

# A PATTERN RECOGNITION SYSTEM FOR THE AUTOMATED TRACKING AND CLASSIFICATION OF METEORS USING DIGITAL IMAGE DATA

Kenny McGarvey, Ricardo Vilalta  
Dept. of Computer Science, Univ. of Houston  
Houston, Texas, USA

Marilia Samara, Robert Michell  
Space Science and Engineering Division  
Southwest Research Institute  
San Antonio Texas, USA

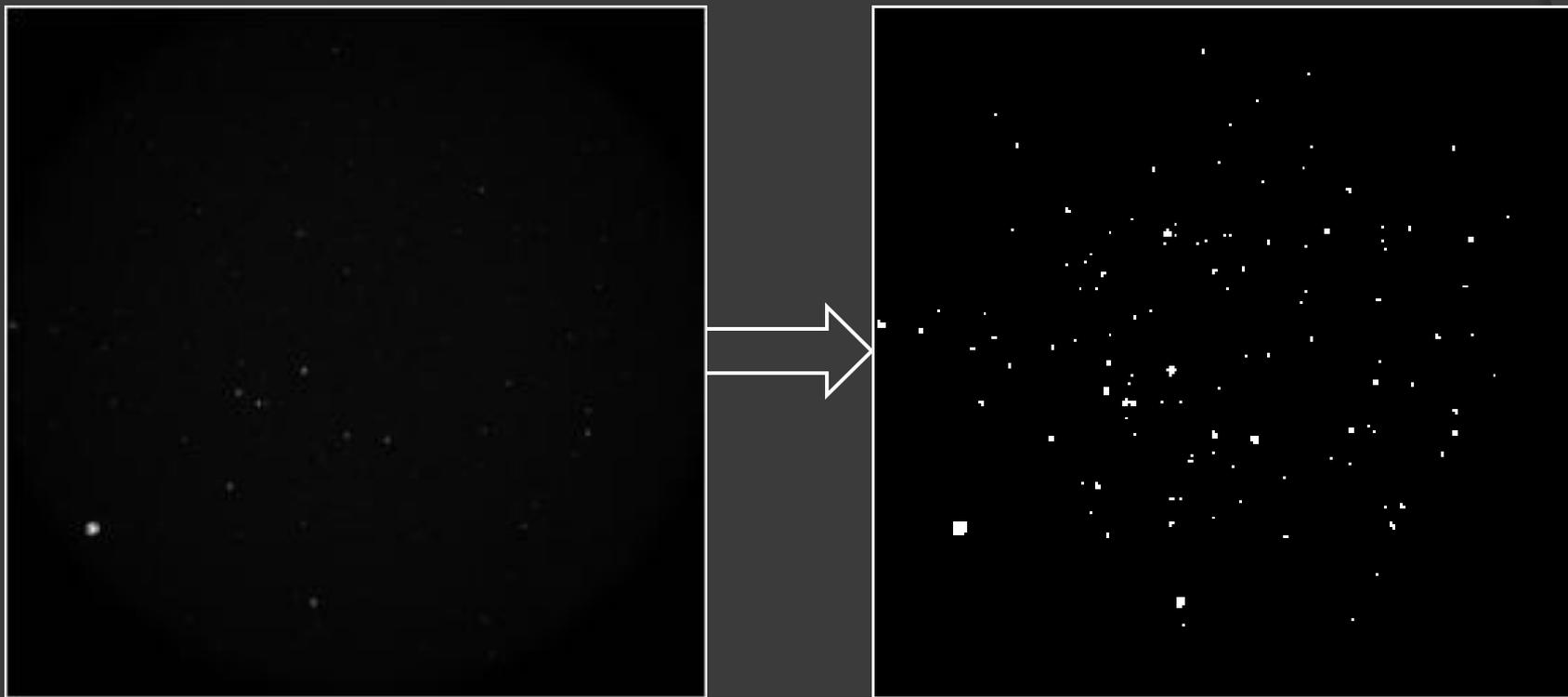
# About the images

- Tiff image
- Each contains 16297 frames representing roughly 8 minutes of data
- Size: 256 x 256 pixels
- Depth: 16 bit Gray scale
  
