Visual Analysis of Corrupted Video Data in Video Event Data Recorders

Youngbin Pyo
Department of Computer Science and Engineering
Korea University, Seoul
Email: vydudqls@korea.ac.kr

Choongin Lee
Department of Computer Science and Engineering
Korea University, Seoul
Email: choonginlee@korea.ac.kr

Heejo Lee
Department of Computer Science and Engineering
Korea University, Seoul
Email: heejo@korea.ac.kr

***Abstract*—With the rapid proliferation of video event data recorders (VEDRs), video file data from VEDRs are often used as the primary evidence in many fields, such as law enforcement. In this paper, we propose a method for reconstructing corrupted video files and capturing key events recorded in the video file for use as valid evidence. The method first extracts image features from each video frame and constructs a multidimensional vector. Subsequently, dimension reduction of these vectors is performed for visualization in low-dimensional space. The proper sequence of the video frames is restored by using a curve fitting technique for the low-dimensional vectors. Then, we calculate the change in the slope of the curve-fitted model to detect key events in video files. The proposed method generates significant results not provided by existing file recovery techniques.**

Keywords—Digital forensics, Visualization, Video sequence reconstruction, Video event detection

I. Introduction

With the increasing popularity of video event data recorders (VEDRs), video files recorded with a VEDR device have been used as primary evidence in field of law enforcement and insurance industries [9]. However, several factors can hinder video files from becoming useful evidence. One is external impact on the device at the time of recording. The power supplied to the VEDR device can be temporarily disconnected owing to the nature of traffic accidents accompanied by physical impact. In such cases, physical damage can corrupt the video being recorded or cause video encoding failure. Another factor is manual intervention by users. Video data can be lost from intentional acts such as the deletion or modification of video data by users. Existing file recovery techniques, such as file carving [8], identify a specific pattern using meta-information in the header of the file system [4]. These methods are applicable only if the files are in contiguous space. However, video files are generally large and easily fragmented. To solve this problem, signature-based recovery techniques [10] [1] have been proposed. Signature-based methods restore individual frames of video data discontinuously stored in the file system using different types of information, such as codec specifications. However, even if the frame is restored, there is still difficulty in restoring the file unit because of the absence of frame sequence information and partial omissions [5].

From this viewpoint, we extended existing methods to provide a method for analyzing video data by visually displaying the correlation between frames based on restored frames. In the case of VEDR video data recorded consecutively, since the context of the image is continuous, the preceding and subsequent frames contain more similar information than the other frames. To understand the relationship between these frames, we extracted image features containing information from each frame and constructed a multidimensional vector for each frame. In order to show the relationship between frame vectors, a dimension reduction method for the frame vector was applied to visualize in two dimensions. The distribution of the vectors through visualization shows the approximate structure of the image file to a human. Then, the frame sequence information is determined based on the data distribution, and the individual frames are reconstructed into one video file. The flow of the video can also be observed through the visualized result, and the point at which an event occurs, such as a collision, is detected.

The proposed method overcomes the fact that existing video restoration techniques cannot recover corrupted video files owing to limitations in available information. In the experiment based on frame data extracted from various video files, the proposed method successfully reconstructed visually recognizable video and automatically detected the event occurring in the video. In this paper, we demonstrate the proposed method using two datasets from the National Forensic Service and eight datasets captured by various VEDR devices in the market. Recorded video files in datasets were preprocessed by deleting meta-information so that the proper sequence of the frames is not given. As a result of analyzing the image features extracted from individual frames and reconstructing the videos through the proposed method, 88 percent of the frames were aligned with the original. With this method, we can reconstruct video files that are visually perceptible without meta-information. It is also possible to perform event detection based on the image feature change rate of the video frame unit. Existing file recovery techniques only analyze meta-information or signatures in a file system. The proposed method overcomes the limitations of other technologies by using visual aspects in video data.

