



Annual ADFSL Conference on Digital Forensics, Security and Law

2018
Proceedings


May 17th, 11:15 AM - 11:50 AM

Live GPU Forensics: The Process of Recovering Video Frames from NVIDIA GPU

Yazeed M. Alabtain
Purdue University, yalabta@purdue.edu

Baijian Yang
Purdue University, byang@purdue.edu

Follow this and additional works at: <https://commons.erau.edu/adfsl>

 Part of the [Computer Law Commons](#), [Defense and Security Studies Commons](#), [Forensic Science and Technology Commons](#), and the [Information Security Commons](#)

Scholarly Commons Citation

Alabtain, Yazeed M. and Yang, Baijian, "Live GPU Forensics: The Process of Recovering Video Frames from NVIDIA GPU" (2018). *Annual ADFSL Conference on Digital Forensics, Security and Law*. 3. <https://commons.erau.edu/adfsl/2018/presentations/3>

This Peer Reviewed Paper is brought to you for free and open access by the Conferences at Scholarly Commons. It has been accepted for inclusion in Annual ADFSL Conference on Digital Forensics, Security and Law by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

EMBRY-RIDDLE
Aeronautical University™
SCHOLARLY COMMONS

(c)ADFSL



LIVE GPU FORENSICS: THE PROCESS OF RECOVERING VIDEO FRAMES FROM NVIDIA GPU

Yazeed Albabtain, Baijian Yang

Purdue University

Department of Computer and Information Technology, Purdue University 401 N. Grant
Street, West Lafayette, IN 47907
{yalbaba, byang}@purdue.edu

ABSTRACT

The purpose of this research is to apply a graphics processing unit (GPU) forensics method to recover video artifacts from NVIDIA GPU. The tested video specs are 512 x 512 in resolution for video 1 and 800 x 600 in resolution for video 2. Both videos are mpeg4 video codec. A VLC player was used in the experiment. A special program has been developed using OpenCL to recover 1) patterns that are frames consist of pixel values and 2) dump data from the GPU global memory. The dump data that represent the video frame were located using simple steps. The recovery process was successful. For 512 x 512 resolution video, the frames were partially recovered but it shows enough information for the forensics investigator to determine what was viewed last. The research indicates that it is harder, but not impossible, to obtain a viewable frame from higher-resolution video

Keywords: Forensics, GPU forensics, Video forensics, OpenCL, NVIDIA GPU, Global memory

1. INTRODUCTION

Video forensics has not been addressed widely due to the difficulty of the process. Considering how complicated the GPU structure and the volatility of the data, the process of recovering the evidence is challenging. One aspect of video forensics involves analyzing, measuring, and comparing video scientifically as part of a criminal investigation. With the introduction of advanced media players, face-recognition software, and biometric-face matching technology, video forensics has been moved a notch higher. Reyes et al. indicates that

live forensics is essential nowadays especially with the evolution of technology (Reyes, Steele, O'Shea, & Britton, 2007). As it is a common practice by law enforcements to pull the plug the amount of information lost there is unbearable (Reyes et al., 2007). Live forensics will provide more information to discover how, when, and who did what in a criminal event, especially when handling volatile data (Rogers, Goldman, Milan, Wedge, & Debrot, 2006; Hale Ligh, 2014). Rogers et al. introduced an on-site forensics model that allows investigators to identify, analyze, and interpret digital evidence without the need to go to the lab

for more inspection (Rogers et al., 2006). The Rogers forensics model involves practicing live forensics on volatile memory (RAM), which has been used by law enforcement for many years. The technique introduced in this research is limited to video forensics and involves live forensics on GPU volatile memory. To the best of the author's knowledge, this is the first GPU live forensics technique that harvests the information directly from the GPU memory. The amount of information stored in a GPU is massive and could benefit the forensics field from many aspects. This research is significant because the method presented can be applied on a real-case scenario if the video player used is a VLC player and the computer is still powered on when investigators arrive. The technique presented will allow the forensics investigator to recover the last video frame seen by the user by using patterns that are generated from the suspect machine. NVIDIA GTX560M was used for the test along with a 340.43 driver. Recovery from 512 x 512 video was successful, but the results from 800 x 600 video were not as encouraging because only a limited number of patterns could be generated. However, this issue may be addressed by generating more patterns, thereby increasing the chance of having viewable video frames. The rest of the paper is organized as follows. Section 2 discusses previous work related to the topic. Section 3 discusses the recovery process of the video frames. Section 4 discusses the results of the research. Section 5 discusses future research and offers a conclusion.

