A Framework for Transient Objects in Digital Libraries

Tjalve Aarflot[1], Cathal Gurrin[1,2], Dag Johansen[1]
[1]*Dept. of Comp. Sci, University of Tromsø, Norway,* [2]*Centre for Digital Video Processing, Dublin City University, Ireland*

Abstract

Digital Libraries maintain information lifecycles for the digital objects contained therein. Typically these information lifecycles stipulate that a digital object is deleted at the end of its lifecycle. We propose a transient framework for digital libraries in which suitable objects are transformed through a state transition lifecycle thereby reducing the need to ever delete the digital object from the digital library, Transience works for certain media types and we describe and positively evaluate a prototype deployment in the domain of security video.

1. Introduction

In 2007 a key milestone was reached in the history of computing. The amount of information available surpassed the amount of available storage space [1]. In this paper we are concerned with how we can address the problem of the widening gap between the amount of digital information that is created and the amount that can be stored in a digital library. The obvious solution is to view a digital library as a pipe - as new content is added to a 'full' library, the oldest content is deleted - however, we propose a framework in which digital libraries examine the nature of the digital objects themselves and selectively transform the digital objects as required (to reduce the storage overhead of the object), thereby minimizing the need for data deletion.

This work presents a framework for digital libraries that supports the idea of transience applied to the digital objects. Some digital objects may be transient in nature, and because of this, we propose to automatically alter the representation of the objects in a digital library as they age or become less important. As a result, it becomes possible to dramatically increase the amount of data that can be stored in a fixed size digital library, for certain data types, such as video or sensor data. For example, in a PVR (Personal Video Recorder), a recorded football game need not be stored in its entirety, rather we can (over time) progressively reduce the content stored, eventually retaining only an automatically generated video highlight summary. Likewise, security video data can be transformed (over time) from raw video to short clips of notable content and eventually be represented by metadata of the key events and (perhaps) still images extracted from the video. Once again, a huge data reduction. In this paper, we build a digital library for security video, though we note that there are many scenarios in which our framework can be applied; for example a promising area is sensor networks, which will produce huge quantities of data being produced by abundant, low-cost sensors.

2. Background and motivation

In 2008, there are a billion digital cameras or camera phones capturing data [1], there are 900 million personal computers and over half a billion audio players. Looking forward two years (by 2010), it is expected that the installed base of such devices will be 50% larger. Much of this information will be stored in digital libraries and how we index, search and access this information in personal and shared digital libraries is a subject of ongoing research.

The storage space problem for digital libraries will eventually affect us in our everyday lives, and already does to some extent. While storage of text and to some degree audio data are less likely to be affected by this problem, storage of sensor output, image data and video data are more likely to be affected. Take a 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. A similar situation occurs with security video content where the captured content is typically stored only for a short period of time.

Motivation for this work comes from the authors' experience of developing large-scale digital libraries for both digital video [2] and pioneering distributed sensor networks [3]. Our experience suggests that an ad-hoc information lifecycle management of digital objects is not sufficient. By viewing objects in a digital library as being transient (can be transformed from one representation to another) in nature and not fixed, we can address the problem of maintaining

large digital libraries for storage space hungry multimedia data. We now identify background research in the areas of digital libraries, digital objects, and information lifecycle management, digital video libraries, before discussing our new digital library framework in section 3 and our implementation and results in section 4.

2.1 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' [4]. While traditionally digital libraries have been large digital representations of conventional libraries, personal digital libraries represent an alternative deployment that is gaining increasing attention, due to the fact that people are now gathering increasing amounts of personal digital information, whether videos, photos, emails, and so on [5]. One of the key research points for digital libraries is by the phenomenon of Googlisation, where search technologies are applied to digital libraries to support information access [6].

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 objects stored in the digital libraries.
- Metadata concerning these digital objects.
- Software to access the digital objects in the 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 for multimedia or sensor network digital libraries, as the amount of data produced can, at some point, surpass the amount of storage space available. Clearly, a framework within which digital libraries can address the storage space issue needs to be considered, and this is the focus of this paper.

