Journal of Contemporary Archival Studies

Volume 6 Article 16

2019

Low-Cost 8mm/Super 8 Film Digitization Using a Canon 9000F II Flatbed Scanner and Photoshop: A Case Study

Kenneth Eckert
Hanyang University (ERICA), Korea, keneckert@hotmail.com

Low-Cost 8mm/Super 8 Film Digitization Using a Canon 9000F II Flatbed Scanner and Photoshop: A Case Study

Kenneth Eckert
Hanyang University, Korea

Introduction

For some fifty years, between its introduction in 1932 and its obsolescence with the growth of consumer video cameras in the 1980s, 8mm/Super 8 movie film was widely used for home movies and by amateur hobbyists and film students. It occasionally was used in TV news or even for in-flight movies and arcade kiosks, and possibly its most unfortunately famous example is the Zapruder 8mm film of John F. Kennedy's assassination. Yet its users have shown even less loyalty to small-gauge movie film than to still film, for while a specialist industry thrives for 35mm photo film, 8mm/Super 8 are largely extinct and forgotten apart from niche enterprises and a tiny following of determined users.

For archivists or filmmakers interested in digitization of old film this presents a peculiar problem, for 8mm/Super 8 movie film does not enjoy the economies of scale that larger professional formats do. Commercial photo film scanners are designed for negative strips or slide positives, not for continuous film. Dubious-quality consumer gadgets for small-gauge film transfer that largely utilize webcam-grade cameras or reflection boxes can be found online, but industry-grade video telecine equipment is now decades old, and high-quality digital transfers require scarce and expensive machinery. As a home or independent user, digitization facilities are increasingly inaccessible or beyond an individual's budget.

As a boy I found my first 8mm Brownie movie camera at a garage sale in October 1980, and over some twenty years shot some 2,000 feet or two hours of family movies, short plotted films, and animation. Even in the 1980s the cameras and projectors were aging and subject to frequent breakdowns and hack repairs, and now in a digital world it is far better to have these films preserved in a digital format where they can be shared online and enjoyed. Digital media also allows music and sound effects to be added to once-silent film.

A small network of websites exists where home users have patched together their own cottage digitization systems. The general consensus is that flatbed scanners are inadequate for such purposes, and so such systems usually employ some means of photographing either the projected image or the film frame itself onto a digital camera, with some sophisticated users engineering motorized systems to auto-advance the film and trigger the camera for partly unsupervised scanning. While such processes may be optimal, they require a high order of equipment, technical expertise, and expenditure. The purpose of this paper is to demonstrate a workflow solution that makes flatbed scanning feasible for the small-budget archivist, by using a Canon 9000F Mark II flatbed scanner with a transparency adapter, along with Adobe Photoshop scripting, to successfully scan and render optimum-quality digitized 8mm/Super 8 silent film.

Project Costs and Time Outlay

- Canon 9000F Mark II scanner, approximately US$299
- Adobe Photoshop, pricing varies

- Support arms for film reels can be removed from an old film projector, or made from dollar-store tripods. Film negative sleeves, cardboard, hobby plastic squares as needed.
- Scanning film time: approximately 0.5 frames per second (fps), two to three hours per fifty-foot roll
- Straightening and manual cleaning of film files time: varies
- Rendering film files into 16/17 single frames time: on Intel i5 8600 with SDD, 3 fps

What Does *Not* Work

Analog Videotape Capture. About once every decade for thirty years I attempted to transfer my films to analog video with minimal success, and I usually ended up limping back to a professional telecine transfer service. Eight millimeter runs at 16 fps and silent Super 8 runs at 18 fps, and North American NTSC video runs at 29.97 fps (or more accurately, every other line of resolution is scanned every 1/60 of a second and a complete image is assembled 30 times a second comprised of two fields of scans), resulting in a hopeless flickering image if a consumer video camera is aimed at the projector screen. European PAL is 25 fps, again resulting in a sync mismatch.

