Software for viewing and converting digital cinema materials

Charles Fenimore and Alexei Nikolaev

NIST

NIST

Software for viewing and converting digital cinema materials

Charles Fenimore and Alexei Nikolaev

NIST

Software for viewing and converting digital cinema materials

Charles Fenimore
and
Alexei Nikolaev

Abstract

Digital cinema (d-cinema) is the highest quality electronic motion imagery for entertainment. The cinema is presented in a theatrical environment, with high brightness projectors, high rate data transfer, and high resolution color imagery. The entertainment industry uses formats that are specific to film-based electronic imagery. This report describes software tools for viewing digital cinema specific imagery (DPX format) and converting it to and from other common image formats, such as TIFF, BMP, and JPG. It includes sample test imagery, which has been generated in connection with recent d-cinema system tests. The d-cinema viewer/converter is available on line.

Keywords
digital cinema evaluation, DPX, format conversion, image format, test patterns, viewer

1. Introduction

The introduction of digital projection as an alternative to film for entertainment cinema raises a number of challenging issues: transfer of large data files, maintaining the security of the digital movie, its storage and preservation, and the quality of the imagery, among others. The DC2001 Conference Proceedings (http://digitalcinema.nist.gov/agenda.html) provides an introduction to these challenges. This report describes one part of the quality puzzle for electronic projection systems. We present software tools for viewing and manipulating d-cinema test imagery. In addition, example software is included for generating the patterns. A viewer/converter that runs on PC-platforms is available for download, as are a number of test patterns.

The test materials are intended for use in characterizing electronic systems used to project imagery. The measurements are based on the IEC [1] and VESA [2] projector display measurement standards. The characteristics of the systems may depend on the projectors, the storage system, interfaces, and converters. In the June 2001 MPEG digital cinema compression tests, two display systems were used with differing interfaces and image formats. We used the IEC and VESA display measurement standards to characterize the display systems, not simply the projectors. The test patterns were stored and played out using the same components as for the subjective viewing tests. A more detailed report on these measurements is available [3].

Test imagery intermediates are generated in 10-bit per sample TIFF format. DPX and YUV formats are required for presentation. This report describes the basic characteristics of d-cinema image formats and presents ImageViewer for viewing and converting d-cinema materials on PCs.

2. Image Characteristics

For the recent MPEG projection system characterization [3], the characteristics of the test patterns are the same as those of the materials for the subjective tests. While the patterns described in *Electronic Projection* [1] are given by analog specifications, the present patterns are the corresponding digital signals as specified in the relevant SMPTE standard [3]. The digital characteristics of the test patterns for the two projectors used in the MPEG d-cinema tests are detailed in Table 1. The NIST software supports conversion of color imagery at either 8- or 10-bits per color sample.

Table 1: Image Formats for Projection Systems

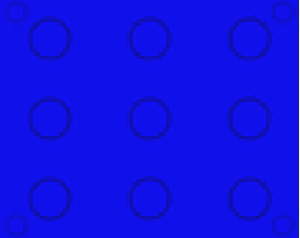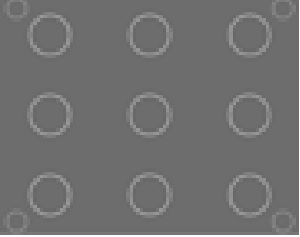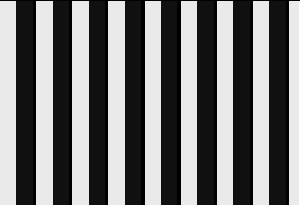| Image Attribute	DLP[#] Projection system	CRT Projection System
Image file format	DPX [5]	YUV (SIF)
Color-coding components [4]	R, G, B	Y, P_B, P_R
Sampling structure	4:4:4	4:2:2
Addressable active pixels per frame	1280 X 1024	1920 X 1080
Stored pixels per frame	1920 X 1080	1920 X 1080
Bits per color sample	10	10
Code value range	[64 - 940] for R, G, and B	[64 - 940] for Y [64 - 960] for P_B, and P_R
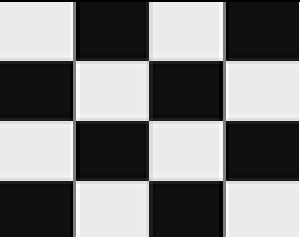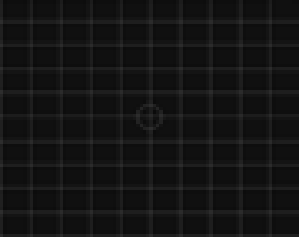
3. Test Images