2.1.1 Digital Objects

Digital objects are the key underlying data stored in digital libraries. Digital objects typically have been fixed and permanent (non-transient) with the focus of the digital library on maintaining an archive and supporting user access. The concept of a fixed and permanent digital object is based on the concept of a physical book from conventional libraries. However, an advantage of digital objects in digital libraries is that they are malleable, mutable and mobile [7] and hence it is not always necessary to consider them to be fixed and permanent. For example, some digital objects (e.g. videos, sensor streams) can be so large that they will need to be either replaced by smaller versions or deleted from the library. Our contention is that such digital objects (not only the metadata), can over time be modified and even deleted and replaced, yet still remain in the digital library and accessible by the end user. Hence, we consider such objects to be transient in nature, and we can define a transient digital object to be *a digital object within a digital library that can alter state and still remain published within the library.*

Exactly how a digital object gets modified or deleted is governed by the information lifecycle of the digital object. Information Lifecycle Management (ILM) is the practice of applying certain policies to the effective management of information throughout its useful life. It typically includes every phase of a digital object, from its creation to its deletion [8], and in some cases particular attention is paid to the storage and preservation of digital objects in a lifecycle model [9]. However most of the previous research has focused on lifecycle management of digital objects in order to maintain long-term availability and at the end of the lifecycle, the objects are simply deleted. In our framework, we also envisage that every digital object has an information lifecycle, from object pre-import/create, through object import, object publish and ending with eventual object delete (if necessary). However, we include in this information lifecycle a revised publish phase of the lifecycle, the purpose of which is to transform the transient digital objects from one representation to another (more storage space efficient) representation.

Our conjecture is that applying a transience property to digital objects is going to become essential in some domains (such as video, sensor network libraries, or lifelog libraries) to allow the digital library to expand to include new digital objects, even though the capacity of the library would conventionally suggest otherwise.

2.2 Digital Video Libraries

As the volume of digital video data in existence constantly increases, the resulting, vast archives of professional video content and UCC (User Created Content) are presenting an opportunity for the development of digital libraries of video information (both personal and WWW). Content-based video retrieval system development was initially led by academic research such as the Informedia Digital Video Library [10] from Carnegie Mellon and the Físchlár Digital Video Suite [2] from Dublin City University. Both of these systems operated over thousands of hours of content. Now however, digital video search has become an everyday WWW phenomenon, with millions of items of digital video

being indexed by the major WWW search engines and video upload sites. The Googlisation of such archives goes some of the way towards managing the content; however one important aspect of these archives is how much digital content they can contain. It is not economically possible to simply keep adding digital objects into the digital video library. If we examine the nature of digital video objects themselves, we can cleverly convert a large digital object (e.g. a TV show) into a form that is less disk-space hungry, thereby allowing a digital library to grow beyond the assumed limitations of disk space vs. raw video storage.

2.2.2 Digital Video Objects

Upon first examination of digital video content, one would assume that the digital video object is a fixed and permanent object in a library, which is an ideal scenario, but this view will increasingly be challenged as the volume of content in the libraries grows and the available storage fails to keep pace. In our past experiences of building the Físchlár digital video library we maintained an archive with a maximum number of about 300 programs, due to a limit on disk space. A naïve solution would be to simply transcode the video objects to save diskspace, but over-transcoding reduces quality, is limited and can affect the user experience. If we could assume a transience property for the video objects then applying video reduction or summarisation techniques could vastly increase the capacity of the Físchlár digital library. For example, we could remove advertisements or summarise content such as sports or news shows. Exactly how this is performed is dependent on the ILM of the digital objects.

3. A Framework for Transient Digital Object Transformation and Reduction

We have developed a framework for a digital library where objects are transient instead of being static/fixed. Our goal in doing this is that we wanted to allow for dynamic and optimal use of digital library resources; this required the examination of the lifecycle of the digital objects.

3.1 A Revised Digital Object Lifecycle

The information lifecycle of a digital object identifies a number of states through which the object progresses through its lifecycle. For example, Mazurek and Werla identify the conventional four state lifecycle from pre-import, through insertion, publishing and finally deletion [8], which is also the basis of our work. Such objects 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 through import, publish and finally delete.

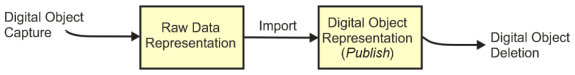

Figure 1. Conventional Digital Object Lifecycle

As stated, it is our contention that in digital libraries, some objects may need to be transient in nature; therefore a revised 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 transient digital objects, rather than be maintained in a fixed *publish* state, should enter an statefull cycle where each cycle results in a revised digital form of the digital object, until the digital object is eventually minimally represented in the library, or deleted. Figure 2 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.