The rest of this paper is organized as follows. In Section II, the background and related works on file recovery techniques and visualization methods is given. In Section III, the proposed method is described. Experimental results are presented in Section IV. and conclusions are drawn in Section V.

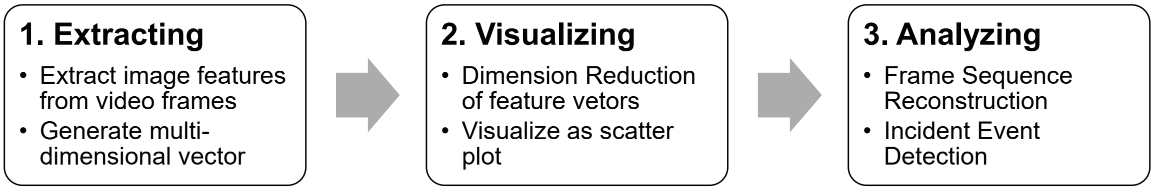

Figure 1: The process of visual analysis in the proposed method

II. BACKGROUND AND RELATED WORK

A. Video File Carving

File carving is a data recovery technique that reconstructs files based on a database of headers and footers for specific file types. Thus, file carvers can retrieve files from raw disk images without using file system metadata. File carving is an important technique in digital forensic cases because it provides recover data independent of the type of the file system.

File carvers use databases containing headers and footers, and search the disk image for occurrences of the headers and footers. In case of video files, the video file encoded by video codec store decoding header information at the start or end of the video file. Na et al. [7] restores the video file using a combination of frame data and decoding header information.

However, file carvers cannot restore files if data are not in contiguous space. Since a video file typically has a large volume of the data, it is highly likely to be split into several fragments. To resolve this problem, various techniques have been proposed by using a file meta-information such as a size of the file. Although those techniques produce good results in certain situations, it is still difficult to use it generally.

B. Visualization in Data Analysis

Data analysts need to understand a structure of data while analyzing complex data, and visualization methods can be helpful at this time. Recently, stochastic neighbor embedding (SNE) [3] and its extensions have drawn the attention of researchers for conducting dimensionality reduction and visualization tasks. SNE converts the high-dimensional Euclidean distances between data points into a conditional probability distribution related to Gaussian, which represents the pairwise similarity, and then requires the low-dimensional data to retain the same probability distribution. t-stochastic neighbor embedding (t-SNE) [6] is an extension of the SNE. SNE assumes normal distribution when measuring similarity between data, and t-SNE assumes t-distribution with 1 degree of freedom when calculating similarity between data. Since the range of the gradient descent value of the cost function that measures whether or not the high-dimensional data is well mapped to the low dimension is changed by distribution types, t-SNE represents the cluster relation in high-dimensional data at low levels better than SNE [14].

III. VIDEO DATA ANALYSIS USING VISUALIZATION

Existing technologies tend to rely on the remaining meta-information in video files. This means that if the meta-information is lost, the recovery rate is significantly lower. However, the proposed method is based on the analysis of frame-by-frame fragments obtained by existing techniques. The proposed technique can be divided into three phases, as shown in Fig. 1.

- Extracting phase: We extract image-based features in Fig. 2 from the frame and construct a feature vector to represent the frame.
- Visualizing phase: The feature vectors are visualized using a dimension reduction technique to provide visually meaningful results, while maintaining the relationship between the frames.
- Analyzing phase: Through the above results, we reconstruct the sequence of video frames and recover the original video file. In addition, specific driving events can be identified through an analysis of the feature vector space.

