BBC

Research White Paper

WHP 155

September 2007

File-based Production: Making It Work In Practice

Stuart Cunningham and Philip de Nier

BRITISH BROADCASTING CORPORATION

White Paper WHP 155

File-based Production: Making It Work In Practice

Stuart Cunningham and Philip de Nier

Abstract

Many organisations are moving from video tape based television production to file-based production. A number of difficulties arise during this migration including unexpected costs, workflow complications and lack of equipment interoperability. Such difficulties can be solved in practice using novel applications of low-cost IT equipment, software tools and industry standards for file formats.

This paper describes a number of problems of file-based production which arise in practice and demonstrates the techniques used to solve them using real-world examples from BBC productions.

The application of commodity PCs to achieve flexible real-time software encoding in a multi-camera TV studio is shown. Software tools to overcome incompatibilities between different MXF implementations are described. The design and format of MXF files for long term programme archival on LTO-3 data tape is explained. We show how file-based systems can be interfaced to legacy broadcast equipment. Finally, the problem of migrating from SD to HD production is shown to be mitigated by the HD capable nature of many IT components.

This document was originally published in the Proceedings of the International Broadcasting Convention, September 2007.

Additional key words: Ingex, Avid, AAF, archive, libMXF, FFmpeg, open source

White Papers are distributed freely on request.

Authorisation of the Head of Research is required for publication.

FILE-BASED PRODUCTION: MAKING IT WORK IN PRACTICE

Stuart Cunningham and Philip de Nier

BBC Research, UK

ABSTRACT

Many organisations are moving from video tape based television production to file-based production. A number of difficulties arise during this migration including unexpected costs, workflow complications and lack of equipment interoperability. Such difficulties can be solved in practice using novel applications of low-cost IT equipment, software tools and industry standards for file formats.

This paper describes a number of problems of file-based production which arise in practice and demonstrates the techniques used to solve them using real-world examples from BBC productions.

The application of commodity PCs to achieve flexible real-time software encoding in a multi-camera TV studio is shown. Software tools to overcome incompatibilities between different MXF implementations are described. The design and format of MXF files for long term programme archival on LTO-3 data tape is explained. We show how file-based systems can be interfaced to legacy broadcast equipment. Finally, the problem of migrating from SD to HD production is shown to be mitigated by the HD capable nature of many IT components.

INTRODUCTION

File-based production offers potentially significant savings in many areas of programme production, particularly for time spent on non-creative tasks. While file-based production solutions are available in the marketplace, many productions do not have the necessary budget. Where production budgets are low, an alternative approach is to use commodity IT hardware and software where possible throughout the production chain.

In practice, a number of difficulties arise when attempting to interoperate commodity IT equipment with existing broadcast production equipment. Equipment may not use standardised file formats, making interoperability challenging. Where equipment does support a standard format such as MXF (1), implementation limitations can prevent the full range of interchanges anticipated.

This paper describes a number of areas where difficulties have arisen and been resolved or mitigated by applying novel combinations of software tools and commodity IT hardware. The Ingex (2) suite of software applications and libraries was written by BBC Research to solve practical problems encountered by BBC productions. Combined with commodity PC hardware and SDI I/O adapter cards, Ingex software has been used in multi-camera studio productions, fast turn around sports editing and to transfer a significant archive of legacy video tapes into accessible file-based formats.

STUDIO FILE-BASED ACQUISITION

Acquisition in a studio environment frequently involves multi-camera operation. Typically four video streams are required to be recorded with corresponding audio and timecode data.

Commercial studio file-based acquisition products exist but can be beyond the budget of productions.

An early version of the Ingex studio recording system was first used for a BBC Childrens' programme called BAMZOOKi in 2005 (3)(4). Since then, Ingex has been developed to perform the more demanding tasks required by a BBC serial drama production.

The Ingex studio system, outlined in Figure 1, captures four studio SDI signals containing video and embedded audio and creates online and offline editing video formats in real-time. The online and offline formats are stored as MXF files on a commodity NAS server and are available immediately to Avid Media Composer editing workstations. A convenient method for loading the appropriate clips into an Avid Media Composer bin is provided through an AAF (5) file which is drag-and-dropped into an editor's bin.