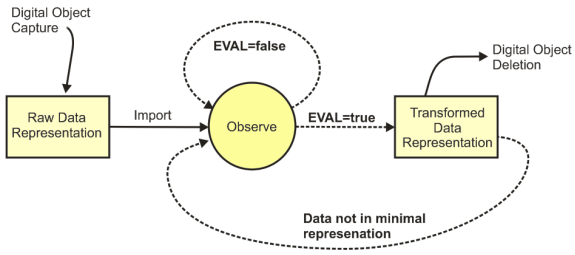

Figure 2. A Statefull Lifecycle of Transient Digital Objects

In our framework, a digital object is *observed* when it is 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 until the object is transformed to a minimal representation or is deleted from the library. The tine at which an object is transformed is dependent on a set of transformation rules that are manually generated rules that govern the process.

The key factor in this framework and lifecycle is that the digital object, once published, is not persistent in the published state, rather the object undergoes a sequence of transformations (typically data reductions), the necessity for which is decided by the digital library itself. In addition, the nature of the transformation rules need 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 takes place at night and has no movement events.

3.2 Architecture

In a conventional digital library, objects are imported, metadata generated and published through a user interface. The framework we propose differs from a conventional digital library in that it contains an *Observer* and a *Transformation Engine* which control the internal lifecycle of the digital objects in the library. The architecture for our digital library framework is shown in Figure 3.

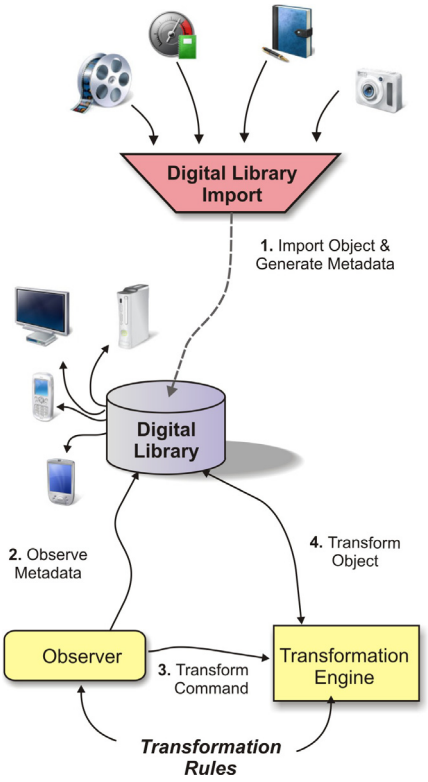

Figure 3. Digital Library Framework supporting Transient Digital Objects and Object Transformations

In a typical import-to-publish usage scenario for a digital library based on our proposed framework, the following describes the processes involved:

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.

There are a number of core components of this digital library framework that we now describe.

3.2.1 Digital Library Import

Importing objects into our proposed digital library framework is not different from importing into a conventional digital library, except that in addition to generating conventional metadata, such as creation date, ID, and indexable content, this metadata must also contain the current transform-state of the object, initiated to a default state upon object creation. This metadata will trigger digital object transformation.

3.2.2 Observer

The Observer component observes metadata of objects in the digital library and triggers a transform command upon positive evaluation of the transform-state of the object. Evaluation is performed by comparing object metadata against transformation rules. The observer triggers an event for the Transformation Engine, which will utilize the transformation rules to alter object from transform state S to transform state S'. By integrating external sources of information, the trigger event could also be generated by requirements such as reduced storage capacity or system resources.

3.2.3 Transformation Rules

In our proposed framework, every digital library type (e.g. multiple-media, video, image, text etc.) would require a bespoke set of transformation rules that are specific to the requirements of the digital library, the digital object types and the transform-state of the objects. A transformation rule will contain:

- Digital object type and transform state identifiers.
- Digital object metadata trigger requirements that describe what metadata element values are required to trigger the transformation.
- Digital object transformation specification which will define exactly how to transform the transient digital object from one transform-state into another transform-state. These specifications may be as simple as a 'delete' command for aged digital objects or could be algorithms that actively examine the digital object and create a new object and/or a new representation of the transient object. Some transformations may only alter the metadata of the object, for instance by increasing some iterative value periodically (e.g. implement an ageing of the object).