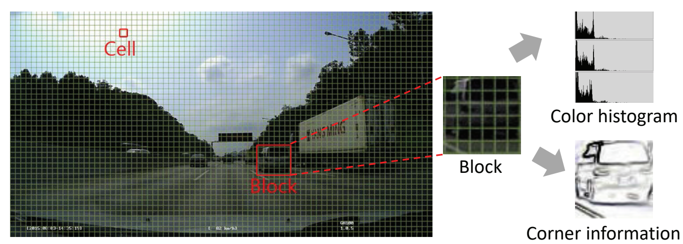

Figure 2: Components for the frame analysis

A. Extraction of Features from Video Frames

A video frame is the smallest semantic unit of a video file. However, it only has information about one image when there is no meta-information in the video file. Thus, we cannot know the correlation with other frames. In order to understand the correlation between video frames, we tried to construct a feature vector representing a frame by extracting image-based features [12]. The feature vector was constructed by considering the following points:

- Image feature: The VEDR device records the running of the vehicle consistently, so it records continuous scenes. Therefore, it is possible to use a global feature such as a color histogram to compare the similarity between frames. However, the global feature does not distinguish the background from the object, and does not detect the geometric change caused by the movement of the object. To overcome these problems, we combined local features to achieve complementary

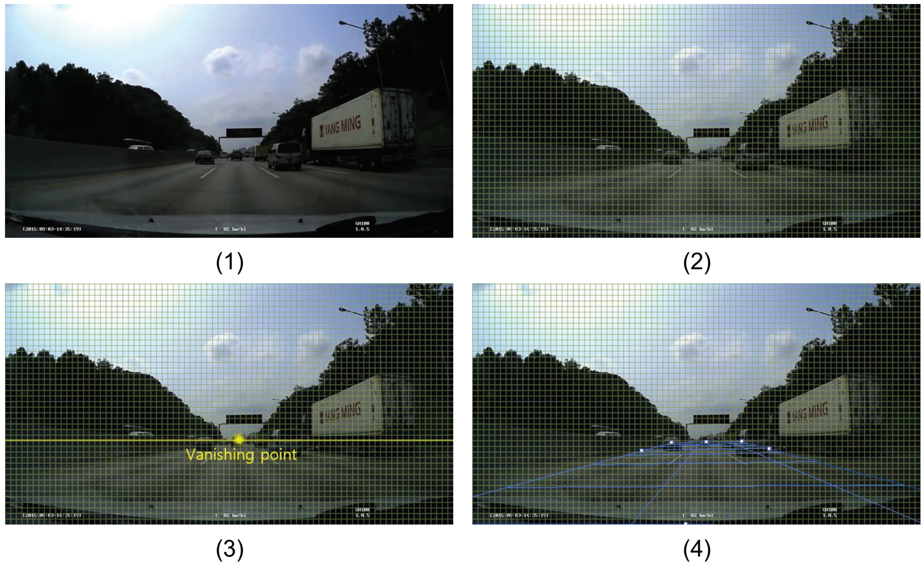

Figure 3: Creating blocks based on the vanishing point: (1) Raw image, (2) Divided with cells, (3) Set the vanishing point, (4) Blocks generated by the vanishing point

effects. In the case of the local features, the VEDR video should be robust to translation and rotation of the object because it is the main content of the vehicle driving. Thus, we applied the FAST corner detection method which satisfies the above characteristics as the local feature used in our method.

- Cell: Since typical VEDR devices basically record at least 30 frames per second, the computations for feature extraction increase exponentially with recording time. In this method, each frame was divided into small pieces with fixed sizes called cells. The cells were used as the minimum unit of each vector in the calculation process.
- Vanishing point: One of the unique characteristics of a VEDR is that the device is attached to a fixed location in the vehicle. Therefore, the vanishing point is at the same position in every frame. While an object moves among the frames, a geometric change in the object will be proportional to its distance from the vanishing point. Hence, these changes must be calculated relative to the location.

