Semantic Video Carving Using Perceptual Hashing and Optical Flow

Sijin Li, Guikai Xi, Zoe Jiang, Siu-Ming Yiu, Liyang Yu, Xuan Wang, Qi Han, Qiong Li

Chapter 13

SEMANTIC VIDEO CARVING USING PERCEPTUAL HASHING AND OPTICAL FLOW

Junbin Fang, Sijin Li, Guikai Xi, Zoe Jiang, Siu-Ming Yiu, Liyang Yu, Xuan Wang, Qi Han and Qiong Li

Abstract Video files are frequently encountered in digital forensic investigations. However, these files are usually fragmented and are not stored consecutively on physical media. Suspects may logically delete the files and also erase filesystem information. Unlike image carving, limited research has focused on video carving. Current approaches depend on filesystem information or attempt to match every pair of fragments, which is impractical. This chapter proposes a two-stage approach to tackle the problem. The first perceptual grouping stage computes a hash value for each fragment; the Hamming distance between hashes is used to quickly group fragments from the same file. The second precise stitching stage uses optical flow to identify the correct order of fragments in each group. Experiments with the BOSS dataset reveal that the approach is very fast and does not sacrifice accuracy or overall precision.

Keywords: Digital forensics, video carving, perceptual hashing, optical flow

1. Introduction

The amount of video encountered in digital forensic investigations has increased significantly over the past decade. The digital evidence includes surveillance camera and mobile device video files, forged video files and erotic video files [15, 16]. However, video files are usually broken into segments due to large file sizes and filesystem storage mechanisms such as file scattering and wear leveling [10]. Additionally, criminals may attempt to erase the files that may have recorded their actions. Indeed, it is common for digital forensic investigators to only obtain (deleted) raw video fragments extracted from storage media. In such instances,

video carving is needed to reassemble the fragments to create the original video files for further investigation, especially when filesystem information related to file organization is lost [6].

In principle, video carving should only consider the content of video fragments instead of the filesystem structure or other metadata [13]. However, most research assumes that video fragments are stored sequentially or some type of metadata is available to help reorder file fragments [3, 9, 11, 18, 19]. Without these assumptions, the only option is to apply an exhaustive matching method, which compares the content of every pair of fragments and concatenates one fragment to another when the two fragments have the highest adjacency likelihood. This procedure is analogous to assembling a jigsaw puzzle using brute force.

The computational effort for the brute force content-based video carving grows quadratically with the total number of fragments. Specifically, for n fragments, the algorithm requires $O(n^2)$ steps for reassembly [11]. Garfinkel [6] notes that many video files are typically recovered from storage devices during a digital investigation and these files are often very large, resulting in a massive number of fragments and, thus, significant computational costs. Therefore, an automated semantic video carving approach with high efficiency and precision is sorely needed to support digital forensic investigations.

Content-based video carving is complicated because fragmentation shuffles the constituent parts of a video file; additionally, the fragments from multiple video files are mixed together. The approach described in this chapter is designed to semantically carve video fragments from multiple video files, especially in the case of surveillance videos, which are commonly encountered in digital forensic investigations. The novel approach involves two stages that reduce the computational complexity while maintaining high precision. Instead of performing pairwise matching of all the fragments, the proposed approach employs perceptual grouping to collect fragments that originate from the same video file. This step is followed by content-based precise stitching that restores the video file by assembling the out-of-order fragments from a group corresponding to a single video file.

The proposed approach first calculates the perceptual hash (P-hash) value [12] (i.e., compressed digest) of each video fragment, following which the Hamming distances between pairs of hashes are computed. Two fragments whose Hamming distance is within a threshold are clustered into the same group and are deemed to originate from the same video file. The second stage precisely evaluates the adjacency likelihoods of the raw content of fragments in each group using optical flow; this

enables the fragments to be reordered correctly based on their motion feature.

The overall computational complexity of semantic video carving is reduced significantly because the scale of the precise stitching computations is decreased by the perceptual grouping stage. For example, if all n fragments from m different video files are mixed together, the proposed approach requires $O(mn)$ grouping computations plus $O(m(n/m)^2)$ computations to compare fragments for reassembly instead of $O(n^2)$ computations required by the brute force method. Note also that the computational cost for perceptual grouping is much less than the cost for content-based precise stitching.

Experimental results obtained for the BOSS dataset [2] reveal that increasing the number of video files captured by the same camera from one file to ten files yields a final precision rate greater than 96%. Moreover, increasing the number of cameras from one to nine, all of them recording the same scenario, yields a final precision rate greater than 98%. The execution times range from two seconds to 15 minutes (for 10 to 100 fragments), demonstrating that the proposed approach is practical.

2. Related Work

File carving approaches can be classified as: (i) file-signature-based carving [17]; (ii) mapping function carving [5]; and (iii) graph theoretic-carving [10]. Graph-theoretic carving, which is often referred to as semantic carving, exhibits better performance than the other two approaches, especially for text carving and image carving [13].

However, while graph-theoretic carving approaches have constantly improved, they are not as effective on video images; this is because relatively little research has focused on semantic video carving. Most research has leveraged file signatures, file headers of video formats, codec specifications, etc. Additionally, the direct application of graph-theoretic carving to video fragments has high computational complexity. Table 1 lists the principal video carving methods described in the literature.

Poisel et al. [14, 16] have proposed file carving approaches for carving fragmented multimedia files. The approaches involve preprocessing, collating and reassembly. However, their work only focuses on image fragments.

Yoo et al. [19] have developed a file carving approach for multimedia AVI, WAV and MP3 files compressed by NTFS. The main contribution is a recovery method for deleted NTFS compressed files. The approach assumes that multimedia files are continuously allocated and that the files can be carved based on file header signatures.

Table 1. Comparison of video carving methods.