- Location: Poker Flat Research Range in Fairbanks, Alaska

# Apply a threshold

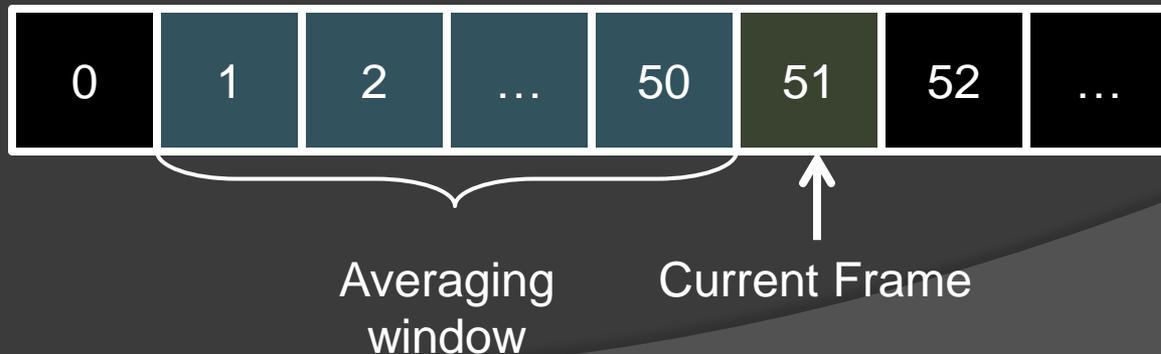
- ⦿ A binary threshold of 850 is applied to each frame
- ⦿ Pixels with a intensity below 850 will be set to 0, else set to 8000
- ⦿ This creates a uniform black background and removes some of the background noise

# Threshold example



# Building a background image

- By taking the average of many frames we can isolate the background
- A queue of the previous 50 frames is stored



# Building a background image

- ⦿ When an image is added to the queue, each pixel is added to a matrix representing the running total of intensity values
  - $\text{Sum}[i, j] += \text{Image}[i, j]$
- ⦿ Constructing the averaged image
  - $\text{AveragedImage}[i, j] = \text{Sum}[i, j] / \text{queue size}$

