

AN AUTOMATED IMPLEMENTATION OF BEAMLETS TO CLASSIFY FRAMES OF TRIGGERED LIGHTNING

John A. Martiney
martiney@mil.ufl.edu

Uriel Rodriguez
uriel@mil.ufl.edu

Michael Nechyba
nechyba@mil.ufl.edu

Machine Intelligence Laboratory
Department of Electrical and Computer Engineering
University of Florida, Gainesville, FL 32611-6200

Abstract

The automation of data extraction and classification from video/images expedites the data acquisition process of a research project. In the case of a triggered lightning video, information is extracted from each frame to classify it as a frame in which lightning is either present or absent. This paper first presents the conventional method of lightning data extraction performed by a human operator. Then, it discusses the successful development of a system that automates this process using simple beamlets. The simple beamlet method can be expanded to use Recursive Dyadic Partitioning with beamlet chains in order to classify naturally occurring lightning.

1. Introduction

The University of Florida Lightning Research Group conducts lightning experiments at their Camp Blanding facility. They have equipment that allows them to gather information on lightning bolts that they trigger using special rockets. Among the information gathered is video footage of the lightning bolt strike. Currently, researchers must manually scan through the video to isolate the portions containing visible lightning bolts and then create smaller sized videos from these frames. Although this data extraction is simple, it requires that a human operator manually

crop the data. This process easily lends itself to automation.

This project automates this process as follows:

- Convert the movie file into a sequence of images
- Prepare the images for data extraction
- Classify each image (lightning or no lightning)
- Return images and frame numbers containing lightning bolts

2. Data Extraction

The Camp Blanding video footage is NTSC standard 240x352 MPEG and each frame contains four different views of the trigger site as shown in Figure 1.



Figure 1: Sample Input Frame

The MPEG video is converted into a sequence of PPM images. Every image is then divided into quadrants. The bottom third of each quadrant is cropped in order to reduce processing time by removing pixels that will never contain lightning. Finally, each quadrant is converted to a gray-scale image to obtain the intensity of each pixel. This is done using the following equation for NTSC video:

$$BW = 0.299 * Red + 0.587 * Green + 0.114 * Blue$$

Each quadrant can then be processed as a separate image. Processing the image in Figure 1 will produce the output in Figure 2.

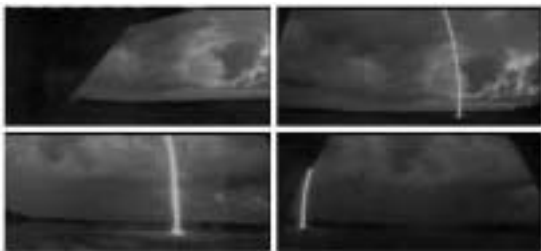


Figure 2: Sample Processed Frame

The following data extraction techniques are performed on each quadrant of the processed frame.

Because triggered lightning will always be characterized by high intensity pixels arranged in a nearly straight line, it is important to further process the image to filter out the low intensity pixels. This process borders on classification because it essentially categorizes each pixel as either a lightning or no lightning pixel based on intensity. However, it should be noted that this process is blind to the arrangement of the high-intensity pixels, and thus, it is not able to classify the image as containing a lightning bolt.

The black and white representation of the image assigns a value ranging from 0x00 (low-intensity) to 0xFF (high-intensity) to each pixel based on its intensity. A threshold value (for example 0xA0) can be used to classify each pixel as high-intensity or low-intensity. The threshold value functions as a high-pass filter that preserves pixel values equal to or above it and clears the values below it.

Using intensity to classify pixels introduces the problem of lighting conditions. Although most videos present an environment with overcast skies, there is no guarantee that the lighting conditions are constant. Thus, the intensity filter threshold is also not constant. The solution is to calculate an appropriate threshold for each image.

A “Smart Filter,, was developed in order to properly identify the high-intensity pixels with varying lighting conditions by automatically selecting an appropriate threshold. This is done by applying predefined threshold values (0xF0, 0xE0, ..., 0x60) and choosing the highest valued filter that will allow at least 40 high-intensity pixels in the image. The 40 pixel value was determined after running several trial and error experiments; therefore, this value is only applicable to the cropped images of size 80x176. If no threshold value returns at least 40 high-intensity pixels, the 0xF0 filter is applied in order to clear most of the bits.

