GLM Photogrammetry (Digital Image Processing)

Yuanming Suo (UAH ECE) and Larry Carey

Outline

I. Introduction
II. Background
III. Data & Methodology
IV. Preliminary Results
V. Preliminary Conclusions and Future Work

I. Introduction

Target:

- A target (or object) is a lightning event, group, flash, area or background.

Objective:

- Improve object location accuracy by using digital image processing techniques.
- Since lightning is a spatially and temporally varying object, the image processing of the lightning “blobs” might pose some interesting complications.
  - It has not been done before.

Research Procedure:

1. *Define Object and Location*: candidate objects at different temporal and spatial scales such as group, flash, area, etc.
2. *Digital Image Processing Technique*: various centroid detection methods, model fitting or interpolation (e.g., ellipse fitting), edge detection, others (?).
3. *Validation*: validate results with proxy data such as LIS, LMA data, etc.
4. *Improvement*: use validation result to improve image processing technique.

II. Background – Proxy data

Simulation Data (Lightning Imaging Sensor, LIS) and its characteristics:

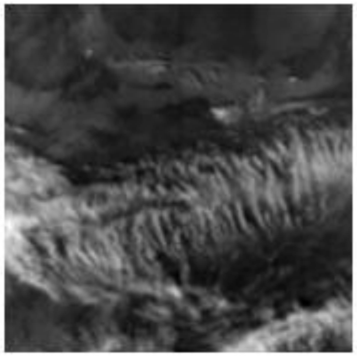
1. Spatial resolution (CCD pixel size): 5+ km
2. Spatial scale (CCD FoV): 580 km *580 km
3. Temporal resolution (frame rate): 2 ms
4. Temporal scale (longest monitor time):80 s.

Event (Vset)			
Variable Name	**Type**	**Order**	**Description**
TAI93_time	float64	1	Time of the event
location	float32	2	Latitude/longitude of event (deg)
radiance	float32	1	Calibrated event radiance
footprint	float32	1	Size of event (km2)
x_pixel	int8	1	X pixel location
y_pixel	int8	1	Y pixel location
bg_value	int16	1	Estimated raw background value(12bits)
bg_radiance	int16	1	Estimated calibrated background value
amplitude	int8	1	Event raw amplitude

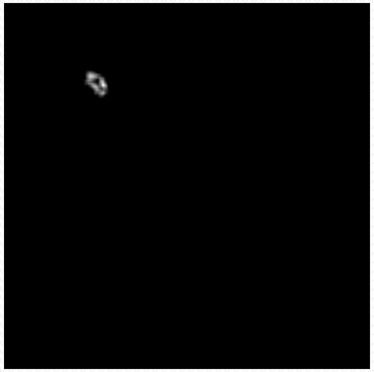
Group/Flash/Area (Vset)			
Variable Name	**Type**	**Order**	**Description**
Location	float32	2	Latitude/longitude of the area centroid (deg)
net_radiance	float32	1	Total calibrated radiance of the area(sum of all flashes for this area)
Footprint	float32	1	Size of the area (km2)
Delta_time(for flash and area)	float32	1	Time between first and last event(s)

*All data from LIS ATBD

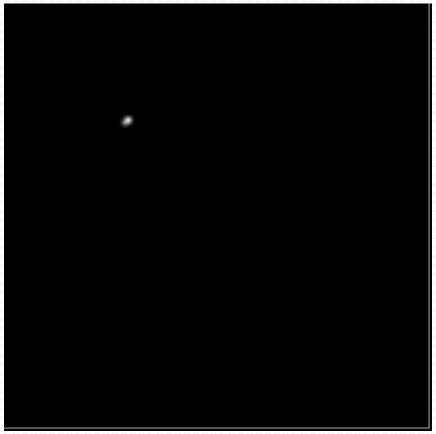
II. Background – Data example and LIS location



Background image example



Flash image example



Area animation example

The location of objects provided by LIS is the centroid location. And it's derived by radiance* weighted centroid function.

$$X = \sum_{j=1}^{n}\sum_{i=1}^{m} j * I_{i,j} \Big/ \sum_{j=1}^{m}\sum_{i=1}^{n} I_{i,j}$$

$$Y = \sum_{i=1}^{m}\sum_{j=1}^{n} i * I_{i,j} \Big/ \sum_{j=1}^{m}\sum_{i=1}^{n} I_{i,j}$$

Where $I_{i,j}$ are the weight at latitude and longitude location (i,j), and m, n are the dimensions of the window in which the centroid is being computed.

