10-12-2018

Real-Time Video Processing of Arcing Events to Determine Coincidence

JR Dennison
Utah State Univesity

Gregory Wilson
Utah State University

Jonh Mojica Decena
Utah State University

Brian Wood
Utah State University

Real-time Video Processing of Arcing Events to Determine Coincidence

JR Dennison, Gregory Wilson, Jonh Mojica and Brian Wood

USU Materials Physics Group
Utah State University, Logan, UT 84332-4414

APS physics 4CS FOUR CORNERS SECTION

Abstract

A system has been developed and tested for real-time monitoring of environmentally-induced electrostatic discharge events to test spacecraft component and material survivability. Simultaneous detection by several parallel methods in coincidence, enhances event detection, minimizes false signals, and collects complementary information to determine arc location, intensity, and timing. This research focuses on four computer-interfaced video cameras which provide spatial and temporal detection of visual arcing from the surface of various elements. A real time processing solution was developed which can calculate integrated intensities, sensitively detect intensity threshold events, and store relevant video frames from these threshold events. Post-processing of this data can generate activity maps and give detailed threshold event information. This selective approach not only saves vast amounts of disk space and post-processing time, but it facilitates real-time monitoring of month long experiments. An experiment which induces arcs on insulating dots on a conductive substrate using high-energy beta radiation from a Sr^{90} source in the Space Survivability Test vacuum chamber was performed to demonstrate the quality of captured data and the effectiveness of the analysis methods.