Figure 3 shows the output of the Smart Filter with the images of Figure 2 as the input. It also includes the number of pixels above a given threshold and the threshold value used to filter the image. Different quadrants in the same frame may require different threshold values in order to normalize the intensity across images.

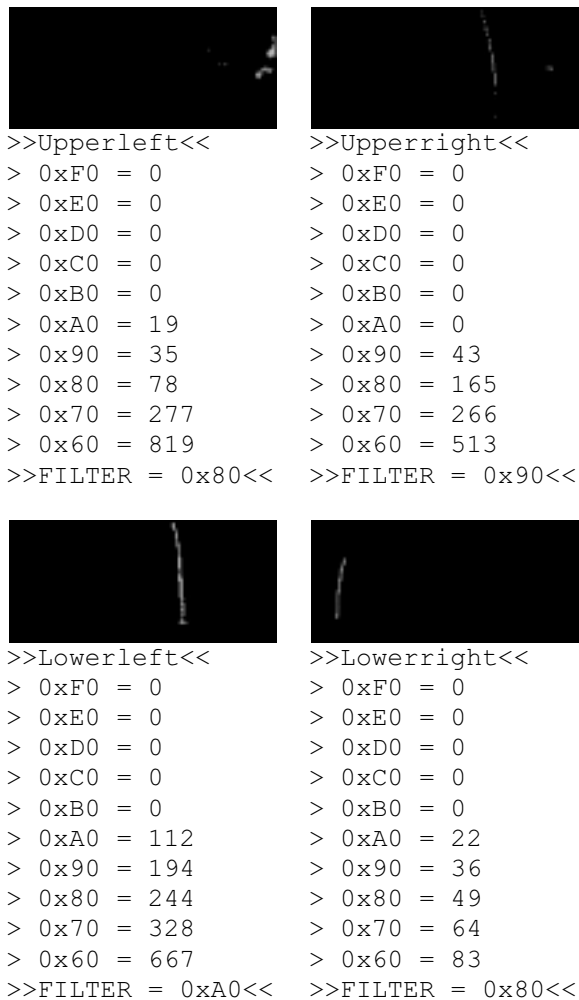


Figure 3: Filtered Images

After extracting the useful features (intensity values) from the image, the data is now ready to be passed to the classification step. The data for each image is now an array of pixels that are either low-intensity or high-intensity.

3. Data Classification

After the data extraction step, the remaining high-intensity pixels in each image are scanned to detect a pattern that follows a straight line (lightning). The scan method in this project is a simple case of Beamlet Analysis.

Beamlet Analysis is simply the process of scanning any area of a picture by using connecting vertices that form straight lines (beams). Connecting a series of beams forms the structure called a beamlet. The coordinates of these vertices form another structure called the beamlet dictionary. In simple Beamlet Analysis, each beamlet is actually composed of a single beam and the vertices in the dictionary must be on the edges of the image. This means that the entire picture is scanned by using straight lines that go from each vertex in the beamlet dictionary to all of the other vertices in the dictionary. However, in the case of lightning, it is not useful to scan using vertices originating from the left, right, and bottom edges of the picture because triggered lightning travels straight down. Thus, the simple Beamlet Analysis is further optimized by removing extraneous computations of beamlets originating from the left, right, and bottom edges of the dictionary. An example of this process is shown in Figure 4.

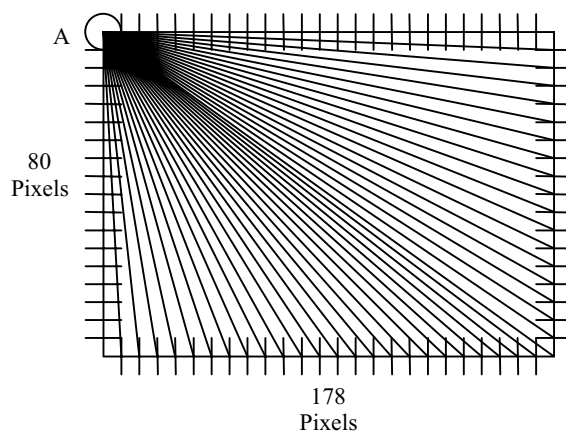


Figure 4: Example beamlet dictionary with beamlets originating at point A

The beamlet dictionary used in this project contains one vertex every two pixels. Since the processed images are of size 80x176, the dictionary contains a total of 88 (top) + 88 (bottom) + 38 (left) + 38 (right) = 252

vertices. It should be noted that the corner vertices of the images should only be included once. Since the beamlets only originate from vertices located at the top, there are a total of 14,354 beamlets scanning the picture. The width of the beamlets is fixed at three pixels. This ensures that every pixel in the image is a member of at least one beamlet.