2. RELATED WORK

Zhang attempted to discover a forensically sound method to recover data about an image from an NVIDIA GPU by creating software to run several tests to request and store information directly from the GPU

(Y. Zhang & Yang, 2015). The first test checked the availability of such information. The second one checked its integrity. The third locates the test image data and extracts it from GPU memory. The recovery process was divided into three sections: recovery of color data, validating color data against the original, and testing whether such a procedure could be used in different environments (Y. Zhang & Yang, 2015).

The results of Zhang study shows that it is possible to locate and access information on GPU memory on full volume without damaging it. The examiner can even replicate the data pattern. However, a certain number of such patterns need to be recovered before it is possible to reconstruct the image. The forensic soundness of such a method was not proved because the examiner was not able to repeat the results on the same image because of the way the data is stored in GPU. Nonetheless, a new test called Color Depth Map was used in the process in order to restore the data from GPU by creating a number of image data conversion matrices. According to Zhang that the method presented in the study cannot yet be called forensically sound because the data in the GPU memory is not directly linked to the original image but instead it represents the image that will be shown on the monitor. However, this is not the case for video files. In the end, Zhang hopes that with some modification, this new test will be able to become forensically sound or at least be able to guide the research in the right direction. The research is extremely important because of the absence of reliable forensic methods of retrieving and researching data from GPU (Y. Zhang & Yang, 2015).

During court cases, digital videos form a very important part of the evidence. Assessment of the accuracy of these digital videos helps in deciding whether the videos are authentic and can be used as evidence. Tamer

Shanableh focused his study on how much a court can rely on the digital videos as evidence and how to detect whether the video was tampered with. Shanableh found that machine learning technique (MLT) is reliable and is being used all over the world to determine if a digital video was altered (Shanableh, 2013). Shanableh concluded that these machine learning techniques are viable and feasible to detect if the video was tampered with and are forged irrespective of the fact that how many frames were deleted. The restraining fact was that it should not be a multiple of length of the group of the pictures. The technique was trained and laden with both forged and unaltered video samples. Shanableh used three different MLTs to show that the results generated are authentic and accurate. The three techniques that were used included KNN (or K Nearest Neighbor), Support Vector Machines (SVM), and logistic regression technique (Shanableh, 2013). The author affirmed that the technique is extremely reliable in detecting forged videos and helping the court in deciding the validity and accuracy of the digital videos (Shanableh, 2013).

Van Houten and Geradts discussed a new method that can be used to determine the source of a video, such as videos uploaded on YouTube. Determining the source of a video is becoming increasingly necessary, especially if the case is about child pornography or abuse (Van Houten & Geradts, 2009). People use the Internet to propagate their ideas, and in some cases people such as pedophiles may upload illegal images over the Internet. Therefore, tracing videos back to a source is important, especially for law enforcers (Van Houten & Geradts, 2009).

Currently, device identification is achieved by extracting undetectable sensor pattern noise from image residues of the image sensor in the image structure. The sensors act as unique signatures and each device has its

own unique pattern; therefore, the pattern can be compared with the source camera even in a database. When two or more patterns are similar, it indicates that both images could have the same origin. Van Houten and Geradts therefore suggested that it is necessary to create a database of images, such as images of child pornography, that can be linked together to find a perpetrator (Van Houten & Geradts, 2009). The source of these videos can also be determined from YouTube, which, after compression, lowers the level of noise (Van Houten & Geradts, 2009). However, this method has some limitations. Changing the aspect ratio or resizing negatively affects the sensor noise, which may lead to erroneous identification of the source device. Furthermore, it is impossible to tell the initial codec setting or resolution in which a video was uploaded. Finally, as manufacturing standards improve, the pattern noise will likely decrease. In addition, YouTube and other cloud platforms keep upgrading their websites, which may in turn alter the settings of a video. Still, while the method is not perfect, it can be used to solve many crimes and stop illegal activity, such as child pornography (Van Houten & Geradts, 2009).

