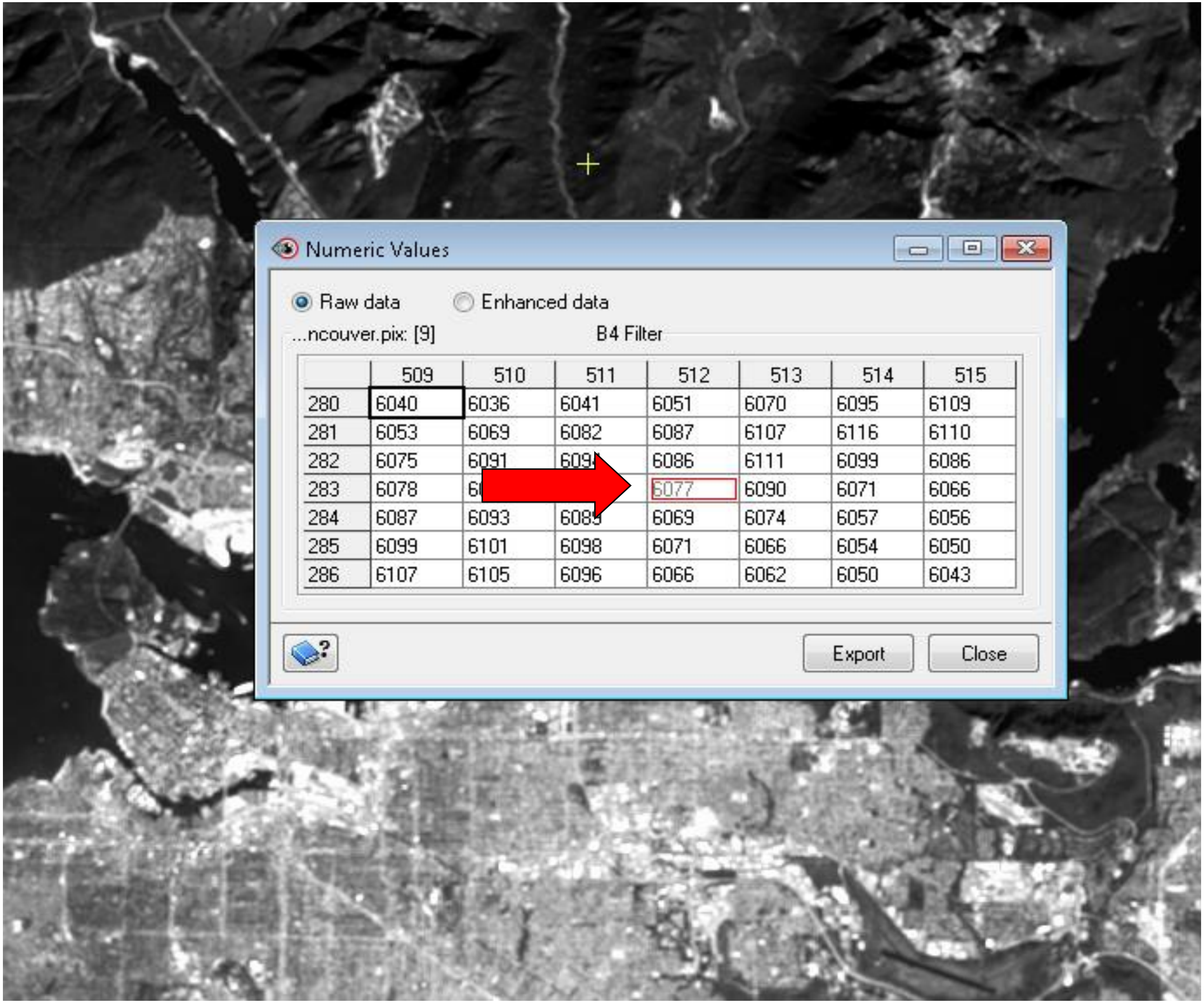
- Filter = 3x3
  - Coefficient is 1
  - smooths the image data
    - eliminate noise.
  - A 3 x 3 average filter (kernel)
    - computes the sum of all image pixels under filter kernel and then divides the sum by the total sum of the kernel
- Image = Band 4 landsat 8 Vancouver
- Result = smoother band 4



	509	510	511	512	513	514	515
280	5970	6019	6048	6056	6088	6122	6161
281	6092	6115	6082	6097	6088	6153	6136
282	[illegible]	[illegible]	6128	6115	6078	6165	6050
283	6077	[illegible]	6066	6082	6041	6178	6004
284	6106	6094	6084	6064	6039	6049	6031
285	6088	6142	6090	6092	6065	6059	6046
286	6104	6152	6051	6109	6044	6073	6078

Pixel/Line 283, 512

6128+6115+6078+6066+6082+6041+6084+6064+6039 = 54,697 / 9 = 6,077



	509	510	511	512	513	514	515
280	6040	6036	6041	6051	6070	6095	6109
281	6053	6069	6082	6087	6107	6116	6110
282	6075	6091	609[illegible]	6086	6111	6099	6086
283	6078	6[illegible]	[illegible]	6077	6090	6071	6066
284	6087	6093	608[illegible]	6069	6074	6057	6056
285	6099	6101	6098	6071	6066	6054	6050
286	6107	6105	6096	6066	6062	6050	6043

Convolution

- Calculations are preformed on entire image
- One pixel at a time
- Known as: Convolution
- Demo in example filtering pdf