Fig. 3 shows the steps for extracting the feature vector from frames. In step 1, each frame is divided into connected cells with the same specific size. In step 2, the vanishing point is extracted from an arbitrary frame in order to apply the characteristics of the vanishing point as mentioned above. Since the VEDR device is mounted in a fixed location, such as the dashboard, the same vanishing point is applied to all the frames. To reflect the absolute movement of an object regardless of its distance from the vanishing point D_v, we performed inverse perspective transformations based on the vanishing points. The calculated region was divided into grids of a certain height P_h and width P_w, and then reflected onto the original region to create a block that bundles the cells N_c included in each grid.

$$N_c = D_v * P_w * P_h$$

In step 3, the features from each block were extracted. The features to be extracted are a color histogram as a global feature and corner information as a local feature. The color histogram was constructed for every pixel in each block. However, since the amount of computations increases exponentially with the number of color channels K_n^c, the histogram was produced first by discretization of the colors into a number of bins N_b. Therefore, the size of the histogram N_h can be calculated as

$$N_h = N_b * K_n^c$$

For the corner information, we applied the FAST corner detection method [11] to the entire image. Since the corner information was used to determine whether the same object is detected in each image, it was configured in units of frames different from the local features stored in block units. Finally, the color histogram of each block and the corner information of the frame were arranged to form a multidimensional feature vector for each frame. The feature vector of each frame containing i blocks with j features is written as

$$[X_{1,1}, X_{1,2}, \cdots, X_{i,1}, X_{i,2}, \cdots, X_{i,j}]$$

B. Visualization of Feature Vectors Using Dimension Reduction

The multi-dimensional vector generated in the previous step does not show the relationships between each frame by itself. For perceptually meaningful results, dimension reduction was performed for the visualization of the frames. However, it should be noted that there are many dimension reduction techniques, and each one has different features; hence. we

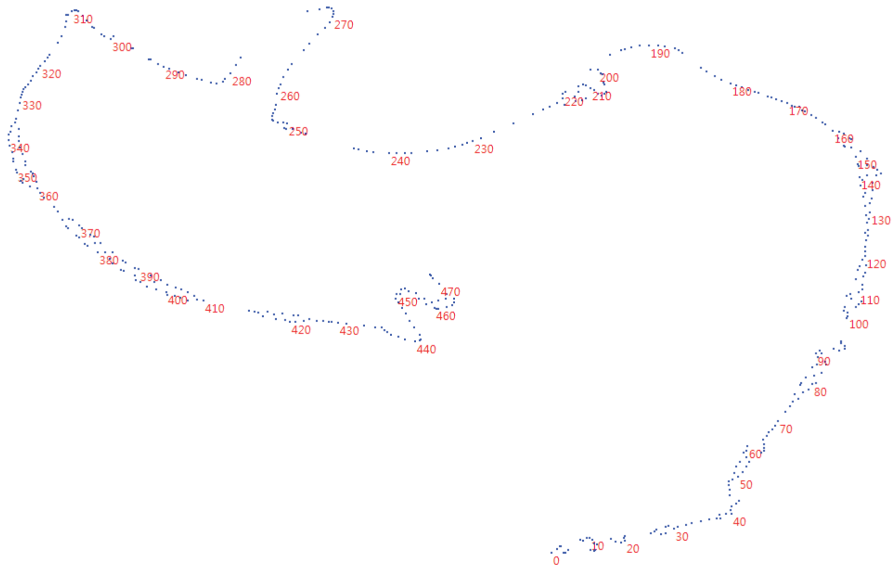

Figure 4: Feature vectors projected in 2D scattered points

should consider the characteristics of VEDR video data. First, video data is the sum of discrete frames. This means that VEDR video data has a nonlinear structure, but second, it still has a continuous context. All frames before and after contain similar image information. Lastly, it can cause a crowd problem. When a vehicle is stationary, the recorded image will be almost same during that period, making it hard to find differences between frames. With these considerations, t-SNE is the most suitable method to reduce the dimensionality of VEDR feature vectors [13]. t-SNE is a dimension reduction technique for nonlinear data that preserves the local properties of the data manifold in the low-dimensional representation and reflects the continuous context of frames. It is also robust in the crowd problem by using a student t-distribution.