Each projection system requires multiple test patterns for the measurements reported to MPEG. The patterns, together with illustrative images of the active image area, are summarized in this section. Most patterns appear in two versions, one with and one without circles. Circle patterns are used to align the measurement apparatus. Measurements are made on patterns without alignment circles.

[#] Certain commercial products are mentioned in this report. Such mention is not an endorsement of the products by NIST, nor is it meant to imply they are the best for the purpose of the reported research.

Table 2: Test patterns

| Pattern identifier (1280x1024 = Small and 1920x1080 = Big images.)	Reference: IEC [1] or VESA [2]; Pattern shown.	Illustrative image (Small pattern.)
Blue Red Green Blue with alignment circles Red with alignment circles Green with alignment circles	VESA [2] section 302-4: Color uniformity Pattern shown: Blue with circles	
Gray64 Gray189 Gray314 Gray439 Gray564 Gray689 Gray815 Gray940 With/without alignment circles	VESA [2] section 302-5: Used for gray scale, gamma, sequential contrast, and uniformity. Pattern shown: Gray439 + circles	
Alternating lines patterns (horizontal and vertical) Hor_1x1, Vert_1x1 Hor_2x2, Vert_2x2 Hor_3x3, Vert_3x3 Hor_4x4, Vert_4x4 Hor_5x5, Vert_5x5	IEC [1] Sec. 6.1, VESA [2] Sec 303-2: Resolution Illustration suggests a vertical pattern having line width >> 5 pixels.	
Chess Chess with alignment circles	IEC [1] Figure A.3 Used for "ANSI" contrast. Pattern shown: Chess	
Grille64 GrilleMid Grille940 GrilleCircle GrilleMultiCircles	IEC [1] Figure A-1 VESA [2] section 501. Image size, aspect ratio Pattern shown: Grill Circle	

4. Image viewing and conversion.

The NIST ImageViewer supports conversion of the patterns to 32-bit DPX. Conversions from YUV to RGB are presented briefly in Section 5.1. YUV uses subsampled color and has lower resolution than DPX or TIFF. The test patterns were generated as 48-bit TIFF RGB (packed 10-bit color samples) using MatLab software.