Filters with coefficients

$$DN_{ij} = INT\left(\frac{\sum_{j=1}^{nrows}\left(\sum_{i=1}^{ncolumns} C_i \times DN_i\right)}{n}\right)$$

- where
  - i = row location
  - j = column location
  - Ci = coefficient of kernel at position i, j
  - DNi = of the original data at position i, j
  - n =
    - the sum of the coefficients of the kernel or
    - 1 if the sum of coefficients is zero
    - In example on right n=8
  - DNij = output pixel value

-1	-1	-1
-1	16	-1
-1	-1	-1

Example: centre pixel calculation

Original image sample:

8	6	6	5
2	**8**	6	8
2	2	8	8
2	2	6	8

Kernel:

-1	-1	-1
-1	16	-1
-1	-1	-1

Result image sample

?	?	?	?
?	**11**	?	?
?	?	?	?
?	?	?	?

	j=1		j=2		j=3		
i=1	(-1)(8)	+	(-1)(6)	+	(-1)(6) =	-8 -6 -6	= -20
i=2	(-1)(2)	+	(16)(8)	+	(-1)(6) =	-2 +128 -6	= 120
i=3	(-1)(2)	+	(-1)(2)	+	(-1)(8) =	-2 -2 -8	= -12

(n) Sum = 88

n = 16 - 8 = 8

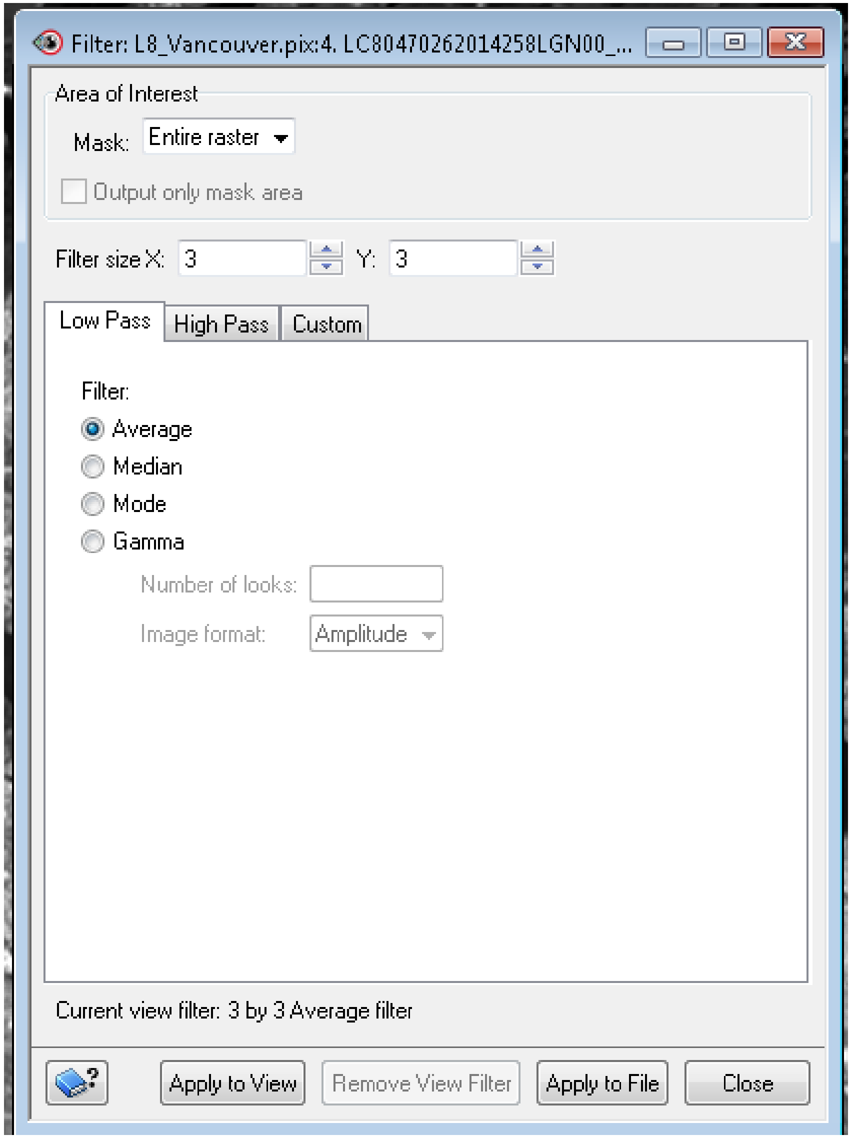
output $DN_{i,j}$ = 88 / 8 = 11

Frequency Enhancement

- High frequency enhancement
  - Emphasize difference
  - High contrast
  - High pass
- Low frequency enhancement
  - Emphasize sameness
  - Low contrast
  - Low pass
- See handout for examples