Milani et al. discussed the strengths and weaknesses of several video forensic techniques (Milani et al., 2012). Video forensics can use several techniques to authenticate the originality of a video, including video sharpening, video stabilization, masking, interlacing, and demultiplexing. Milani et al. also discussed camera artifacts, image compression, and geometric/physics inconsistencies (Milani et al., 2012). These methods have existed for a long time and can be considered basic methodologies for establishing authenticity. But Milani et al. also discussed the methodologies of signal analysis, including ballistic fingerprinting, detection of reacquisition, and detection of copying (Milani et

al., 2012).

Milani et al. provided an overview of the tools used for video compression, network footprint identification and video compression anti-forensics (Milani et al., 2012). When it comes to video forensic techniques, inter-frame analysis is an important technique used extensively. Because a video is composed of multiple frames, there is always a possibility to insert, delete, and duplicate frames, thereby modifying the content. Thus, effective techniques should be developed to identify these forgeries. The authors also mentioned aspects of video forensics, such as block detection, quantization step detection, and identification of motion vectors. Videos can also be forged during transmission. To detect such actions, footprint identification is used (Milani et al., 2012).

According to Gironi et al., a digital video (DV) is a series of motionless images shot at an adequately high rate (Gironi, Fontani, Bianchi, Piva, & Barni, 2014). The resulting signal may be compressed by lowering the temporal and spatial redundancy through the use of video coding algorithms such as H.264 and MPEG-2 (Gironi et al., 2014). These algorithms utilize a block-oriented combination video coding technique and split images into varying kinds, such as intra-coded images or I-frames, as well as predictive-coded images referred to as B-frames and P-frames (Gironi et al., 2014).

DVs are utilized for security reasons because of their higher reliability as evidence compared to still pictures. It is much harder to forge a video than a picture, although it is still possible to tamper with a video through the insertion, deletion, or duplication of frames, which may make the video worthless (Gironi et al., 2014).

The goal of video forensics is to analyze the procedural history of a digital video, which is generally stored in a compressed format. Thus, multiple techniques have been

developed to ascertain whether a DV has been encoded more than once, which may indicate forgery in some cases. Additionally, video forensics examines the integrity of digital videos by attempting to disclose manipulations such as inter-frame and intra-frame forgeries (Gironi et al., 2014).

One method of detecting video forgery depends on the fact that a double compression suggests a forgery where the first compression occurred while attaining the video and the second compression occurred after manipulation. Thus, the double encoding detection technique involves alterations to the video, making it robust to frame elimination between the pair of encodings. Afterwards, an algorithm is developed to repeatedly utilize this technique to monitor whether a misalignment exists in the frame setup between the pair of encodings. If there is a misalignment, the next step involves locating its position and grouping the attack as either frame insertion or elimination (Gironi et al., 2014).

Bress, Kiltz, and Schaler carried out a study to investigate the key challenges as well as countermeasures associated with forensics on GPU coprocessing in databases (Sebastian Bress, 2013). The research also provides an analysis of how data can be retrieved from the GPU RAM. In earlier studies, this was achieved by utilizing other processes whereby the memory dump of a device memory would be created. This provided a gap for bypassing the access controls of the database management systems (DBMS) (Sebastian Bress, 2013).

According to Bress, Kiltz, and Schaler, one of the key challenges in GPU coprocessing in databases is that access restriction makes it difficult to carry out forensic investigations. In addition, the authors indicates that direct access to GPU may cause security issues. Another challenge is the closed nature of the application programming interface APIs, which limit access con-

trol (Sebastian Bress, 2013). Bress, Kiltz, and Schaler opine that there is a need to conduct forensic inspections of GPUs so that information systems processing confidential data can take advantage from the GPU acceleration. The key challenges for secure co-processing that can result from bypassing the access controls of the DBMS include the loss of confidentiality, violation of data integrity, as well as availability. Furthermore, it also leads to non-repudiation and lack of authenticity. In addition, GPU coprocessing on databases also presents major challenges when it comes to forensics. These challenges come in the form of the lack of appropriate approaches to conduct data recovery, memory interpretation, and the reconstruction of program flow (Sebastian Bress, 2013). To obtain a memory dump from the GPU, Bress, Kiltz, and Schaler executed a program that writes a fixed string specified by the user to the GPU RAM, which is followed by memory allocation after the initial step is terminated. The allocated memory is then copied from the GPU RAM to the CPU RAM, which creates a data and memory dump (Sebastian Bress, 2013).