Method	Auxiliary Information Used
AVI Carver [19]	NTFS compressed signature
Lewis Method [9]	Cluster boundaries in storage media
Robust Video Carver [18]	Frame and sequence headers
NFI Defraser [3]	MPEG structure and semantic checks
DC3carver [3]	File format characteristics
Frame-Based Recovery [11]	Codec specifications and STSZ box information

Lewis [9] has proposed an improved video fragment reassembly method that leverages the cluster boundaries in storage media. Because files are generally saved on storage media by cluster, all the data in a single cluster belongs to a single file, except for the last cluster of a file, which may also contain data from other files in its slack or uninitialized space. The method relies on cluster configuration information. However, it is challenging to reliably detect clusters that contain video file data. Another challenge is to connect clusters that belong to the same fragmented file.

Yannikos et al. [18] have proposed the combination of two forensic techniques – video file carving and robust hashing – to automate the identification and recovery of video content. Their video frame carving approach analyzes frame information in order to extract and decode single intracoded frames (I-frames); this results in more robust recovery than traditional header/footer identification. However, the method carves video slices by searching for I-frame headers backwards and forwards, assuming that all the video fragments are allocated in sequence.

Casey and Zoun [3] have compared the Defraser and DC3Carver carving tools. They also discuss the trade-offs of using carving tools in digital forensic examinations.

Na et al. [11] have proposed a frame-based video carving approach that leverages codec specifications for surveillance video. Their approach restores corrupted video files at the frame level. However, it essentially performs extended signature-based file restoration because it relies on sample-to-size (STSZ) box data in MPEG-4 files, which records the length of each frame set. Without STSZ information, the approach has to match frames one by one, resulting in a significant time complexity of $O(n^2)$.

3. Proposed Video Carving Approach

A successful reassembly of video fragments implies that all the fragments are placed in the same sequence as in the original video file. Fig-

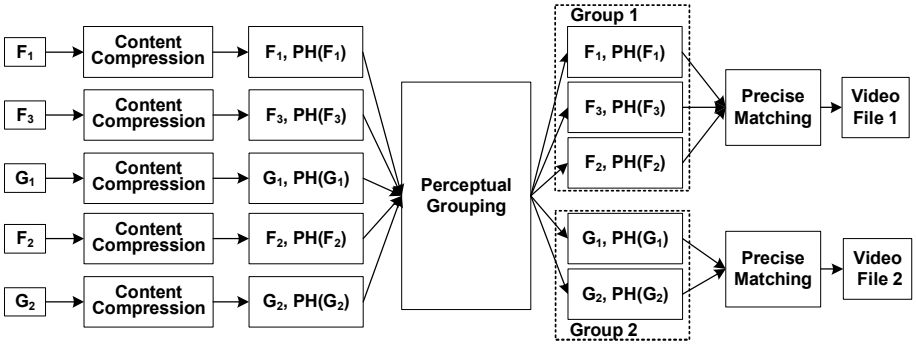


Figure 1. Proposed video carving approach.

ure 1 illustrates the proposed video carving approach using a simple example. Let F_1, F_2, F_3, G_1 and G_2 be video fragments that must be carved. Assume that F_1, F_2 and F_3 originate from one video file while G_1 and G_2 originate from a second file. Note that, in general, data fragments are extracted from physical storage using a digital forensic tool and the video fragments are identified via data type classification.

In the proposed video carving approach, each fragment is pre-processed via content compression to produce a P-hash value. Next, the video fragments (e.g., F_1) and their P-hash values (e.g., $PH(F_1)$) are input to the perceptual grouping stage. This stage clusters the fragments based on the Hamming distances between their P-hash values. For example, the grouping stage clusters the fragments F_1, F_3 and F_2 into Group 1, and the fragments G_1 and G_2 into Group 2.

Next, each group (now with a substantially smaller number of fragments) is input to the precise stitching stage to calculate the adjacency likelihoods of fragments in the group. Note that no effective measure exists for this step. In the proposed approach, optical flow is used to estimate the similarity of the frames around the fragmentation points of video fragments in same group. Based on the motion vectors computed for the image frames, an improved graph-theoretic carving algorithm is used to position the fragments in a group correctly to reconstruct the original video file. For example, fragments F_1, F_3 and F_2 are reordered in the correct sequence (F_1, F_2, F_3) and are subsequently concatenated to produce Video File 1. Likewise, fragments G_1 and G_2 are reordered as (G_1, G_2) and concatenated to produce Video File 2.

3.1 Perceptual Grouping

Since the video carving input is a large number of video fragments from different video files, exhaustively matching the fragments is an ex-

tremely time-consuming task. Instead of conducting precise comparisons of all the video fragments directly, a coarse grouping algorithm is employed to cluster the fragments originating from the same file in a single group, without considering the order of the fragments.

Generally, video fragments from the same video file source have more common features or scenes than those from different video files. Specifically, video fragments from the same file are more similar semantically than those from other files. Utilizing this characteristic, the grouping problem can be transformed to a clustering problem, where the distance between objects represents the dissimilarity of video fragment content and fragments originating from the same file tend to gather around a cluster center. The nearer the objects, the more similar the fragments and the greater the grouping likelihood. Centroid-based clustering is used to divide the video fragments into groups. Note that the cluster center can be initialized as the first fragment of a video file, which is easily identified because it usually contains a number of specific codes. For example, the two popular codecs, MPEG-4 [7] and H.264 [8], have the header codes `0x000001` and `0x00000001` or `0x000001`, respectively, which help identify the header fragment.

Three techniques are employed to implement this approach efficiently. The techniques are described in the following paragraphs.

Perceptual Evaluation. The first technique helps choose an appropriate measure to evaluate the similarities or dissimilarities of video fragments with low computational complexity and a high recall ratio. Figure 2 demonstrates this grouping technique. Start Fragment 0 indicates the first fragment of a video file, which is obtained by simply searching for the unique magic number of a file in the storage media [11]; this fragment is marked as the initial cluster center. Next, the image frames around the fragmentation points of Start Fragment 0 and Candidate Fragment are compressed into binary descriptors (i.e., P-hash values corresponding to the white and black square patterns in the figure) through perceptual hashing as described below. The clustering distance is measured as the Hamming distance between two P-hash values. In the example, because the Hamming distance between Start Fragment 0 and Candidate Fragment is lower than the threshold, Candidate Fragment is assigned to Group 0, which contains the best available fragments that originate from the same video file as Start Fragment 0.