* *Centroid weighting function not specifically documented in ATBD.*

II. Background – Centroid location method comparison

- Different ways of doing centroid location? Change the weight options.
  1. Equal weight (Average of perimeter)
  2. Radiance weight
  3. Squared Radiance weight
  4. Delta time weight (time difference of each event with respect to first event in each flash)
  5. Squared delta time weight

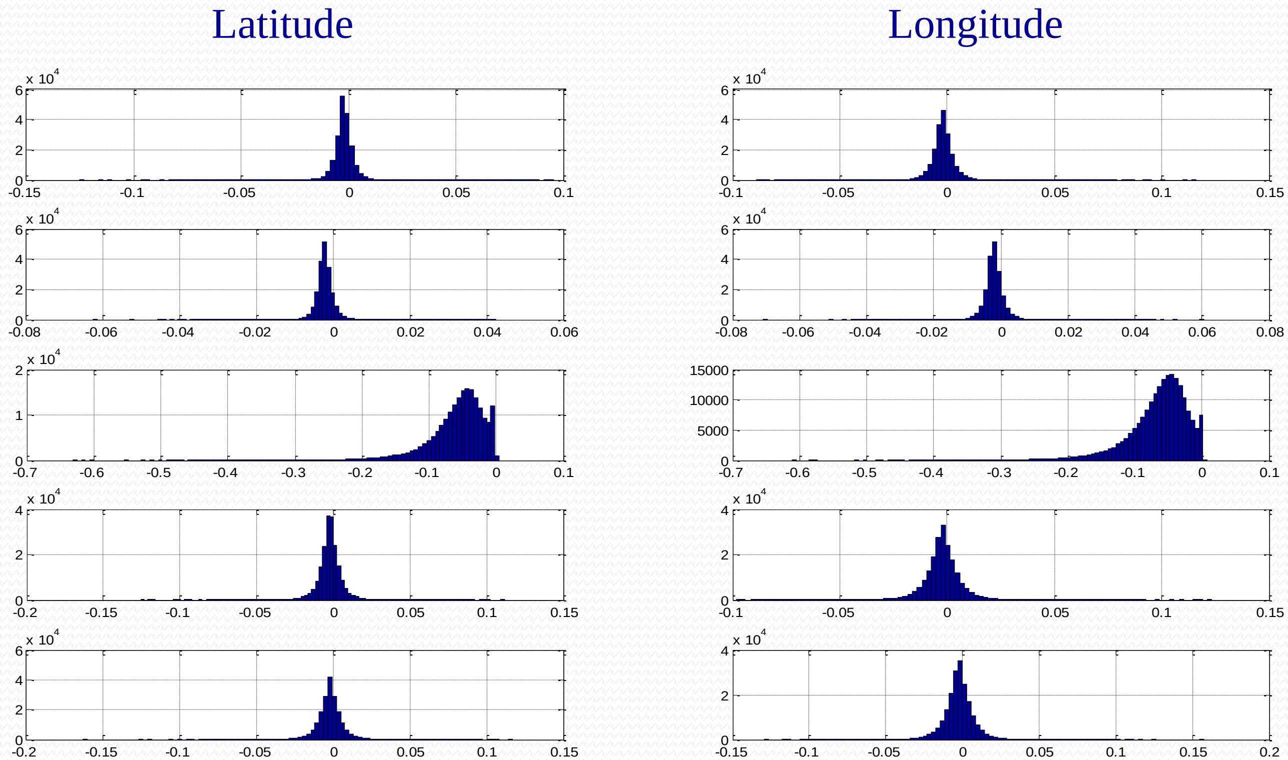
1. Average of perimeter
mean_lat = -0.0022 std_lat = 0.0051
mean_lon = -0.0019 std_lon = 0.0059

