

Videntifier™ Forensic: A New Law Enforcement Service for Automatic Identification of Illegal Video Material

Herwig Lejsek^{†‡}
Ársæll Þór Jóhannsson[†]
†Eff2 Technologies ehf.
Kringlan 1
IS-103 Reykjavík
Iceland
herwig@eff2.net

Friðrik Ásmundsson[†]
Björn Þ. Jónsson[‡]
‡School of Computer Science
Reykjavík University
Kringlan 1, IS-103 Reykjavík
Iceland
bjorn@ru.is

Kristleifur Daðason^{†‡}
Laurent Amsaleg[§]
§IRISA–CNRS
Campus de Beaulieu
35042 Rennes
France
laurent.amsaleg@irisa.fr

ABSTRACT

Tracking down producers and distributors of offensive video material, in particular child pornography, has become an ever growing focus of the world's law enforcement agencies. We describe Videntifier™ Forensic, a new service which radically improves the forensic video identification process, by providing law enforcement agencies with a robust, fast and easy-to-use video identification system. Using this service, a single mouse-click is sufficient to automatically scan an entire storage device and classify all videos. We give an overview of the service and the underlying technology components. We then describe an acceptance test, performed by the Icelandic police forces, which demonstrates the robustness of the service.

Categories and Subject Descriptors: K.4.1 [Computers and Society]: Public Policy Issues—*Abuse and crime involving computers*; H.2.4 [Database Management]: Systems—*Multimedia Databases*

General Terms: Algorithms, Performance.

Keywords: Videntifier™ Forensic; Video Identification; Service; Robustness; Scalability.

1. INTRODUCTION

The proliferation of the Internet has revolutionized distribution and consumption of video material. As mass storage devices have become extremely cheap and broadband connections more and more popular, the passion for collecting and distributing video material has exploded. While most of the material is harmless, a significant portion is illegally produced and distributed. Unfortunately, this includes videos showing sexual violence against children, which has become a significant underground industry. Tracking down producers and distributors of such material has, in turn, become an ever growing focus of the world's law enforcement agencies.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MiFor'09, October 23, 2009, Beijing, China.

Copyright 2009 ACM 978-1-60558-755-4/09/10 ...\$10.00.

As an example, a recent visit to Denmark's High-Tech Crime Center revealed that, in this small country alone, the police forces investigate over 1,500 such cases each year. Roughly two-thirds of the center's 55 employees are devoted to performing video identification. As this work is manual, it is a very expensive process; as some of the material is very offensive, it is also stressful work. Despite such massive efforts from law enforcement agencies, there is general consensus that the amount of material that needs to be identified in each case is rapidly growing beyond their capacity.

1.1 The Traditional Identification Process

When the police suspect someone of distributing illegal material they seize the suspect's computer and all storage devices. A copy is then made of each device to avoid tampering with original evidence. Once the copies have been scanned to find video files, the identification process begins.

The investigation methods for this process, however, are still very poor. Information about the video files is typically entered into a forensic administration system (Encase by Guidance software appears to be the most common system). The investigator must then manually open each and every file and inspect its contents. Note that the entire file must be scanned, as illegal material is often embedded within legal material to avoid identification. Of course the same videos are found over and over again, but still they must be inspected in each separate case, as file names and contents are regularly reorganised and changed.

In order to reduce the workload of the investigator, several departments have introduced the use of MD5 checksums to their investigation process. An MD5 checksum is a hash value that is computed from a video and compared to a collection of known MD5 checksums—for both legal and illegal material. If the checksum matches the checksum of a legal video, no further action is taken. Otherwise, the video must be scanned. Unfortunately, however, checksum matching is prone to failure. For example, by changing even a single bit of a file, the checksum becomes different. Furthermore, an illegal video file can be manipulated so that its checksum matches that of a well known legal video while still retaining the illegal content, causing the file to be overlooked.

1.2 Videntifier™ Forensic

Videntifier™ Forensic is a new service which proposes to radically improve the forensic video identification process,

by providing law enforcement agencies with a robust, efficient and easy-to-use video identification system. Using this service, a single click is sufficient to automatically scan an entire device faster than can be done manually. A summary report is produced, detailing which videos have been identified as either legal or illegal, and which videos must be manually scanned; these videos can then be classified for recognition to reduce the work on future cases.