P-hashing is used to compress the contents of all the fragments before running the clustering algorithm. A number of hashing functions have been proposed based on histogram, discrete cosine transform (DCT), singular value decomposition (SVD), local color features and random

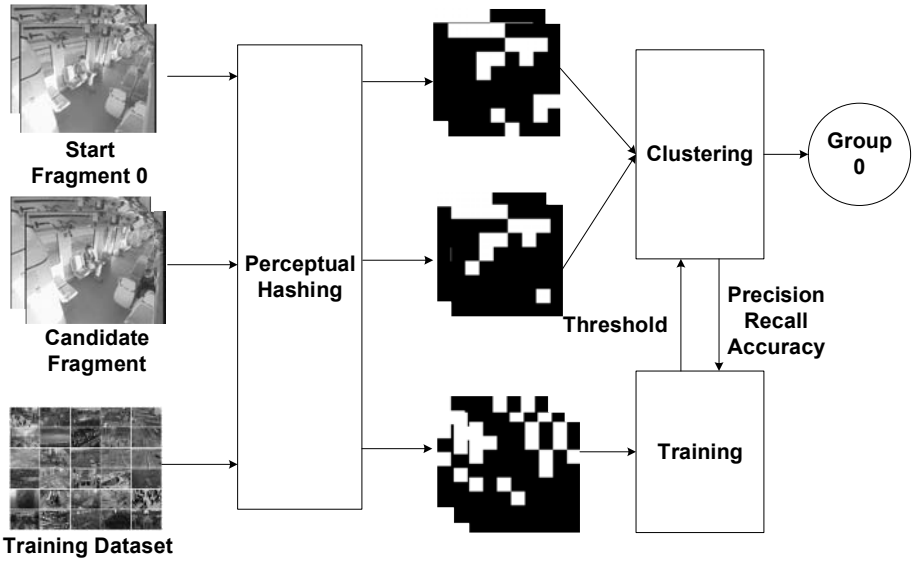


Figure 2. Perceptual grouping stage.

transformation methods [12]. The proposed approach generates P-hash values using a discrete cosine transform hashing function, which involves four steps:

- **Step 1 (Grayscale Transformation):** A color frame contains redundant information for processing. The grayscale transformation reduces the information content and, thus, the subsequent computational effort.

- **Step 2 (Resizing):** Resizing reduces the computational cost of discrete cosine transform hashing. In this case, video frames are resized to a fixed resolution of 32×32 pixels via linear interpolation.

- **Step 3 (Discrete Cosine Transformation):** This transformation converts video content from the spatial domain to the frequency domain, causing the primary information of a frame to converge to the low-frequency area, which means that the magnitudes of the low-frequency discrete cosine transform coefficients are robust to slight/invisible changes to video frames. Experimentation revealed that the first 64 coefficients are suitable for computing hash values that enable robust comparisons.

- **Step 4 (Hashing):** Binary hash values are computed as:

$$H(i) = \begin{cases} 1, & c[i] > c_{threshold} \\ 0, & otherwise \end{cases} \quad (1)$$

  where $c[i]$ is the i^{th} discrete cosine transform coefficient ($i = 1..64$) and $c_{threshold}$ is a threshold value for the discrete cosine transform coefficients; it is typically the average of the 64 discrete cosine transform coefficient values.

This processing reduces the computational cost of comparing two 1024×768-pixel video frames to the cost of comparing two 64-bit binary values, a dramatic reduction in the computational complexity of the grouping stage.

Cluster Optimization. The second technique involves optimal and self-adaptive clustering. Clustering is formulated as a multi-objective optimization problem. A training module shown in Figure 2 is employed to optimize the output groups. Since the Hamming distance between a pair of P-hash values is chosen as the weighting parameter in cluster analysis, adjusting the cluster radius affects the clustering performance when the inter-member distances are small.

An optimized threshold for the clustering radius is significant to the performance of the algorithm. If the threshold value is too small (i.e., too strict), then some candidate fragments may be excluded (false reject or false negative errors). On the other hand, a large threshold may lead to the addition of outliers (false accept or false positive errors). Therefore, before the clustering process is initiated, a training dataset must be input to the clustering algorithm to determine the optimal threshold by adjusting the clustering radius until optimal groups are produced that maximize the true positives and minimize the false positives in each group. To accomplish this, clustering performance metrics such as precision, recall ratio and accuracy are fed back to the training module in order to self-adjust the clustering radius threshold.

Note that if the fragments are restricted to being in exactly one group, then when a fragment is assigned incorrectly to a group, the precise stitching stage performance deteriorates because the correct group and the incorrect group both have the wrong output. Therefore, fragments are permitted to belong to multiple groups. This reduces the number of false negatives while increasing the number of false positives. Thus, more computations are performed during the precise stitching stage than in theory, but they are still much less than those required by the exhaustive matching algorithm with an optimal threshold setting.

Fissile Clustering. The third technique seeks to improve the traditional clustering algorithm to fit the characteristics of the video fragments. One problem with directly performing traditional clustering is that after the starting fragment is set as a fixed cluster center, when the video file is highly fragmented, the clustering radius – Hamming distance in this case – should be large enough to include all the video fragments (e.g., fragments around the end of the file). This occurs because of the inherent "chain-like" property of file carving. However, the larger the clustering radius threshold, the greater the number of incorrect fragments from other files included in each group. To address this problem, a "fissile clustering" algorithm is employed that compares the front-end frames of candidate fragments and the back-end frames of current fragments in similarity evaluations.

The fissile clustering algorithm involves the following steps:

- **Step 1:** For each group, begin with the start fragment and set it as the current fragment.
- **Step 2:** Calculate the Hamming distance between P-hash values of the back-end frame(s) of the current fragment(s) and the front-end frames of all the remaining fragments.
- **Step 3:** Compare the similarity likelihoods based on the Hamming distances. If the distance is below the threshold, then the candidate fragments are collected into a group and become the current fragment(s).
- **Step 4:** Select all the clustered fragments one by one, and repeat Steps 1 through 3 until there are no more available fragments.