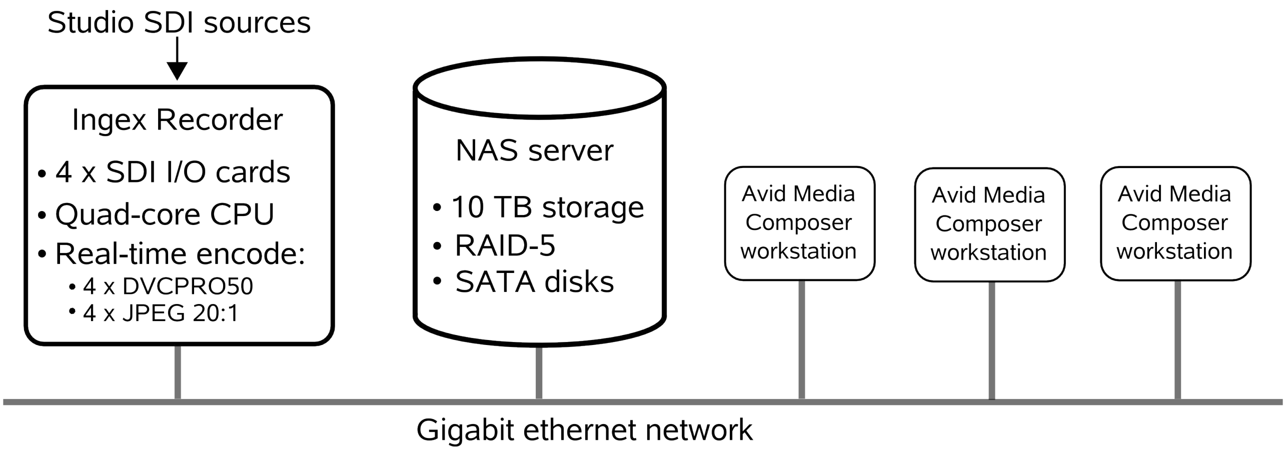

Figure 1 – Ingex system for studio recording and post-production

Real-time encoding and wrapping as MXF files

Commodity PC workstations capable of encoding four streams of SDI video to an online video quality became available in 2006. Computers with four CPU cores can encode four video streams to DVCPRO50 or JPEG 2:1 in real-time, with CPUs such as the quad-core Intel X5355 capable of encoding offline video, such as JPEG 20:1, in real-time in addition to online video for each input.

The real-time video encoding software used was provided by the FFmpeg (6) and libjpeg (7) open source software packages. For the JPEG formats, the Ingex mjpeg_compress software module was written to restructure the libjpeg created JPEG images into the form required by Avid Media Composer.

Wrapping a stream of video essence in MXF required an MXF software library capable of accepting a stream as input. The newly developed libMXF (8) software library was used to accept a stream of video or audio essence and directly write the MXF OP-Atom files required by Avid Media Composer. When the MXF files are written directly to a network drive which is also mounted by editing workstations, the MXF files become available for review or editing as soon as the clip has finished recording, enabling a real-time editing workflow.

Multi-camera metadata capture

The Ingex database stores the configuration of studio camera sources and microphone sources that are connected to the recording PC. Users configure this data using a web browser interface. When a recording is made the database records the start timecode and duration of each clip, which is also stored in the MXF files. When an editing workstation user is ready to load clips into their editor, they select a range of scenes or takes using the web browser search interface. The Ingex software creates an AAF file on the fly, representing not only all the clips chosen but also the multi-camera grouping of the four video sources. This AAF file is then drag-and-dropped into a bin in the Avid editor and the multi-camera group is immediately ready for use.

The newly written software implementing this functionality consisted of Perl CGI code for the web application and a C++ application for creating AAF files. The following open source components were used to complete the system; PostgreSQL for the clip metadata database, Apache for the web server and the AAF SDK (9) library for creating AAF files.

Network and storage considerations

The commodity NAS server was required to provide a level of reliability typically found in editing storage servers but at a much lower cost. Commodity SATA disk drives have been shown by large scale studies such as Schroeder (10) to be just as reliable as more expensive SCSI and Fibre Channel disk drives. The NAS server chosen consisted of 16 750GB SATA 300 disk drives configured for RAID 5 to provide an acceptable level of redundancy. After formatting with the XFS file system, the disk storage space available was 10 TB for a cost under €8,000.