Albabbain and Yang introduced a method that recovers images and web-pages from the GPU global memory using OpenCL (Albabbain & Yang, 2017). The authors indicated that the recovery process is not possible without cleaning the GPU global memory first which is a huge hurdle when it comes to forensics, as this step will alter the original data and therefore become dismissed in court (Albabbain & Yang, 2017).

Most authors used the GPU to accelerate or enhance the forensics process but very few used forensics to extract evidence from the GPU. Considering the fact that the GPU holds vital information stored in the volatile memory and the fact that the chip is programmable, it is important to develop a method that extracts these informa-

tion. Law enforcement officers can benefit from such information because it can help in solving cases especially child pornography cases.

3. EXPERIMENT

This experiment was conducted in an unaltered setup to bring it closer to reality. The GPU model used in this test is GTX560M with driver number 340.43. The test consisted of three main stages:

The process starts with viewing the video using VLC player then collecting evidence using OpenCL. Once the evidence is secured, then the second stage starts by generating patterns from the same machine. The process of generating patterns can be repeated to recover as much patterns as possible, because this will increase the chances of having a better clear picture of the recovered frame (Albabbain & Yang, 2017).

3.1 Stage one (preparing the evidence):

Multiple videos were prepared for the test. Each video had different specs.

1. Video 1: A 512 x 512 resolution video with mpeg4 codec of NVIDIA CEO Jensen Huang presenting at an NVIDIA conference.
2. Video 2: An 800 x 600 resolution video with mpeg4 codec of NVIDIA CEO Jensen Huang presenting at an NVIDIA conference.

Both videos were viewed using a VLC player, which is a well-known media player in the market. Stage one involved simply opening the video file for a few seconds, then closing the program. After that, the author collected the dump data from the GPU global memory. The author used a specially developed OpenCL code that extracts the dump