Because Videntifier™ Forensic is based on state-of-the-art video identification technology, it performs its identification based on the actual contents of the videos. The service can detect modified versions of the same video and identify short clips from a longer video, both of which easily fool checksum-based systems. Videntifier™ Forensic can even tolerate quite severe modifications, which significantly impact the contents of the video file.

And because Videntifier™ Forensic is also based on state-of-the-art in multimedia database technology, the identification process is extremely efficient, even for very large video collections. In standard usage on moderate hardware, the system inspects each hour of video in 30 seconds, with greater accuracy than manual identification, and runs continuously day and night.

The remainder of this paper is organized as follows. We first provide an overview of the Videntifier™ Forensic service, from the point of view of an investigator. Then, in Section 3, we briefly describe the underlying technology and its performance. In Section 4 we describe the outcome of an acceptance test performed by the Icelandic police forces. Finally, we discuss related work and conclude.

2. VIDENTIFIER™ FORENSIC SERVICE

The Videntifier™ Forensic service is built on technology components that enable a computer to remember the visual content of videos, even for a huge collection of video files. This is based on extracting a set of visual “fingerprints” that together uniquely identify the video, and storing them inside a state-of-the-art central database for comparison. As soon as Videntifier™ Forensic has “seen” a video once and its fingerprints have been stored, it is capable of identifying subsequent copies of that video, even when the video has suffered radical changes to the video content, such as compression, camcorder rips, subtitles, and mirroring.

Videntifier Forensic is split up into three major components. These consist of the Videntifier™ Forensic Client, which is installed on the investigator’s workstation and scans the video files; a secure Fingerprint Extraction Unit (FEU) which computes fingerprints for all video scans of the law enforcement agency; and the central Videntifier Database Server hosted by Eff2 Technologies. The workings of each component are described in the following.

2.1 Videntifier™ Forensic Client

The Videntifier™ Forensic Client is the tool that investigators use to interact with the service. In order to make the service appealing to users, Eff2 Technologies has developed an easy-to-use graphical interface that can be installed on both Windows and Linux. The client was designed to be as unobtrusive as possible, by allowing the investigator to minimize the application to the system tray. From there, all major information and configuration of the identification process is accessible, such as: the configuration of identification accuracy; the estimated time left for a running scan;

and a detailed report, once a scan is finished. Furthermore, the investigator can import lists of video files for evaluation from Encase and export the identification results back.

In order to start an investigation process, the investigator simply connects a copy of the suspected storage device (hard drive, USB stick, DVD, or cell phone) to the workstation. As soon as the device is mounted and appears as an icon on the user’s desktop, the investigator can right-click the icon and choose “Investigate” from the menu. Videntifier™ Forensic now starts the identification process. As the identification process works in the background and requires only minimal resources from the workstation, the investigator can continue to work on other things in the meantime.

Internally, the identification starts by scanning the device for video files. Since video files can have any filename or extension, the scan process peeks into each file above a certain size, and examines its header data to identify which files are videos. When all video files have been found on the device, the actual identification starts.

The client now seeks to a random position near the beginning of the video and extracts a set of frames from a short segment of the video (the number of frames depends on configuration settings); The set of frames is then transferred to the FEU for processing. Then the client jumps forward in the video to another random starting point (again, configuration settings determine the exact jump length) and the frame extraction process starts again. Once a match has been found, or the video has been completely processed, the client turns to the next video in the list. The coverage density of the scan can be varied from 1% to 100% of each video file through a configuration menu.

When all videos have been processed, the videos have been classified into three different categories:

Legal videos: Videos that could be successfully identified, but have been classified as not containing any illegal content.

Illegal videos: Videos that were identified as containing (at least partially) illegal content.

Unidentified videos: Videos that have not yet been seen by the system and could therefore not be identified.

The investigator is now offered to export these results (e.g., into Encase) or to print a separate report. Furthermore, the investigator can use the system to scan the unknown material in order to classify it and insert it into the central database, so that the video can be identified in the future.