Laboratory tests showed that a commodity gigabit Ethernet network was sufficient for supporting six editing workstations under real-world editing conditions. However, once deployed in the target post-production environment, network speeds between the NAS server and the editing workstations were found to be 200 Mbps for a CIFS network file transfer and was inadequate for multi-camera online editing. Testing using open source tools such as ttcp (11) revealed hardware or network driver problems in the editing workstations. Moving an editing workstation's gigabit network adapter onto a different PCI bus or replacing the network adapter was necessary to improve the performance to around 450 Mbps for a CIFS network file transfer. A commodity gigabit network switch was sufficient to complete the network installation.

Supporting legacy post-production equipment

File-based production may require later versions of software products than the versions encountered in typical post-production environments. We discovered several instances where old software products could not be immediately upgraded and therefore required workarounds to be implemented by the software creating the MXF files.

To interchange MXF files with early versions of Avid editing software we needed to add a number of dark metadata elements to the MXF file, otherwise the Avid software would not recognise it as an MXF file. The Avid extensions to the MXF file included an AAF-style meta-dictionary, which defined the classes, properties and types used in the header metadata, and an object directory, listing the file offsets of every KLV set in the MXF header. The meta-dictionary enables MXF file readers to correctly interpret header metadata, even when extensions are present. Although the primer pack defined in MXF contains a subset of the meta-dictionary information, it can be insufficient to correctly interpret all header metadata.

When creating an audio export for use with an old version of Pro Tools another interoperability issue arose. The libMXF software creates a SMPTE 330M (12) compliant UMID for the MXF Package UIDs. The Avid software preserves these Package UIDs when exporting OMF material for use in separate audio editors. When the Pro Tools version 5.3.1 software read the OMF file it reported “OMF1_ERR ERR:MobID not present in the file”. To resolve the incompatibility we changed the format of the MXF Package UIDs to resemble Avid UMID generation, using an old version of AAFUtils.cpp from the AAF SDK before it was modified for SMPTE 330M compliance.

EDITOR TO FILE-BASED CAMERA INTEROPERABILITY

To improve the workflow of BBC Sport editors during the 2006 FIFA World Cup, a need to transfer completed edits from an Avid editor to a Panasonic P2 camera arose. Material was captured on the P2 camera, copied to a laptop running Avid editing software, and when a completed edit was conformed on the laptop it was copied back to the P2 camera to be played out over its SDI interface.

Since the Avid editor and the P2 camera both used MXF OP-Atom files it appeared to be a straightforward task, but constraints in the P2 camera required a precise structure of MXF OP-Atom file to be followed to achieve interoperability. A custom software tool named TransferToP2 (13) was written to extract the raw DV or DVCPRO50 essence from the Avid MXF file and re-wrap it into a P2 compatible MXF file following these constraints:

- The first sample of essence (video or audio) must be positioned at file offset 0x8000
- The KLV BER encoded lengths must be 8 bytes for essence, 4 bytes for other KLVs
- The header must contain an IndexTableSegment (not required by OP-Atom)

The open nature of the MXF format, and the availability of software tools such as MXFDump (14) and libMXF, allowed us to investigate the interoperability problems and develop effective solutions. The information gathered in this process was documented and publicised via the Ingex website to encourage commercial adoption. Subsequently, commercial products have included this interoperability feature.

ARCHIVE AND PRESERVATION

In 2006 the BBC video tape archive contained around 380,000 tapes in the obsolete D3 video tape format. The D3 tapes included programs recorded in earlier video tape formats as far back as 1967. The BBC D3 transfer project, outlined in Figure 2, was created to preserve the content of the D3 tapes by creating a file-based video archive, putting an end to the previous practice of transferring video from one obsolete video tape format to another video tape format whenever old video tape formats became difficult to support.