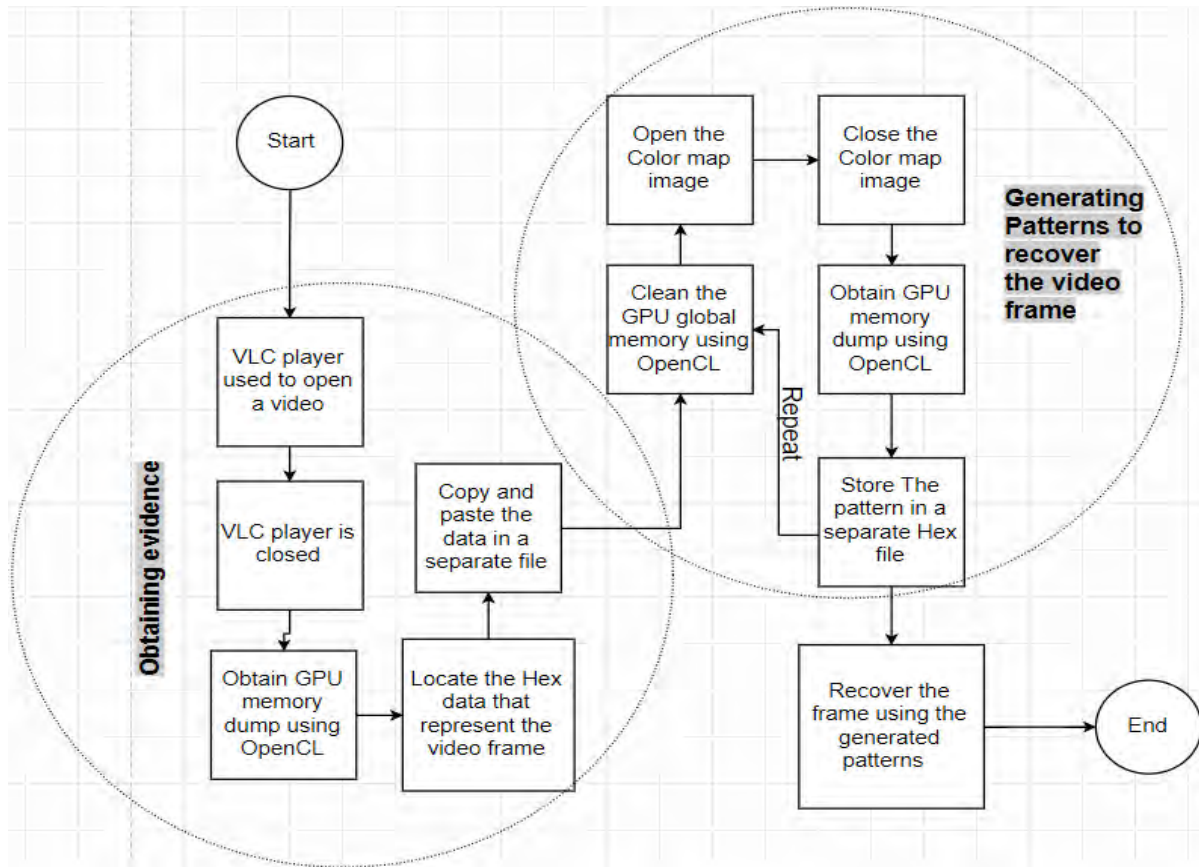


Figure 1. Video frame recovery roadmap.

data from the GPU global memory. The structure of the GPU dump data has been studied thoroughly, and the hex file consists of six main parts. Parts one through five hold parts of the last viewed frame, but the sixth part of the hex file holds the full frame that was displayed. To locate the sixth part in the hex file, the following steps must be followed:

- Because the used application is a VLC player, there is a unique Hexadecimal value that represents the start of the VLC dump data.
- For this experiment, the start of the frame corresponded with 13A20000 offset. The hexadecimal values in that offset are: "10 10 10 00 10 10 10 00 10 10 00 10 10 10 00" which represents

the start of the VLC video player data. This represents part one, which holds a partial frame and is not important, although it does help in locating the full frame which is located in part six, which is in this case, the offset 13F20000.

- Copy the data starting from 13F20000 to the end of the dump file and paste it in a new document. These data represent the last viewed frame.

Once the data have been obtained, the investigator now can generate patterns that will help reconstruct the recovered frame to a viewable image.

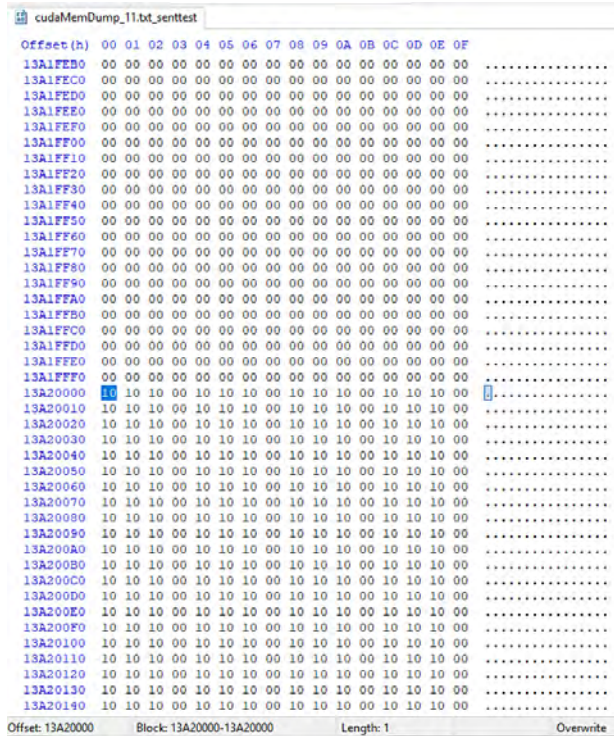


Figure 2. The beginning of the dump data that represent the video frame .

3.2 Stage two (generating patterns):

A 512 x 512 and an 800 x 600 color map pattern were created using Adobe Photoshop. Zhang used the color map pattern method to establish a correlation between pixels used for evidence and those in the original image (Y. Zhang & Yang, 2015). As a result of the test, it will be possible to reconstruct the original image using the data that were stored in an array. Colors in the array were stored as decimal values of RGB colors. Us-

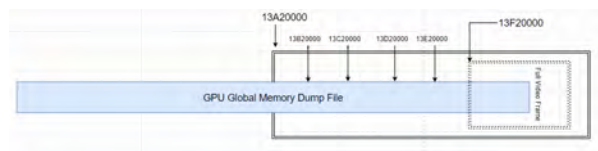


Figure 3. GPU global memory dump file structure .