2. Radiance centroid
mean_lat = -0.0022 std_lat = 0.0026
mean_lon = -0.0020 std_lon = 0.0032

3. Squared radiance centroid
mean_lat = -0.0593 std_lat = 0.0444
mean_lon = -0.0656 std_lon = 0.0467

4. Delta time weight centroid
mean_lat = -0.0022 std_lat = 0.0078
mean_lon = -0.0021 std_lon = 0.0088

5. Squared Delta time weight centroid
mean_lat = -0.0022 std_lat = 0.0092
mean_lon = -0.0022 std_lon = 0.0103
Units: degree



Relative difference for flash centroid by using five different weight and compare with LIS result. Sample size: 1000 orbits including 196535 flashes

III. Data & Methodology – Preprocessing I

Mapping:

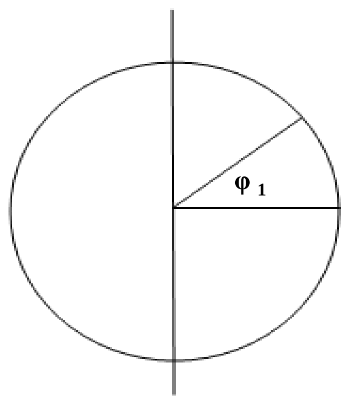
- Motivation
  - Because LIS is moving at a speed of 7 km/s, so for long duration object such as area the satellite has an smearing effect. So we use the geo-location of object rather than the pixel location provided by LIS and map it into a newly created x-y plane.

- Method*

$$y = (\varphi - \varphi_1) * R;$$

$$x = (\lambda - \lambda_1) * R * \cos\varphi;$$

where (λ, φ) is an arbitrary longitude and latitude and (λ_1, Φ_1) is selected point to be the origin of the new created Cartesian coordinates system.



Select events for this object ⇒ Find min lat&lon of these events ⇒ Mapping all events into xy plane

In implementation we choose this method for its mathematical simplicity.

*Data Retrieval Algorithms for Validating the Optical Transient Detector and the Lightning Imaging Sensor, W. J. KOSHAK AND R. J. BLAKESLEE

III. Data & Methodology – Preprocessing II

Gridding:

➢ *Motivation:*

1. The LIS provides the centroid. So interpolation is needed to make the data continuous.

➢ *Method:*

1. Using bi-cubic interpolation to do gridding on group level. The grid interval is 1 km.
2. If the group has less than 3 events, then use the event centroid location and footprint to calculate a square area with the center at event centroid and equal size.
3. When there's overlap, use the average of previous and new intensity value as the final intensity value for that grid.

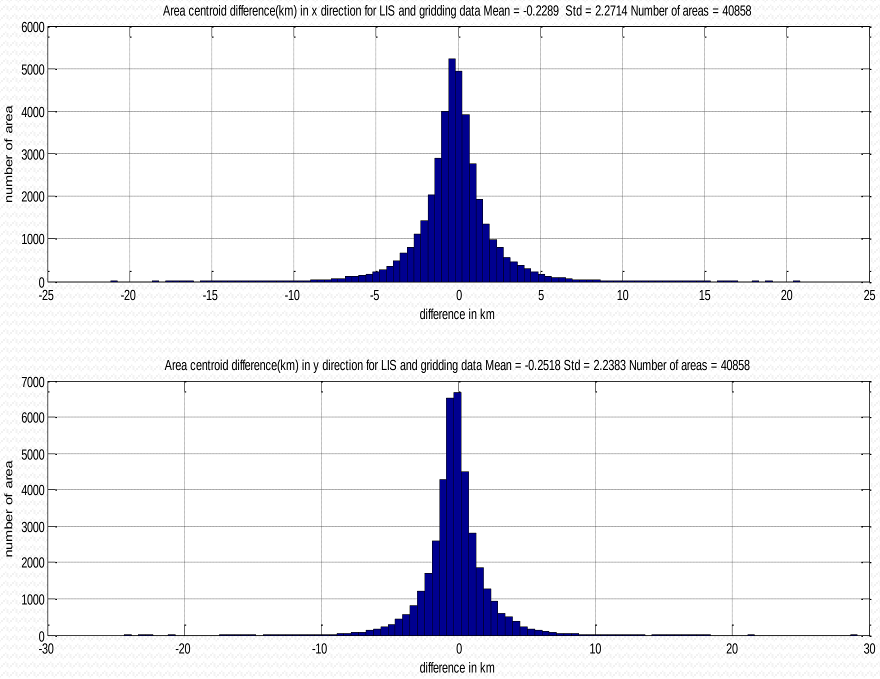
Smoothing:

➢ *Motivation:*

1. To mitigate the noise and smooth the gap between LIS pixels.

➢ *Method:*

1. Using 7 km*7km average filter.



Area centroid relative difference between LIS and gridded and smoothed data (Sample size of 1000 orbits, 40858 areas): Mean_x = -0.2289km, Std_x = 2.2714 km, Mean_y = -0.2518km, Std_y = 2.2383 km

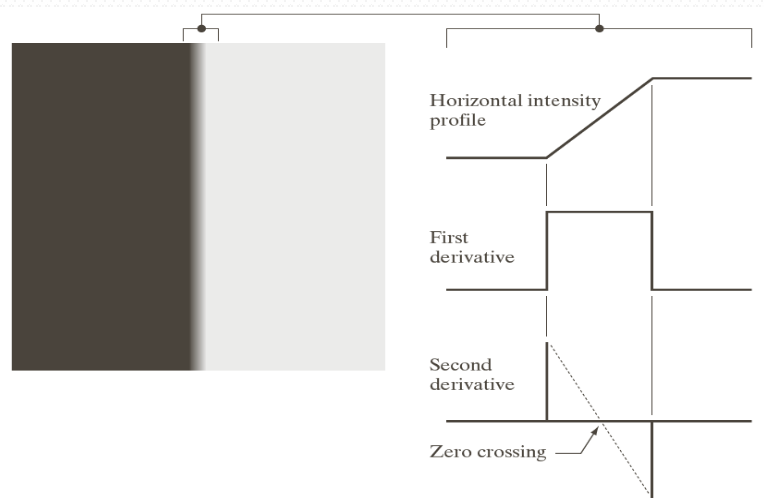
III. Data & Methodology – Edge detection overview

Basic concept:

➢ Edge is a set of connected pixels that lie on the boundary between two regions which have different intensity levels.

Model of an ideal digital edge (left) and a ramp edge (right) with their intensity profile

➢ First and second order derivative can be used to detect the presence of edge.



$$G_x = \frac{\partial f}{\partial x} = f(x+1) - f(x)$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

Gonzalez and Woods (2008)

III. Data & Methodology – Edge detection overview

Motivation:

- Provide location (possibly at sub-pixel accuracy) while
  1. significantly reducing the amount of data to the end users and
  2. preserving the important structural properties of an image.

Approach:

- Edge detection result may reveal the structure of object base on the change in the image intensity (radiance, flash rate). The extracted feature can be used to identify the location of object.

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0		0	−1
0	1		1	0

Roberts

−1	−1	−1		−1	0	1
0	0	0		−1	0	1
1	1	1		−1	0	1

Prewitt

−1	−2	−1		−1	0	1
0	0	0		−2	0	2
1	2	1		−1	0	1

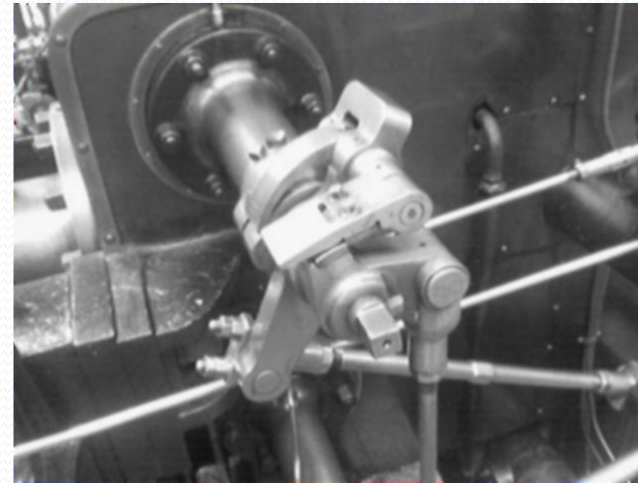
Sobel

Digital gradient detection operators

Gonzalez and Woods (2008)

Example using gradient detection operator to find horizontal, vertical and diagonal edges.