2.2 Fingerprint Extraction Unit

Eff2 Technologies provides each law enforcement agency with a secure Fingerprint Extraction Unit (FEU) which acts as a secure gateway between the clients and the central Videntifier Database Server. The FEU is a small server computer which is equipped with one or more graphics processing units (GPUs). These GPUs, which are commonly used for producing high-end graphics in computer games, are actually highly efficient parallel processors capable of extracting fingerprints an order of magnitude faster than a standard CPU [4]. The GPU processing must be performed on the FEU as a specific hardware configuration is needed, which is outside of the range of usual computer workstations.

During identification, the FEU receives the frames to be identified from the clients over the local network. The FEU

transfers the frames to the GPU and extracts several hundred fingerprints from each frame. Each fingerprint is a sequence of numbers, extracted from a small area of the frame, which encodes the visual content of that area into a 72-dimensional vector. Once all the fingerprints are computed they are used to query the central database server.

During insertion of videos into the central database server, a slightly different process is used. In this case, the client submits all the frames of the video to the FEU, which generates the fingerprints for each frame as before. Since subsequent frames are typically very similar, however, the fingerprints are likely to also be very similar. The FEU therefore applies a filtering step to the descriptor stream, to remove such redundant descriptors and store only very representative descriptors in the descriptor collection. Through this filtering step, the majority (90–98%) of all descriptors are removed, allowing the service to describe a typical hour of video content using only 100,000–300,000 descriptors.

2.3 Videntifer Database Server

During identification, the extracted fingerprints from each query frame are sent from the FEU over the Internet to the Videntifer Database Server. Once the fingerprints reach the database server they are compared to those already registered in the database. If there is a strong correlation between the order of matching descriptors in the query frames and the result, a match is declared. The database server then looks up the details of the video, such as the name and the classification of the content. This information is then sent, through the FEU, back to the client.

Note that since these fingerprints are a one-way encoding of small parts of the visual content of the video they cannot be used to reconstruct the visual content. This is important, as law enforcement agencies must ensure that none of their data leaks to the public. To add another level of security on top of that, and also to prevent denial-of-service attacks, Videntifer™ Forensic uses IP whitelisting and a secure SSL connection between the FEU and the database server.

3. UNDERLYING TECHNOLOGY

In this section we briefly describe four key technology components of the Videntifer™ Forensic service. First, in order to identify videos, we use fine-grained local image descriptors which are highly optimized for performance [4]. Second, during insertion of videos, an advanced filter is used to remove redundant descriptors. Third, the descriptors are stored in, and retrieved from, a large-scale multidimensional NV-tree index [8]. Fourth, during query processing, a correlation-based decision process is used, which determines the outcome of the retrieval depending on the correlation between the query and the potential match with respect to time. Note that since many aspects of this process are protected by a pending patent application, we only outline the components here; a more detailed article is in preparation.

3.1 GPU-Eff² Descriptors

The fingerprints used by the system are the GPU-Eff² descriptors, which are derived from the well-known SIFT descriptor family proposed by David Lowe [11]. The SIFT descriptors, which are commonly used in robotic vision applications, are computed around specific interest points in the image which are typically found in areas of high contrast. The SIFT descriptor itself is a 128-dimensional vec-

tor which encodes the contrast changes around the interest point. While SIFT descriptors have been shown to be effective at copy detection, Eff² descriptors, an improved version of SIFT, work significantly better for this application [7]. The Eff² descriptors encode visual features into 72-dimensional vectors and can identify altered images much better than SIFT, especially in large collections.

As the extraction of the descriptors has shown to be a performance bottleneck compared to query processing, the computation of the descriptors has been offloaded to the GPU. NVIDIA has recently developed a general programming interface, CUDA, for its GPUs. CUDA integrates parallel programming mechanisms into the C language and has become a de facto standard for GPU programming. By migrating the Eff² descriptor extraction to CUDA, resulting in the GPU-Eff² descriptors, a 12-fold speed increase was obtained compared to Eff² descriptors and up to 40-fold increase compared to standard SIFT descriptors, with some optimization work still to be done [4]. Using this implementation, we can currently process up to 60 video frames per second on a single GPU.