3.2 Precise Stitching

After all the video fragments have been clustered into smaller groups, the second stage of the proposed approach evaluates the adjacency likelihoods or similarities of the fragments in each group and attempts to stitch them together in the correct order. For each group, a graph-theoretic carving algorithm can be applied with an appropriate weight function. The idea is to find the shortest path for the Hamilton path problem (i.e., optimal order of the fragments in a group). Compared with the original graph-theoretic carving method [10, 13], the scale of the proposed algorithm is reduced to the number of candidates in a group instead of the total number of fragments. The smaller scale also reduces the numbers of false positives in the groups, excluding outliers from the final restored video file.

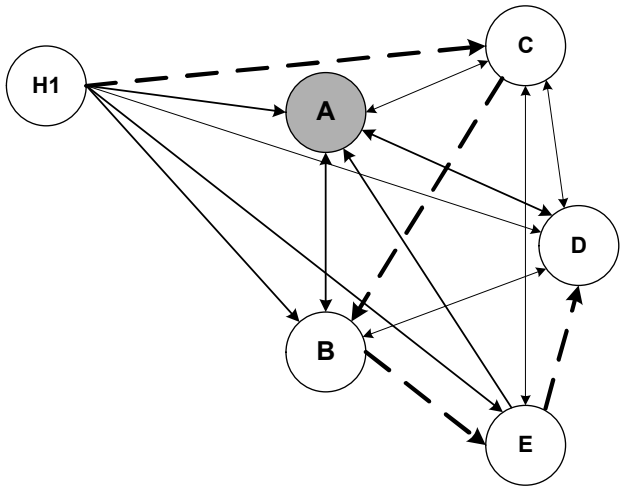


Figure 3. Graph of six video fragments with a disjoint path and rejected outlier A

Consider two possible situations in which the algorithm mistakenly concatenates an outlier in the restored video file. First, if the content of an outlier fragment has a higher adjacency likelihood than those of the other candidate fragments, it may be mistakenly chosen as the best adjacent fragment. Since this erroneous concatenation occurs due to the high similarity between the outlier fragment and preceding fragment, it cannot be distinguished by a graph-theoretic algorithm, including the proposed precise stitching stage and the original graph-theoretic carving method. In this case, a double-check procedure may be needed.

On the other hand, even if the outlier fragments have lower adjacency likelihoods than other candidate fragments, the algorithm may still keep appending them to the end of the video file if the exhaustive method is not terminated based on some other constraint. Therefore, an additional constraint is required to identify the ending vertex of the optimal path and terminate the algorithm.

Figure 3 shows a simple example. A group with six fragments is represented as a complete graph of six vertices, where each edge has a weight corresponding to the adjacency likelihood between the fragments. The header fragment in the group is the vertex H1 and assume that fragment (vertex) A is an outlier that is mistakenly clustered during the previous stage. The problem of reconstructing the original video file is equivalent to finding the optimal path (i.e., shortest path in the graph). Thus, the algorithm should attempt the best adjacent fragment for each fragment and also try to avoid passing through the outlier vertex A.

Otherwise, the restored video file would contain incorrect content. In the example, the complete disjoint path is H1-C-B-E-D while the outlier vertex A is rejected.

Since a video is a series of frames (images) in time sequence and the objects in the video have spatial consistency between frames, the motion field between two frames can be leveraged as a similarity measure by the precise stitching algorithm. In the proposed method, the optical flow is selected for the relative motion analysis of frames at the fragmentation point of two candidate video fragments.

Suppose that two pixels from the frames of a preceding fragment F_P and a candidate fragment F_C have same pixel value, although the pixel positions may be different. In other words, $p_{preceding}(x, y) = p_{candidate}(x + \Delta x, y + \Delta y)$. Then, the motion distance is calculated as:

$$D(F_P, F_C) = \frac{\sum_{i=0}^{n_{pixels}} \sqrt{\Delta x^2(i) + \Delta y^2(i)}}{n_{pixels}} \tag{2}$$

where n_{pixels} is the number of pixels in each frame, $p_{preceding}(x, y)$ is the pixel value of the point $(x,\ y)$ in a preceding frame, $p_{candidate}(x + \Delta x, y + \Delta y)$ is the pixel value of point $(x + \Delta x, y + \Delta y)$ in the adjacent candidate frame and $D(F_P, F_C)$ is the average distance between the preceding frame F_P and the candidate frame F_C; this is used to evaluate the adjacency likelihoods of the available candidate fragments to the preceding fragment.

Stitching processing is the straightforward application of a greedy approximation algorithm that is commonly used to solve edge- and vertex-disjoint problems [10]. To start with, the header fragment is chosen as the current fragment and its adjacency likelihoods with the remaining fragments in the current cluster group are computed. The best available fragment is stitched to the current fragment and this best available fragment is set as the current fragment. This process is repeated until the entire video is reassembled.

4. Experimental Results

The performance of the proposed video carving approach is evaluated using the BOSS public surveillance dataset [2].

The BOSS dataset has fifteen scenarios: two no-incident scenarios, three specific-incident detection scenarios and ten incident scenarios, such as "cell phone theft," "disease," "harassment" and "panic." Each scenario was concurrently recorded by nine surveillance cameras installed in a single train car (from nine different angles). Therefore, each scenario should have nine video clips. However, five scenarios do not have nine

(a) Cell phone theft.

(b) Disease.

(c) Harassment.