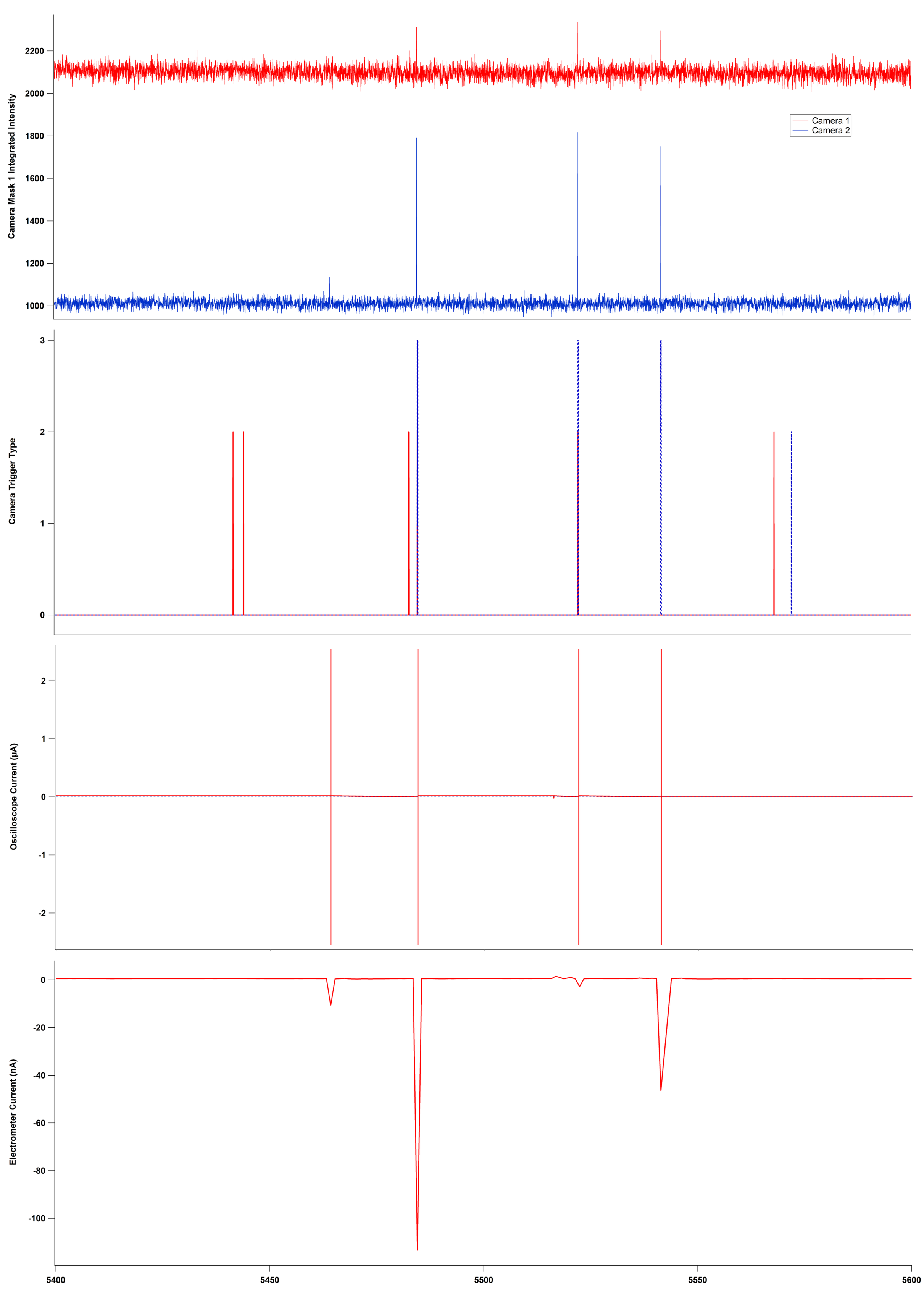

Figure 1. Comparison of timing of arcs between (a) Camera integrated intensity, (b) Camera triggers, (c) Oscilloscope current, (d) Electrometer current.

I. Introduction

A *Windows* program was developed to analyze and record the intensity versus time of multiple video capture devices in real time. Because data runs could take several days to weeks, raw video capture became very memory intensive. In order to make the program more efficient, a real time approach was developed. Because frames of interest still needed to be captured, an automatic triggering system was developed to save frames of interest. This system not only saves vast amounts of data, but it also provides near real time data analysis of a data run. By combining this data with other measurement devices, a detailed picture of the data run can be formed as shown in Figure 1.

II. Data Processing

1. Data is captured using either *openCV* or the *VideoCapture Library*. They both use either *VFW* or *DirectShow* libraries to accomplish this.
2. Image is captured in the *CameraCapture* thread as a 24 bit BGR image and saved to the image buffer along with its time stamp.
3. In the processing thread, an image is then pulled from the buffer as well as its time stamp. The image is then converted to a 16 bit grayscale using one of the following formula:
   - Intensity: Gray = (B*G*R)/253.016
   - Avg Intensity: Gray = (1/3)*(B+G+R)*257
   - Luminance: Gray = (0.299 * B + 0.587 * G + 0.114 * R)*257
   - Value : Gray = Max(B,R,G)*257
   - Geo Intensity = $148.379\sqrt{B^2+R+G}$
4. The intensity is then integrated over the entire frame and divided by the number of pixels to get an average intensity from 0 to 65535.
5. The average intensity is then calculated for any configured masks.
6. These data are then saved to a text file.
7. The average intensity for each mask is then compared to its 3 point moving average and if the difference between these two values is greater than the Intensity Trigger Threshold a T1 trigger is recorded and the frame is saved as a png.
8. Each pixel's intensity is also compared with the pixel intensity of the previous frame and if their difference is greater than the Pixel Intensity Trigger than a T2 trigger is recorded and the frame is saved as a png.

