

Addressing the Challenge of Managing Large-scale Digital Multimedia Libraries

Cathal Gurrin^{1,2}, Tjälve Aarflot¹, Sorin Sav², Dag Johansen¹

¹Dept. of Comp. Sci,
University of Tromsø,
Norway
tjalve.aarflot@uit.no, dag@cs.uit.no

²Centre for Digital Video Processing
Dublin City University,
Dublin, Ireland
cathal.gurrin@computing.dcu.ie, sorinsav@eeng.dcu.ie

Abstract: *Traditional Digital Libraries require human editorial control over the lifecycles of digital objects contained therein. This imposes an inherent (human) overhead on the maintenance of these digital libraries, which becomes unwieldy once the number of important information units in the digital library becomes too large. A revised framework is needed for digital libraries that takes the onus off the editor and allows the digital library to directly control digital object lifecycles, by employing a set of transformation rules that operate directly on the digital objects themselves. In this paper we motivate and describe a revised digital library framework that utilises transformation rules to automatically optimise system resources. We evaluate this library in three scenarios and also outline how we could apply concepts from this revised framework to address other challenges for digital libraries and digital information access in general.*

Categories: H. Information Systems

H.3 INFORMATION STORAGE AND RETRIEVAL

H.3.7 Digital Libraries

Subject descriptor: Systems issues

Keywords: Design, Experimentation, Human Factors

1. Introduction

The volume of digital information in the world is increasing every day. Our mobile phones can capture and stream digital video content, our digital video recorders can record hundreds of hours of content, our cameras can take thousands of photos, and some individuals have even gone so far as recording everything that they see and experience [1]. Never before have such content creation tools been in the hands of so many. Truly, we are witnessing the dawn of a new age of content creation, where we are transforming from a generation of content consumers to a new generation of content creators. The challenge to be addressed is not how to capture these oceans of data, rather it is how we store and locate relevant information from our archives.

Conventionally, content creators have assumed the principle of 'store everything' and that a googlisation [3] phase of software development will eventually address the problems of locating content from our vast archives. However our conjecture is that the 'store everything' principle is far from ideal and we propose a different solution, that of self-regulating digital libraries. Although it has mostly gone unnoticed, in 2007 a key milestone was reached in the history of computing, when the amount of information available surpassed the amount of available storage space [4]. The 'store everything' principle cannot be upheld, therefore the onus of digital library maintenance either rests on the shoulder of a human editor, or the digital library itself must take some responsibility for optimally managing the data contained therein. Assuming the second of the two options, then this poses a new challenge for digital libraries, namely how to effectively self-manage vast quantities of heterogeneous data, avoiding the imposition of a significant user overhead in the data management process.

2. Background and Motivation

In 2008, there were a billion digital cameras or camera phones capturing data [4], more than 900 million personal computers and over half a billion audio players. Looking forward to 2010, it is expected that the installed base of such devices will be 50% larger. Much of the information created by these devices will be stored in digital libraries and how we store, search and access this information in personal and shared digital libraries is a subject of ongoing research. Motivation for this work comes from the authors' experience of developing large-scale digital libraries for both digital video [5] and pioneering distributed sensor networks [6]. Our experience suggests that the conventional (ad-hoc) information lifecycle management of digital objects in digital libraries is no longer sufficient. There are 'off-the-shelf' digital library solutions that adhere to the 'one-size-fits-all' approach [7], however these libraries will rely heavily on metadata-based digital object organisation and will require significant editorial input to the management process. Constructing digital libraries specifically to suit the requirements of a particular scenario, such as managing archives of digital video content [8], or archives of written text, such as Project Gutenberg [10], is one approach. Such an approach however introduces significant overheads for both digital library management and for the creation of the digital library in the first-place, both of which are beyond the scope of many organisations, and certainly most individuals.

In this paper we present a revised framework for digital libraries where the digital objects are transient (can be transformed from one representation to another) instead of fixed in nature. In addition, our framework is a self regulating framework which means that the overhead of organizing data in the digital library is maintained by the digital library itself, and does not rely on the user to manage the digital library content. We will make use of three application scenarios to describe the impact of the revised framework for digital libraries: two everyday use scenarios, and another scenario that is likely to become important in years to come as more and more individuals begin to maintain personal digital libraries of life experiences. These three scenarios are:

- **Security Video:** There are millions of digital security video cameras in existence, and each is recording onto a fixed size storage infrastructure. At present, unless there is an immediate and pressing need to access the video footage, it is simply deleted after a prescribed period of time.
- **PVR:** Take a Personal (digital) Video Recorder (PVR) which typically has a fixed size hard disk that can store in the order of hundreds of hours of recorded content in a personal digital video library. Due to this upper bound on storage resources, eventually at some point the user will have to delete content.
- **HDM:** a Human Digital Memory (HDM), though a very new concept, is concerned with capturing and storing life experiences digitally, as exemplified by Gordon Bell's MyLifeBits [1] project at Microsoft or the author's experiences of gathering visual HDMs [2]. Over the course of a lifetime, one can imagine the enormous data requirements for these (extremely) personal digital libraries, whereby upwards of a million digital photos can be captured annually. This is a scenario that is likely to become more common in the coming years.

The framework we present in this paper is ahead-of-the-curve in that we have examined challenging digital library scenarios from everyday experiences, but also the HDM scenario and in so doing we are in a position to present a revised digital library framework that has been guided by both the real-world needs of today, and the challenging needs of tomorrow.

We now identify background research in the areas of digital libraries, digital objects, information lifecycle management and digital video libraries, before discussing our new digital library framework in Section 4 and our implementation and results in section 5.

3. Digital Libraries

A digital library is defined as a 'focused collection of digital objects, including text, video, and audio, along with methods for access and retrieval, and for selection, organization, and maintenance of the collection' [11]. While traditionally digital libraries have been large digital representations of conventional libraries, nowadays with the advent of ubiquitous content creation, digital libraries are managed by individuals and they are incorporating increasing amounts of personal digital information, for example, videos, photos, digital memories and emails [12].

Regardless of whether the library is a personal library or a conventional shared repository library, the building blocks of digital libraries are as follows:

- Digital objects, which are the key underlying data stored in digital libraries. A digital object adheres to a pre-specified digital object lifecycle.
- Metadata concerning these digital objects. Often this metadata is used as the sole underlying content to support resource discovery within the digital library.
- Software to access the digital objects in the libraries, including management tools, search tools and interfaces.