4.1 Image Viewing and Conversion
The viewer/converter, **ImageViewer**, is adapted from a viewer distributed with the Borland C++ Builder package. This section describes the use of **ImageViewer** including Figure 1 showing the graphical user interface and a Manual Page. Because the native display format for PCs is 24-bit RGB, the 10-bit per sample d-cinema imagery is displayed at 8-bits per sample, which is at less than the available bit depth.
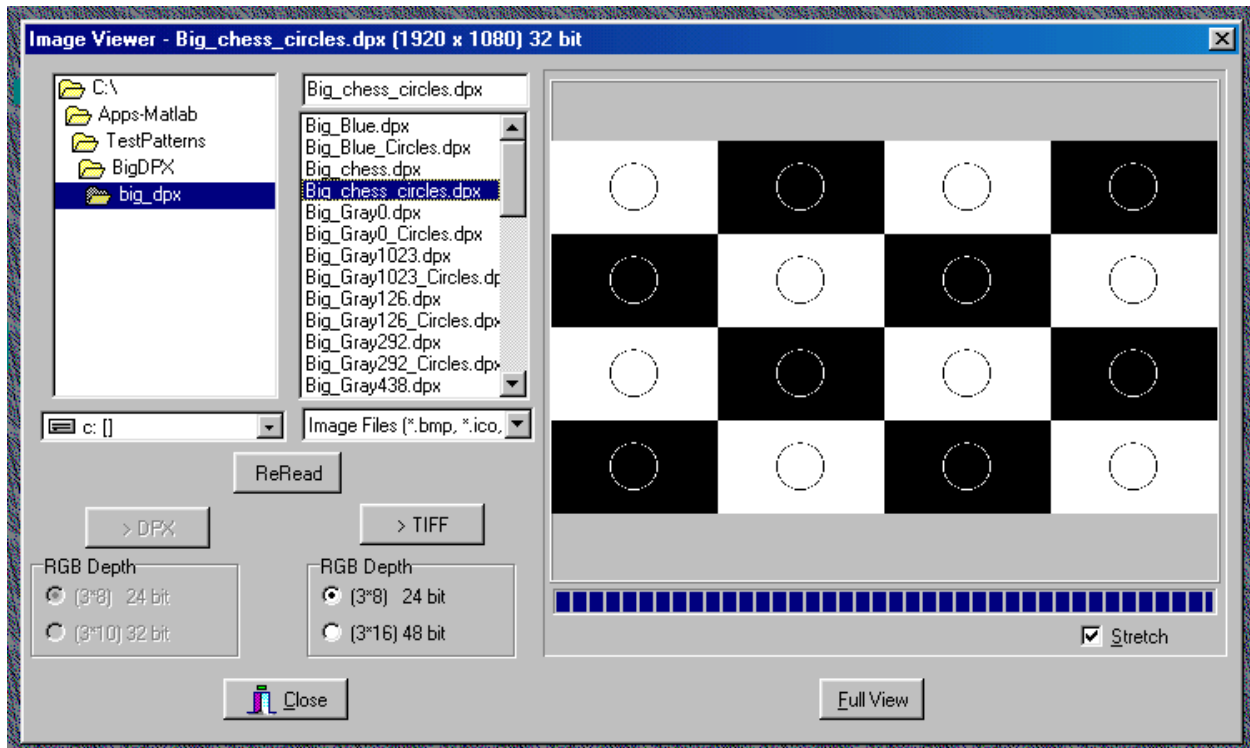
Figure 1: The GUI for ImageViewer includes windows for file viewing, file access, and display/output controls.

4.1.1 Manual page for ImageViewer

FUNCTION: **ImageViewer** displays and converts images in several electronic cinema formats. While the viewer supports the reading of imagery that is compressed (JPEG) or sub-sampled (YUV), the outputs are in uncompressed TIFF or DPX. **ImageViewer** reads images with 8 or 10 bits per sample. Down conversion to reduced bit-depth is supported.

PLATFORMS**:   ImageViewer** runs under MS Windows-95, -98, and -2000.

FORMATS (for details see Section 5):
  Input: `DPX` (8- and 10-bit/sample, RGB)
       `TIFF` (8- and 10-bit/sample, RGB)
       `YUV` (8- and 10-bit/sample, YUV in 4:2:2 sampling)
       `BMP` (8-bit/sample, RGB)
       `JPEG`

  Conversion (output) formats
       `DPX` and `TIFF` (both at 8- and 10-bit/sample, RGB)

  Display formats
       The images are displayed on a PC in 8-bit per sample RGB.

GUI BUTTONS AND SUB-WINDOWS: The GUI has a split Read/Write panel, a Viewing panel, and an Image Viewer topbar as follows.
  TopBar
       The **ImageViewer** top bar displays the image name, dimensions, and bits per pixel.
  Reading
       Single click on a filename, or scroll up and down, to load an image in one of the supported input formats. The image will appear in the right side viewing panel.
  Writing with **>DPX** and **>TIFF**
       Select the bit depth of the image to be written, and then click on the appropriate output format button to convert and write a new image file. The GUI only activates buttons that will convert to DPX or TIFF from another image format. **RGB Depth** permits the user to convert the format at a reduced bit depth. Increasing the bit depth is not supported.
  ReRead
       Update the directory listing in the left side windows.
  Viewing
       Imagery is displayed in the right side panel, which is 480 X 360 pixels in size. For images larger than the window, a central portion is displayed.
  Stretch
       Click on **Stretch** to scale the image to the window.
  FullView
       Click on **FullView,** or double click on the right side image panel, to open a full resolution window.