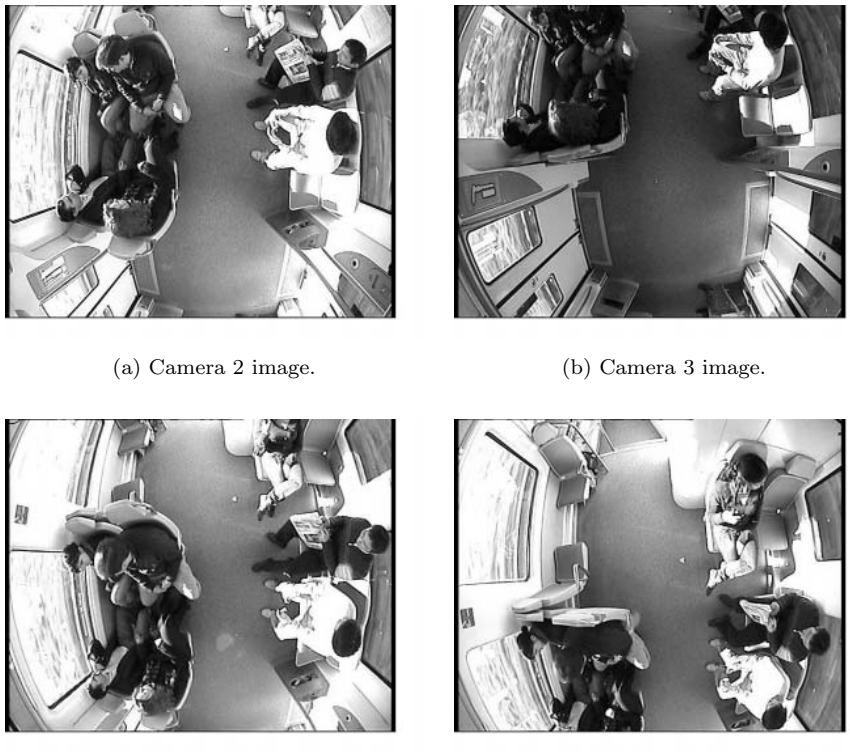
(d) Panic.

Figure 4. Four video scenarios recorded by Camera 1.

video clips; therefore, the other ten scenarios for which nine video clips exist were used in the experiments. Figure 4 shows images from video clips taken by Camera 1 for four scenarios. The parameters of the video clips in BOSS dataset are:

- **Frame Rate:** 25 fps interlaced.
- **Resolution:** 720×576 pixels.
- **Video Container:** AVI.
- **Codec:** MJPEG 4:2:2 (Cameras 1 through 9).
- **Bit Rate:** 30 Mbps.

To evaluate the proposed approach, all the video files were randomly sliced into video fragments, which were then mixed. The set of mixed video fragments was used as the experimental input. Since the proposed



(a) Camera 2 image.

(b) Camera 3 image.

(c) Camera 5 image.

(d) Camera 8 image.

Figure 5. A single video scenario recorded by different cameras.

video carving approach focuses on the efficiency of reassembling video fragments, all the fragments were present in the input.

The following three metrics were used to evaluate the performance of the video carving approach:

- **Recall:** $TP/(TP + FN)$.
- **Precision:** $TP/(TP + FP)$.
- **Accuracy:** $(TP+TN)/(TP+TN+FP+FN)$ where TP, TN, FP and FN are the numbers of true positives (inliers), true negatives, false positives (outliers) and false negatives, respectively, and $n = (TP + TN + FP + FN)$ is the total number of video fragments.

Note that the dataset is quite challenging for the video carving algorithm because some of original video files recorded for the same scenario are extremely similar. For example, as shown in Figure 5, the same scenario recorded by Cameras 2, 3, 5 and 8 generates four different video

frames, but the images are very similar, especially the images from Cameras 2 and 5. The great similarities of the video files increase the difficulty of video carving when the files are fragmented and the fragments are mixed together.

A series of experiments under different conditions were conducted to optimize the clustering threshold of the first stage and to investigate the performance of the proposed approach. The computing platform used in the experiments was a desktop computer with an Intel I5-3317U 2.60 GHz CPU and 6 GB memory.

Optimizing the Clustering Threshold. As discussed above, the clustering threshold has a significant impact on the performance of the perceptual grouping stage and the overall video carving approach. If the threshold is set too large, each group could include several outliers, increasing the recall ratio of the grouping while decreasing its precision and accuracy. In contrast, a small threshold could reject some inliers from each group, increasing the precision and accuracy, but decreasing the recall ratio. Therefore, the threshold should be selected carefully to optimize the overall performance.

The clustering threshold was optimized by training. Two public datasets were used for this purpose, the CAVIAR surveillance dataset [4] and the crowd segmentation dataset provided by the Center for Research in Computer Vision at the University of Central Florida [1].

Forty video clips from the two training datasets were randomly sliced into fragments to create training samples that were input to the perceptual grouping algorithm. Each video clip was divided into two to 30 fragments randomly. Figure 6 shows the relationships between recall, precision and accuracy versus the clustering threshold for the training datasets. When the threshold is larger than 16, the recall ratio of grouping reaches 100% while the accuracy drops to about 30% and precision is only 10%, meaning that the number of false positives is about nine times the number of true positives. At the other extreme, when the threshold is set to below 2, the recall ratio of grouping is less than 50%, meaning that about half the fragments are clustered in the correct group.

As mentioned above, video fragments are allowed to belong to multiple groups; this relaxes the restrictions on the precision and accuracy of a grouping. However, if the grouping precision is too low, the increase in the number of false positives contributes to increased computations in the subsequent precise stitching stage. According to the curves in Figure 6, the optimal threshold should be in the range 7 to 11 because the recall ratio and accuracy have high values and are flat within this range. Moreover, the recall ratio of grouping reaches 98% when the

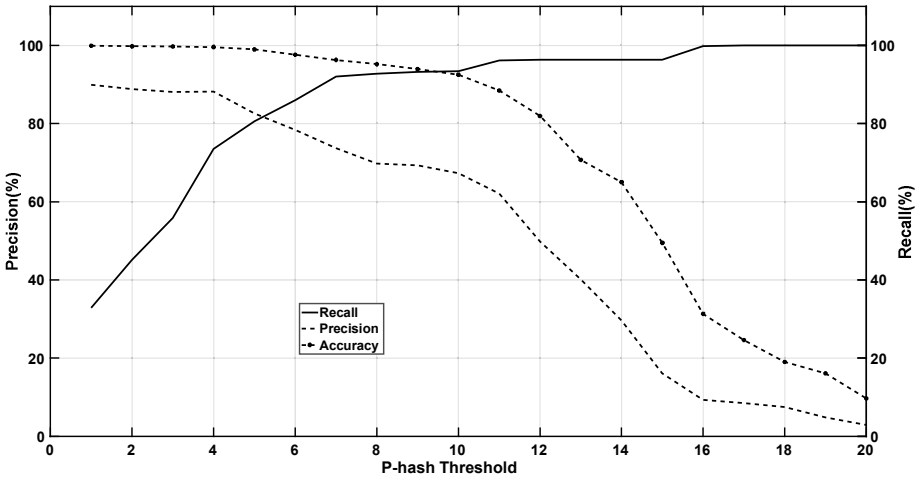


Figure 6. Performance metrics versus clustering threshold for the training datasets.