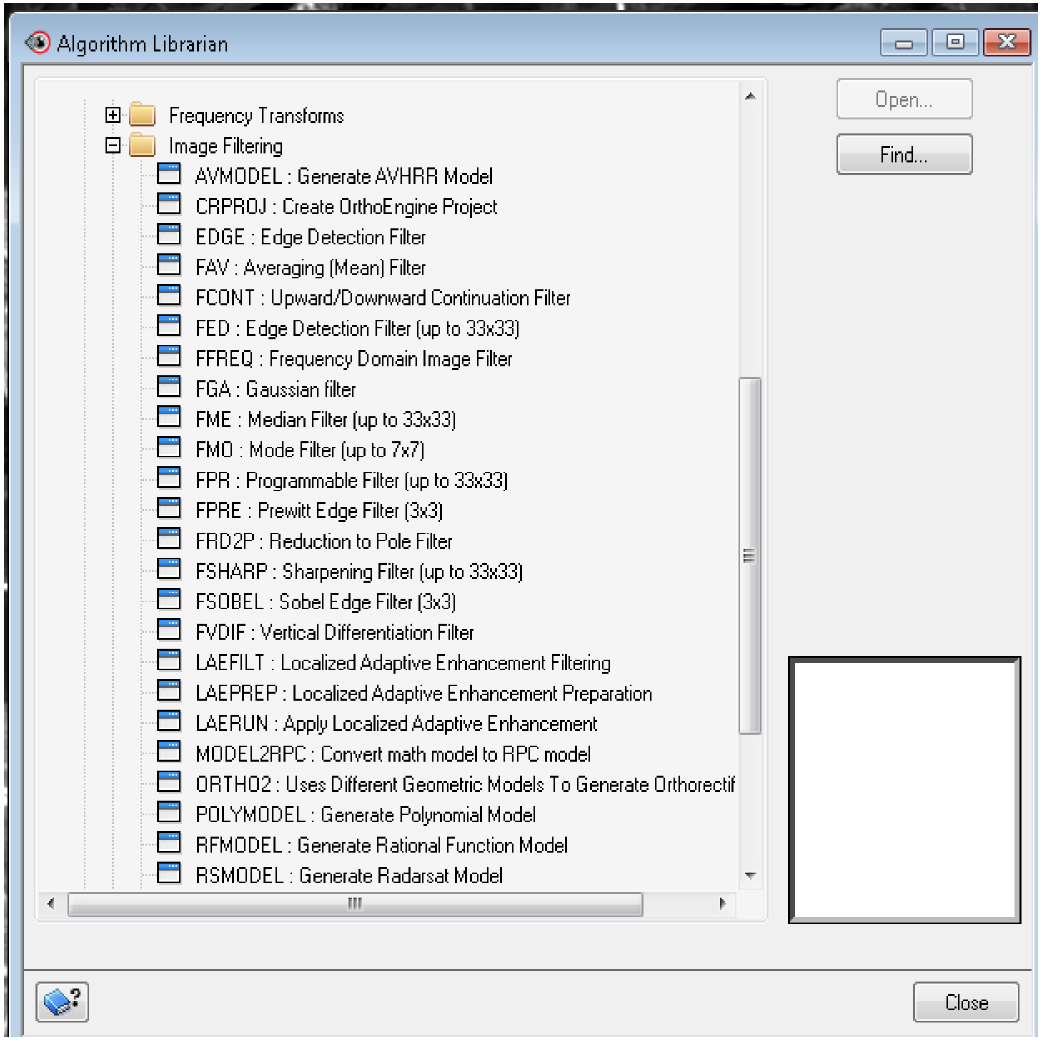
Filters in Geomatica

- Focus
  - Filter dialog from context menu of layer



Filters in Geomatica (2)

- Algorithms in library
- Modeler



Create a LINE network e.g. Shoreline (road)….

- First
  - Edge detection
    - filter
- Second
  - Edge Enhancement
    - Math
- Third
  - Edge Mapping
    - THR algorighm
- Fourth (optional)
  - Convert to Vector
    - RTV – algorighm
- Fifth (optional)
  - Change format
    - Export to other format e.g. SHAPEFILE

- Look at your results

Edge detection

- Smooth out areas of low spatial frequency and highlight edges
  - local changes
- Possible methods:
  - 1) calculating spatial derivatives (differencing)
  - 2) edge detecting template (Zero-sum kernels):
    - A - directional (compass templates)
    - B - non-directional (Laplacian)
    - All are Edge enhancement kernels
  - 3) subtracting a smoothed image from the original

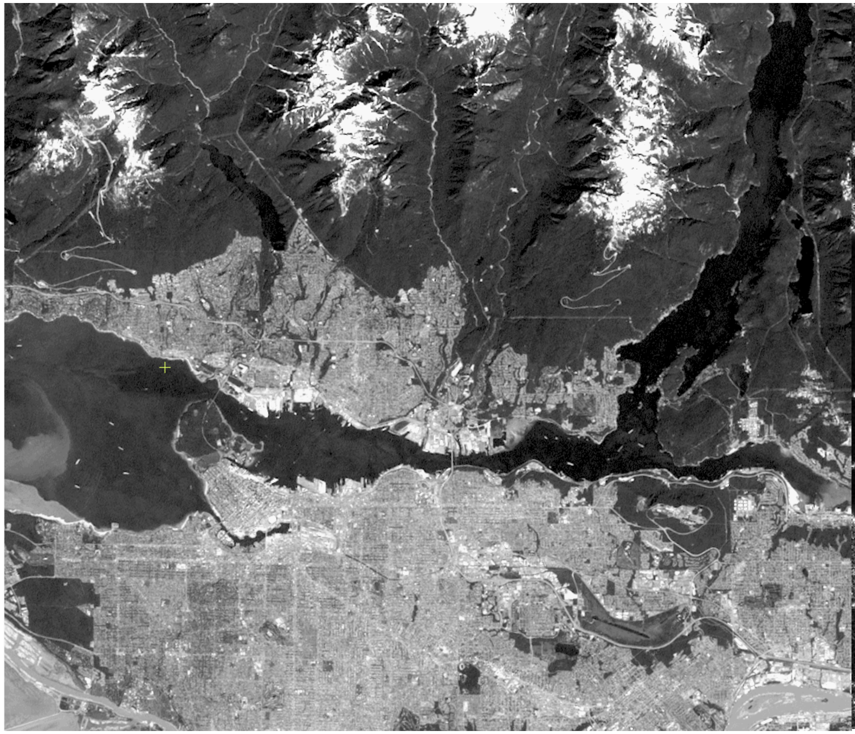
Method 1 - Spatial differencing

- Vertical:
  - $DN_{i,j} = DN_{i,j} - DN_{i,j+1} + K$
- Horizontal:
  - $DN_{i,j} = DN_{i,j} - DN_{i-1,j} + K$
- constant K added to make output positive