High-dimensional feature vectors are reduced to two dimensions and visualized with a scatter plot graph in Fig. 4. Each point corresponds to each frame of the VEDR video and shows that similar frames are placed closed together. We can recognize intuitively the order of video frames and categorize an individual video with the curve-linear shape of a point cloud.

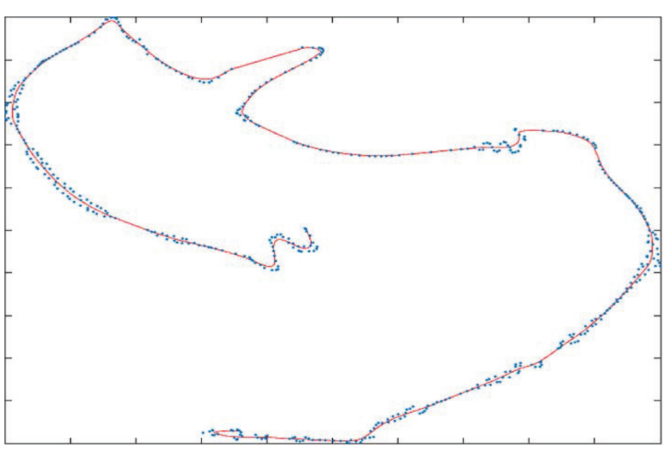

Figure 5: Curve-fitting result of feature vectors

C. *Reconstruction of Video Sequences*

The feature vectors in the low-dimensional domain maintain the gradually changing shape that they had in the higher dimension owing to the continuous context of the VEDR video mentioned above. As a result, the visualized result in the two-dimensional graph has a curved-shaped distribution. This means that the feature vectors of the frame are gradually changing, that is, the flow of the video. Therefore, finding the representative model fitted to this distribution will give a rough indication of the proper sequence of the video.

The Principal curve [2] is one of the nonlinear curve fitting methods that satisfy these conditions. This method is defined as "self-consistent" smooth curves that pass through the middle of a data cloud. In Fig. 5, we can fit the standard model of the video sequence using this method. With a given curve-fitted model, in order to reconstruct the sequence, we move from the beginning to the end of the curve and reconstruct the order by calculating the nearest frame.

D. *Detecting of Driving Events*

By using the previous curve-fitted model, we can detect certain events in the video. Particular events such as car crash or sharp turn can occur during driving. These events can cause sudden changes in the image data of frames, and these are reflected in the relation between feature vectors.

By using the curve-fitted model, we can compute the direction of each frame vector and the rate of change. With this result, performing second derivative on the curve can determine where sudden changes occurred and that suspicious event point can be noted. Whether the change is rapid or not is

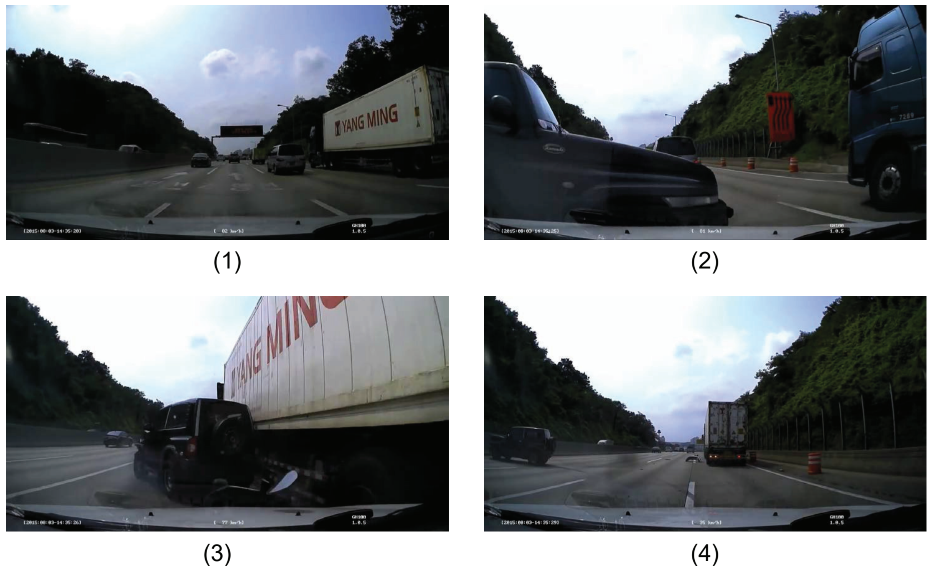

Figure 6: Sample frames of a video file: (1) Driving state, (2-3) Accident state, (4) Stop state