The transformation rules are used by both the observer to trigger the transformation, and by the

transformation engine to actually apply the transformation.

3.2.4 Transformation Engine

The Transformation Engine receives a transform command from the observer identifying objects that are triggered for transformation. The transformation engine will read the object from the digital library, transform it using the transformation rules, and write it back to the library. This set of operations should be performed atomically as a single transaction, so as to maintain the data integrity of the digital library. As shown in both Figure 2, the transformed object is returned to the publish lifecycle state, though the (internal) transform-state of the digital object is different, since it has undergone transformation.

4. Details of Implementation

When implementing our framework we chose the domain of surveillance video. At the end of 2006 in Britain alone, there were estimated 4.2m CCTV cameras – one for every 14 people [11]. If all cameras are recording 24 hours a day and if no data is discarded, the CCTV's of Britain would generate roughly 42 petabytes of data each day, which obviously results in most of this video having a very short lifecycle. However, 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, this is even beyond the scope of a manual process.

Therefore, security video makes an ideal subject area for prototyping our framework, in which the video data is reduced through a number of cycles from raw video objects to a sequence of important events, which can be minimally and permanently stored in a digital library and not require deletion, thereby extending the capacity of the library greatly.

4.1 Environment and Interface

For the security video scenario, we recreated a small-scale environment in which security cameras constantly monitor an area. To achieve this we equipped a multi-person office location with a digital video camera and a PIR sensor overlooking the sole access point (door) to the office, as in Figure 4. The rationale for having a PIR sensor was to support low-cost and very effective visual analysis (red flash) of people entering and leaving the room, and this could be picked-up by the camera that points directly at the door. Without using the PIR sensor and relying purely on visual analysis, it is possible that some events would be missed and false positives injected. The PIR sensor helps to ensure that we do not miss any security events by always flashing red when someone passes through the door. In our implementation, the camera captures people entering and leaving the room, as well as identifying when the PIR sensor indicates a security event.

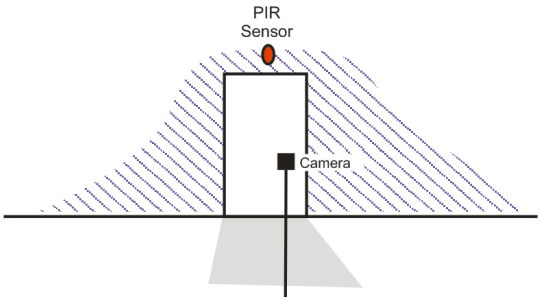

Figure 4. Event Detection Set-up of Prototype

The video stream captured from the camera was segmented into one-hour segments (*capture* state) and imported into the digital library (*import* state) as a sequence of digital objects (one-hour long segments), each with metadata for the date & time of capture and a set of keyframes. When in the *publish* state, the digital objects are transient objects and can be transformed from one representation to another and at the end of the lifecycle, although it is probably not necessary given our framework, the digital objects can eventually be deleted (*delete* state) from the library. For this prototype, we have developed a web interface supporting multi-modal access via web browsers in a desktop environment and via mobile devices for ubiquitous access to the archive. The web interface is based on our extensive experience of developing prototype digital video libraries [12] and the security events for one morning is shown in Figure 5, in minimal *publish* state (S4).

As can be seen from Figure 5, any day can be chosen using the calendar control at the top left of the screen. In addition to the calendar, the weekly overview control shows the current week's security events, color coded by the automatically assigned importance of the event. The main part of the interface is taken up by a listing of the events (digital objects in state S2) of the chosen day. On top of the main view is the date of the selected day, the number of detected events for this day, and a bar showing all the events of the day on a timeline, which allows for rapid visualization and access to any part of the day. Emphasis is given to the most important events, which are shown using larger keyframes (images chosen from the video that include the person entering or leaving the room) and event metadata. Colors are employed to give emphasis to digital objects based on their importance. In this scenario, events with unknown persons are the most important. The events for the day are shown in chronological order and divided into sections per hour. Where many events are aggregated into one color, as in the calendar and the weekly overview, the color of the object of most importance is shown.