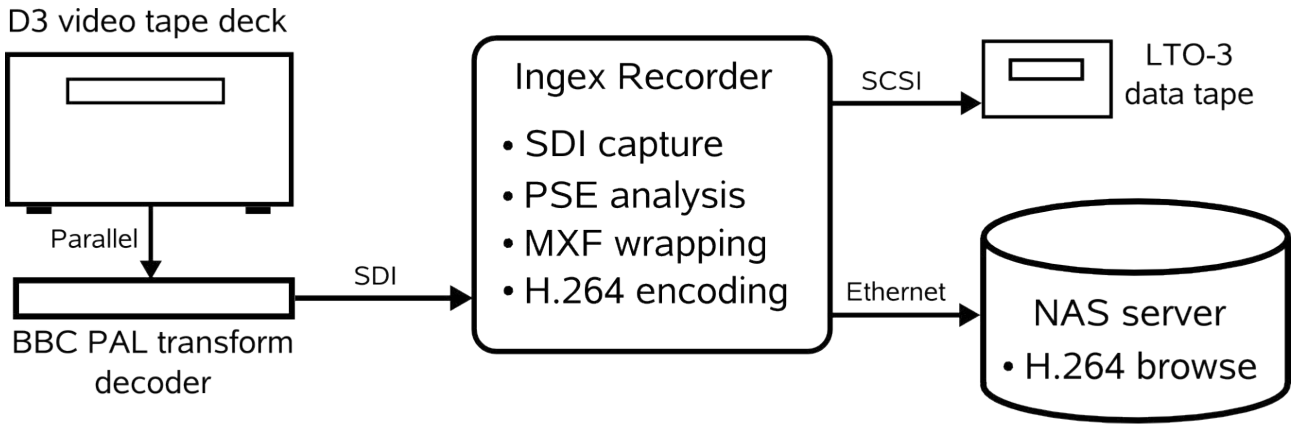

Figure 2 – D3 transfer system for transferring D3 video tapes to MXF files on data tape

The design of the file format to contain the BBC's entire D3 video archive had to fulfil the requirement of long term preservation. Once a programme is transferred from D3 video tape to a file, the expected lifetime of the file is several decades or longer. Although the file storage medium is likely to change, the file itself would not change as it is moved from a legacy file medium to the next file medium.

To help ensure video files would be readable in the distant future, open standards were chosen at all levels of the design. MXF OP-1A was chosen as the file container to store the video, audio and timecode data which would reproduce as closely as possible the information on the D3 video tape. Holding the entire archive of MXF files permanently on hard disks was too expensive so the open standard LTO-3 data tape (15) was chosen as the initial file storage medium.

The OP-1A MXF file contained a sequence of content packages, one for each frame, containing video, audio and timecode, in the following formats:

- A video item of SMPTE 384M uncompressed 4:2:2 video at 8 bits per sample in UYVY format
- An audio item with 4 tracks of SMPTE 382M uncompressed PCM audio at 48kHz and 24 bits per sample
- A system item containing an array of SMPTE 12M timecodes representing the VITC and LTC timecodes read off the video tape

In addition to the essence, the OP-1A MXF file contained the following metadata in the MXF Header:

- Data items from the BBC Infax archive database record for the TV programme, identified by SMPTE Universal Labels registered under the BBC node of SMPTE metadata dictionary
- An array of D3 error records to facilitate future analysis, preservation and restoration
- An array of Photosensitive Epilepsy (PSE) failures if any
- The barcode of the LTO-3 data tape the MXF file is stored on

In designing the way MXF video files were to be stored on data tape, consideration was given to the way archived tapes were to be used. LTO-3 tapes can store 400GB of uncompressed data, corresponding to about 10 uncompressed archive programmes. They are stored on shelves in an archive vault initially, and intended to be retrieved by human operator rather than a robot. A standalone Quality Check (QC) station is used to check the contents of the LTO-3 tapes after transfer, and to make copies of programmes as either video tape copies or file copies. The format of files on the data tape had the following attributes:

- POSIX.1-2001 archive format (16) (also known as pax Interchange format), a superset of the tar format which overcomes the 8 GiB limitation of tar format, to ensure future access to programmes
- No spanning of tapes, only complete MXF files were stored on any single tape
- No tape level compression, to mitigate data loss in the rare event of media errors
- File marks between programmes, to aid identification and improve seeking
- Tape block size of 256 KiB, to optimise LTO-3 data transfer speeds
- An ASCII text index file at the beginning of tape listing each MXF file with its file size and a detailed text description of the tape and MXF structure. A hard disk cache was required to support the construction of the index file.

To make the most of the D3 video tape to MXF file transfer process, several other processes were run during the MXF file creation stage. A low bitrate 1.2 Mbps browse video consisting of H.264 video and AAC audio encoding was performed in real-time using the open source software libraries x264 (17) and FAAC (18). This browse copy was written over gigabit Ethernet to a NAS server to be maintained as online storage and used for archive searching. PSE analysis was performed in software in real-time using a proprietary library and any failures were recorded in the MXF file and in an XHTML report. Current developments include processing for video shot change detection and video feature extraction to enhance archive search capabilities.