3.1 Limits of Current Generation Digital Libraries

Conventional wisdom concerning digital libraries suggests that the digital objects contained therein are fixed in nature (static) and, while the metadata concerning the objects can (and should) be updated over time, that the digital objects themselves are not, except for necessary format transformations to support prevailing technologies. While this is a worthy goal for digital libraries, this will not always be possible due to limited storage space and limited editorial resources.

The concept of a fixed and permanent digital object is based on that of a physical book from conventional libraries. However, an advantage of digital objects in digital libraries is that they are malleable, mutable and mobile [13] and hence it is not always necessary to consider them to be fixed and permanent. Consider a PVR for example, where hard disk space of the PVR is naturally limited, therefore (very quickly) a user has to delete content, so as to be able to record new TV shows. Our contention is that such digital objects (not only the metadata), can over time be modified and replaced, yet still remain in the digital library and accessible by the end user. The current onus on the individual to manage their own archive of personal content is thereby supplied by automated policies of the system.

The practice of applying certain policies to the effective management of information throughout its useful life is called Information Lifecycle Management (ILM). Mazurek and Werla identify the conventional four state lifecycle from pre-import, through insertion, publishing and finally deletion [14]. Such a lifecycle model for digital objects aims to maintain long-term availability within the digital library and at the end of the lifecycle, the objects are simply deleted. Usually, digital objects in this lifecycle will be fixed in nature and will likely not change between import and deletion, as seen in the example lifecycle in Figure 1, which illustrates a four state lifecycle from *capture* (1) through *import* (2), *publish* (3) and finally *delete* (4). In this lifecycle, digital objects are usually only visible to the end user when in the publish phase. This is a generally accepted lifecycle for digital libraries, and also forms the basis of our work; however there is one inherent limitation in this lifecycle: the need for editorial control of the process.

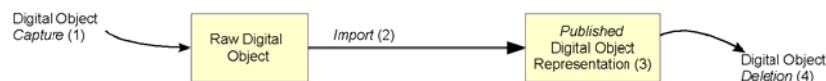


Figure 1. Event detection set-up of CCTV experiment.

However, there has been some research in related areas, for example Mazurek and Werla [14] propose a lifecycle of a digital object that allows for a number of editions of the digital object to be maintained and made available, which would be very useful for different display devices. Other work to describe a digital object and repository architecture [15] addresses concepts such as mixed data types and specifying multiple content disseminations of digital objects. Lagoze and Davis [16] are concerned with issues relating to distributed, decentralized multi-format document collections and Curino et al.[17] present the Panta Rhei Framework to manage frequent database schema changes.

3.2 Addressing limitations in traditional library systems

Our conjecture is that by making digital objects malleable (transient) in nature and by having the digital library self-regulate the digital objects contained therein we can remove the need for constant editorial control of the objects in the library. It is our considered opinion that applying a transience property to digital objects is going to become essential in some domains (such as video, sensor network libraries, or HDM libraries). In previous work we have examined digital library frameworks with regard to supporting the automatic reduction of disk space requirements by exploiting content redundancy [18]. In this paper, we significantly extend this work by supporting transient digital objects and self-regulation of the digital objects. We propose that the digital library can be significantly more user friendly for busy or non-professional editors, be able to store significantly larger quantities of data given the same limited resources, and be able to automatically transform the digital objects between storage formats to reflect the popular formats of the time.

4. A Revised Framework for Digital Libraries

The framework for a digital library that we present is based on the concept that digital objects are transient instead of being static/fixed and that software is constantly managing the digital objects to ensure that they are always in the most suitable format for the digital library. This allows us to better meet the new challenges for digital library systems that are posed as we face ever-increasing quantities of digital content. Digital objects in our framework can be dynamically transformed into new representations to allow for flexible and optimal use of the information contained in them.

4.1 Architecture

In a conventional digital library, objects are imported, published to the library, and accessed and managed through a user interface. The framework we propose differs from a conventional digital library in that it contains two new concepts, an *Observer* and a *Transformation Engine* which control the internal lifecycle of the digital objects in the library, where typically these digital objects would be simply in the 'publish' state and accessible to the end user. The architecture for our digital library framework is shown in Figure 2.

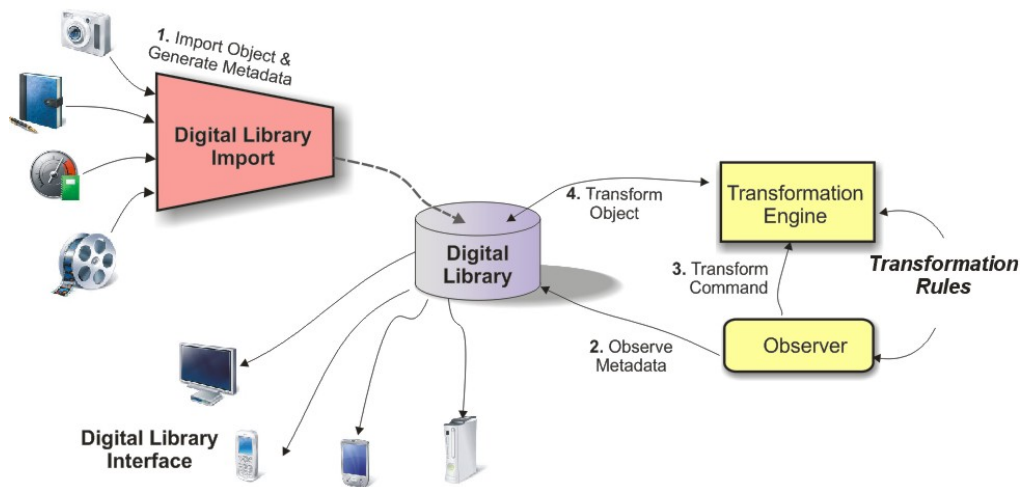


Figure 2. Components of extended framework for digital libraries.

The typical import-to-publish scenario in our framework (shown in Figure 2) can be explained as follows:

- 1) Upon importing an object into the digital library, metadata is imported/generated to support user access.
- 2) The Observer's role in the framework is to examine the digital objects and their metadata, looking for a trigger state (a pre-defined object state), so as to apply transformation rules.
- 3) Upon positive evaluation of a trigger state, the Observer sends a transform command which identifies the digital object and the transformation rule to be applied.
- 4) Finally, the digital object is transformed by the Transformation Engine, which will convert the object from one state/representation into another state/representation; hence it is a transient digital object. The digital object remains in the library and the process continues.

We will now look in detail at the digital objects, their lifecycle, and transformation rules of our digital library framework.