# Averaged background example

50 Frames



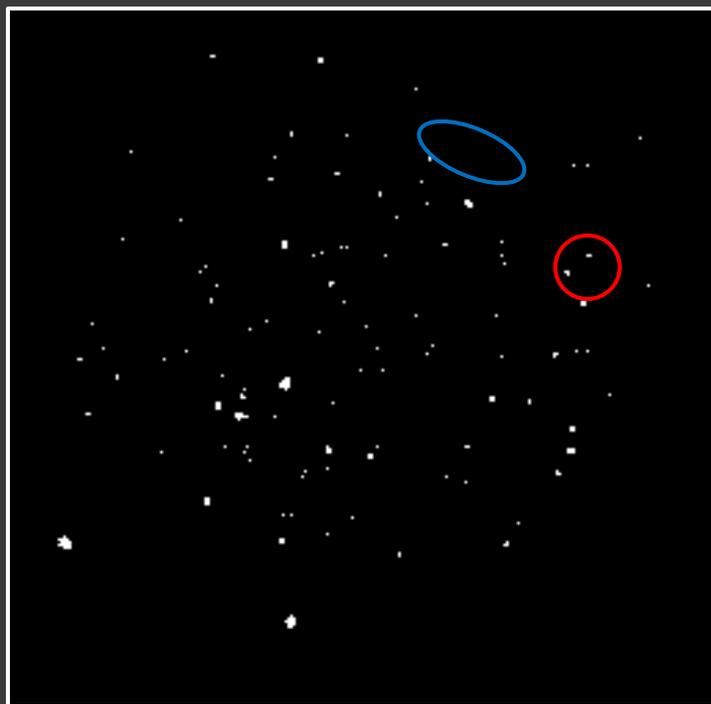
Average



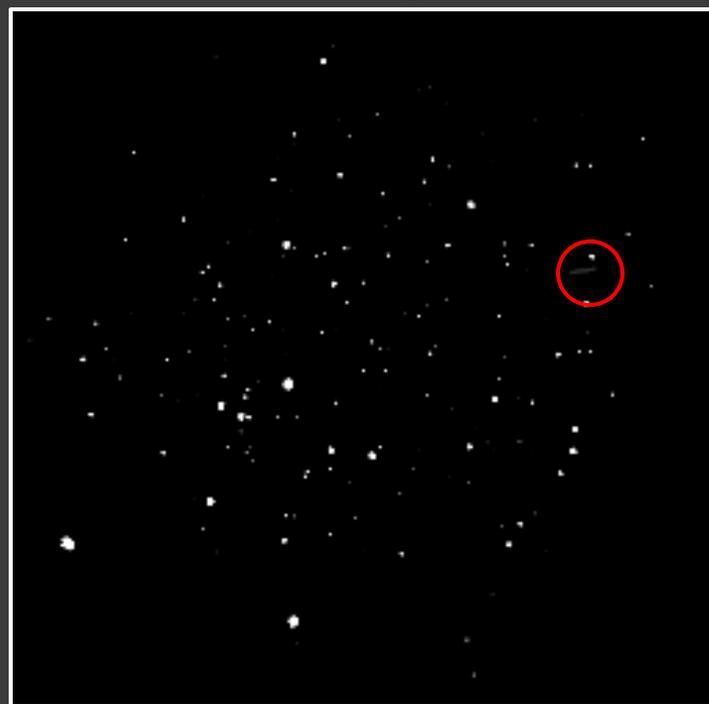
# Averaged background example

Notice the **meteor** and the slow moving **satellite**

50 Frames

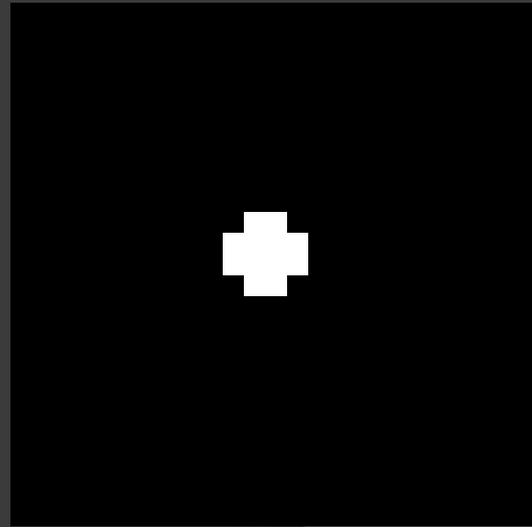
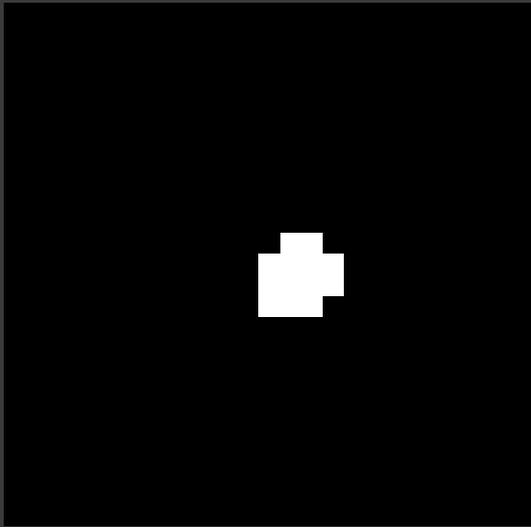