PLAYOUT OF FILES TO BROADCAST EQUIPMENT

Despite the progress of file-based production a significant requirement remains to interface file-based media to broadcast equipment using interfaces such as SDI. We have developed a software application named IngexPlayer, a part of the Ingex project, to playout file-based material over SDI using an SDI I/O card in a commodity PC. The user interface to IngexPlayer has been designed to accommodate users from post-production, from a studio gallery environment and those in an archive environment.

IngexPlayer supports a USB jog-shuttle controller (see Figure 3) providing a familiar interface to those users experienced with video tape machines. However, since the video is stored as a file, users have been surprised at the speed at which video can be randomly accessed. In addition, there is no "tearing" of video during fast forward or rewind operations.

IngexPlayer can display its output on a computer monitor, SDI output using an SDI I/O card, or both. When output is over an SDI signal, video is accompanied by up to 8 tracks of embedded audio and VITC and LTC timecodes, allowing a file-based media asset to be cloned onto video tape for compatibility with existing broadcast equipment.

IngexPlayer can decode and playout in real-time the following video formats:

- DV (25 Mbps), DVCPRO50 and DVCPROHD
- Uncompressed SD and uncompressed HD
- IMX 30, 40 and 50

Figure 3 – USB jog-shuttle controller used by IngexPlayer

Use in a studio environment

Using file-based capture in a studio environment requires the provision of a file playout facility to provide the user with feedback that the correct video was recorded. IngexPlayer was used to playback the recorded MXF files containing DVCPRO50 video onto SDI monitors in the studio gallery. A quad-split feature was also provided which plays four sets of MXF files representing four video sources, combining them into a single frame. The quad-split feature requires real-time software decoding of four DVCPRO50 streams, provided by the FFmpeg software library, and filtering of the down-sampled frames to avoid aliasing and interlace artifacts.

Use in an archive environment

As part of the BBC D3 transfer project described above, a Quality Check (QC) step was required once an LTO-3 tape was written with MXF files representing items dubbed from D3 video tapes. The QC step is a manual process where an operator checks that the video contained on the data tape is free from transfer errors and is labelled correctly. A logging feature was built into IngexPlayer to allow the user to log problems in the transfer while watching the programme playback on an SDI monitor. The user sets marks using buttons on the jog-shuttle control. The marks are stored in a log file which can be reloaded at a later stage to review the problems logged. PSE failures and D3 replay errors are automatically marked up for review.

The archive environment requires a restore from tape facility. IngexPlayer streamlines this process using a user friendly interface and disk cache. When an LTO-3 tape is inserted into the tape drive, the index file of MXF programmes is immediately read and the contents displayed on the user interface (see Figure 4). The jog-shuttle control is used to scroll through the list of programmes stored on tape and once selected, a programme is read off tape, stored on the disk cache and simultaneously played out over SDI. The user interface can be locked to avoid accidental interruption of playout when restoring the programme to video tape.

Partial restore from data tape is also made available. Given a list of in and out timecodes a partial restore can be made by creating a new MXF file containing the sections of programme specified. Although all data up to the last end point of a programme must be read from tape to complete the partial restore, the data can be read off tape faster than real-time reducing the overall time to perform the restore. The MXF file or files created can be tailored to suit a particular editor or playout server, including a provision to transcode from uncompressed video to a required compressed essence format.