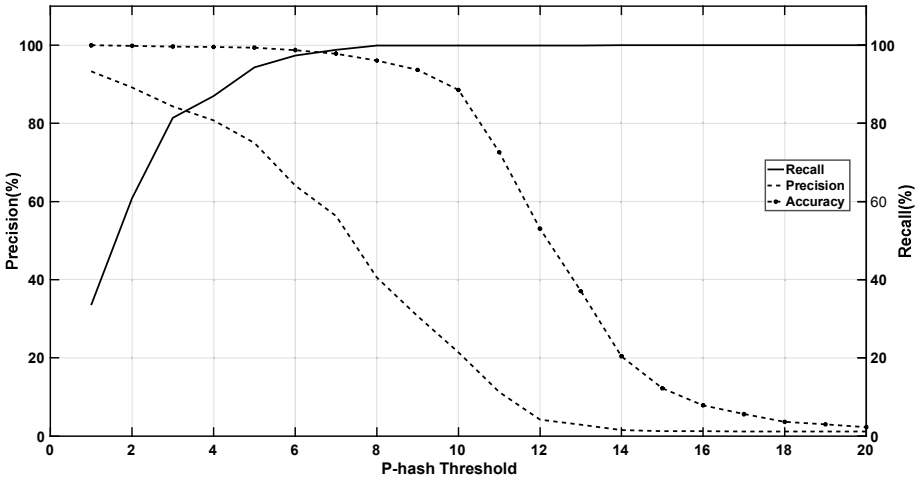


Figure 7. Performance metrics versus clustering threshold for the BOSS dataset.

threshold is 11 and only increases slightly after this value. Therefore, since all the datasets involve surveillance videos, a Hamming distance of 10 was chosen as the clustering threshold, which turns out to be adequate for the BOSS dataset to yield a 100% recall ratio. This is confirmed in the experimental results obtained with the BOSS dataset (Figure 7).

Carving Fragments from the Same Camera. An experiment investigated the performance of the proposed approach for video fragments originating from the same camera. Such a situation is commonly en-

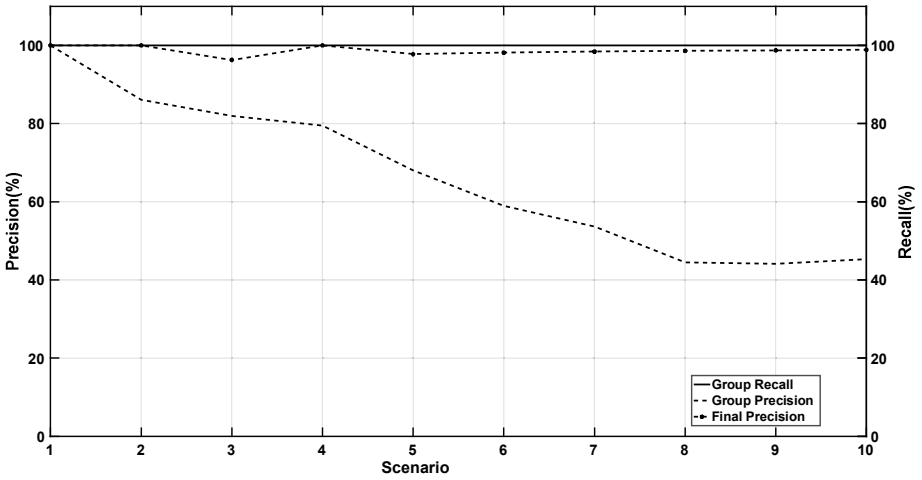


Figure 8. Video file carving performance for the same camera.

countered in digital forensic investigations. For example, this may occur when videos captured and stored locally by a surveillance camera are extracted as fragments due to the lack of filesystem information or overwriting by the storage mechanism. Because the videos were recorded by the same camera at different times, their backgrounds should be similar, which increases the difficulty of video carving when the fragments are mixed together.

Digital videos in the BOSS dataset that were recorded by the same surveillance camera were chosen for this experiment. The video files with different scenarios were randomly divided into four pieces, giving rise to a total of $4n_f$ fragments, where n_f, the number of video files, varied from two to ten, yielding a total number of mixed fragments ranging from eight to 40. Since the BOSS dataset has nine surveillance cameras (recording sources), the experiment was performed on nine sets of video files. Figure 8 shows the average video file carving performance for the experiment.

Since the clustering threshold was set to 10 to achieve a 100% recall ratio in the perceptual grouping stage, all the fragments could be grouped correctly with some outliers. The graph of final precision versus the number of scenarios in Figure 8 shows that more than 96% of the video fragments were correctly reassembled by the proposed approach even when the number of scenarios (i.e., video files) was increased to ten. The 4% error rate for $n_{scenarios} = 3$ is due to the fact that the video file of the No_Event scenario has almost stationary pictures, which makes

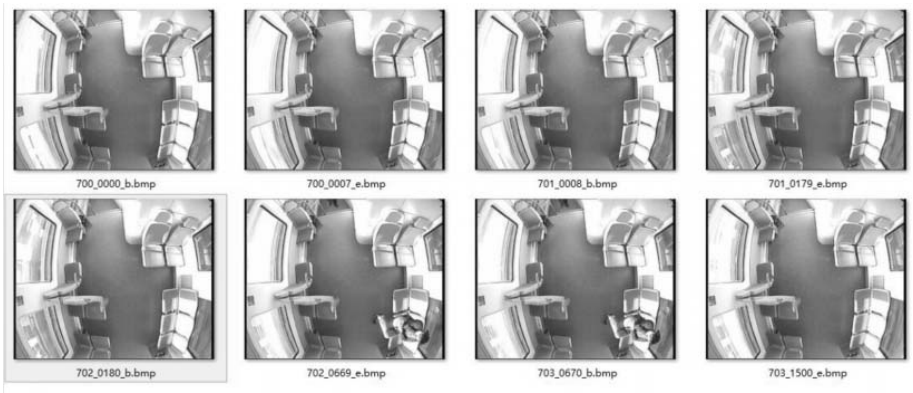


Figure 9. Frames shot in the No_Event scenario.

it very difficult to judge the order between fragments because all the frames are almost the same as shown in Figure 9.