My digital captures of videotapes from the telecine laboratories were better, but were limited by the original quality of the tape format: A 1985 transfer was onto standard VHS, rendering only about 240 vertical lines of analog resolution, and a 2001/2006 transfer was onto S-VHS and DV-cam 8mm tapes with about 400 lines of resolution. With computer video capture from the players the latter tapes provided a viewable 640 × 480 pixel video file, which was better than VHS but still unsatisfactory.

Frame Photography with a Digital Camera. By 2015 prosumer digital cameras were advanced enough to experiment with photographing movie frames directly off an illuminated stand. The workflow I designed was to have two support reels with a small sheet of translucent white plastic lit by an LCD lamp underneath, and to shoot single frames onto the camera, later stringing them sequentially into a movie file with Virtual Dub. An alternative was to shoot into the film gate of the projector. Neither trial worked. Why?

The first problem is focusing. Eight millimeter frame images are *very* small, measuring 4.8 mm × 3.5 mm; about seven frames will sit on a North American penny. Eight millimeter movie film is actually just 16mm film in twenty-five-foot rolls, which is run through the camera twice and shot on the left half. During processing the film is sliced down the middle and connected to make one fifty-foot roll of about four minutes running time. Super 8 was introduced in 1965 and was simpler to use because it came boxed in a cartridge and required no reloading; it also yields a larger film frame as the sprockets are narrower. The result is still only 5.8 mm × 4.0 mm, a challenging size for macro focusing.

Attempts with my Sony DSC-RX100, Canon PowerShot G10, and Nikon 1 J5 were unsuccessful, as the cameras could not get close enough; a macro attachment on the J5 rendered images clear in the middle but blurry on the edges. What would be required is a true digital single-lens reflex camera (DSLR) with a macro bellows, which would move the project into a far higher price range. The second problem was registration. I had over ten 200-foot reels of film, and shooting 168,000 single frames with minimal jiggling as the tiny film was manually advanced was impossible.

Fig. 1. A penny and a strip of Super 8 film

Shooting the Projected Image with a Digital Camera. In a trial in 2001 I captured the screen from my Fujicascope M25 film projector onto a PC webcam at 640 × 480 pixels, and processed the image with Virtual Dub's deflicker filters to smooth out the frame desynchronization. Results were mediocre in quality and still not frame-accurate, but they were promising. Might this now work with a modern digital camera and Photoshop's improved image processing?

Ideally, the film would be advanced and photographed one frame at a time. This required that I remove the original tungsten lamp from the projector, a CXR/CXL 8V 50W bulb, for if the film sat stationary it would melt within seconds. In January 2018 I found an Energizer LED bicycle lamp with 300 lumens and affixed it in place, and belted to the flywheel a small metal crank welded by my father to advance the film. Using a remote release for the Canon PowerShot G10, I began cranking and photographing frame by frame. Results were acceptable but capture was interminably slow. The next improvement was to motorize the advance, and the crank was replaced with a geared 12V GM-36B motor that rotated at 200 rpm and advanced the film at 3 fps. This speed was now too rapid to photograph individual frames, but I reasoned that I could capture digital video at 30 fps and then select correct frames or remove bad ones in File Explorer.