REVISION HISTORY: Version 1.0, July 2001; Version 1.1, 9/4/2001.

AUTHOR: Alex Nikolaev, alexei.nikolaev@nist.gov, phone (301) 975-8265

4.2 Matlab test pattern generation

The Appendix, Section 7, lists sample Matlab image generation code for the *Chess with alignment circles* pattern. The simplicity of the test patterns permits straightforward coding. Circles are generated as annular regions of points (`i,j`) satisfying an inequality:

```
abs( (i-XCenter)^2 + (j-Ycenter)^2 - R^2 ) < Thickness^2
```

The output of 48-bit TIFF imagery is handled by a Matlab function call that packs each 10-bit color sample into a 16-bit word.

```
imwrite(uint16(65535*pattern1),'chess.tif','tif','Compression','none');
```

While there are a variety of picture editors that can be used to generate the patterns, doing so for d-cinema requires output at 10-bit color sampling, which is supported by the TIFF library.

5. Image Formats

The NIST `ImageViewer` supports conversion of imagery with linearly scaled RGB into TIFF and DPX at both 10 and 8 bits per color sample. For the most part, these conversions reformat without a change of color space. The conversions from YUV to RGB involves mapping from one of two possible YUV spaces, the different color spaces of digital standard definition television [7] and that for HDTV production [5]. In addition, YUV uses subsampled P_R and P_B and so has lower resolution than DPX or TIFF. The `ImageViewer` can be found on the web site http://www.itl.nist.gov/div895/TestPatterns/index.html. The source code for these conversions is available from the authors.

5.1 YUV

YUV or SIF is a headerless image file format that carries YP_RP_B at 8- or 10-bits per sample. For each 4 Y samples there are 2 P_R and 2 P_R samples, designated as 4:2:2 sampling. Conversion to RGB at 4:4:4 sampling involves upsampling and a linear conversion.

YUV has different colorimetry for 8- and 10-bit formats. For most imagery of interest, 10-bit YUV conforms to the colorimetry of ITU's high definition Recommendation 709 [5]. The conversion to RGB is given by the system of equations:

```
R = 0.999685*Y + 0.000130416*(PB-512)  + 1.57516  *(PR-512)
G = 0.999907*Y - 0.187054*(PB-512)     - 0.4167983*(PR-512)
B = 1.00186*Y  + 1.85562 *(PB-512)     + 0.00100598*(PR-512)
```

Similarly, almost all of the 8-bit YUV imagery conforms to Rec. 601 colorimetry [7].

```
R = 1.16438356*(Y-16)                            + 1.59602679 *( P_R-128)
G = 1.16438356*(Y-16) - 0.39176652*(P_B-128) - 0.812971875*( P_R-128)
B = 1.16438356*(Y-16) + 2.01723214*(P_B-128)
```

5.2 DPX
DPX is a flexible image format. The header is about 2K bytes and specifies characteristics of the image including image format, film-related information, television-related information, and user-defined data. The format accommodates various color systems including RBG and YUV (or YP_RP_B) in 8-, 10-, and 12-bit color sample size. **ImageViewer** supports input and output of DPX in 24- and 32-bits per RGB pixel.

5.3 TIFF packing of RGB
TIFF is a flexible tagged image file format. In the present implementation, the test patterns are in 48-bit per pixel formats. The TIFF library is large and is available on line from a number of sources, including http://www.libtiff.org/. **ImageViewer** supports reading and writing of TIFF in 24- and 48-bit RGB files.

6. Conclusions

The present tool set for generating, viewing, and converting digital cinema imagery is PC-based and freely available. The Matlab pattern generators and the C++-based **ImageViewer** are two pieces in a number of elements needed to explore the interchange of visual information across platforms ranging from the very highest available quality down to television, computers, and even lower resolution devices. The interfaces for image I/O are available from the authors. In its present form **ImageViewer** should enable users to convert small clips of digital cinema in a flexible manner.