3.2 Descriptor Filtering

The descriptor filter resides on the FEU and is used during insertion of a new video into the collection. Instead of sampling, as during identification, the insertion process needs to extract fingerprints from the whole video file to cover all its content. As the content of consecutive frames often changes only very slightly, many redundant descriptors are extracted. These redundant descriptors typically create small clusters in the high-dimensional space, as the interest point they originate from changes only slightly over time. The task of the descriptor filter is to choose the best candidate from each such cluster and sieve out all further redundant copies of that fingerprint. Depending on how rapidly the content is changing inside a video, up to 98% of the extracted fingerprints can be removed, retaining only the most representative ones for each scene of the video.

3.3 NV-tree Index

The NV-tree database is a novel indexing method which allows very fast and effective approximate nearest neighbor search in large collections of high dimensional vectors. Such high-dimensional vectors typically appear in fields such as multimedia retrieval, comparison of financial data and searches in organic chemical structures. In 1961, Richard Bellmann coined the term “curse of dimensionality” [2] to describe the problem of partitioning a large collection of such vectors for efficient nearest neighbor search. While several methods (e.g., the R-Tree) for indexing low dimensional spaces (up to 12-15 dimensions) were developed in the 1980s and 1990s, none of them was successful in indexing collections with dimensionality above 20. The NV-tree technology has been shown to deliver a very good approximation to this problem, with the benefit of a very short response time unaffected by collection size or dimensionality [8].

While older versions of the NV-tree [8] stored descriptor identifiers redundantly to improve recognition, and were disk-bound as a result, we have since eliminated this redundancy while improving the quality of the result. This change allows the index to reside completely in main memory even for very large descriptor collections, resulting in very significant efficiency gains. Note, however, that the NV-tree

adapts gracefully to disk IOs if the index outgrows memory. The NV-tree also supports efficient insertions and basic write-ahead logging prevents critical loss of data in the event of crashes and allows recovery to a consistent state.

The NV-tree has been tested on image descriptor collections ranging from 5 million up to 3 billion feature vectors (equivalent to about 25,000 hours of video content), with average recall of 70-82% for a given query vector. As the whole index structure can now reside in memory, it can perform up to 5,000 queries per second per CPU core, and scales nearly linearly with additional cores. One interesting property of the NV-tree is that it does not work well with small collections (below several hundred thousand descriptors) as then a high number of false positives is returned. This is due to the probabilistic nature of the index, where the precision is dependent on the number of video scenes residing in the database; if there are few scenes, many descriptors are randomly found to match each scene, while if there are many, these random matches are divided amongst all those many scenes. The more scenes that are inserted into the NV-tree, the higher overall precision is seen.

3.4 Correlation-Based Decision Process

The final component is a signal detector, which decides whether a query matches with a particular video in the database or not. The signal detector resides on the database server and processes the result lists of every individual query frame from the same video. It runs the stop-rule presented in [7] on each frame and aggregates the winners over time from consecutive query frames. In case winners are referring to nearby scenes, the single results are grouped together in a special set. While aggregating more and more query results these sets eventually grow larger. With each update, a correlation score is calculated. Once this score exceeds a certain threshold, the specific scene-group is declared as a match and information on the video it belongs to is retrieved from a relational database. By building up the matching criteria on top of this correlation property the likelihood of false positives can be strictly controlled.

4. EVALUATION

We now describe an acceptance test that was run by the Icelandic police forces on the VidentifierTM Forensic system to evaluate the detection rate and robustness of the system to standard video modifications. This section first describes the video collection and queries used in this evaluation, before discussing the results of the acceptance test.

4.1 Experimental setup

All experiments were run on a desktop machine (containing both client and FEU software) with a 3GHz Intel Core 2 E8400 processor, 4GB RAM and NVIDIA GTX 280 GPU.

As a starting point we used the free video collection from the MUSCLE Video Copy Detection benchmark. This collection contains 101 videos from different sources, with a total length of 80 hours [12]. We extracted 72 million GPU-Eff² descriptors from those videos and created three non-overlapping NV-tree indices on a central server.