ing the same sequence to restore the colors to the same places to create an image means that there is a certain pattern by which colors are written into dump memory (Y. Zhang & Yang, 2015). The researcher used RGB mode because it represents the three colors that an average monitor uses to create all other colors from color palette.(Q. Zhang & Xiao, 2014) Eight initial images were used to represent all possible interactions of these three colors. This test was used to test the hypothesis that it is possible to create a forensically sound method of extracting information from GPU in an undamaged and unaltered state. Even though the hypothesis was not proved, Color Map Pattern tests revealed a clear correlation between evidence and the original data. The author managed to successfully map and extract data and reproduce it; however, it was not an exact copy of the original because the color pattern was not the only data that were written to memory about the image. In stage two, the author used the same computer and the same GPU GTX 560M to generate patterns and save them in an attempt to reconstruct the frame recovered from stage one to a viewable status that could help forensic investigators. A total of 11 patterns were generated by cleaning the GPU global memory using OpenCL, then opening and closing the color map image presented in figure 4 using Windows photo viewer, and then ob-

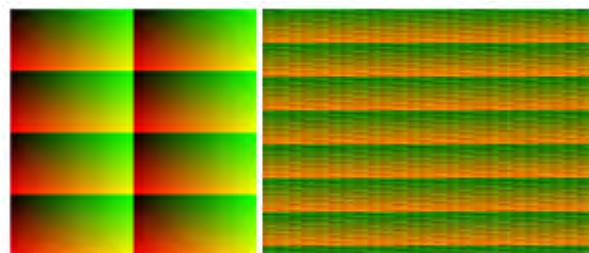


Figure 4. 512 x 512 and an 800 x 600 Color map patterns.

taining the dump data from the GPU using OpenCL. The functions used to obtain the dump data are `clEnqueueReadBuffer` and `clCreateBuffer`. The process was repeated in the hope of generating more unique patterns.

3.3 Stage three(recover the evidence):

In this final stage, the generated patterns were used to reconstruct the frame to its original shape (or close enough for forensics investigators). The patterns consisted of pixels, and each pixel holds a pixel ID. The program mapped the recovered frame to its original shape using the pixel values stored in these patterns.

4. RESULTS

The original frame to be recovered is shown in figure 5. The video has been opened using a VLC player for three seconds.

A total of 11 patterns were recovered after obtaining the evidence dump data. The process of generating patterns and recovering the evidence took 3.8 minutes. The results showed what was viewed last using the VLC player, which was a clip of NVIDIA CEO Jensen Huang presenting at an NVIDIA conference. Some patterns were not clear, but combining the results of the 11 patterns helped the investigator see what was viewed last.

The original 800 x 600 video frame is



Figure 5. 512 x 512 original video frame.



Figure 6. 512x512 recovered evidence.

shown above. Because of the increase in the resolution, the author increased the number of generated patterns to 14.

The results of the 800 x 600 video frame is not as encouraging as the 512 x 512 resolution. The recovery process took 4.9 minutes, almost 20 seconds per image. The results indicate that many more patterns have to be generated in order to reconstruct the frame to a viewable status. In general, video frames can be recovered from the GPU global memory even in a real-case scenario, but certain conditions have to be met:

- The suspect must use VLC player.
- The computer is still powered on to perform live forensics.
- Video resolution must be known, which is not a problem in most cases considering the computer is in the authorities' hands.

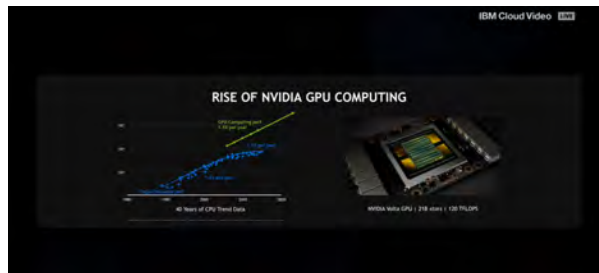


Figure 7. 800 x 600 original video frame.



Figure 8. 800x600 recovered evidence.

Certainly, there are some drawbacks for this technique, such as these:

- Running the OpenCL code on the suspect computer to obtain data from the GPU global memory will update the system's time stamp.
- Video resolution must be known to generate the correct patterns for reconstructing the frame to a viewable status.
- Only the last processed frame can be recovered.