judged through adoptive threshold calculation. Since the state of the recorded video varies depending on the characteristics of the road or the flow of surrounding vehicles, an adoptive threshold value was used.

IV. Experimental Results

There are no large-scale datasets for vehicle driving videos recorded with VEDR devices. Therefore, we manually collected eight datasets containing normal driving scenes captured using the most popular VEDR devices . The other two datasets were evidence of traffic accident cases from the National Forensic Service. Fig. 6 shows sample frames of these datasets.

To verify the results of our method, each video file in the dataset was manually divided into frames, and each frame was labeled in the proper sequence to compare the reconstructed sequence with the original video. Video scenes were also manually classified to three types: “Driving state”, “stop state”, “accident state”. For each video scene, video files were fragmented into 5 pieces in any size, and 50 percent of each video file was overwritten.

State	Matching Ratio	Mean Mislocated Distance	Standard Derivation
Overall	0.89	0.26	0.61
Driving	0.92	0.34	0.65
Stop	0.71	0.42	0.82
Accident	0.94	0.1	0.3

Table I: Match ratio and mean mislocated distance of the reconstruction result

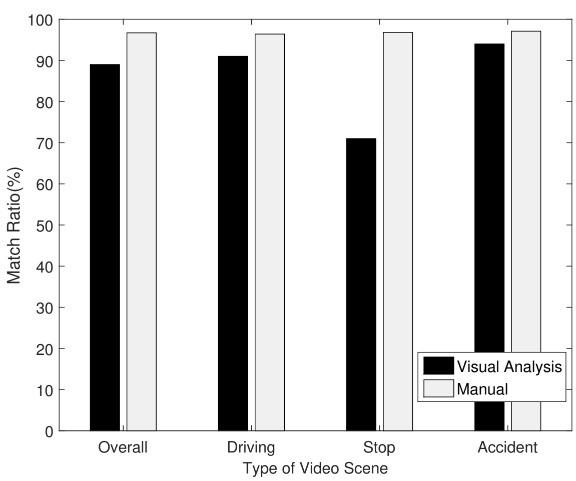

Figure 7: Performance of the frame sequence reconstruction technique

Table I shows the results of the frame sequence reconstruction. The mean distance shows how much the reconstructed frame sequence differed from the original. The closer a value is to zero, the greater the similarity between the order of the reconstructed and original results. In the three types of scenes, the reconstructed results were better in the “normal driving state” and “accident state.” This is because when the vehicle is stationary, the extracted image features have similar values because of the low scene transitions.

Fig. 7 shows the match ratio according to types of a video scene. Manual technique connects frames using the video file meta-information including the size information of each frame called Sample-to-Size (STSZ) box. Thus, In case of the manual technique, the type of the video scene did not affect the match ratio. The match ratio was up to 90 percent for the driving state and accident state in both case. However, In case of the stop state, the proposed method had lower performance than manual technique. This is because of similar visual elements in stop state scenes. Although the overall performance of the proposed method is slightly lower than the manual technique, there