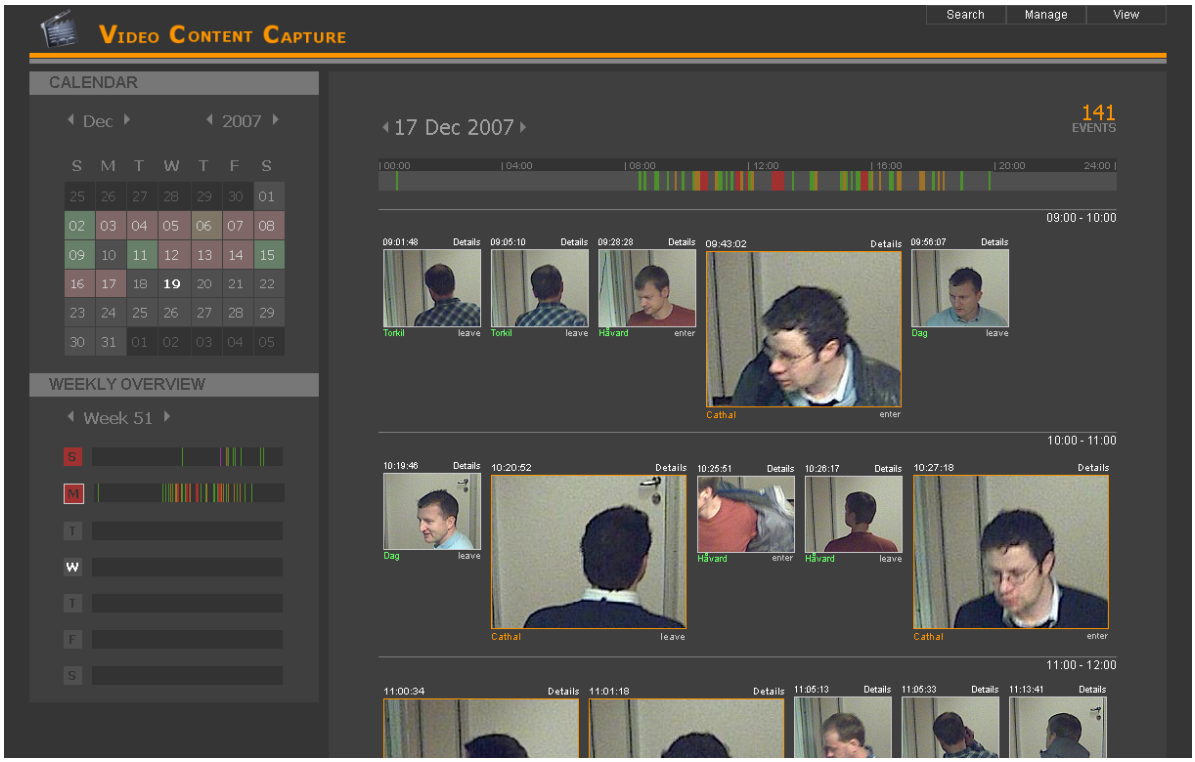

Figure 5. Prototype Digital Video Library Interface in the minimal 'publish' state (S4), one morning shown

4.2 Prototype Lifecycle of Digital Video Objects

Aside from the conventional lifecycle states of a digital object (*capture*, *import, publish* and *delete*), when in the *publish* state, the digital object enters its transform lifecycle. At import time the digital library has segmented the video data into one-hour segments (the transient digital objects) which remain in the *publish* state. Transformation rules can be applied to these transient digital objects, based on automatic visual analysis of the content, the result of which is to reduce the amount of redundancy in the data. In our prototype, over the lifetime of the digital objects, they (and their metadata) will be altered three times, each time reducing redundancy until a minimal representation of the important content is all that remains in the library for any individual digital video object. The set of transformation rules will transform published objects, through four transform-states, from an object containing one-hour of video (S1), to an object containing just the face and body of a person walking through the door (S4). The transform-states of a published object are:

- *Raw video state (S1)*, where the raw one-hour segments of video are stored.
- *Event video state (S2)*, where the video object has been transformed from a one-hour segment (S1) into a number of short video event objects representing each important security event (S2).
- *Event keyframe state (S3),* where an object from S2 has been transformed into a keyframe image object, automatically selected to show the individual walking through the door (S3).
- *Face and body state (S4)*, where the keyframe image object (S3) has been transformed into a cropped image of the face and body (S4), as shown in Figure 5.

Given this description of the transform states of the transient digital objects, it is necessary to examine how the transform rules govern transient digital object transformation from S1 to S4.