4.2 Digital Objects

Digital Objects in our framework can be comprised of any sort of data (video, image, audio). In addition, each object will have a set of metadata tied to it, together with the object's lifecycle state in the digital library (capture, import, publish, delete). Typical metadata consists of creation and access time of the object, data size of the object as stored on a storage medium, and the location from where the object's data can be fetched. Other metadata may be created by the digital library system to help indexing and searching over the objects, or to improve access. Typically, such metadata is object-specific. Additional metadata may also be added either manually or during transformation.

4.2.1 Digital Object Lifecycle

We consider objects in our framework to be transient in nature, as a benefit of digitalization. An example of how we use this property is when objects are transformed with the purpose of data reduction, to occupy less space on

a storage medium, but still contain the important information. Transformation is not limited to data reduction however; rather it allows for any kind of alteration of the objects. Examples may be to convert video data from one format to another to make it suitable for access on certain systems, change the resolution of an image to make it fit in a certain context, or perhaps to do speech recognition from audio data and add the text transcript to the object metadata.

To manage the transient digital objects in our library, a revised lifecycle state description needs to be available to each object. After creation of the digital objects when they are imported into the library, our framework proposes that digital objects, rather than be maintained in a fixed capture, import or publish state, can transition into new forms according to dynamic configurations. Although this can be used to, for example, remove private data from an object during capture or import, most transitions will normally occur in the published state, where objects can enter a cycle of transitions. As long as the object stays published and does not get deleted it can make a new transition cycle where each cycle results in a new digital form of the object. The cardinal example is when objects are revised for each cycle, reducing the data or altering it into a more suitable form. Figure 3 shows a transient digital object lifecycle, in which a digital object is captured, imported, published (cycles through a number of phases of data transformation/reduction) and finally deleted.

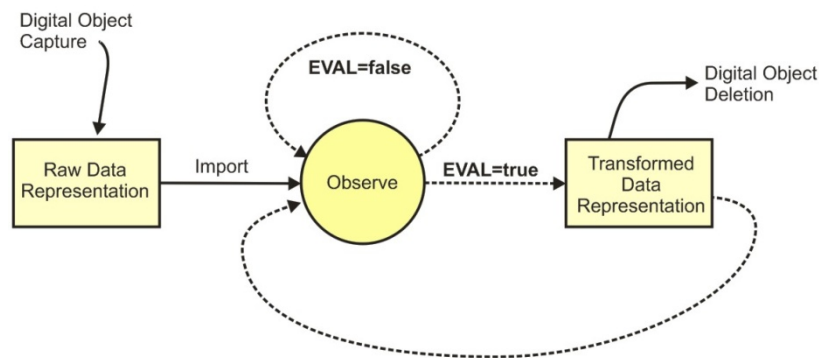


Figure 3. A stateful lifecycle of transient digital objects.

When a digital object is observed for example in the publish state, and when the observing device evaluates the status of the object to be true (given certain requirements) the object is transformed into another representation in the publish state. This process continues as long as the object stays in the published state, or the digital object is deleted from the library. By default, an object goes back to being observed after a transformation. The time at which an object is selected for transformation is dependent on a set of manually generated transformation rules that govern the process.

The key factor in this framework and lifecycle is that the digital object is not persistent in a certain state, a key difference from other digital libraries. Rather, the object undergoes a sequence of transformations, the necessity for which in our framework can be decided by the digital library itself. In addition, the nature of the transformation rules needs to be specified for the type of library and the nature of the digital objects therein. For example, the transformation rules for a PVR library would differ from the transformation rules for a security video library or personal photo library. In a PVR, a transformation rule could be to remove adverts, while in a security video library a transformation rule may be to delete any content that occurs at night and has no events with video motion.

4.2.2 Digital Object Transformations

In our framework, object transformations are specified by a set of rules that are extensible and can be viewed as plug-ins into the digital library. These rules operate differently on each type of digital objects. Digital objects can then have either their data or the metadata (or both) transformed (altered) by these rules. Objects are triggered for transformation by one or more conditions, specified in the transformation rules. For example, one rule could have as a pre-condition that a digital object is in a published state prior to execution; another could check object metadata entries such as timestamp or geographical location. When such a condition is observed, a trigger command is sent to the transformation engine which will then apply the corresponding transformation on the respective object.

To maintain data integrity, object transformations are atomic (similar to transactions in a DBMS) and performed in a single transaction where the object change from state S1 to state S2. During a transaction the digital object is

not in the publish state and can only be accessed by the transformation engine. Upon completion of the transformation, the object is returned to its previous state if nothing else is specified in the transformation rule.

4.3 Transformation Rules

Transformation rules consist of *trigger conditions* and *transformation software*. Trigger conditions define one or several metadata fields to observe, threshold values for each field, and the comparison method to be applied on the threshold value (equals, less than, greater than, etc.). A field can also appear several times with different threshold values. *AND* or *OR* operators can be used to combine more than one metadata field in order to create constructs such as a range ("greater than x *AND* lower than y"). A simple example of this would be a trigger condition for all objects created between 1st and 31st of March 2007.

Transformation software describes the operations to be performed on the digital object once triggered. The simplest software will only alter some of the metadata values of a digital object. One such rule can for example have as its only task to move objects from *import* state to *publish* state. On the other end of the scale is software that performs deep analysis over the object data itself, for example on low-level image features. A transformation rule may alter the object's metadata to indicate that the object has been transformed by this rule, or in a more advanced example will also insert a timestamp when transformation was last performed by this rule.

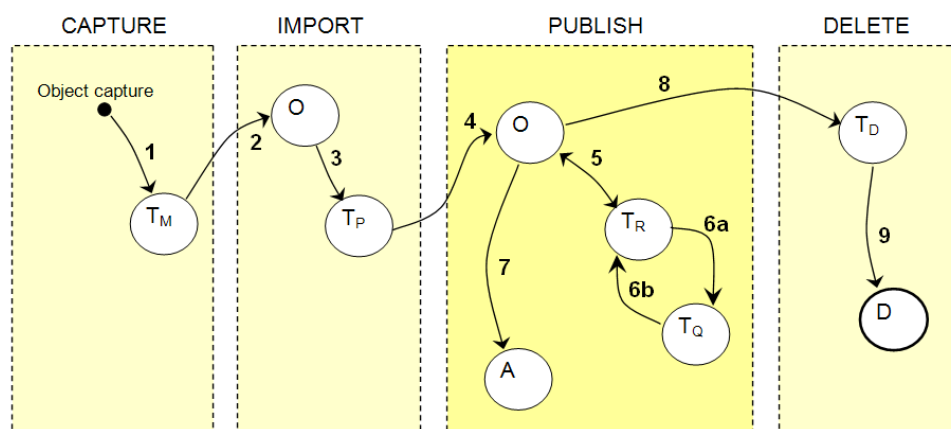


Figure 4. Example transition path diagram of an arbitrary framework implementation.