III. Data & Methodology – Canny Edge Detection Steps



Advantage:

- *Good Detection*: the ability to locate and mark all real edges.
- *Good Localization*: minimal distance between the detected edge and real edge.
- *Clear Response*: only one response per edge.

Method:

Noise reduction (convolve with a smoothing filter)

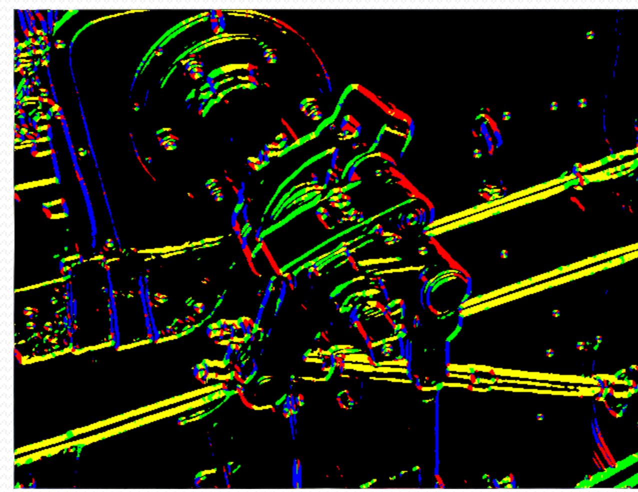
↓

Find intensity gradient and direction of edge(round the angel)

$$\mathbf{G} = \sqrt{\mathbf{G}_x^{\,2} + \mathbf{G}_y^{\,2}}$$

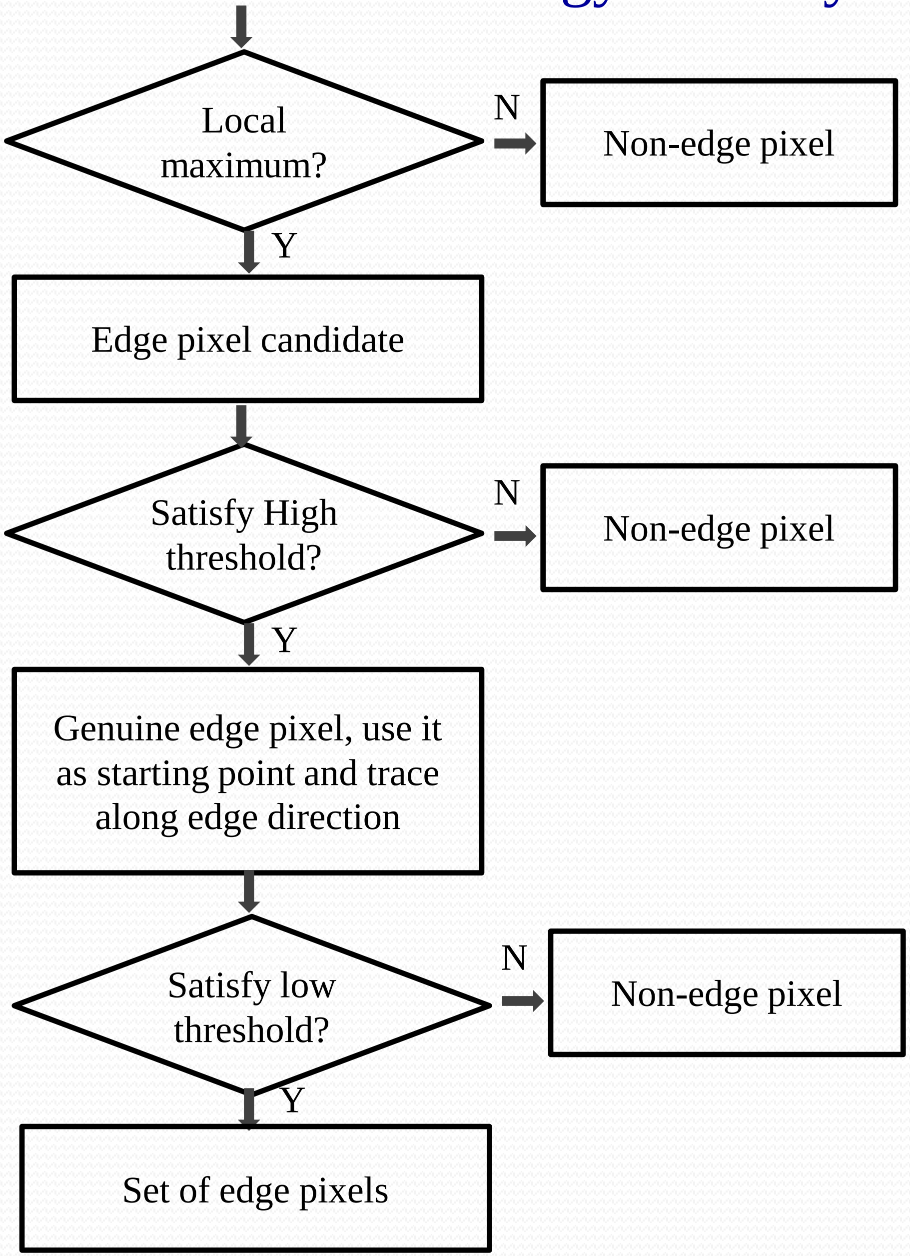
$$\mathbf{\Theta} = \arctan\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$