4.3 Transformation Rules

The transformation rules that govern the lifecycle of the digital objects are triggered one after another in our prototype implementation. As stated, the digital objects are taken from their initial state S1, through three transformations, to a final state S4, with each transformation working on the digital object generated in the preceding state. For each transformation rule we have developed image analysis software that operates on the digital object to produce a new representation of the object, or a new set of objects entirely. In this way, objects propagate through the internal states within the digital library. This is the essence of the transience property of digital objects in our framework. There are three transformation rules, which are briefly described below:

- Video Event Transformation Rule

- Event Keyframe Transformation Rule
- Face & Body Transformation Rule

4.2.1 Video Event Transformation Rule

An event in our prototype is defined as the situation when someone walks through the door (in or out), as captured by the PIR sensor. Hence the first transformation rule observes a digital object in state S1 and transforms the one-hour clip into a number of smaller digital objects representing each event during this one-hour period. On average, the digital objects in state S2 contains about 4.5 seconds of video.

4.2.2 Event Keyframe Transformation Rule

The digital objects in state S2 can be further processed to transform them into smaller (disk space) objects by applying the event keyframe rule onto the digital object in state S2, producing one object in S3. The keyframe of an event is defined to be the image where the person going through the door is in the middle of the doorframe. This is detected by analyzing the pixels that makes up the doorstep of the room, in addition to the area outside and inside of the door step. By comparing values from this analysis between the images that makes up an event, the best keyframe is found. This process also detects whether the person is entering or leaving the office, and added to the metadata of the S3 object.

4.2.3 Face & Body Transformation Rule

This is the last of the transformation rules, which crops the keyframe (from S3) to a small, rectangular image containing the head of the person going through the door (S4). To detect the area that should be cropped, pixels inside the door frame are compared to default values. Two small images surrounding the head and upper body are then extracted. In addition, the height of the person is calculated and stored as metadata. This rule, being purely visual in nature achieves a recall of 0.84; this could be improved with additional visual processing. The transient digital object has now been transformed from state S3 to S4.

4.3 Observations on the Framework

The rationale for developing our new framework for digital libraries was so as to support very long term storage of a representation of each imported digital object. 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.

The 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 1280x720 resolution. At reasonable compression rates, one second of video requires 635kBs of disk space. On a per hour basis, we could extract up to 2.18 GB of keyframes at 4 fps.

The amount of important data (represented by important events) in our imported video content is highly dynamic, varying based on 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.

Recall the four different stages of published objects in the digital library:

- S1. Raw video in one-hour segments
- S2. Short Event clip
- S3. Keyframe of person in door frame
- S4. Face and body of person in event

We can compare the storage capacity the objects require in each of the four stages for a given day as shown in Figure 6.

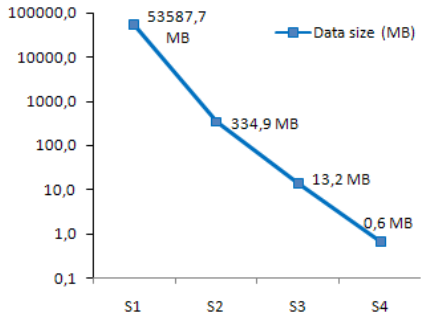

Figure 6. Disk Space Reduction per day through S1-S4

Figure 6 shows disk usage by the digital objects at their different stages (S1-S4), for one day. Note that 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. Note that the transformation from S3 to S4 does result in only 84% of the events being represented by an accurate picture; however this is only a feature of the face detection algorithm employed, and an improved algorithm will tend towards 100% accuracy.

Another benefit of this reduction in the data stored is the support (through the interface) for rapid visualization and searching for an important event in the archive. Even more significant is the time taken for visual analysis software tools to examine the content; for example, when seeking a particular face in the video data, like the face of a known criminal. 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 7, but the figures are algorithm dependent.

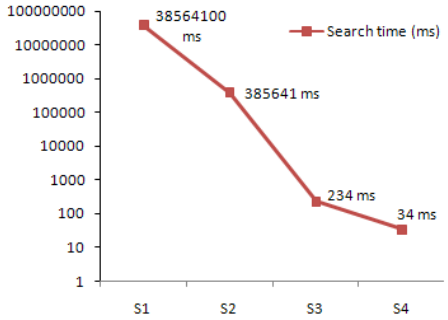

Figure 7. Search Time Reduction through S1-S4

The results from the prototype implementation show that our framework is successful at drastically reducing the amount of data that can be stored in a digital library; indeed when comparing S1 to S4, there is a 85,000 times reduction in the quantity of data stored, making it possible to store effectively an almost infinite representation of data capture. Improved algorithms will convert the 84% precision of the S3-S4 transformation to close to 100%.