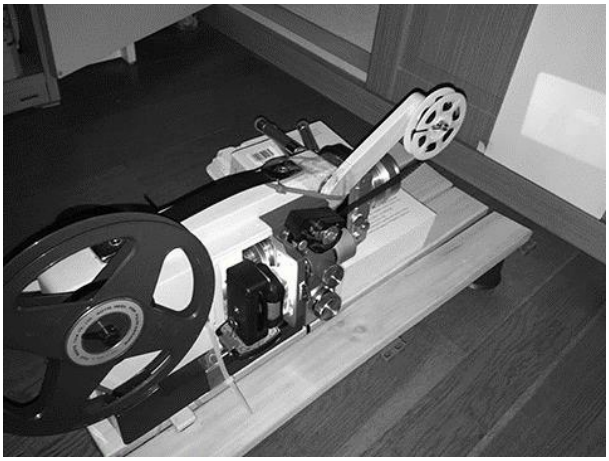

Fig. 2. Fujicascope M25 projector with motor and bicycle LED lamp

Yet by now the multiple problems involved were making this approach an untenable time sink. The projector was over forty years old and long past its day. The film gate persistently filled with dirt and the aged rubber flywheel belt kept snapping. Worse, the gate sprocket was not advancing the film to the exact same vertical slot position each time. I tried different Virtual Dub and Adobe Premiere filters to deshake and stabilize the results, and tested some user scripts for AviSynth (http://www.super-8.be). Optimally one could use scripting to align images by the sprocket hole for perfect registration, but the projector gate, in addition to blocking off the edges of the frame image, also hides the sprocket hole. Distinguishing correct frames in File Explorer *and* manually verifying their correct vertical registration to eliminate the off-kilter ones would be intolerably painstaking.

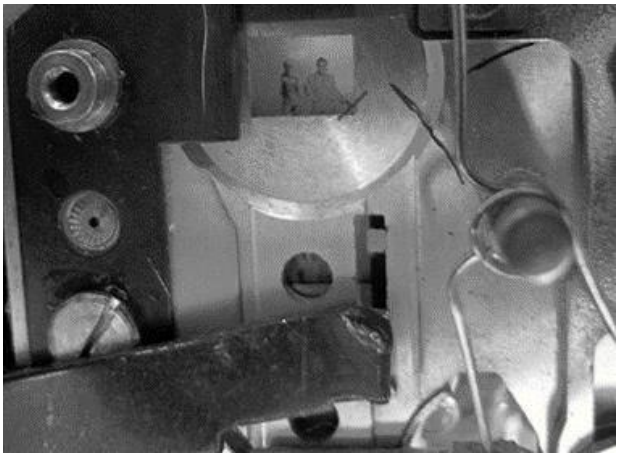

Fig. 3. Projector gate with sprocket tooth

What *Does* Work: Flatbed Scanning

The objection in online specialist forums to flatbed scanners is that they lack quality and resolution, with their transparency functions often being a product afterthought. In 2004 I purchased a Canon LIDE model with a transparency adapter, and this worked well for 35mm negatives; but for 8mm/Super 8 film it was cumbersome, and with a maximum 2400 dpi it rendered a 500 × 390 pixel image, which was sharper when juxtaposed against images from a videotape capture, but even smaller in pixel dimensions.

Yet over the last decade such scanners have steadily improved. In February 2018 I purchased the Canon 9000F Mark II flatbed scanner, which has a built-in 8 × 27 cm transparency adapter. The scanner is capable of 9600 dpi transparencies, but it is far more rapid at 4800 dpi, and apparently this is its "true" optical maximum. At 4800 dpi the scanner produces a Super 8 frame of about 1140 × 800. It is difficult to measure the effective pixel size of analog film, but these dimensions appear to approach the maximum definition of 40 ASA 8mm/Super 8 film. Although 1140 × 800 is not high compared to even smartphone video, one advantage of film is that because the grain is in random places as opposed to a fixed pixel grid, at 16/18 fps our eyes interpolate the missing information to give the appearance of a better image.

Other scanners can presumably also be employed for 8mm/Super 8 scanning, although some do not permit nonstandard transparencies. Happily, while the Canon comes with trays for different media such as slide positives or 35mm negatives, it allows anything to be placed on the scanner as a transparency. However, movie film cannot run continuously up-and-down through the

scanner (fig. 4 left), as it reserves a small horizontal strip at the top of the transparency area to align and check itself every time it operates, and this area needs to be kept empty. Thus a vertical line of film going upwards across the scanner and in and out its lid will not work unless the film is cut into segments, a non-starter option for an archivist or filmmaker.