Figure 4Error! Reference source not found. shows an example transition path diagram for a typical framework implementation. Objects are here shown to be either observed **O**, transformed **T**, accessed **A**, or deleted **D**. After capture (1), objects are transformed by T_M , an example transformation rule generating additional metadata before import. Objects are then sent to the import state (2), where they are observed **O**. Before publish, objects are transformed by T_P , an example privacy rule which filters out data of a private nature (3) before making the objects accessible from the library (published, 4). In the published state, observed objects can either be accessed (7), deleted (8), or transformed by T_R , a data reduction transformation rule. Objects can circle between this rule and T_Q , a quality rule filtering out bad quality data (6a and 6b), or go back to being observed (5). Objects are deleted by T_D , a deletion transformation rule, sending objects to the terminal state **D** (9). Note, once published an object cannot go back to the capture or import state.

Our framework library implementation can be extended by adding new transformation rules to deal with new information sources and data types. Each transformation rule is in reality a plug-in, defined to work under certain conditions and on specific type of data.

5. Framework Evaluations

We employ our proposed framework on three real-world use cases where we look to dynamically reduce the amount of data stored in the digital library via transformation rules. First use case covers a video surveillance system, the second a personal video recorder while the third addresses the case of human digital memories.

5.1 Video Surveillance Use Case

At the end of 2006 in Britain alone, there was an estimate of 4.2m CCTV (Closed-circuit television) systems used for security surveillance – one for every 14 people [18]. If all cameras were recording 24 hours a day and no data was discarded, the CCTV's of Britain would generate roughly 6.9 petabytes (7.2m GB) a day, calculated for an image resolution of 2CIF (704x288) and a frame rate of 25 frames-per-second. Obviously, this amount of data cannot be feasibly stored; only a small fraction of this data is likely to be useful into the future so it should ideally be stored. However, with so many security cameras, selecting which images to store is beyond any manual driven process. CCTV therefore makes an ideal subject area for testing our framework.

We created a small scale digital security video library (Figure 6) in which a security camera constantly monitors the only access door into a multi-person office location. The video stream captured from the camera is partitioned into one hour-long segments and imported into the digital library. These segments are the initial digital objects which will later be transformed to achieve storage size reduction.

Transformations are defined via a set of rules that in time will reduce the digital object from a raw video segment into just a face and body patch frame, as shown in Figure 5. Between transformations the objects are published in one of the four states described below:

- **Raw video state** (S_1), the import state of each one hour-long raw video possible containing possible door access events which were not yet detected.
- **Event video state** (S_2), where the video object has been transformed from a one hour segment (S_1) into a number of short video event objects representing individual door security events (S_2).
- **Event keyframe state** (S_3), where a video object (S_2) has been transformed into a keyframe image, automatically selected to show the person stepping through the door frame (S_3).
- **Face & body state** (S_4), where the keyframe image object (S_3) has been transformed into a cropped image of the face and body patch of that person (S_4).

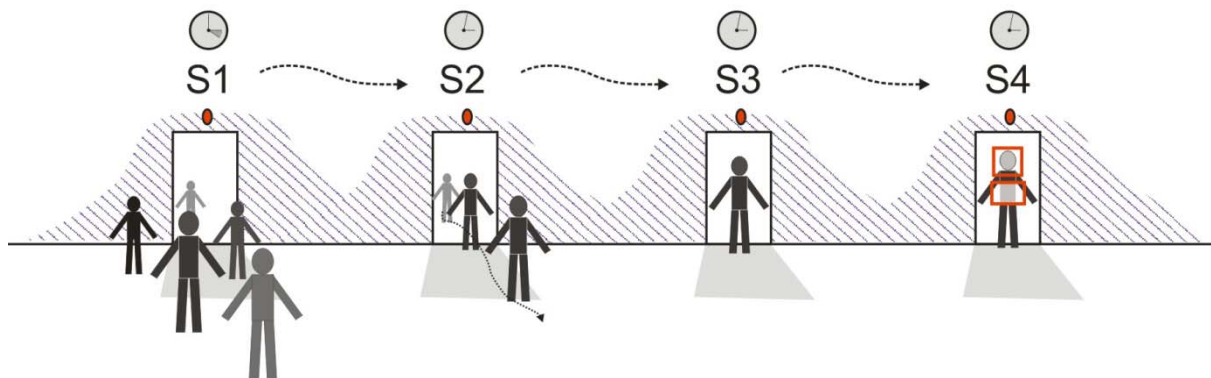


Figure 5. Stages of transformation, from S_1 to S_4 .

The transformation rules that govern the lifecycle of the digital objects are configured as a chain in our implementation, thus they are triggered one after another. This is done in order to allow publishing of a video object immediate after capture since performing the video analysis requires some amount of time. The final *face and body patch state* is the optimal storage size of the object. However a number of processing steps need to be performed in order to obtain it and this takes some amount of time. Without the intermediate publishing stages the object would either be offline or would remain unmodified until the entire visual processing is completed.

If our library framework had followed conventional digital object lifecycles, then the storage requirements for a security video camera in a busy location would be such that the lifecycle of the objects would require object deletion within a short period of time. In our case, a 360 GB storage medium would be able to hold only a week of full quality video data from our prototype installation.

Three transformation rules govern the object transformation chain as detailed below:

- **Video event Transformation Rule** triggering the change of a video segment object from state S1 into a number of objects in state S2. An event in our prototype is defined as the situation when someone walks through the door (in or out), as identified by automated transformation software. As a result of this transformation, one hour of video may be replaced by several events of people walking through the door. On average, the digital objects in state S2 contain about 4.5 seconds of video.
- **Event keyframe Transformation Rule** applied on an object (video clip) in state S2 in order to reduce it to a single event image (keyframe) by selecting a video frame that shows the person in the middle of the door frame. This process also detects whether the person is entering or leaving the office, which is added as metadata to the digital object.
- **Face & body Transformation Rule** that further reduce an object in state S3 from a large image, to just an image area containing the person head and another area corresponding to the person's upper body section.

These transformations save a significant amount of disk space for every digital object. For experimentation, video was captured at 4 fps (frames per second) from the video camera, and keyframes extracted from the video were stored as JPEG images at a resolution of 1280x720 pixels. At reasonable compression rates, one second of video requires 635 kB of disk space and one hour then amounts for up to 2.18 GB. The amount of important data, represented by important event information, is highly dynamic, varying according to the time-of-day and day-of-week. On average, there were 120 events during a day, and most activity was observed between 10 AM and 4 PM from Monday to Friday.