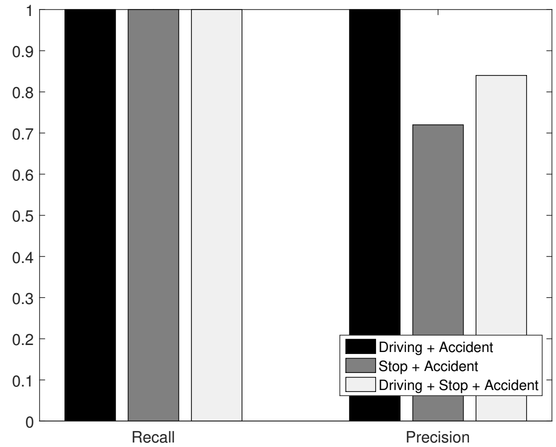

Figure 8: Precision-Recall graph of the event detection result

are two advantages to using proposed method: 1) the manual technique using the meta-information cannot works when the portion of video file meta-information was overwritten. 2) It is relying on the type of the video codec to use meta-information for connecting frames. In the case of the visual analysis, All video files encoded any video codecs always contain visual elements in each video frame, we can use the proposed method without knowing of the video file meta-information.

Fig. 8 shows the results of the "incident event" detection. The second derivative results of the curved-fitted model finds all of the points at which rapid changes occurred, but also included changes due to sudden changes in the background or illumination, regardless of the accident. However, false negative is 0, which detects all incident events in the video file.

V. CONCLUSION

In this paper, we introduced a method for analyzing VEDR video data using visualization. Existing methods have only restored video fragments frame by frame using lost meta-information. Unlike those methods, the proposed technique analyzes video fragments and reconstructs them into one meaningful video file. It also automatically detects significant events, such as accident scenes, in the video file.

ACKNOWLEDGMENT

This research was supported by the Public Welfare & Safety Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future planning (2012M3A2A1051118).

REFERENCES

[1] Simson L Garfinkel. "Carving contiguous and fragmented files with fast object validation". In: *digital investigation* 4 (2007), pp. 2–12.

[2] Trevor Hastie and Werner Stuetzle. "Principal curves". In: *Journal of the American Statistical Association* 84.406 (1989), pp. 502–516.

[3] Geoffrey Hinton and Sam Roweis. "Stochastic neighbor embedding". In: *NIPS*. Vol. 15. 2002, pp. 833–840.

[4] Larry Huston et al. "Forensic video reconstruction". In: *Proceedings of the ACM 2nd international workshop on Video surveillance & sensor networks*. ACM. 2004, pp. 20–28.

[5] Thomas Laurenson. "Performance analysis of file carving tools". In: *IFIP International Information Security Conference*. Springer. 2013, pp. 419–433.

[6] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE". In: *Journal of Machine Learning Research* 9.Nov (2008), pp. 2579–2605.

[7] Gi-Hyun Na et al. "Frame-based recovery of corrupted video files using video codec specifications". In: *IEEE Transactions on Image Processing* 23.2 (2014), pp. 517–526.

[8] Anandabrata Pal and Nasir Memon. "The evolution of file carving". In: *IEEE Signal Processing Magazine* 26.2 (2009), pp. 59–71.

[9] Rainer Poisel and Simon Tjoa. "Forensics investigations of multimedia data: A review of the state-of-the-art". In: *IT Security Incident Management and IT Forensics (IMF), 2011 Sixth International Conference on*. IEEE. 2011, pp. 48–61.

[10] Golden G Richard III and Vassil Roussev. "Scalpel: A Frugal, High Performance File Carver." In: *DFRWS*. 2005.

[11] Edward Rosten and Tom Drummond. "Machine learning for high-speed corner detection". In: *Computer vision–ECCV 2006* (2006), pp. 430–443.

[12] Szabolcs Sergyan. "Color histogram features based image classification in content-based image retrieval systems". In: *Applied Machine Intelligence and Informatics, 2008. SAMI 2008. 6th International Symposium on*. IEEE. 2008, pp. 221–224.

[13] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. "Dimensionality reduction: a comparative". In: *J Mach Learn Res* 10 (2009), pp. 66–71.

[14] Zhirong Yang et al. "Heavy-tailed symmetric stochastic neighbor embedding". In: *Advances in neural information processing systems*. 2009, pp. 2169–2177.