Fig. 4. Film moving up-down, diagonal, and side-to-side through and out the scanner.

In figure 4 (middle) I tried advancing the film diagonally so as to dodge the registration area during transparency scan. This did work and gave long forty-eight-frame scans, but the software saves rectangular boxes and not an angled area, and so the scans were ponderously slow at a massive 3.1 GB 13328 × 41654 pixel file for each frame strip. As well, at this length the film begins to slightly flex. If ten reels of film were to be scanned in one lifetime, compromises were necessary to speed up the workflow. The first reluctant compromise was to turn off Canon's useful but glacier-slow FARE dust detection algorithm; the second was to scan left-to-right horizontally across the scanner, as in figure 4 (right). As the transparency adapter handles 8.2 cm in width, that allows seventeen frames of 8mm or sixteen frames of Super 8 at a time, about one second of film time per scan—effectively scanning at roughly 0.5 fps. This process requires about ten hours to scan a 200-foot (13:00) reel of film, which is not an activity for the impatient or those on a tight time budget.

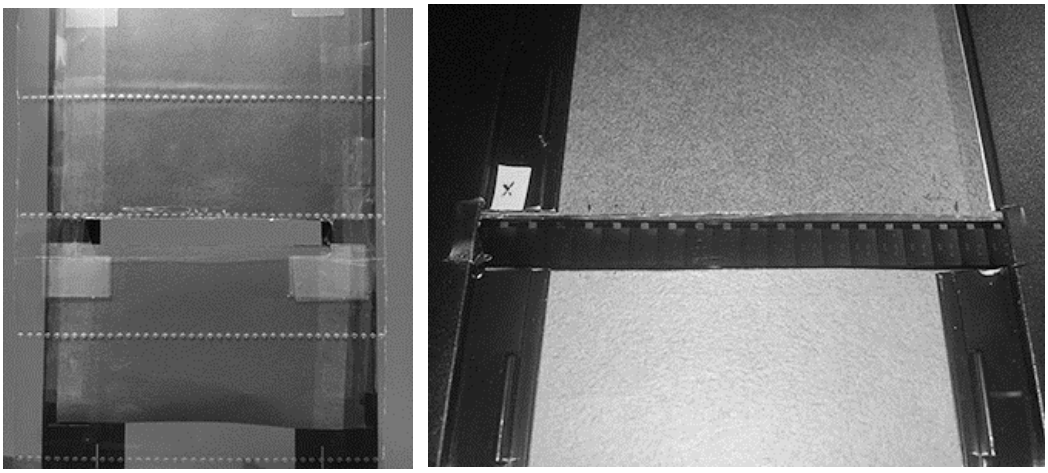

Fig. 5. Film tray, back and front

As there are no bespoke film holders for 8mm/Super 8 for the Canon scanner, they must be improvised. I took the Canon's large-format tray and cut out a centimeter or so of its flaps to widen the visible horizontal area, and then cut a mask out of cardboard and taped it to the underside of the tray, marking out sprocket holes on the cardboard with a felt pen. I then took a transparent acetate 35mm negative album sleeve and taped off its entry and exit points on each end to vertically secure the film and minimize wiggling. At first I also had the sleeve cover the film on top to weigh it down more firmly to the scanner plate, but this area filled with dirt, and so I trimmed it off. It remains important to regularly clean the scan path as dust will persistently accumulate.

Usually movie film in good condition will lie flat anyway, and while the Canon's focus is fixed, it is fairly forgiving about film raised up to a millimeter or so. Occasionally, where the film is damaged or has a rough splice I weighed it down with coins on the edges of the sprocket. In certain cases where I had treated the film with bleach, needle scrapings, or felt pens to simulate gunfire or explosions, or where sprocket holes were torn, keeping the film flat and straight had to be done with care. Otherwise, so long as finger oils were kept off the movie film there was no more apparent stress or risk of scratches or damage than normal projection would cause. The scanner lid did not significantly drag or impede film as it was tugged through.