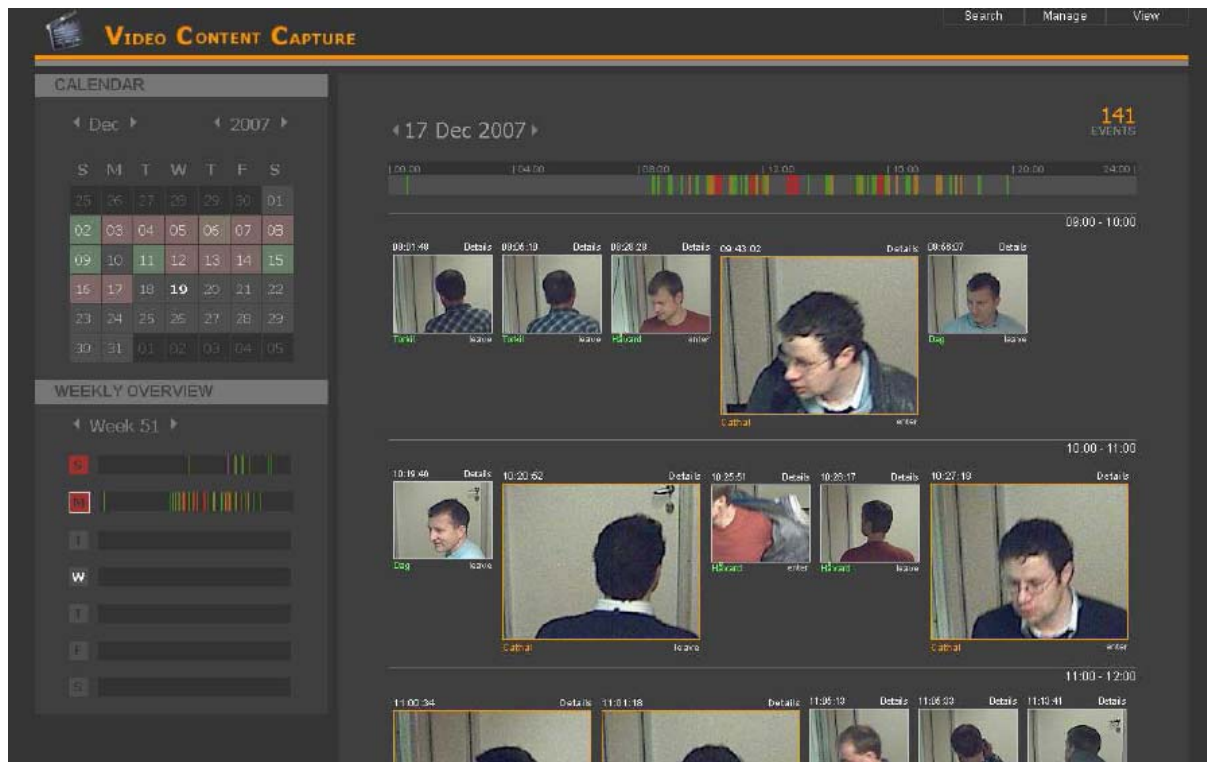


Figure 6. Main interface for the security video implementation.

Figure 6 shows the main interface for our implementation of a video surveillance use case. Objects from S4 (cropped face of person) is displayed in a time-ordered fashion. Images are given weight by importance of person (known, rare, unknown), visualized by a larger image and change of color. Library objects can be browsed by date and/or selected from event-summary bars.

We can compare the storage capacity the objects require in each of the four stages for a given day as shown in Figure 7. The reduction in storage capacity required for any given day is from 52.3 GB (S1) to 0.6MB (S4), yet the digital library still maintains the key events in the archive. Because it relies on a less than perfect face detection algorithm the transformation from S3 to S4 does result in only 84% of the events being represented by an accurate picture. This deficiency could be significantly alleviated by using a more advanced face detection algorithm, and an improved algorithm will tend towards 100% accuracy [20].

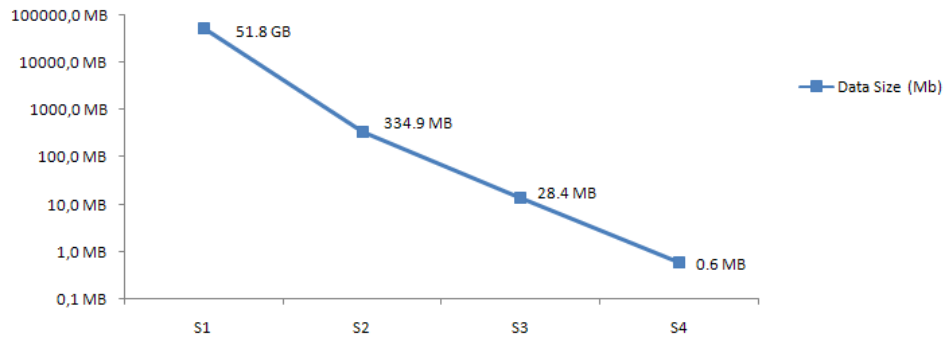


Figure 7. Disk space reduction per day through S1-S4 on a logarithmic scale.

Another benefit of this reduction is in providing speed-up for search and visualization of events recorded in the archive. Even more, the time taken with the low-level processing the content is significantly reduced for example when seeking for a particular face in the video data, such as the face of an intruder. We evaluated our face detection algorithm on a day's objects in each of the four states (S1-S4). Naturally, the time it took to perform the query on a given day's data decreased rapidly, as shown in Figure 8.

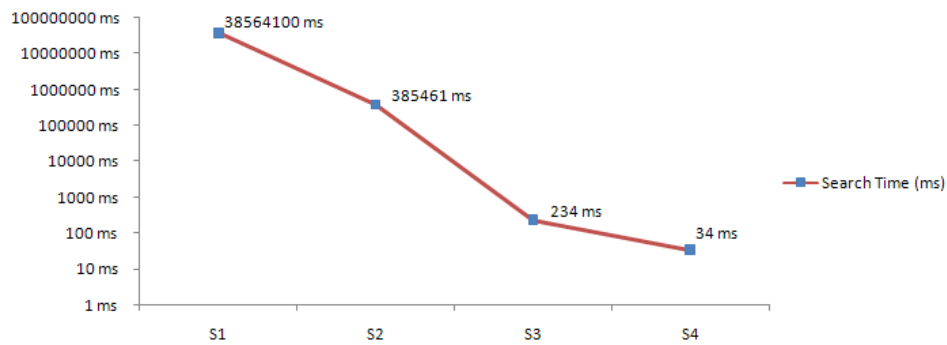


Figure 8. Search time reduction through S1-S4 on a logarithmic scale.