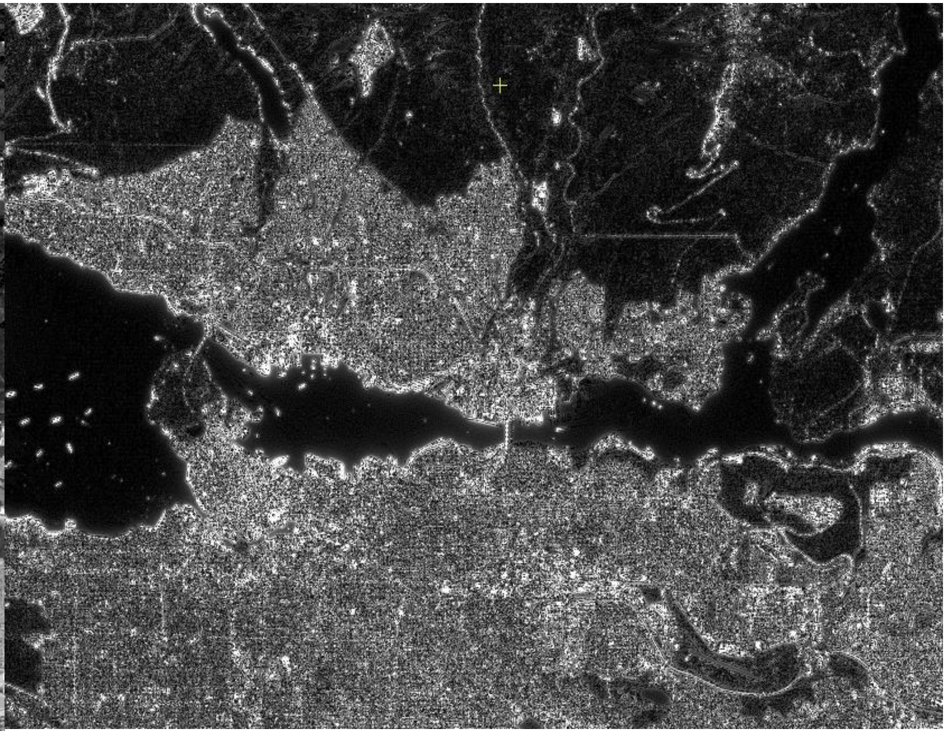
	j1	j2	j3	j4
i1				
i2				
i3				

Example: Vertical Difference

OLI band 4



FVDIF (Vertical Differentiation Filter) - algorithm OLI band 4



Method 2 - Zero sum kernels

- the sum of all coefficients in the kernel equals zero.
- n is set = 1 since division by zero is impossible
- Result
  - zero in areas where all input values are equal
  - low in areas of low spatial frequency
  - extreme in areas of high spatial frequency
    - high values become higher, low values lower

Method 2 A – Directional filters

- Calculate difference in a given direction
  - the directional gradient
- Line features (i.e. rivers and roads)
  - More than one pixel wide
    - pairs of edges
    - Linear Edge detection templates
  - Single pixel wide
    - Linear Line Detecting Templates

Example Templates

Example: Linear Edge Detecting Templates

Vertical: $\begin{matrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{matrix}$

Horizontal: $\begin{matrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{matrix}$

Diagonal (NW-SE): $\begin{matrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{matrix}$

Diagonal (NE-SW): $\begin{matrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{matrix}$

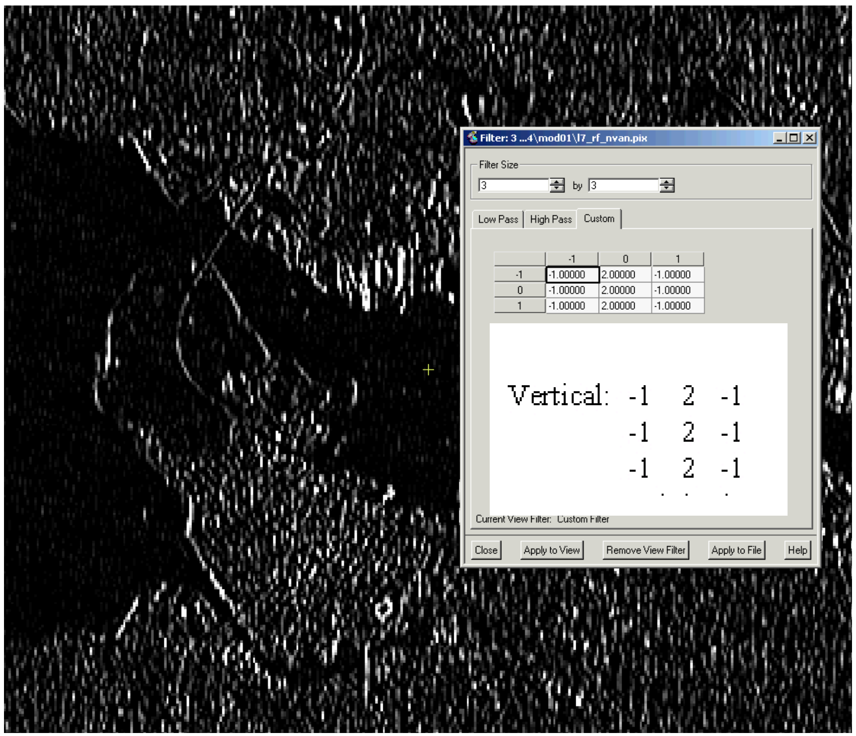
Example: Linear Line Detecting Templates