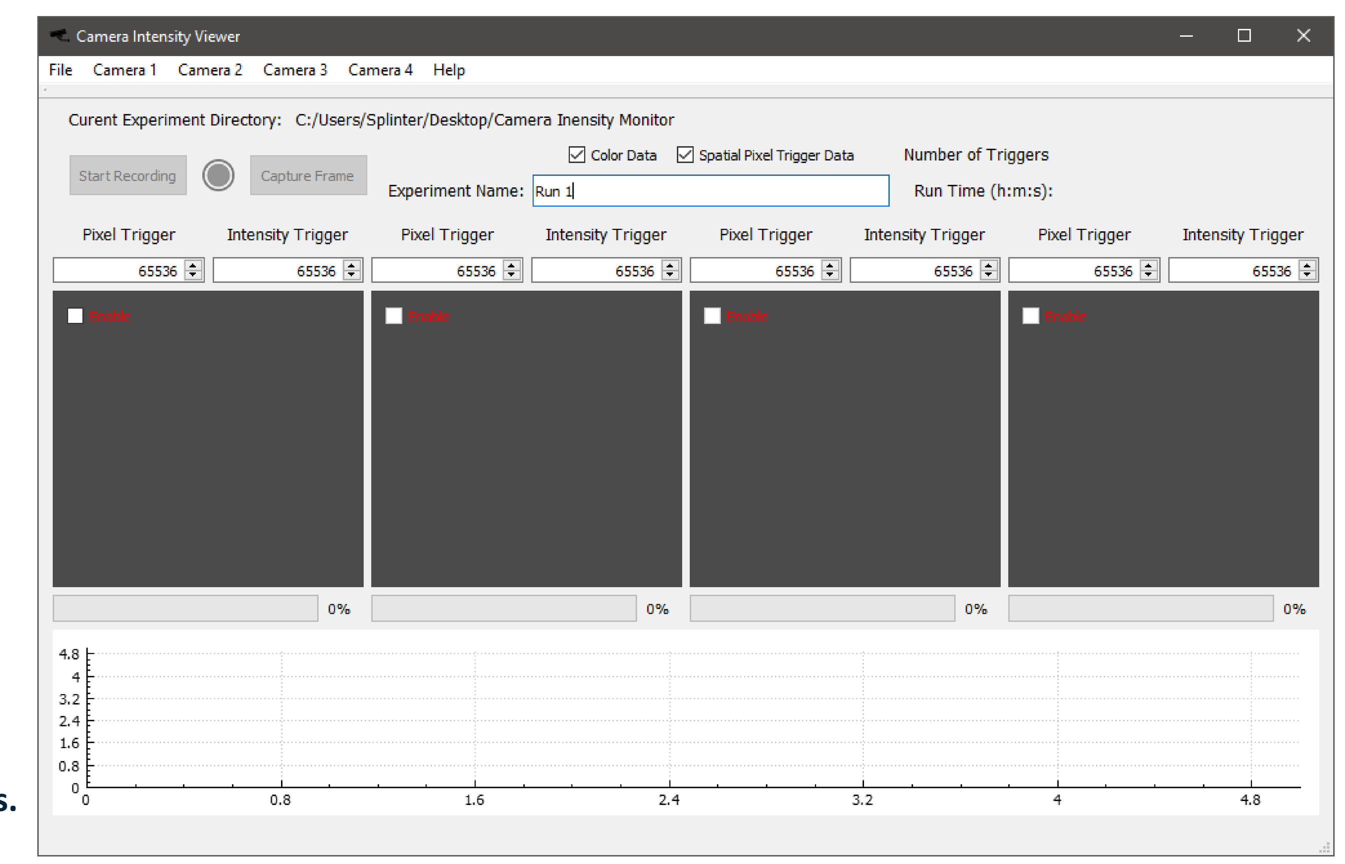

Figure 2. Screenshot of the main program.