Now that the behavior of the beamlet dictionary has been defined, it can be used to count the number of high-intensity pixels in the path of each beamlet. A local variable is used to keep track of the beamlet with the largest number of high-intensity pixels. The contents of this beamlet are then tested against those of the entire image. If the beamlet in question contains at least 75 percent of the total number of high-intensity pixels in the image, it is considered to be a beamlet that encloses lightning. Therefore, the picture is classified as a lightning image and is tagged with the frame number and the quadrant. This process is repeated for every quadrant of every frame in the video.

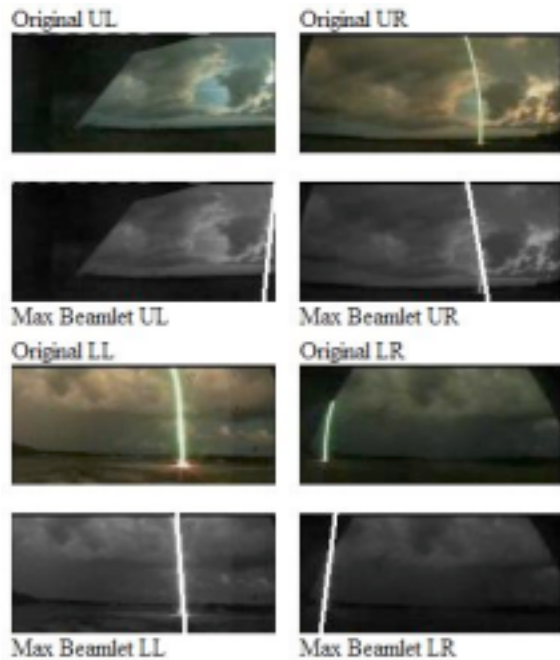


Figure 5: Maximum beamlet for each quadrant

Figure 5 shows the original quadrants of the sample frame. Below each original quadrant, the maximum beamlet is shown superimposed on the grayscale image. In the case of the upper left (UL) image, the beamlet does not contain enough high-intensity pixels to be classified as lightning. Whereas, in the other three cases, the beamlets do contain enough high-intensity pixels to be classified as lightning. It can be seen that the maximum beamlets in those three cases clearly enclose the lightning bolt present in the image.

The following is the statistical information obtained from the frame quarters shown in Figure 5.

- **Upper Left**
 Number of "lightning" pixels in max beamlet = **39**
 Number of total "lightning" pixels in the image = **78**
 Ratio (beamlet/total) = **0.500000**
- **Upper Right**
 Number of "lightning" pixels in max beamlet = **37**
 Number of total "lightning" pixels in the image = **43**
 Ratio (beamlet/total) = **0.860465**
- **Lower Left**
 Number of "lightning" pixels in max beamlet = **101**
 Number of total "lightning" pixels in the image = **112**
 Ratio (beamlet/total) = **0.901786**
- **Lower Right**
 Number of "lightning" pixels in max beamlet = **47**
 Number of total "lightning" pixels in the image = **49**
 Ratio (beamlet/total) = **0.959184**

4. Conclusion

In conclusion, the lightning image classification system described in this paper had a success rate of 91% when identifying images with lightning. Moreover, it had a 0% misclassification rate. That is, no non-lightning frames were incorrectly classified as lightning frames.

This project can be further improved by adding elements that will create a video file using the frames classified as lightning. Such a video should include all of the frames between the first and last identified lightning frames and an extra ten frames before and after the first and last lightning frame. This output format will guarantee that lightning frames will be captured with an almost 100% accuracy.

5. References

[1] EEL6825: Lightning Image Processing System website [online]. URL: <http://ieee.ece.ufl.edu/~lightning/>. Retrieved on April 9, 2003.

[2] Lightning Research Laboratory (UF) website [online]. URL: <http://www.lightning.ece.ufl.edu/>. Retrieved on April 9, 2003.