Vertical: $\begin{matrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{matrix}$

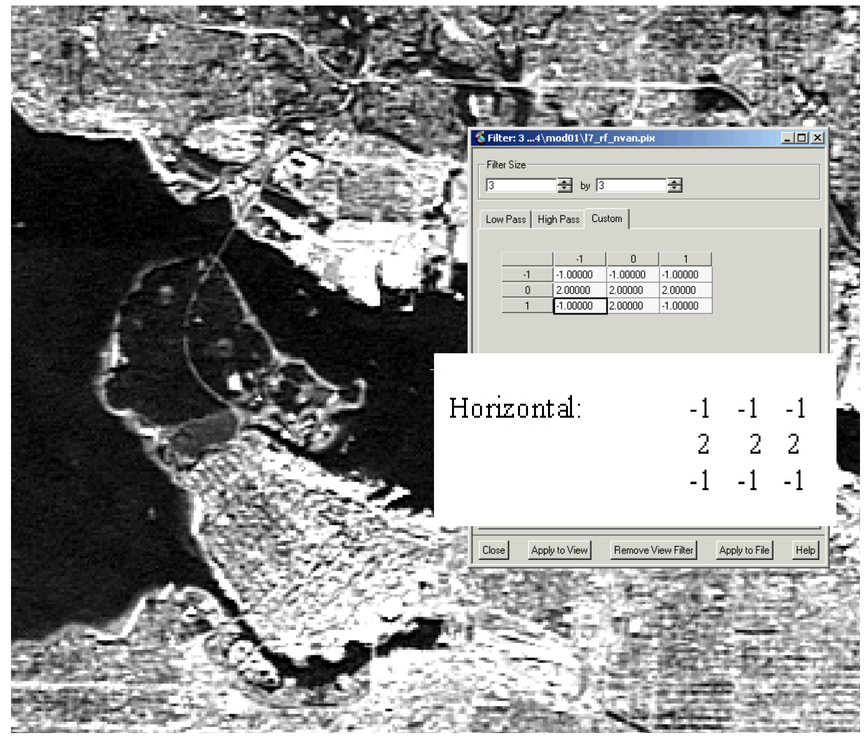
Horizontal: $\begin{matrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{matrix}$

Linear Line detection example

Vertical Line



Horizontal Line



Line Detection

	515	516	517	518			
561	84	91	89	71			
562	65	73	86	58	36	46	71
563	34	40	64	62	32	37	62
564	27	29	58	67	30	33	54
565	27	23	62	65	32	31	48
566	27	23	61	62	33	29	48
567	27	26	57	67	33	30	49
568	26	30	55	77	29	30	38
569	26	28	55	74	27	29	29
570	27	25	57	68	30	28	31
571	28	25	55	66	33	26	31
572	29	27	49	70	33	29	31
573	29	30	45	76	29	30	29
574	29	30	47	73	25	27	28
575	26	28	50	67	31	25	31
576	26	28	50	69	38	22	32
577	26	27	47	73	40	23	29
578	27	27	46	77	39	30	32
579	28	27	49	80	45	42	49
580	27	30	53	80	51	40	52
581	72	53	57	72	52	35	38
582	118	68	59	72	49	29	31
583	81	53	60	73	48	32	31
584	57	37	55	71	45	34	41