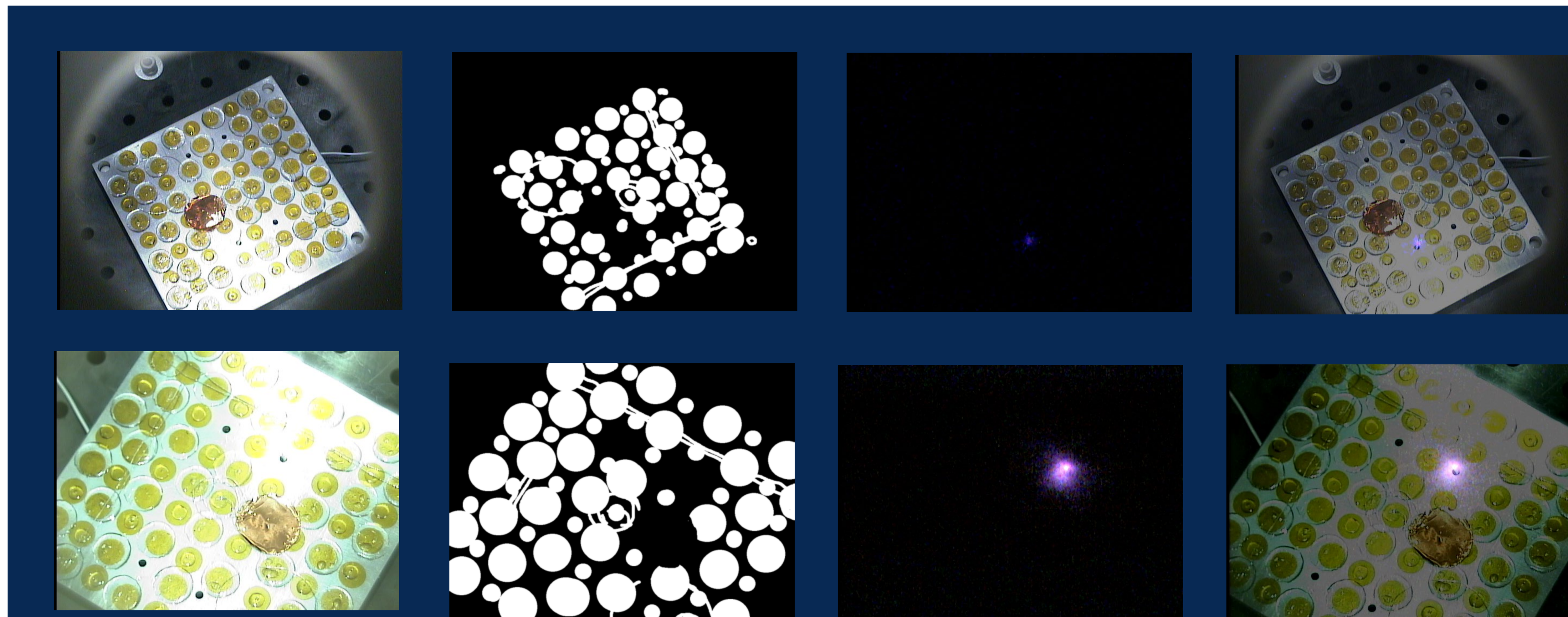

Fig. 3. (a,e) Reference images of the test sample. (b,f) The masks which defined the regions of interest for the two cameras. (c,g) An example arc as shown in both cameras simultaneously. (d,h) An overlay of the arc on the reference image.

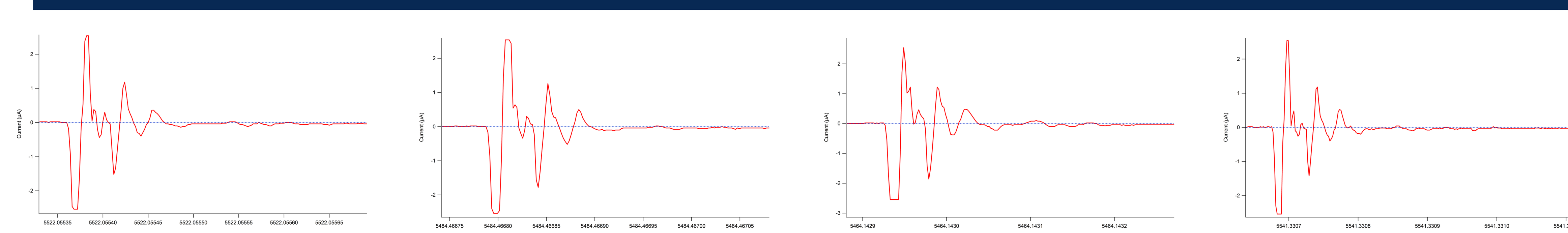

Figure 4. Oscilloscope traces of the four arcs shown in Figure 1.

III. Video Trigger Types

There are two types of triggers within the program. T1 triggers occurs when the integrated intensity of the full frame or the integrated intensity of any of the mask regions exceeds a three point moving average set by a user defined threshold. T2 triggers occurs when a pixel value exceeds its previous value by a user defined threshold value.

You can then adjust the "Pixel Trigger" and "Intensity Trigger" values from a scale of 1 to 65536. The "Pixel Trigger" adjustment is the T2 threshold value for the change in intensity for individual pixels from one frame to the next. The "Intensity Trigger" T1 threshold value for the average intensity change from the current frame from a three frame running average. As you adjust the trigger value while previewing a video source, a red box around the trigger text will flash, indicating that a trigger has occurred. This can be used to help tune the trigger settings before a run.

When either trigger occurs, the current frame, the frame before and the frame after is recorded. For pixel triggers, the pixel location and color values are also saved. Using the pixel location, a heat map, or activity map, can be created which shows the number of times a given pixel trigged over the duration of the run (see Figure 5).