An interesting point is that the perceptual grouping precision has less impact on the final precision. Although the grouping precision drops to 50% when the number of scenarios (video files) is increased to ten, the final precision is still greater than 96%. This is because the relaxed requirement for grouping precision is compensated for by the precise stitching stage. Of course, an increased number of outliers in the grouping stage increases the computational cost of the stitching algorithm.

Carving Fragments from Different Cameras. This experiment investigated the performance of the proposed approach when the mixed video fragments come from different cameras, although the scenario in the video files may be the same because the nine cameras monitored the same spot concurrently. This situation is frequently encountered in the real world because videos from surveillance cameras are usually uploaded to central servers or the cloud for storage, backup or analysis. When the servers are involved in a digital forensic investigation, it is common to recover a huge number of mixed video fragments.

In another experiment, digital videos of the same scenario that were recorded synchronously by different surveillance cameras were selected for analysis. In particular, video files of a scenario recorded by each of the nine cameras were chosen. Each video file was randomly divided into four pieces. Ten scenarios in the BOSS dataset were selected; therefore, ten sets, each with nine camera videos, were used in the experiment.

Figure 10 shows the average video file carving performance for the experiment. The results reveal that the grouping precision is much better than in the previous experiment; the final precision, which is higher than

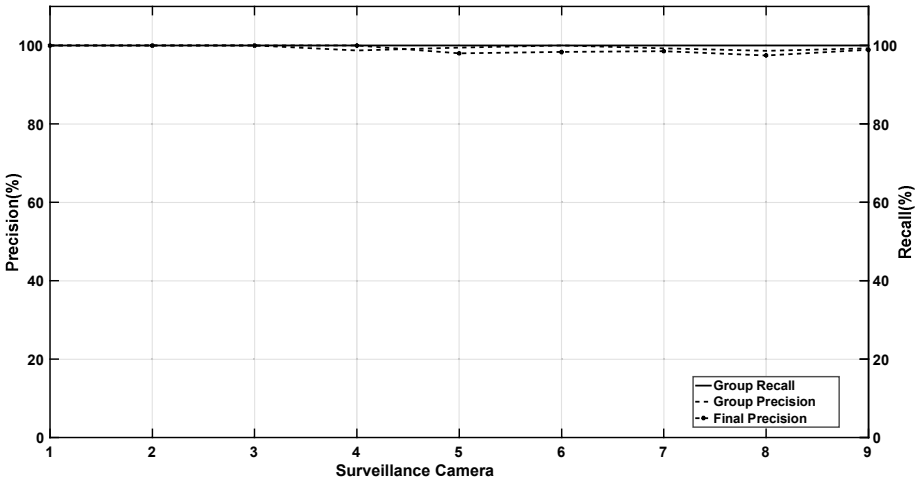


Figure 10. Video file carving performance for different cameras.

98%, is also much better. The principal reason is that the video files from different cameras have similar, but slightly different backgrounds, which can be distinguished more robustly by the video carving approach, contributing to the good results.

Carving Fragments with Various Fragmentation Degrees. This experiment investigated the impact of the fragmentation degree on the proposed video carving approach. Five video recordings of different scenarios recorded by the same camera were selected for analysis; each video file was randomly divided into two to 20 pieces. Since the BOSS dataset has nine surveillance cameras, the experiment was performed on nine sets of video files.

Figure 11 shows the average video file carving performance. When the fragmentation degree and number of video fragments increase, the performance in both stages drops, especially the final precision of the precise stitching stage. When the number of fragments in each video file is not greater than four, the grouping precision is greater than 76% while the final precision of the video carving approach, the final restored rate, is 100%. Despite the fact that the number of fragments goes up to 20 per video, the final restored rate is still as high as 67%.

Computational Time. Since computational cost is positively correlated with the number of fragments, the computational time for the two stages was measured versus the number of fragments. Table 2 shows the results. As expected, the computational time $T_{grouping}$ for the percep-

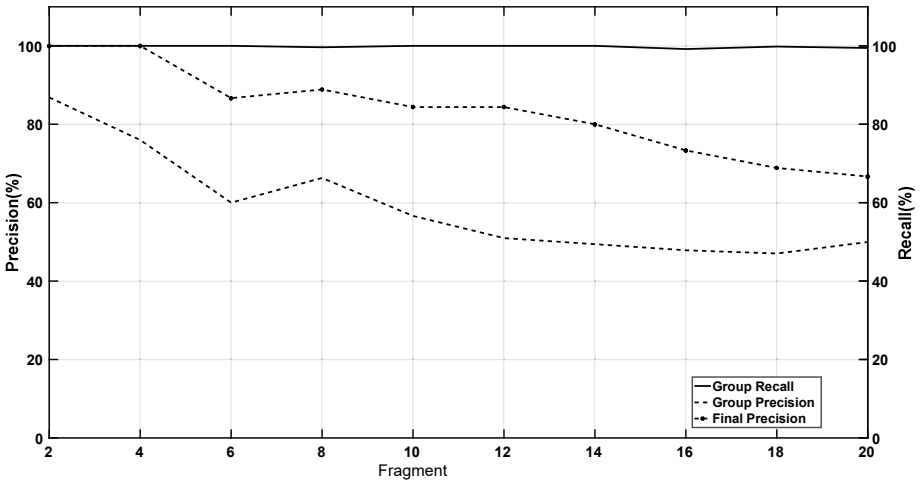


Figure 11. Video file carving performance versus fragmentation degree.