7. References
[1] IEC, 61947-2 ED 2/CDV, Draft International Standard, *Electronic Projection – Variable Resolution Projectors*, Draft 2, [August 2000].
[2] VESA, *Flat Panel Display Measurement Standard Version 1.0,* [May 1998].
[3] P. Boynton and C. Fenimore, *Report to MPEG of the Characterization of Projectors for the MPEG Digital cinema tests, NISTIR 6YYY* [September 2001].
[4] SMPTE Standard 274M, *1920 X 1080 Scanning and Analog and Parallel Digital Interfaces for Multiple Picture Rates*, [1998].
[5] ITU-R BT.709-2, *Parameter Values for the HDTV Standards for Production and International Programme Exchange*.
[6] SMPTE 268M-1994, *File Format for Digital Moving-Picture Exchange*.
[7] ITU-R BT.601-4, *Encoding Parameters of Digital Television for Studios*.

8. Appendix – Test pattern generation in TIFF (Matlab code).

```matlab
  % Chessboard test pattern
clear all;
close all;

hor_pix=1921;                    %Horizontal number of pixels
ver_pix=1081;                    %Vertical number of pixels

hor_win = 1920; %1280; %large and small patterns
ver_win = 1080; %1024;

Y_win = round((ver_pix-ver_win)/2);
X_win = round((hor_pix-hor_win)/2);

squareX = 4;
squareY = 4;
depth_D = 0;
depth_L = 1;


dY=round(ver_win/squareY);
dX=round(hor_win/squareX);

R = min(dX,dY)/4;
thickness = 12;

%------------------------------------------------%
%Creation of frame, 100%
pattern(1:ver_pix,1:X_win)= 1;
pattern(1:ver_pix,X_win+hor_win:hor_pix)= 1;
pattern(1:Y_win,X_win:X_win+hor_win) = 1;
pattern(Y_win+ver_win:ver_pix,X_win:X_win+hor_win) = 1;

%patternD(1:dY,1:dX)=depth_D;
%patternL(1:dY,1:dX)=depth_L;

%Creating circles for light (L) and dark (D) board squares
for i = 1:1:dX
    for j = 1:1:dY
        patternD(j,i) = depth_D;
        if(abs((i-dX/2)^2 + (j-dY/2)^2 - R^2)) < thickness^2
            patternD(j,i)=depth_L;
        end
    end
end
for i = 1:1:dX
    for j = 1:1:dY
        patternL(j,i) = depth_L;
        if(abs((i-dX/2)^2 + (j-dY/2)^2 - R^2)) < thickness^2
            patternL(j,i)=depth_D;
        end
    end
end
```

```
%Creation of chessboard pattern
for i=1:2:squareX
  for j=1:2:squareY
      pattern(1+(j-1)*dY+Y_win:j*dY+Y_win, 1+(i-1)*dX+X_win:i*dX+X_win) =
      patternL(1:dY,1:dX);
      if ((j+1)*dY <= ver_win)
            pattern(j*dY+1+Y_win:(j+1)*dY+Y_win, 1+(i-1)*dX+X_win:i*dX+X_win)
            = patternD(1:dY,1:dX);
      end;
      if ((i+1)*dX <= hor_win)
            pattern(1+(j-1)*dY+Y_win:j*dY+Y_win, 1+i*dX+X_win:(i+1)*dX+X_win)
            = patternD(1:dY,1:dX);
            if ((j+1)*dY <= ver_win)
                  pattern(j*dY+1+Y_win:(j+1)*dY+Y_win,
                  1+i*dX+X_win:(i+1)*dX+X_win) = patternL(1:dY,1:dX);
            end;
      end;
  end
end

%Y_win:Y_win+ver_win-1, X_win:X_win+hor_win-1
pattern1(:,:,1)= pattern(:,:);                      %converting to rgb, red
pattern1(:,:,2)= pattern(:,:);                      %converting to rgb,
green
pattern1(:,:,3)= pattern(:,:);                      %converting to rgb, blue


imwrite(uint16(65535*pattern1),'pattern_chess.tif','tif','Compression','none')
;

imshow(pattern1,65535);
```