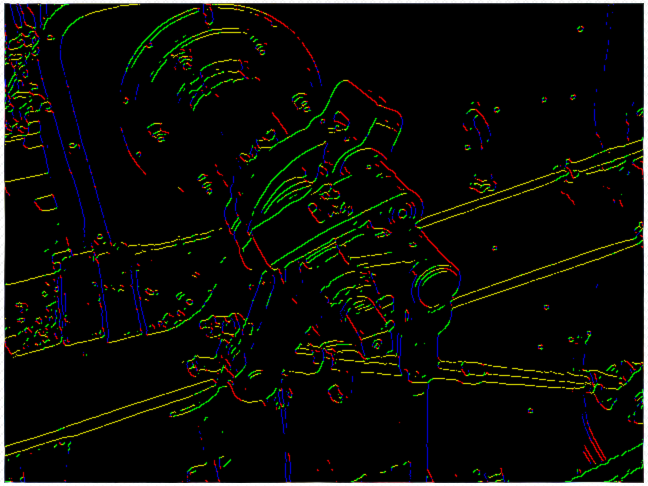
↓

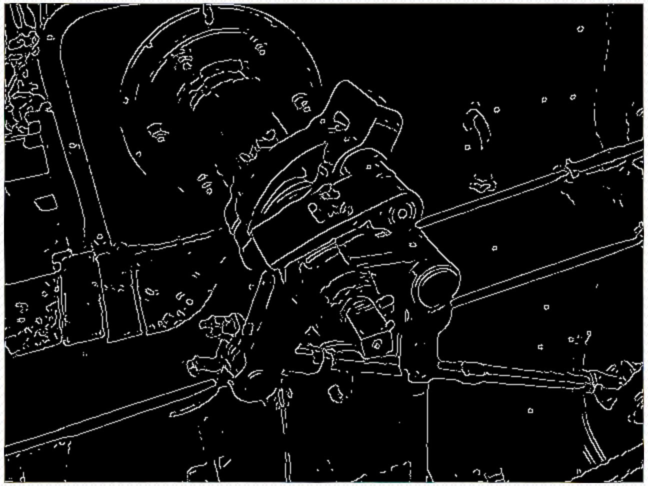


http://en.wikipedia.org/wiki/Canny_edge_detector

III. Data & Methodology – Canny Edge Detection Steps (Contd.)

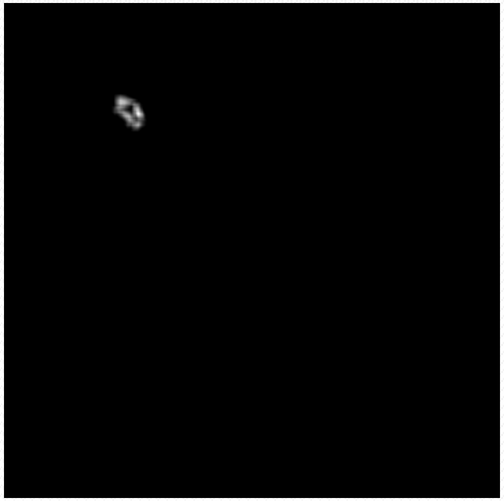




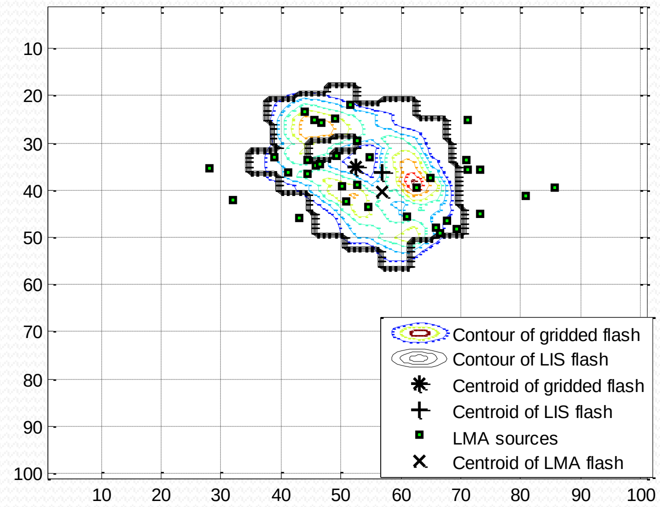


http://en.wikipedia.org/wiki/Canny_edge_detector

IV. Preliminary result – Edge Detection Example I (Flash)

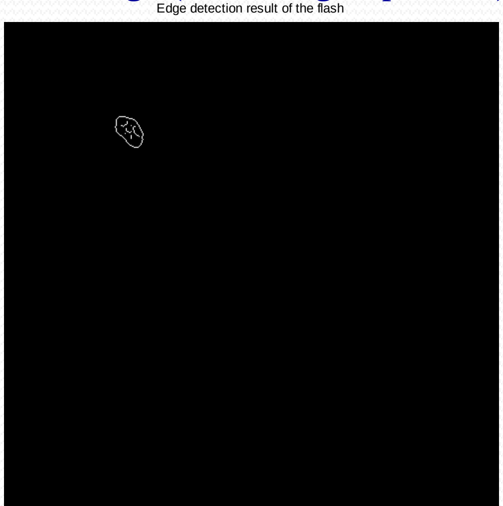


Flash Image (time integral product)