The Icelandic police then selected 112 videos from previous cases, containing both conventional video material (such as Hollywood movies and TV series) and a set of adult videos. From each of these videos we randomly selected a 10-minute clip and inserted this clip into the database.

The insertion process for these 112 clips of 10 minutes each (2,4 million GPU-Eff² descriptors) took about 2 hours and 40 minutes, which is about 7 times faster than real time.

Then we extracted a 1-minute clip from each of the 10 minute clips, and based our queries on those 112 shorter clips. We modified each of the query clips with 33 predefined modifications, most of which are very common video transformations, using the AviSynth video scripting tool. A description of the modifications is found in Table 1.

We stored the resulting $112 \times 33 = 3,696$ query clips in a folder and selected this folder for investigation through the VidentifierTM Forensic Client as outlined in Section 2. The video coverage was set to 100% for these short clips, so each query clip resulted in two query sets, each containing 60 query frames extracted at an interval of 12 frames.

Processing all 3,696 query clips took 3 hours and 55 minutes, for an average identification time of 3.8 seconds per clip. It must be noted, however, that this acceptance test only considered clips with copies already inserted into the collection. As a result, evaluation often stopped early, before processing descriptors from all query frames. Unknown video clips typically need longer processing time.

4.2 Results

Table 1 shows the outcome of the acceptance test. Of the 3,696 query clips, a total of 3,644 clips could be identified by the VidentifierTM Forensic system, for an overall recall of 98.6%. Overall, this is excellent recall. There are, however, 9 different modifications where at least one clip is missed, and we now examine those modifications in more detail. Figure 1 shows five of these modifications for a clip that was detected.

The largest number of failures (16) occurred with clips whose brightness was decreased by 20% (Figure 1(b)). When reviewing the actual clips that could not be identified, we observed that eight were so dark that no actual content could be seen. The scenes happened to take place at night, in a cave, in space or in a dark room, so that a significant decrease in brightness made the content vanish completely. Another seven clips were also very dark for most parts of the clip, so that only a well-established movie-freak might have recognized the content. Only a single video that was missed of this variant could be called acceptable for watching.

The next two modification groups that yield misses are VERTICAL and HORIZONTAL REDUCE (Figure 1(c)). This significant and noticeable change of the aspect-ratio challenges the invariability of the SIFT family of descriptors towards affine variations. As the GPU-Eff² descriptors have not yet been adapted to handle extreme affine changes, the high identification rate is still a very good result.

The two Picture-in-Picture variants (Figure 1(d)) are also difficult. The video is scaled down to 25% of its original size and inserted into a host video. Clearly this challenges even fine-grained local descriptors, as they must match two videos at the same time. Going to the next modification, INFO, which inserts information about the video into each frame (Figure 1(e)), the system failed to detect clips where the information text covered almost the whole screen.

A very special case in videos are the credits. Among the 112 selected video clips 3 contained only text from credits. Because of the type of information they contain they are lost in several modifications (9 times in total). In particular, they are lost in the very strong RESCALE 25 variant (Figure 1(f)), where an aliasing effect makes the content