Figure 9 illustrates an example search performed during evaluation. An extracted body-patch of a person is used as input where the low-level color features of body area represents the query “*find persons wearing clothing of this color*”. The result is a set of face captures corresponding to people dressed similarly. Naturally, more advanced search capabilities will be required in a real-world security system.



Figure 9. Example search result of persons wearing clothing with similar color to that of a sample input.

The experimental results show that the proposed framework is successful at drastically reducing the amount of data that need to be stored in a digital library. Indeed when comparing S1 to S4, there is an 85,000 times reduction in the quantity of data stored, making it possible to store effectively a virtually infinite amount of events.

To assess the effectiveness of our S1-S2 transformation we performed a brief evaluation of event recall, and concluded that all the important events were correctly identified. We note that not all digital library deployments could apply this framework, or could obtain such drastic reductions in disk space. In this use case, the scenario lent itself very well to our framework.

5.2 PVR Use Case

Personal digital video recorders are emerging as the standard gadget to replace the classical VCR, and it is likely that in a few years time the vast majority of homes will contain a PVR. Since these are essentially entertainment devices designed to be used in the lean-back, enjoyment-orientated location of the living room, the need for the PVR personal digital library to be self-regulating and make best use of the limited storage space (a hundred standard-definition shows is typical) is essential. In order for this to be achievable, one needs transformation rules designed to work with broadcast TV content. In our previous research [5] we have developed digital libraries for digital video data recorded from the TV and one aspect of this research was the development of a number of digital video analysis and summarisation techniques, which can make for ideal transformation rules. In applying the framework to a PVR scenario, we can identify a number of transformation rules that can be suitable to such a deployment. These rules typically concern content reduction.

- Format Transformation:** The storage capacity of a DVR device depends on the format in which the content is saved on the disc. Format parameters such as image size, audio and video compression directly influence the size of stored video and also the audio-visual quality of the content. Most DVR appliances use the MPEG-2 encoding format since this is the predominant standard for digital TV broadcast (DVB). For convenience most DVR store directly the DVB bit stream without applying any format transformation to reduce disk space requirements. Transformation rules that decrease the frame size and the visual resolution can pack more content into a smaller bit-rate thus extending the storage-hours capacity. An experiment detailed in [21] shows that the visual quality of the content, as perceived by humans, can be maintained at satisfactory levels even for relatively low bit-rates.
- Advertisement removal Transformation Rule:** A study by Arbitron/Edison Media Research found that "nearly one in three (29%) [DVR owners] say the ability to skip TV ads is the main reason they record TV programs" [22]. Ad detection and skipping has become a standard feature for many personal recorders. However the commercial DVR available on the market allow users only to fast-forward or jump through commercial break but do not consider ad deletion in order to increase storage hours. Advertising densities vary with region, but typically are between 12-16 minutes per hour. Given a advertisement removal transformation rule, this would allow the saving of up to 26%. On an average DVR allowing 200 hours of storage this means 40 to 52 extra hours.
- News story Transformation Rule:** News journals are among the most watched TV programs and probably also widely recorded by DVR users. Typically most viewers are interested only in certain news stories or sections of the journal. Techniques for segmentation of news stories have proven successful in delivering personalized access to news content [5]. Equipping DVRs with content transformation rules based on such technology would allow removal, during or post recording, of news content that does not

match user's interests. The actual space savings introduced by such a transformation-rule is dependent on user's preferences.

- **Sports Transformation Rule:** As it happens with most recorded TV content the user interest in keeping a copies of old games decreases in time especially when storage space becomes a problem. However, often the case is that although pressured for storage space the user will want to at least save the best game moments. Algorithms that automatically detect important moments in the game are available for field sports [23] and can be employed to extract game highlights. For example, the method described in [23] allows for a reduction in stored game content of 65-74% while retaining 90% of top events for field sports such as soccer and rugby. In terms of storage requirements this means that the best moments can be compacted into a summary three times shorter than the game duration, which results in storage savings of about 60 minutes in the case of a soccer match and 55 minutes in the case of a rugby match.

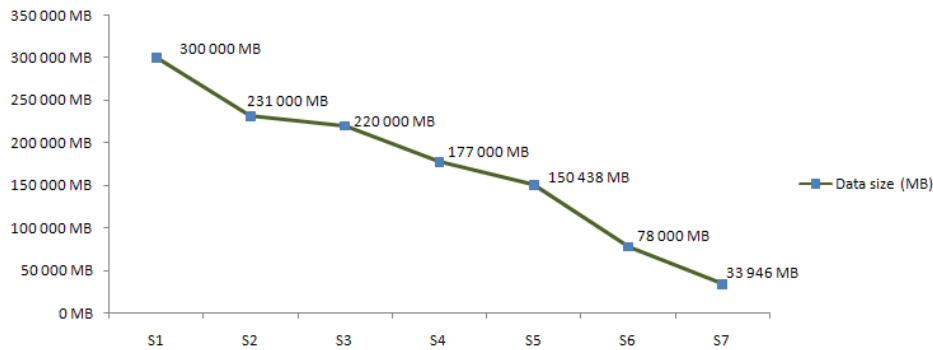


Figure 10. Disk space reduction on 200 hours of TV content.

Figure 10 illustrates the significant savings in storage space obtained for 200 hours of TV content encoded in MPEG-2 TS format (DVB stream encoding). This virtual test corpus amounting for 300 GB of disk space contains 50 hours of news (25%), 50 hours of field sports (25%), while rest of the content is made of other TV programs. As the content is trimmed down objects pass through as chain of states as detailed below:

- **Original state (S1)** before transformation of any objects.
- **Advertisement removal state (S2)** obtained after the ads have been deleted.
- **News story removal state (S3)** where news content videos are processed assuming that only half of the stories featured in a journal are of interests to the user, thus discarding the other half.
- **Sports summarization state (S4)** where the sports content objects are summarised into the top 90% events.
- **Video format conversion state 1 (S5)** obtained when video objects are reduced from the original MPEG-2 TS encoding format to MPEG-4 format without changing the image resolution.
- **Video format conversion state 2 (S6)** after downsizing the image resolution of S4 objects from the original 720x576 pixels to 360x288 pixels. This image resolution still allows a good visual quality despite the encoding low bit-rate.
- **Video format conversion state 3 (S7)** which shows a further reduction of the image resolution of S5 objects to 180x144 pixels. This is a relatively poor visual resolution and is used here only for illustrative purpose.