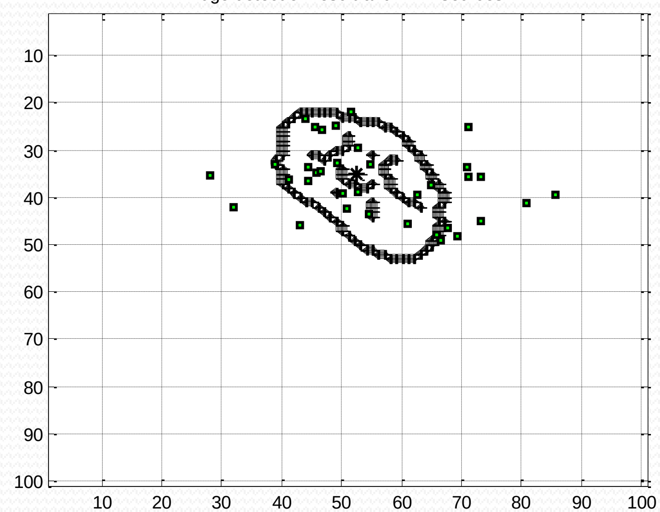
Flash contour and LMA sources

Edge detection result of the flash



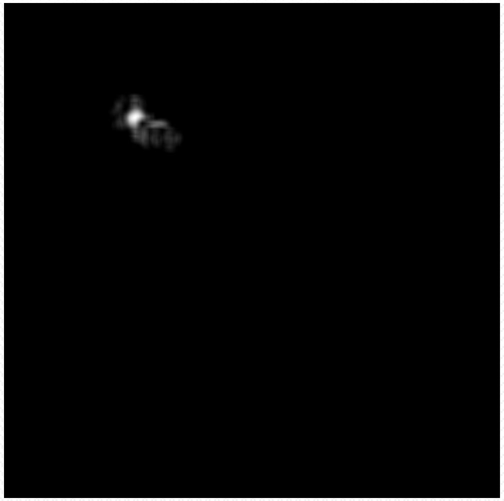
Flash edge detection result

Edge detection result and LMA sources

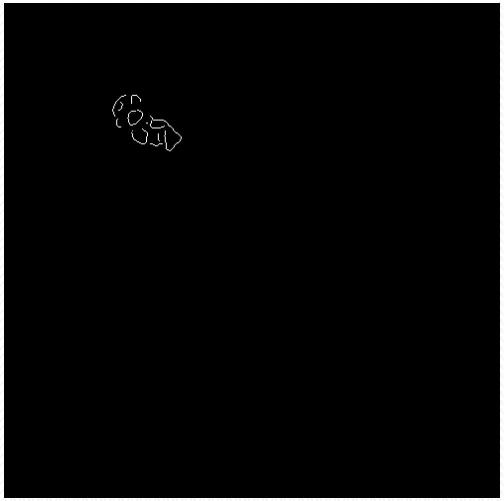


Flash Edge contour and LMA sources

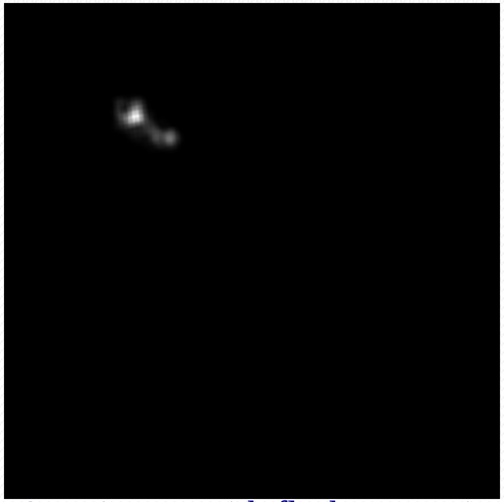
IV. Preliminary result – Edge Detection Example II (Area)



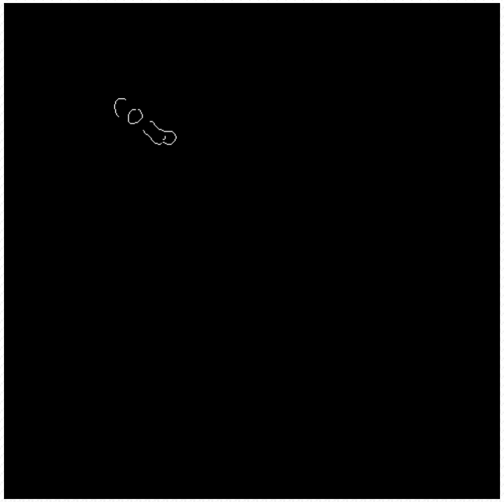
Area image with radiance as intensity (time integral product)



Area edge result with radiance as intensity



Area image with flash rate as intensity (time integral product)



Area edge result with flash rate as intensity

V. Preliminary Conclusions and Future Work

- *Preliminary Conclusions:*

1. Mapping and Gridding data from geo-space to newly created x-y plane is with acceptable mean and standard deviation. For the areas, the mean is around zero and std dev is 2 km.

2. Edge detected can be used to connect and indicate the boundary of the object (flash or area) and edges in the object which refers to the boundary of the high radiance region (cell, etc.)

- *Future Work:*

1. Further improvement can be made to mapping, interpolation and smoothing.
   - Different parameter assumptions and methods should be tried and compared.
   - Sensitivity tests

2. Different edge detection techniques can be applied.
   - Threshold which determines the selection of edge pixels is the key issue in edge detection technique.
   - Adaptive method of setting threshold and sub-pixel accuracy should be obtainable.

3. Standard for validation with LMA (or other) data should be set up to have a quantitative view.
   - Defining "truth" is problematic given lack of ideal, high-resolution data proxies.

4. Super-resolution techniques (using several low resolution images to get a high resolution image) or other image processing techniques might also be applied for the project purpose.