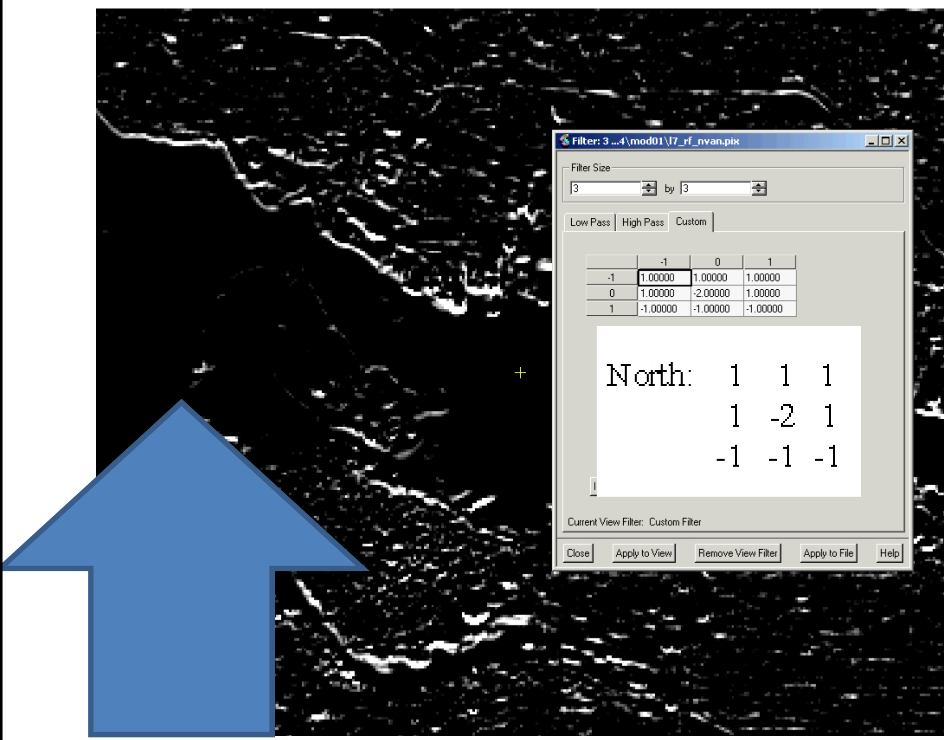
			517	518	519	520	521
				40	0	0	29
				24	0	0	55
				25	0	0	67
563	0	0	87	68	0	0	70
564	0	0	82	110	0	0	54
565	0	0	93	112	0	0	47
566	0	0	94	110	0	0	47
567	0	0	61	144	0	0	37
568	0	0	32	180	0	0	13
569	0	0	32	185	0	0	0
570	3	0	48	159	0	0	0
571	8	0	41	151	0	0	0
572	8	0	4	180	0	0	0
573	4	0	0	210	0	0	0
574	0	0	0	205	0	0	0
575	0	0	0	177	0	0	5
576	0	0	2	162	0	0	9
577	0	0	0	178	0	0	4
578	0	0	0	194	0	0	6
579	0	0	0	191	0	0	24
580	27	0	0	157	0	0	36
581	60	0	0	127	0	0	31
582	68	0	0	109	0	0	10
583	55	0	0	116	0	0	0
584	28	0	3	115	0	0	0
585	7	0	27	99	0	0	0
586	0	0	59	63	0	0	0

2A Directional - Compass gradient

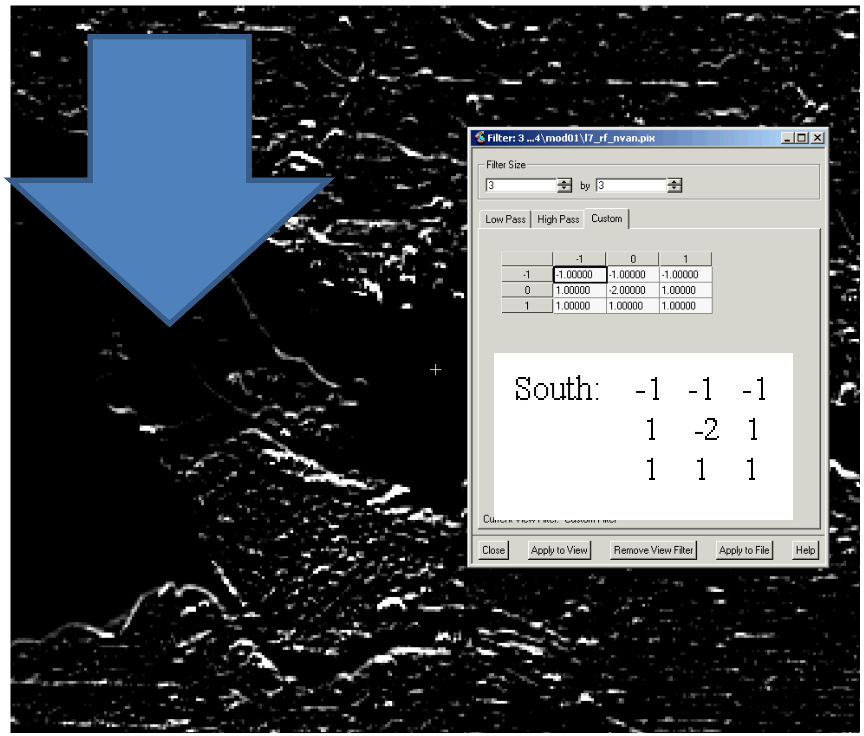
- Produce a maximum output for vertical (or horizontal) brightness value changes from the specified direction.
- For example a North compass gradient enhances changes that increase in a northerly direction,
  - i.e. from south to north:

Example Compass gradient

North (south to north edges)



South (north to south edges)



Method 2B Non-directional

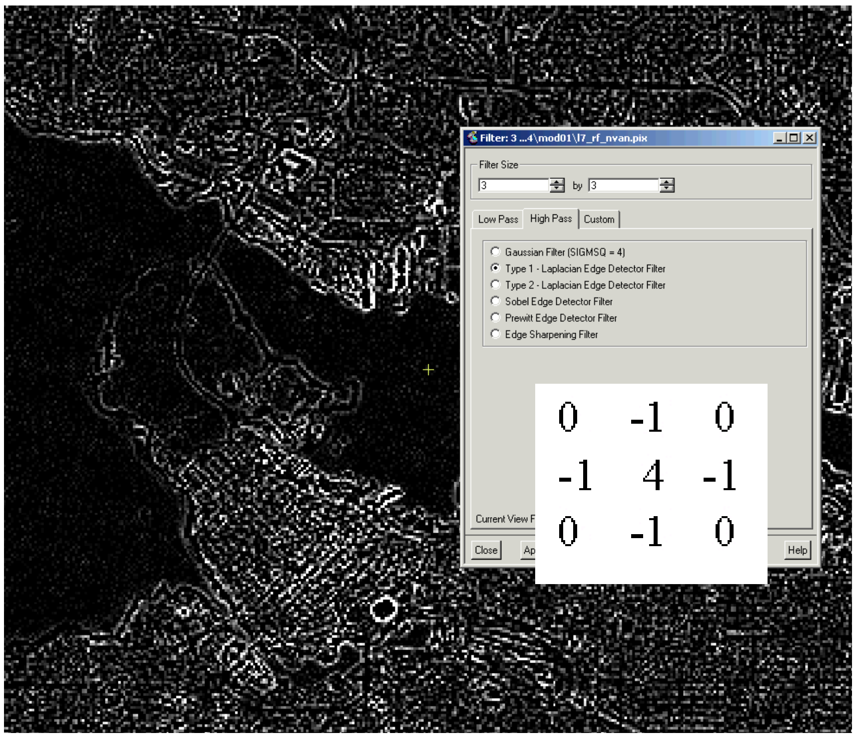
- Laplacian
  - insensitive to direction.
  - highlights points, lines and edges
  - suppresses uniform, smoothly varying regions
  - Result: Single edges