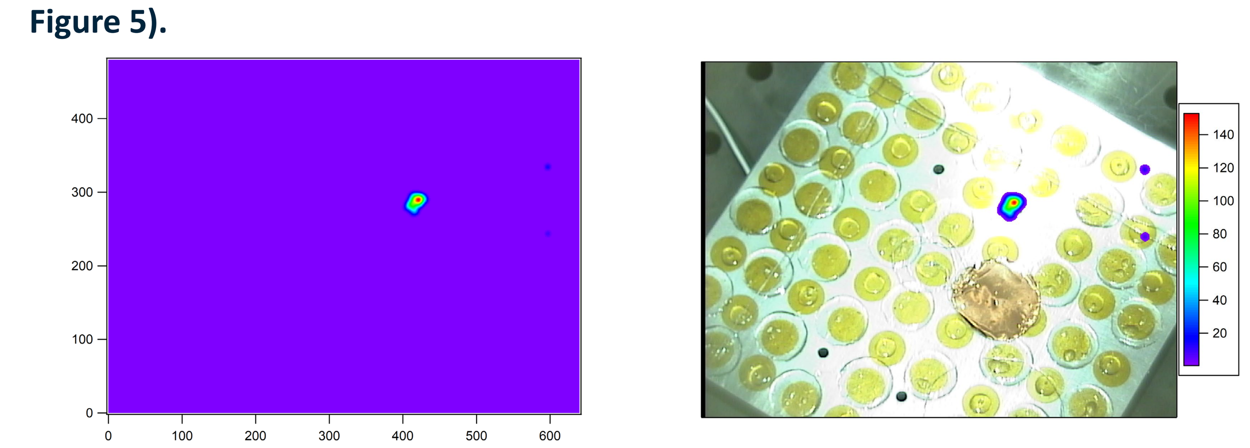

Fig. 5. (a) Heat map of pixel triggers which shows the number of times a pixel at a given coordinate triggered. (b) An overlay of the heat map on a reference image.

IV. Noise and False Triggers

Because runs are generally done under dark conditions, noise can become an issue; this is especially true when a Sr^{90} beta radiation source is used due to stray X-rays hitting the imaging sensor. A long term test was done to determine the number of triggers for a given threshold value. The results of this test show a homogenous image of random noise as shown in Figure 6. Because each camera has slightly different amplification and threshold values, it is important to take dark data before every run to determine false trigger rates.

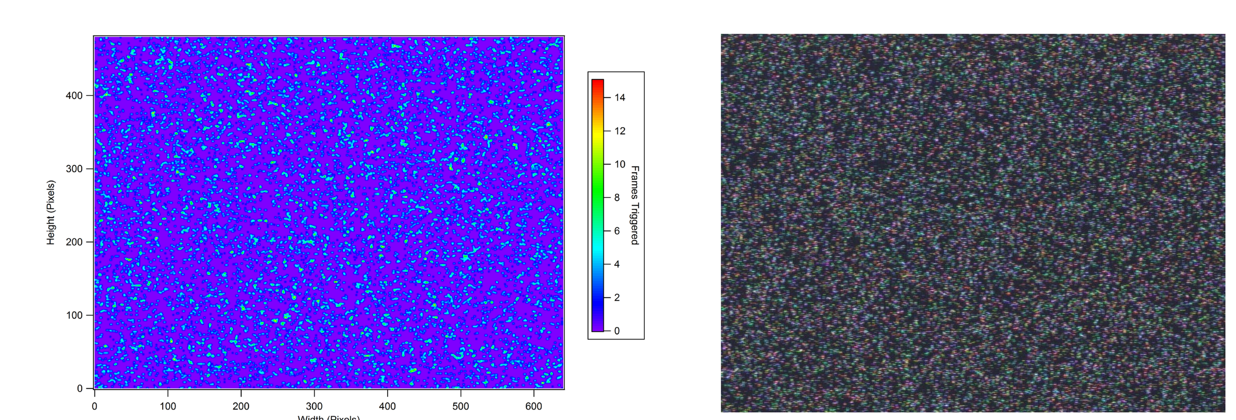

Figure 6. (a) Heat map of pixel triggers for a dark run. Pixel triggers here are due to single pixel events from sensor noise. (b) A composite image showing all of the pixel triggers due to noise.

V. Future Work

Further studies will determine which color to grayscale method is the most reliable and efficient at detecting arcs. This software will be extended to accomadate 16 bit grayscale video cameras to measure cathodoluminescence. This software also has the ability to load in video files previously recorded to perform data analysis on them. This will not only help with the analysis of previously recorded video, but allow different conversion methods and various trigger values to be compared in a controlled environment

UtahState University

We gratefully acknowledge contributions from other members of the Materials Physics Groups. This work was supported through funding a Utah NASA Space Grant Consortium Graduate Fellowship.

Scan code to access the USU Material Physics Group papers and presentations.

UTAH STATE UNIVERSITY MATERIALS PHYSICS GROUP