5. CONCLUSION

In this technologically advanced world, data plays a significant role in everyone's life. The GPU holds rich data for the forensics investigator. Access to these information will certainly help in solving cases. This research is a step forward to reach that goal. According to Rogers et al., live forensics has been applied by law enforcement as some cases are extremely time sensitive (Rogers et al., 2006). It can be concluded that the higher the video resolution, the more patterns need to be created. In the 512 x 512 resolution video, the results were enough with 11 patterns; but for the 800 x 600 video, the results were not satisfying even with 14 patterns. Also, although it is not impossible to recover video frames from the GPU global memory, data volatility can make the process impossible. For instance, if the computer is powered off, data cannot be obtained.

5.1 Future work

For further study, testing different GPUs will certainly add more knowledge to this field and will help investigators understand the limitations. The fact that video resolution must be known to recover the frame is an obstacle that can be overcome, especially if the need is to recover a viewable frame and not a fully recovered frame. Also, it is important to note that this research is not limited to VLC player only. VLC player we chosen to the fact that it is a popular video player. Considering testing other video players will help understand how exactly GPU handles video data, which ultimately will improve this research. Another issue to this research is the data integrity, perhaps creating another level of security by hashing the data or any other method is important to preserve the integrity of the recovered data. Incorporating machine learning to digital forensics is certainly an interesting approach. Bhatt and

H. Rughani introduced a new approach by adding forensic analysis to a machine learning open source program called “DeepQA” (Bhatt & H. Rughani, 2017).

ACKNOWLEDGEMENTS

This research is supported by King Faisal Specialist Hospital and Research Center KFSH&RC and Intel Grant No. 301620

REFERENCES

- Albabbain, Y., & Yang, B. (2017). GPU FORENSICS: RECOVERING ARTIFACTS FROM THE GPUS GLOBAL MEMORY USING OPENCL. Retrieved from https://www.academia.edu/35497121/GPU_FORENSICS_RECOVERING_ARTIFACTS_FROM_THE_GPUS_GLOBAL_MEMORY_USING_OPENCL
- Bhatt, P., & H. Rughani, P. (2017). *Machine learning forensics: a new branch of digital forensics*. Retrieved from <http://www.ijarcs.info/index.php/Ijarcs/article/download/4613/4155>
- Gironi, A., Fontani, M., Bianchi, T., Piva, A., & Barni, M. (2014). *A video forensic technique for detecting frame deletion and insertion - ieee conference publication*. Retrieved from <http://ieeexplore.ieee.org/document/6854801/>
- Hale Ligh, C. A. L. J. . W. A., M. (2014). The art of memory forensics: Detecting malware and threats in windows, linux, and mac memory. Retrieved from <http://www.wiley.com/WileyCDA/WileyTitle/productCd-1118825098.html>
- Milani, S., Fontani, M., Bestagini, P., Barni, M., Piva, A., Tagliasacchi, M., & Tubaro, S. (2012). *An overview on video forensics*. Retrieved from <http://ieeexplore.ieee.org/document/6334348/>
- Reyes, A., Steele, J., O’Shea, K., & Brittson, R. (2007). *Cyber crime investigations - 1st edition*. Retrieved from <https://www.elsevier.com/books/cyber-crime-investigations/reyes/978-1-59749-133-4>
- Rogers, M., Goldman, J., Mislan, R., Wedge, T., & Debrotta, S. (2006). Computer forensics field triage process model. *The Journal of Digital Forensics, Security and Law*. doi: 10.15394/jdfsl.2006.1004
- Sebastian Bress, M. S. (2013). *Forensics on gpu coprocessing in databases -research challenges, first experiments, and countermeasures*. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.382.9573>.
- Shanableh, T. (2013). *Detection of frame deletion for digital video forensics*. Retrieved from <https://www.sciencedirect.com/science/article/pii/S1742287613001102>
- Van Houten, W., & Geradts, Z. (2009). Source video camera identification for multiply compressed videos originating from youtube. *Digital Investigation*, 6(1-2), 48-60. doi: 10.1016/j.diin.2009.05.003
- Zhang, Q., & Xiao, C. (2014). Cloud detection of rgb color aerial photographs by progressive refinement scheme. *IEEE Transactions on Geoscience and Remote Sensing*, 52(11), 7264-7275. doi: 10.1109/tgrs.2014.2310240
- Zhang, Y., & Yang, B. (2015). Recovering

image data from a gpu using a
forensic sound method.