Modification	Description	Detected Clips
ADD BORDERS	Add a black border around the video file (50 pixels wide on each side)	112 (100%)
BLUR	Repeating Blur (3 times) with a 3x3-kernel blurring filter	112 (100%)
BRIGHTNESS 50	Decrease the brightness by 20%	112 (100%)
BRIGHTNESS 150	Increase the brightness by 20%	96 (86%)
CONTRAST 50	Decrease the contrast by factor 0.5	112 (100%)
CONTRAST 150	Increase the contrast by factor 1.5	112 (100%)
CROP 10	Central crop, cropping off 5% of the video’s content from all borders	112 (100%)
CROP 25	Central crop, cropping off 12.5% of the video’s content from all borders	111 (99%)
FLIP HORIZONTAL	Flip the video along the y -axis (horizontal)	112 (100%)
FLIP VERTICAL 2	Flip the video along the x -axis (vertical)	112 (100%)
GAMMA 70	Gamma Correction for $\gamma = 0.7$	112 (100%)
GAMMA 180	Gamma Correction for $\gamma = 1.8$	112 (100%)
GRAYSCALE	Convert the video into gray scale	112 (100%)
HORIZONTAL REDUCE	Change of the aspect-ratio, scale the x -axis down to 75% of its width	105 (94%)
INFO	Print information (text) about the video over the actual content	106 (95%)
NOISE	Add random noise to the video	112 (100%)
PAL DEINTERLACE	Vertical blurring, often seen in television material	112 (100%)
PIP CENTER	The video is rescaled to 25% of its size and placed centrally in front of a second video which plays in the background.	103 (92%)
PIP TOP RIGHT	The video is rescaled to 25% of its size and placed in the top-right vertex of a second video which plays in the background.	109 (97%)
RAIN	Animated raindrops in front of the video	112 (100%)
RESCALE 25	Bi-cubic resizing of the video to 6.25% of its size	109 (97%)
RESIZE 50	Resizing of the video to 25% of its size using a gaussian resizer	112 (100%)
REVERSE	Playing the video in reverse (from the end to the beginning)	112 (100%)
ROTATE 90 LEFT	Rotate the video 90° counter-clockwise	112 (100%)
ROTATE 90 RIGHT	Rotate the video 90° clockwise	112 (100%)
ROTATE 10	Rotate the video 10° counter-clockwise retaining the actual video size	112 (100%)
FRAMERATE 12	Reduce frame rate to 12 fps	112 (100%)
FRAMERATE 2	Reduce frame rate to 2 fps	111 (99%)
SHARPEN	Sharpen the video with a 3x3-kernel sharpening filter	112 (100%)
SHIFT	Shifting the image 50 pixels left and 50 down, retaining video size	112 (100%)
SPOTLIGHT	Set the whole video black except a spotlight wandering from the left margin to the right, radius is 57% of the video’s height	112 (100%)
SUBTITLES	Set subtitles into the video clip	112 (100%)
VERTICAL REDUCE	Change of the aspect-ratio, scale the y -axis down to 75% of its height	106 (95%)
Total	All modifications (3,696 clips)	3,644 (98.6%)

Table 1: A quality comparison of the descriptor schemes for different image modifications.

completely visually distorted. Surprisingly, we also found a lost match among the CROP 25 variants, which local descriptors typically handle well. This missed clip, however, has been taken from a music festival with fast changing visual content recorded in an extra-widescreen (10:3) format. Cropping this format is especially hard as all descriptors lie inherently close to the margin while very few descriptors are found in the central area. As all the query frames are scaled down for processing, this clip fails for five modifications.

5. RELATED WORK

Detecting illegal visual material is very related to the field of copy detection for images and videos, which recently saw a burst of activity. For videos, the typical method is to detect key frames, which are representative of whole scenes, and apply local descriptor creation to those frames [6]. Many very robust local description schemes exist. For example, Ke et al. used a variant of SIFT, called PCA-SIFT, for copy detection, while Berrani et al. used the RDTQ descriptors. In [7], however, it was shown that the Eff² descriptors outperform all three variants, SIFT, PCA-SIFT and RDTQ.

Videos challenge the efficiency of the local descriptor extraction process. A recent GPU-based implementation of SIFT, SiftGPU [14], has shown to be about as effective as Eff² and GPU-Eff², but the GPU-Eff² descriptors are extracted 50% faster than SiftGPU [4].

There are also many proposals for high-dimensional index structures, including LSH [5], Spill-Trees [10], cluster-based approaches [9, 3] and Video Google [13]. The NV-tree has shown to outperform both LSH and Spill-Trees, even with the redundancy of the previous version [8]; the current non-redundant version is an order of magnitude faster.

We are aware of at least one existing system that uses image and video databases to investigate child abuse cases. This system is developed by LTU Technology. While they address rather similar issues, the internals of their image and video description scheme as well as their search engine have not been disclosed in details, making any comparison difficult. We are not aware, however, of any other approach extensively using GPUs for extreme efficiency, neither of any other approach doing approximate nearest neighbor searches at very large scale in constant time as the NV-tree does.