The video format conversion transformations S5 to S7 are possible within a single conversion step; however we use them independently for clarity.

5.3 HDM Use Case

A Human Digital Memory is a form of capturing and storing digitally daily events from an individual's life experiences. Over the years an individual can collect enormous amounts of data in this way and this is difficult to store and manage effectively without automating most tasks. For example the data collected by the author while wearing a Microsoft SenseCam for a period of almost two years [2] amounts for more than two million photos automatically captured. Due to the fact that this camera was configured to automatically take photos about every 50 seconds there can be a significant level of semantic redundancy within any given event, such as while the wearer is taking a half hour lunch there are 36 images showing the same food plate. Two or three photos would be enough to semantically summarise the event in the daily log. The camera also incorporates a set of sensors that will trigger a photo capture for changes of position, motion or environmental parameters. Out of the almost 4500 images captured daily some will be affected by problem such as motion blurring, defocus, semantic

redundancy within events, etc. Three transformation rules can be built in the HDM library in order to reduce an object size. Each object is initially represented in the library as event represented by a sequence of photos together with the associated metadata coming from the camera sensors.

- **Visual Quality Transformation Rule** removes from the sequence object all blurred, defocused and low contrast images.
- **Redundant sequence Transformation Rule** reduces the number of photos representing a given object when there is near duplication within the image sequence. Typical cases are when the camera wearer reads, works at the computer, drives car, etc.
- **Salient object Transformation Rule** designed to remove photos that do not appear to contain any salient foreground object, such as photos of uniform surfaces (close up on walls, white boards, items accidentally covering the camera objective).

Figure 11 shows the data reduction obtained when applying the above transformation rule for a two million photos HDM collection. The state S1 is the original state of the library after importing all events, while states S2, S3 and S4 corresponds to the above transformation rules, as follow:

- **Original state (S1)** is the state of an event object immediately after being imported in the library.
- **Quality check state (S2)** attained after blurred, defocused and/or poor contrasting images have been pruned out of the event.
- **Semantic redundancy state (S3)** obtained once near duplicate photos have been deleted from the event.
- **Salient object state (S4)** where meaningless (non-salient) photos have been discarded.

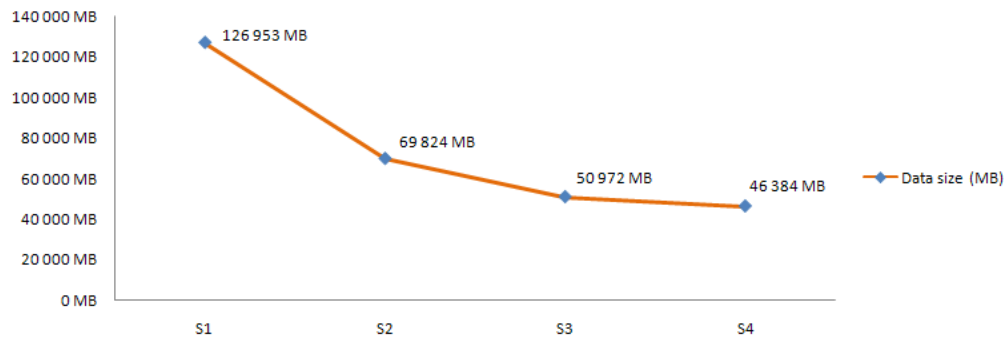


Figure 11. Data reduction via transformation rules on a two million photos HDM.

As the data size statistics show the set of transformations defined on the event objects help in discarding almost 2/3 of the data but the objects still retain the semantics of the event. This is a key challenge for maintaining a long term HDM on the limited storage available to individual users today.

6. Conclusions

Traditional Digital Libraries require human editorial control over the lifecycles of digital objects which imposes an inherent (human) overhead on the maintenance of these digital libraries. In this paper we presented that a revised framework is needed for digital libraries that takes the onus off the editor and allows the digital library to make decisions regarding digital object lifecycles, by employing a set of transformation rules that operate directly on the digital objects themselves. We have shown by means of three use cases how effective the library framework can be at optimising the nature of the digital objects contained in the digital library. The use cases employed in this work were natural choices in that they reflect the current professional digital library environment (security video), the current home user environment (PVR) and the next-generation, extremely challenging HDM environment. In all cases we have shown that our framework addresses the challenge mentioned in section 1, that of effectively self-managing vast quantities of heterogeneous data and avoiding the imposition of a significant user overhead in the data management process.

In this paper, our focus has been on implementing and evaluating the framework using a suite of transformation rules that are designed with the aim of improving the digital library content or reducing the storage space requirements of the content, yet retaining semantic meaning or the utility value of the content. Evaluating transformation rules that are concerned with non-reduction (e.g. transcoding content from one format to another, so as to maintain content in an up-to-date format) was not the subject of evaluation, but this is an inherent aspect of the framework we present. The proposed framework will not suit all digital libraries, and transformation rule specification will always be a domain-specific task. Still, for content rich digital libraries, especially libraries that require the storage of disk-space hungry multimedia data (such as video), our framework will prove very effective.

For future work we will continue to evaluate the proposed framework for different data types and address issues of speed and scale for organizing a lifetime of personal digital data. In addition we will examine how effectively we can employ the transformation rules at query-time to help in answering a user query, in effect, to support multi-modal access to digital objects in a device-agnostic manner by means of presentation transformations, where an object is transformed through a number of iterations until it reaches a suitable format for display on the presentation device. We expect this to be important work, especially for long-lasting or personal digital libraries because access devices will naturally change and it is important to support all likely access devices both today and into the future. Today already we can envisage users accessing a personal digital library using a myriad of devices, from mobile phones through to HD digital TVs. In future, we will likely be using new devices, like video walls and home surface displays. By the framework's use of transformation rules on digital objects it should lend itself well to an extension where it can also support query-time usage of transformation rules for multi-modal access and optimal presentation of objects.

7. Acknowledgments





Funded by the Norwegian Research Council & Science Foundation Ireland under the SFI Stokes lectureship programme. A previous short version of this paper appeared at Third IEEE International Conference on Digital Information Management (ICDIM), November 13-16, 2008, London, UK, Proceedings. IEEE 2008, ISBN 978-1-4244-2917-2.