Table 2. Average time required to carve varying numbers of video fragments.

Fragments	$\mathbf{T_{grouping}}$ **(ms)**	$\mathbf{T_{stitching}}$ **(ms)**
10	2.8	2,281.0
20	16.1	18,214.5
30	40.1	45,889.6
40	79.7	109,242.1
50	127.5	195,378.6
60	197.5	329,275.1
70	245.2	402,303.9
80	304.0	562,519.6
90	451.0	863,913.9
100	584.3	946,265.4

tual grouping stage is very small while the time $T_{stitching}$ for the precise stitching stage is much larger. However, the total time for video carving is still reasonable – ranging from two seconds to 15 minutes as the number of fragments increases from 10 to 100.

5. Conclusions

The proposed semantic reassembly approach for video files with mixed video fragments yields good results in a reasonable amount of time. Experimental results show that most of the videos were correctly reassembled using the novel coarse-to-fine technique. In particular, for one to ten very similar videos originating from the same camera, the final pre-

cision was at least 96%. For the videos of the same scenario taken by one to nine cameras, the final precision was at least 98%. On the other hand, the performance drops when the number of fragments increases. When the fragments per file vary from two to 20, the final precision drops from 100% to about 67%. However, with fourteen fragments per file, the precision is still as high as 80%. Future research will attempt to address the drop in precision that occurs with increasing fragmentation, although the number of fragments seldom goes beyond 20 fragments in real-world scenarios.

The dataset selected for the experiments is challenging because the video recordings are of very similar scenarios. Since the scenarios are similar, more fragments exist in multiple clusters after the first grouping stage, which negatively impacts the effectiveness of the subsequent stitching phase. Future research will investigate how to improve the precision while maintaining a 100% recall in the grouping stage. Also, techniques will be developed to reassemble fragmented video files without any knowledge of the header fragments.

Acknowledgements

This research was partially supported by the China State Scholarship Fund (Grant No. 201506785014), National Natural Science Foundation of China (Grant Nos. 61401176 and 61361166006), Natural Science Foundation of Guangdong Province (Grant No. 2014A030310205), Science and Technology Projects of Guangdong Province (2014B010120002 and 2016A010101017), Project of Guangdong Higher Education (YQ2015018) and NSFC/RGC Joint Research Scheme (N_HKU 72913), Hong Kong.

References

[1] S. Ali and M. Shah, A Lagrangian particle dynamics approach for crowd flow segmentation and stability analysis, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[2] BOSS Project, BOSS Dataset (`www.multitel.be/BOSS`), 2012.

[3] E. Casey and R. Zoun, Design tradeoffs for developing fragmented video carving tools, *Digital Investigation*, vol. 11(S2), pp. S30–S39, 2014.

[4] CAVIAR Project, CAVIAR: Context Aware Vision using Image-Based Active Recognition, School of Informatics, University of Edinburgh, Edinburgh, United Kingdom (`homepages.inf.ed.ac.uk/rbf/CAVIAR`), 2017.

[5] M. Cohen, Advanced carving techniques, *Digital Investigation*, vol. 4(3), pp. 119–128, 2007.

[6] S. Garfinkel, Carving contiguous and fragmented files with fast object validation, *Digital Investigation*, vol. 4(S), pp. S2–S12, 2007.

[7] International Organization for Standardization, Information Technology – Coding of Audio-Visual Objects – Part 2: Visual, ISO/IEC Standard 14496-2:2004, Geneva, Switzerland, 2004.

[8] International Organization for Standardization, Information Technology – Coding of Audio-Visual Objects – Part 10: Advanced Video Coding, ISO/IEC Standard 14496-10:2010, Geneva, Switzerland, 2010.

[9] A. Lewis, Reconstructing Compressed Photo and Video Data, Technical Report No. 813, UCAM-CL-TR-813, Computer Laboratory, University of Cambridge, Cambridge, United Kingdom, 2012.

[10] N. Memon and A. Pal, Automated reassembly of file fragmented images using greedy algorithms, *IEEE Transactions on Image Processing*, vol. 15(2), pp. 385–393, 2006.

[11] G. Na, K. Shim, K. Moon, S. Kong, E. Kim and J. Lee, Frame-based recovery of corrupted video files using video codec specifications, *IEEE Transactions on Image Processing*, vol. 23(2), pp. 517–526, 2014.

[12] A. Neelima and K. Singh, A short survey of perceptual hash functions, *ADBU Journal of Engineering Technology*, vol. 1, 2014.

[13] A. Pal and N. Memon, The evolution of file carving, *IEEE Signal Processing*, vol. 26(2), pp. 59–71, 2009.

[14] R. Poisel and S. Tjoa, Roadmap to approaches for carving of fragmented multimedia files, *Proceedings of the Sixth International Conference on Availability, Reliability and Security*, pp. 752–757, 2011.

[15] R. Poisel and S. Tjoa, A comprehensive literature review of file carving, *Proceedings of the Eighth International Conference on Availability, Reliability and Security*, pp. 475–484, 2013.

[16] R. Poisel, S. Tjoa and P. Tavolato, Advanced file carving approaches for multimedia files, *Journal of Wireless Mobile Networks, Ubiquitous Computing and Dependable Applications*, vol. 2(4), pp. 42–58, 2011.

[17] G. Richard and V. Roussev, Scalpel: A frugal, high performance file carver, *Proceedings of the Digital Forensic Research Workshop*, 2005.

[18] Y. Yannikos, N. Ashraf, M. Steinebach and C. Winter, Automating video file carving and content identification, in *Advances in Digital Forensics IX*, G. Peterson and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 195–212, 2013.

[19] B. Yoo, J. Park, S. Lim, J. Bang and S. Lee, A study on multimedia file carving method, *Multimedia Tools and Applications*, vol. 61(1), pp. 243–261, 2012.