To verify the effectiveness of our S1-S2 transformation we performed brief evaluation of event recall, and concluded that all the important events were identified by our visual analysis on a given day; most probably thanks to the PIR sensor which was very effective and easy to identify in software when it was flashing red indicating a person entering or leaving the door.

We note that not all digital library deployments could apply our framework, or could see such drastic reductions in disk space. In our prototype, the scenario lent itself very well to our framework, with a limited number of events during a day.

5. Conclusions and Recommendations

In this paper we propose, describe and positively evaluate a framework for digital libraries in which published objects are transformed through a transformation cycle thereby reducing the need to ever delete the digital object from the digital library for certain media types; hence we refer to the objects as transient digital objects. Transformation is triggered and performed by a dynamically loaded rule set. A typical digital object will follow a lifecycle with multi-layered reduction while remaining in the publish state. Consequently, in a real-world implementation, any transformation rules and visual processing techniques could be applied to our scenario. The distribution of objects in each transform-state (S1-S4) would be dependent on the transformation rules and the overall status of the digital library.

Our anticipation is that the framework we propose will be employed to reduce redundancy, based on the principle that certain types of data gets less important over time. At the end of the reduction stage of the digital object, it can be maintained permanently (most likely because the storage requirements of the reduced state should be minimal) in the library, or eventually end the lifecycle by being deleted.

Our framework will not suit all digital libraries and transformation rule specification will always be a domain-specific task, but for the new emerging area of personal digital libraries, especially for disk-space hungry multimedia data (such as video), our framework could prove effective. Recent progress in the area of Human Digital Memories (HDMs) [13] whereby people are now in a position to record all aspects of their life in large personal archives is a likely application of our framework. In this area people store vast quantities of personal information, even to the point of visually capturing all that they see, which naturally challenges data storage techniques for conventional digital libraries.

10. References

[1] The Expanding Digital Universe -- A Forecast of World-Wide Information Growth through 2010, IDC WhitePaper, March 2007.

[2] A.F. Smeaton, C. Gurrin, H. Lee, K. Mc Donald, N. Murphy, N. O'Connor, D. O'Sullivan, B. Smyth B and D Wilson, "The Físchlár-News-Stories System: Personalised Access to an Archive of TV News.", RIAO 2004, Avignon, France, 26-28 April 2004.

[3] Johansen, D. 1993. A Distributed Approach to the Design of Applications. In *Proceedings of the Fifth international Conference on Computing and information* (May 27 - 29, 1993).

[4] I.H. Witten, D. Bainbridge, *How to Build a Digital Library*, Morgan Kaufmann, San Francisco, CA, 2003.

[5] N. Beagrie, "Plenty of Room at the Bottom? Personal Digital Libraries and Collections". D-Lib Magazine, 11(6), 2005.

[6] G.E. Gorman, "Giving way to Google", Online Information Review, Vol. 30 No. 2, 2006

[7] A. P. Bishop, S.L. Star, "Social informatics for digital library use and infrastructure". In. Annual Review of Information Science and Technology, NJ, 1996.

[8] C. Mazurek, M. Werla, "Digital Object Lifecycle in dLibra Digital Library Framework", 8th DELOS Workshop on Future Digital Library Management Systems, 2005.

[9] G. Hodge, "Best Practices for Digital Archiving: An Information Life Cycle Approach", D-Lib Magazine, January, 2000.

[10] A. Hauptmann, "Lessons for the Future from a Decade of Informedia Video Analysis Research, Image and Video Retrieval", 4th International Conference, CIVR 2005, Singapore, July, 2005.

[11] "How we are being watched", BBC News Report, 3rd November 2006. Last Visted July 2008, http://news.bbc.co.uk/2/hi/uk_news/6110866.stm,

[12] A.F. Smeaton, C. Gurrin, H. Lee. "Interactive Searching and Browsing of Video Archives: Using Text and Using Image Matching", In: Hammoud, Riad (Ed.), Interactive Video: Algorithms and Technologies, 2006.

[13] G Bell, J Gemmell, "A Digital Life", Scientific American, March 2007.