Up to now, most evaluations have been rather ad-hoc, as each research team has had access to a different reference collection. As a result, comparison between systems has often been difficult. Recently, however, some efforts have been initiated to create standard reference collections, for example through the CIVR Copy Detection event and TRECvid; we have indeed used the reference collection from the CIVR event in our experiments. We believe that such events will be important for the progress of research in copy detection, as they have shown to be in many other research areas.



Figure 1: Selected examples of the modifications evaluated in the Acceptance Test.

6. CONCLUSION

We have presented VidentifierTM Forensic, a new service which proposes to radically improve the video identification process used by law enforcement agencies, by providing them with a robust, efficient and easy-to-use video identification system. Using this service, a single click is sufficient to automatically scan an entire device, resulting in a summary report detailing which videos have been identified as either legal or illegal, and which videos must be manually scanned; these videos can then be classified to reduce the work on future cases. Once identified the videos become a part of the automatic process and need not be watched again.

VidentifierTM Forensic is based on state-of-the-art technology (patent applications are pending), resulting in excellent performance. By using GPU processing for descriptor extraction and filtering, and a novel high-dimensional index for database insertions and retrieval, the system inspects each hour of video in 30 seconds. We have also described an acceptance test run by the Icelandic police forces, where VidentifierTM Forensic detected 98.6% of the query videos. While this is an excellent detection rate, it can be considered artificially low as many of the 52 missed query clips were difficult or impossible to identify, even for a human observer.

We believe that VidentifierTM Forensic is one of the most promising tools available in the fight against illegal video distribution, and child pornography distribution in particular. Despite its novelty, we expect it to become an integrated investigation procedure, so that manual inspection will hopefully be history within a few years.

7. REFERENCES

- [1] F. H. Ásmundsson, H. Lejsek, K. Dadason, B. T. Jónsson, and L. Amsaleg. Videntifier Forensic: Robust and efficient detection of illegal multimedia. In *Proc. ACM Multimedia (demo paper)*, Beijing, China, 2009.
- [2] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.
- [3] F. Chierichetti, A. Panconesi, P. Raghavan, M. Sozio, A. Tiberi, and E. Upfal. Finding near neighbors through cluster pruning. In *Proc. ACM PODS*, Beijing, China, 2007.
- [4] K. Dadason, H. Lejsek, B. T. Jónsson, and L. Amsaleg. Full GPU acceleration of local descriptors using CUDA. Technical report, Reykjavík U., 2009.
- [5] M. Datar, P. Indyk, N. Immorlica, and V. Mirrokni. *Locality-sensitive hashing using stable distributions*. MIT Press, 2006.
- [6] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford. Video copy detection: A comparative study. In *Proc. CIVR*, Amsterdam, Netherlands, 2007.
- [7] H. Lejsek, F. H. Ásmundsson, B. T. Jónsson, and L. Amsaleg. Scalability of local image descriptors: A comparative study. In *Proc. ACM Multimedia*, Santa Barbara, CA, USA, 2006.
- [8] H. Lejsek, F. H. Ásmundsson, B. T. Jónsson, and L. Amsaleg. NV-tree: An efficient disk-based index for approximate search in very large high-dimensional collections. *IEEE TPAMI*, 31(5):869–883, 2009.
- [9] Chen Li, Edward Y. Chang, Hector Garcia-Molina, and Gio Wiederhold. Clustering for approximate similarity search in high-dimensional spaces. *IEEE TKDE*, 14(4):792–808, 2002.
- [10] T. Liu, A. Moore, A. Gray, and K. Yang. An investigation of practical approximate nearest neighbor algorithms. In *Proc. Neural Information Processing Systems*, Vancouver, BC, Canada, 2004.
- [11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
- [12] MUSCLE Video Copy Detection Evaluation Benchmark. www-rocq.inria.fr/imedia/civr-bench.
- [13] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *Proc. IEEE ICCV*, Nice, France, 2003.
- [14] C. Wu. SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). <http://www.cs.unc.edu/~ccwu/siftgpu/>.