$$\begin{matrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{matrix}$$

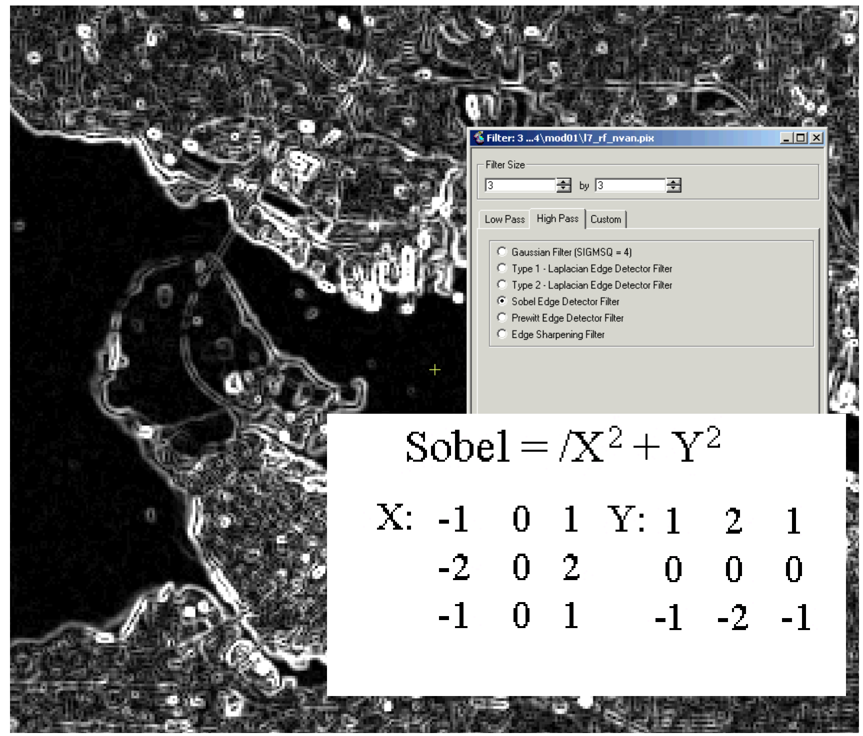
- Sobel
  - Non linear
  - Two kernels
  - Result: Double edges

Examples – Non directional

Laplacian 1



Sobel



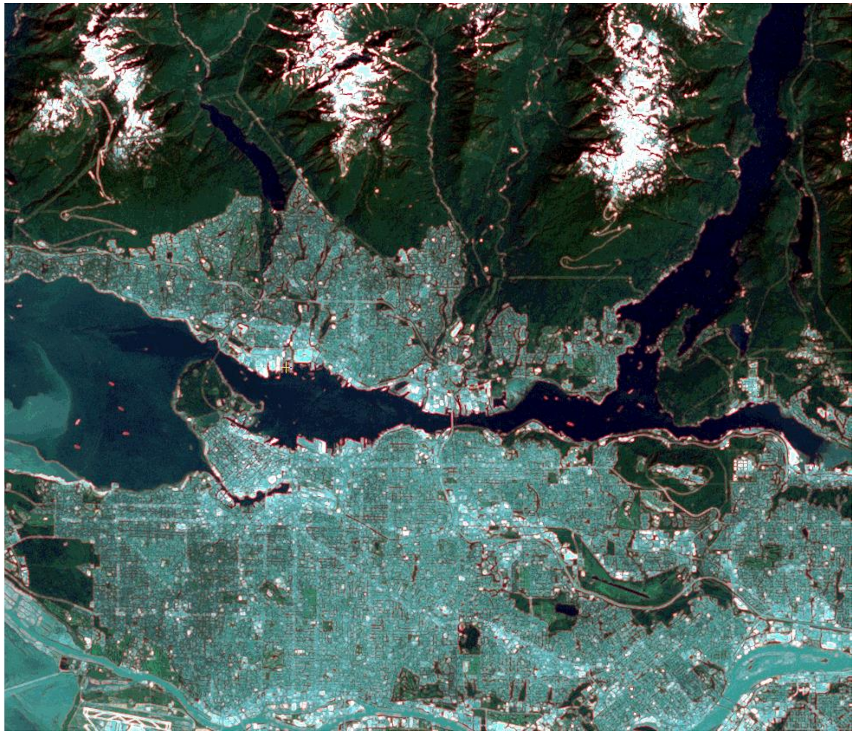
Edge Enhancement

- add the edges back into the original image
- Result:
  - to increase contrast in the vicinity of the edge
  - ARI algorithm
- E.g.
  - Band 4 apply sobel filter output sobel image
  - Sobel image + band 4
    - output = sobel edge enhanced band4 image
  - Display RGB composite with
    - R = SOBEL edge enhanced band3,
    - G = band2,
    - B = band1