8. References

- [1] Gemmell, J., Bell, G., Lueder, R., Drucker, S. and Wong, C. MyLifeBits: Fulfilling the Memex Vision. *In: Proceedings of the tenth ACM International conference on Multimedia*, 2002.
- [2] Gurrin, C., Byrne, D., O'Connor, N., Jones, G. and Smeaton, A. F. Architecture and Challenges of Maintaining a Large-scale, Context-aware Human Digital Memory. *VIE 2008*, Xi'an, China, 29 July - 1 August 2008.
- [3] Gorman, G. E. Giving way to Google. *Online Information Review*, Vol. 30 No. 2, 2006.
- [4] The Expanding Digital Universe -- A Forecast of World-Wide Information Growth through 2010, IDC WhitePaper, March 2007.
- [5] Smeaton, A.F., Gurrin, C., Lee, H., Mc Donald, K., Murphy, N., O'Connor, N., O'Sullivan, D., Smyth, B. and Wilson, D. The Fischlár-News-Stories System: Personalised Access to an Archive of TV News. *In: RIAO 2004 - Coupling Approaches, Coupling Media and Coupling Languages for Information Retrieval*. Avignon, France, 26-28 April 2004.
- [6] Johansen, D. A Distributed Approach to the Design of Applications. *In: IEEE Proceedings of the Fifth international Conference on Computing and information*. May 27 - 29, 1993.
- [7] Stonebraker, M. and Cetintemel, U "One size fits all": an idea whose time has come and gone. *In: Proc. of the 21st International Conference on Data Engineering*, 5-8 April 2005.
- [8] Smeaton, A. F., Gurrin, C., Lee, H. Interactive Searching and Browsing of Video Archives: Using Text and Using Image Matching. *In: Hammoud, Riad (Ed.), Interactive Video: Algorithms and Technologies*. 2006.
- [9] Smeaton A.F, Murphy N, O'Connor N, Marlow S, Lee H, Mc Donald K, Browne P and Ye J. The Fischlár Digital Video System: A Digital Library of Broadcast TV Programmes. *In: ACM+IEEE Joint Conference on Digital Libraries*, Roanoke, VA, 24-28 June 2001.
- [10] <http://www.gutenberg.org/>
- [11] Witten, I. H., Bainbridge, D. How to build a Digital Library. Morgan Kaufmann, San Francisco, CA 2003.
- [12] Beagrie, N. Plenty of Room at the Bottom? Personal Digital Libraries and Collections. *In: D-Lib Magazine* 11(6), 2005.
- [13] Bishop, A. P., Star, S. L. Social informatics for digital library use and infrastructure. *In: Annual review of Information Science and Technology*. NJ, 1996.

- [14] Mazurek, C. and Werla, M. Digital Object Lifecycle in dLibra Digital Library Framework. 8th DELOS Workshop on Future Digital Library Management Systems. 2005.
- [15] Payette, S. and Lagoze, C. "Flexible and Extensible Digital Object and Repository Architecture (FEDORA)," presented at Second European Conference on Research and Advanced Technology for Digital Libraries, Heraklion, Crete, 1998.
- [16] Lagoze, C. and Davis, J. R. 1995. Dienst: an architecture for distributed document libraries. *Communications of the ACM* 38, 4 (Apr. 1995), 47.
- [17] Curino, C., Moon, H.J. and Zaniolo, C. Managing the History of Metadata in Support for DB Archiving and Schema Evolution. *ER Workshops* 2008
- [18] Aarflot, T., Gurrin, C., Johansen, D. A Framework for Transient Objects in Digital Libraries. *In: ICDIM 2008 - IEEE Third International Conference on Digital Information Management*. London, U.K., 13-16 November 2008.
- [19] How are we being watched. BBC News Report, 3rd November 2006. Last Visited July 2008., http://news.bbc.co.uk/2/hi/uk_news/6110866.stm.
- [20] Hauptmann, A. Lessons for the Future from a Decade of Infomedia Video Analysis Research, Image and Video Retrieval. 4th International Conference of CIVR2005. Singapore, July, 2005.
- [21] Verscheure, O., Frossardl, P. and Hamdi, M. User-Oriented QoS Analysis in MPEG-2 Video Delivery, pages 305-315. *Real-Time Imaging* vol. 5, no. 5, October 1999.
- [22] Rose, B., Lenski, J. Internet and Multimedia 2005: The On-Demand Media Consumer. NJ, Januray, 2005. Retrieved March 2009 from <http://internet.arbitron.com/downloads/IM2005Study.pdf>.
- [23] Sadlier, D., O'Connor, N. Event detection in field sports video using audio-visual features and a support vector machine. *In: IEEE Transactions On Circuits And Systems For Video Technology*. vol. 15, no. 10, 1225-1233.

9. Author biographies

	<p>Tjalve Aarflot is a research assistant at the University of Tromsø developing and researching rich interfaces for personalized and context-aware media playback. He has previously researched and prototyped a distributed framework for context-aware image management at the CAIM project, with development of a web interface as a framework client and image management tool.</p> <p>Aarflot received his MSc in Computer Science from the University of Tromsø on improving the usefulness of multimedia archives by applying data reduction techniques.</p>
	<p>Cathal Gurrin is a lecturer in Information Retrieval at the School of Computing at Dublin City University, and also maintains a visiting scientist position at the University of Tromsø.</p> <p>Gurrin received his PhD from Dublin City University, in the area of large-scale web search algorithms. Currently his research is focused on Information Retrieval from digital libraries, HCI factors in accessing information on mobile devices and the capture and organisation of Human Digital Memories (HDMs). He is the a long-term maintainer of a SenseCam HDM, containing over three million location stamped photos and at present has a H-index of twelve.</p>
	<p>Sorin Sav is a postdoctoral researcher at Dublin City University. His research interests are in the area of image and video analysis, and multimedia retrieval. Sav received his PhD from Dublin City University.</p> <p>Sav has been involved on European research projects such as: Cost- 211 and Cost-292, SCHEMA - Network of Excellence in Content-Based Semantic Scene Analysis and Information Retrieval. Sav has eighteen publications in the area of visual image and video processing.</p>
	<p>Dag Johansen is a professor in the Department of Computer Science, University of Tromsø, Norway. His research interest is distributed computer systems, currently focusing on large-scale information access systems. Johansen is currently involved in the iAd project, a Norwegian Research Council supported Centre for Research-based Innovation.</p> <p>In 1997, Johansen co-founded D.A.G. In 2000, this company merged with Fast Search & Transfer. Johansen has been chief scientist in FAST since then, a company acquired by Microsoft in 2008.</p> <p>Johansen received his MSc and Doctor Sc degrees in Computer Science from the University of Tromsø.</p>