Average



# Flickering stars

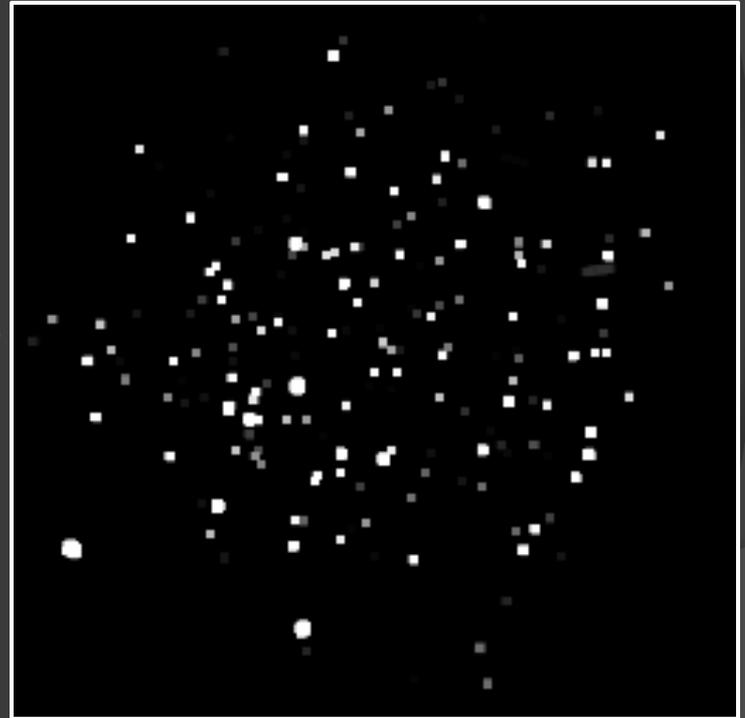
- Some of the stars are constantly changing shape by 1 or 2 pixels as seen below
- This will add noise later in the process



# Removing this noise

- ⦿ We dilate the averaged background image
- ⦿ This increases the radius of the luminous objects by 1 pixel

# Dilation example



# Background removal

- ① The averaged background image is subtracted from the current frame
- ② This will remove luminous objects that are in both the current frame and the background (i.e. removes the background)

# Background removal example

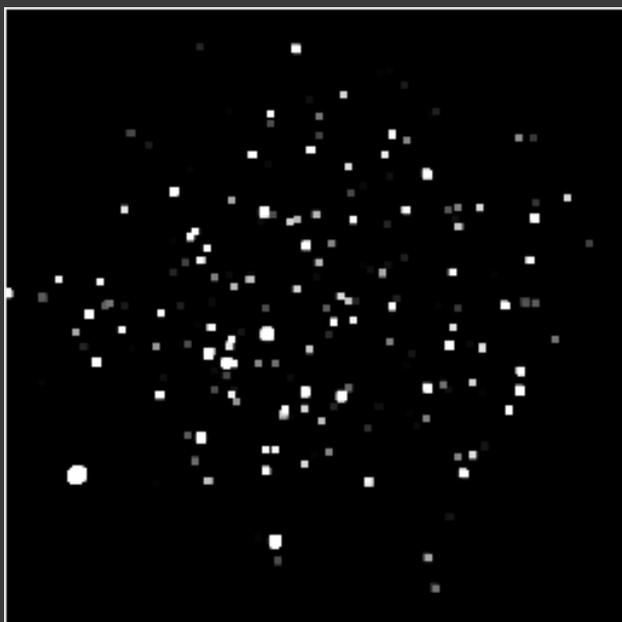
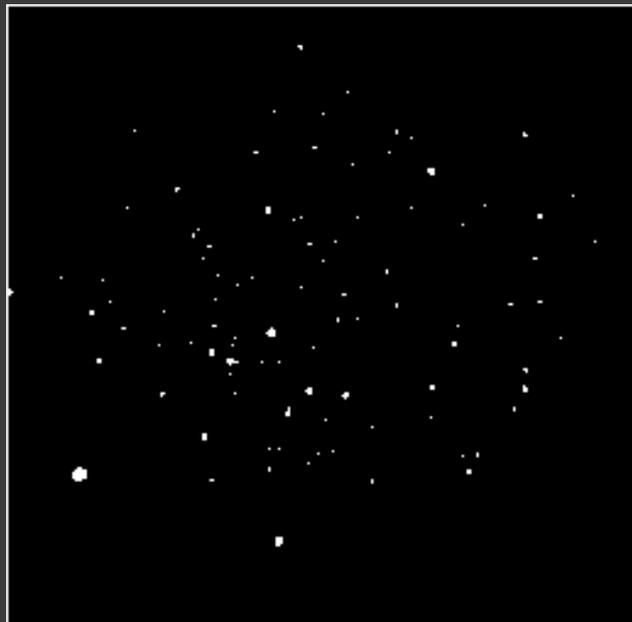
Current Frame