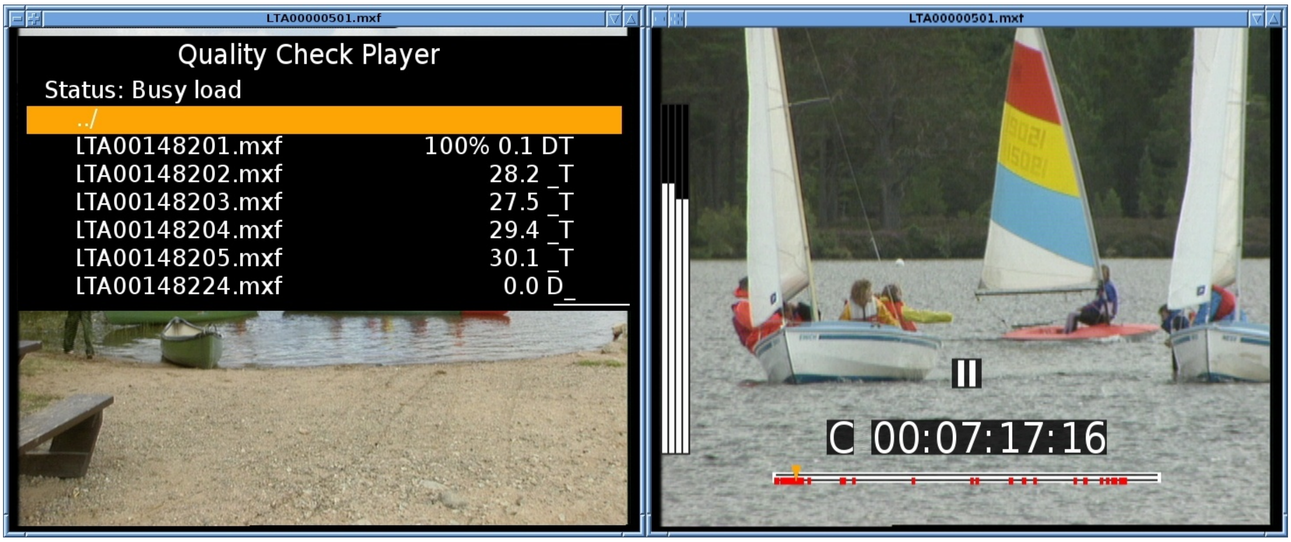

Figure 4 – IngexPlayer user interface showing data tape index file contents (left) and playback screen with timecode, audio monitor and marked-up progress bar displays (right)

MIGRATION FROM SD TO HD EQUIPMENT

An advantage of file-based production is the comparative ease of migration from SD to HD capability for file-based components. The Ingex studio recording system described above is capable of real-time encoding and recording of MXF wrapped DVCPROHD or uncompressed HD video, by using an HD capable SDI I/O card. A real-time SMPTE VC-3 software codec is currently under development for use in an Ingex recorder.

Existing gigabit networks and commodity NAS servers are capable of storing and serving file-based compressed HD content in typical editing environments. File-based compressed HD bitrates we have encountered in BBC productions range from 100 Mbps for DVCPROHD to 185 Mbps for SMPTE VC-3. Compared to online SD bitrates of 50 to 70 Mbps, hard disk storage needs to be increased by a factor of two or three. Editing compressed HD content over gigabit networks is still feasible without requiring a migration to 10 gigabit networks.

LTO-3 data tape, with a data transfer rate of 640 Mbps, is already capable of storing compressed HD video formats, such as DVCPROHD or SMPTE VC-3, several times faster than real-time. Experiments with real-world HD content showed that MXF wrapped 8-bit uncompressed HD is compressed at a ratio of 2:1 using the lossless Lempel-Ziv class 1 (LZ-1) data compression algorithm of LTO-3 tape drives, reducing its bitrate from 830 Mbps to 415 Mbps, making real-time HD recording possible on data tape. In practice, a disk cache may be required to accommodate the lower compressibility of worst-case video when using LTO-3 data tape for real-time uncompressed HD recording. The LTO-4 data tape format (15), due to be available in the second half of 2007, has an uncompressed transfer rate of 960 Mbps, higher than the 830 Mbps of 8-bit uncompressed HD video.

CURRENT DEVELOPMENTS

At the time of writing a number of software tools are under development which build upon the libMXF software library and other features of the Ingex software suite.

Verify and backup for file-based cameras

As file-based cameras are increasingly used in television productions, the need to verify and backup valuable media assets becomes vital. A software tool is under development which copies MXF files from removable camera media and stores them on a network file server and optionally on LTO-3 data tape. In addition to copying, the MXF file structure is verified and the essence stream checked for errors where possible. For example, MXF wrapped DV essence can have each DV frame's timecode and other metadata checked for consistency. The urgency for such a tool was underscored by the corruption of 15 minutes of video during a BBC file-based camera production in May 2007. After recording several video clips the camera appeared to playback the clips correctly through its viewfinder, but when the video clips were later copied onto a laptop, the video was too corrupted to be usable. Every eight or so frames of DV video was corrupted as shown in Figure 5, while other frames were intact.