Example with OLI B4

Original RGB 4,3,2

Result – (Band 4 +sobel),3,2

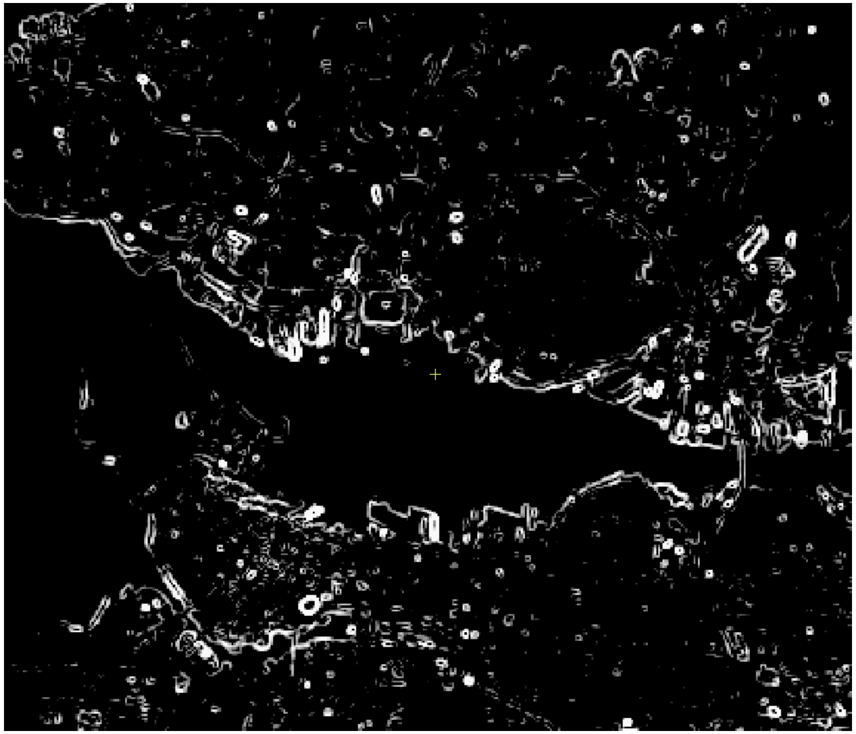


Edge Mapping

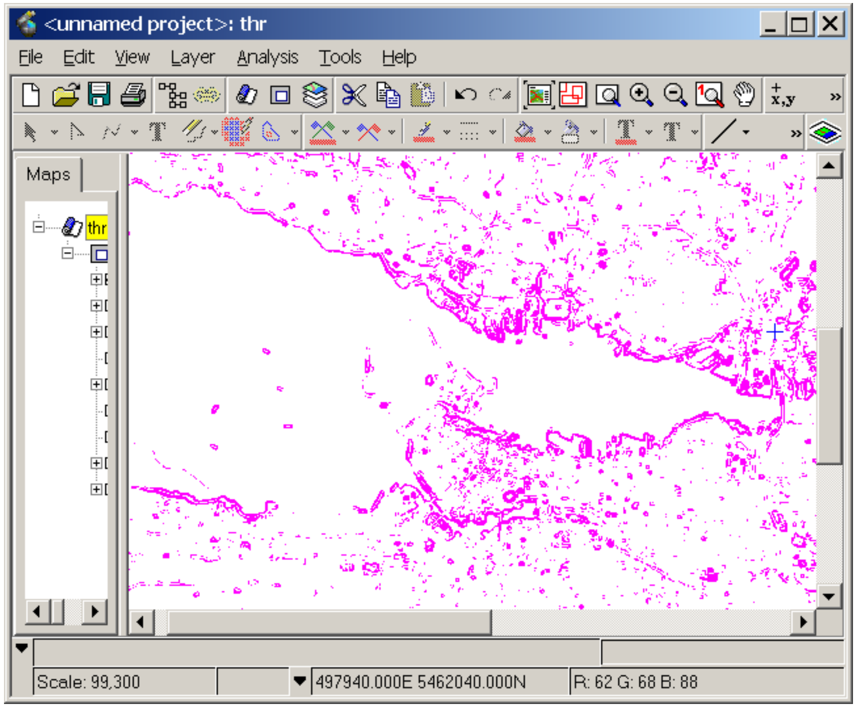
- Find Threshold
  - DN value of edge detection output image
- Result:
  - a binary map of edges vs. non-edges
    - 1 = edge
    - 0 = non edge
- THR algorithm
- Your Caution
  - Threshold too low:
    - too many isolated pixels classified as edges and edge boundaries too thick
  - Threshold too high:
    - boundaries will consist of thin, broken segment

Example Acquire Lines

Edge Map - THR



Convert to Vector



Next Step

- Clean up data to have functional line network
  - Shorelines
  - Roads
- Use editing tools
  - PCI Geomatica
  - ArcGIS

Spatial vs. Spectral Enhancement

- Spatial-based Enhancement
  - modifies a pixel's values based on the values of the surrounding pixels
  - local operator
- Spectral-based Enhancement
  - modifies a pixel's values based solely on the pixel's values
  - point operator
- Difference = ???