-

Averaged Background

=

Subtracted Frame



# Blob detection

- ① A simple blob detection algorithm is run on the subtracted image
- ② The blob detection starts on the subtracted image, but continues expanding the blob using the original (threshold applied) frame

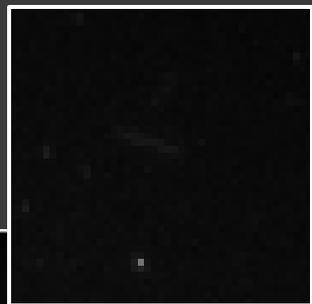
# Blob detection cont.

- ① From the blob detection we get the following:
  - Size
  - Centroid
  - Shape (rectangular or not)
- ① We can remove small blobs from the list of potential meteors, and blobs that do not have a rectangular shape

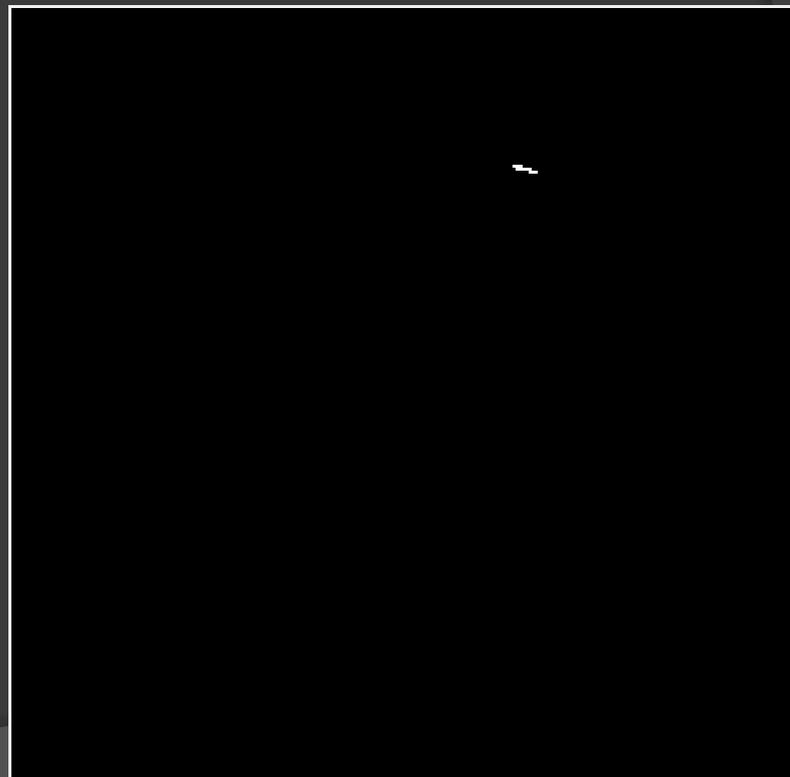
# Meteor examples

Visible for 7 frames

Input



After processing



# Meteor examples cont.

Visible for 1 frame

Input



After processing



# Future work/Features

- ⦿ Threading the meteor (blob) detection to another process
- ⦿ Parallelizing the entire process by starting at different frames
- ⦿ Estimating the coordinates and actual speed of the detected meteors
- ⦿ An adaptive/optimal threshold calculation