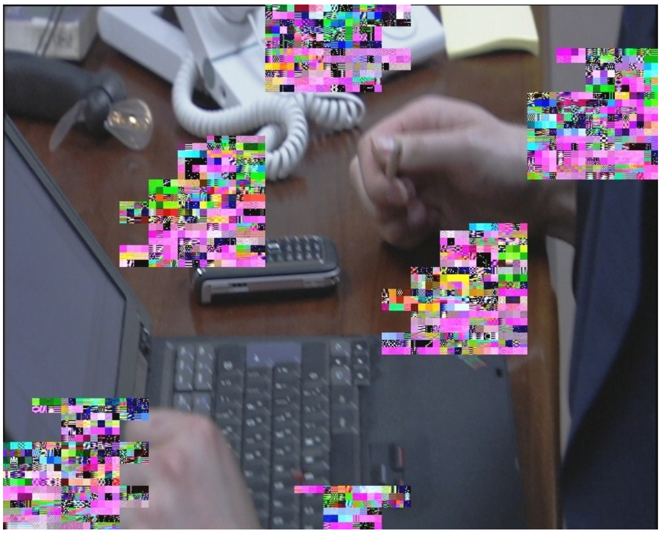

Figure 5 – Corruption of DV video frame after transfer from file-based camera

Conversion between OP-Atom and OP-1A MXF files

The MXF file format has become an accepted standard for file-based broadcast equipment. Problems in file interchange arise when two pieces of equipment use different MXF Operational Patterns. For example, a file-based camera stores OP-1A MXF files, but the editor being used requires OP-Atom files. Although the video essence is compatible with the editor, the essence needs to be re-wrapped into the required OP-Atom form. A simple software tool is under development which converts an OP-1A file to a set of OP-Atom files, preserving all MXF metadata including dark metadata items. The tool also performs the reverse operation, re-wrapping a set of OP-Atom files into a single OP-1A file, which may be a necessary step to take a finished edit and send it to a file-based playout server.

CONCLUSION

Practical problems which arise in real-world file-based productions can be solved using commodity IT equipment and software tools. The IT components which have proved successful include; PC SDI I/O cards for SD and HD capture and playout, SATA disk based NAS servers, gigabit Ethernet adapters and switches and LTO-3 data tape.

Standard file formats, such as MXF and AAF, have been the enabling technology to achieve practical interoperability between broadcast equipment.

To encourage adoption of the techniques described, all the software tools written by BBC Research and described in this paper have been released as open source software and made available from the Ingex website (1).

REFERENCES

1. SMPTE, Material Exchange Format, SMPTE 377M

2. BBC, Ingex project home page: http://ingex.sourceforge.net/

3. Tudor, P. N. and Cunningham, S. H., 2006, Improving workflow in practice for low-cost programme-making using MXF & AAF file formats, NAB Broadcast Engineering Conference 2006

4. Fletcher, J., Kirby, D. and Cunningham, S., 2006. Tapeless and paperless: automating the workflow in TV studio production, Proceedings of 2006 International Broadcasting Convention.

5. Advanced Media Workflow Association (AMWA), AMWA home page:

http://www.amwa.tv/

6. FFmpeg project home page: http://ffmpeg.mplayerhq.hu/

7. Independent JPEG Group (IJG), IJG home page: http://www.ijg.org/

8. BBC, libMXF project home page: http://ingex.sourceforge.net/libMXF/

9. Advanced Media Workflow Association (AMWA), AAF SDK home page:

http://aaf.sourceforge.net/

10. Schroeder, B. and Gibson, G. A., 2007, Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you?, 5th USENIX Conference on File and Storage Technologies (FAST '07)

11. TTCP project home page: http://www.pcausa.com/Utilities/pcattcp.htm

12. SMPTE, Unique Material Identifier (UMID), SMPTE 330M

13. BBC, TransferToP2 project home page: http://ingex.sourceforge.net/TransferToP2.html

14. Advanced Media Workflow Association (AMWA), MXFDump application source code:

http://aaf.cvs.sourceforge.net/aaf/AAF/DevUtils/MXFDump/

15. Linear Tape Open (LTO) Consortium, LTO home page: http://www.ultrium.com/

16. The Open Group, pax Interchange Format,

http://www.opengroup.org/onlinepubs/009695399/utilities/pax.html

17. x264 project home page: http://www.videolan.org/developers/x264.html

18. FAAC project home page: http://sourceforge.net/projects/faac/