Advancing the film in counts of 16/17 frames on the cardboard markers quickly becomes wearisome and eye-straining. To accelerate the process I built a small stand next to the scanner with a furniture riser from a cabinet and a small piece of plastic. Then I took two CD jewel cases and cut them so that their edges face each other and glued them so that they form a path for the film to move along. After marking out visual guides for the 8mm and 16 Super 8 frame points, I used a precision screwdriver to pull the sprockets from the first to the last mark.

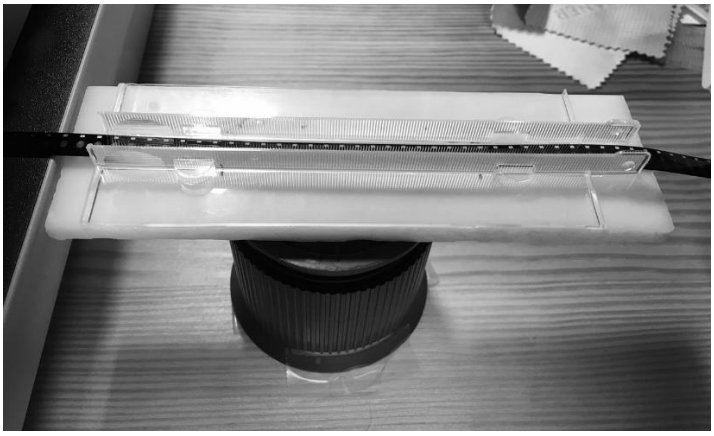

Fig. 6. Measuring stand

Last, I tore off the support arms from the Fujicascope to make a stand to hold the supply reel, and took apart a dollar-store smartphone tripod to make a take-up reel stand. The scanner interface was set to a selection of 2.77" × 0.43" at 48-bit color, 4800 dpi, with unsharp mark on. This gives a TIF scan of 13328 × 1873 pixels at 128 MB. The Canon occasionally hiccupped when saving to a hard drive, but not after switching to a solid state drive. It otherwise ran like a top with no observable deterioration of function after some ten thousand scans.

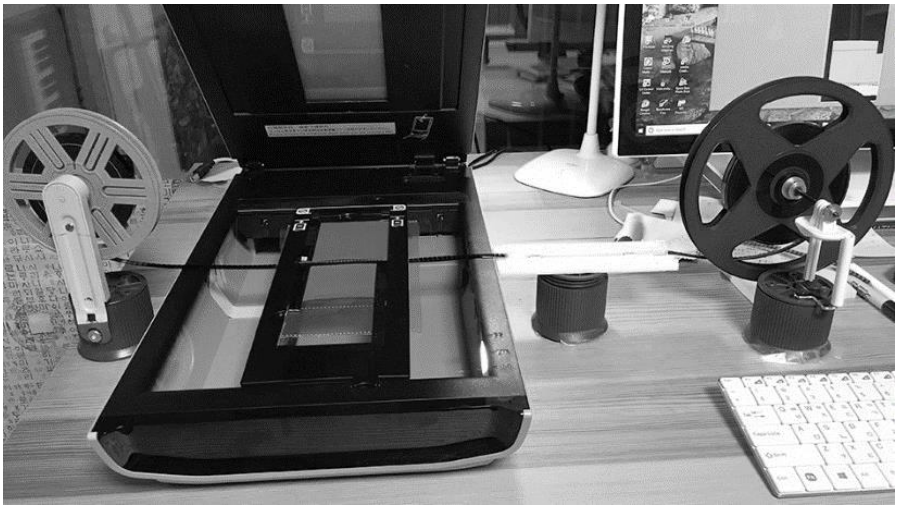

Fig. 7. Full digitization system

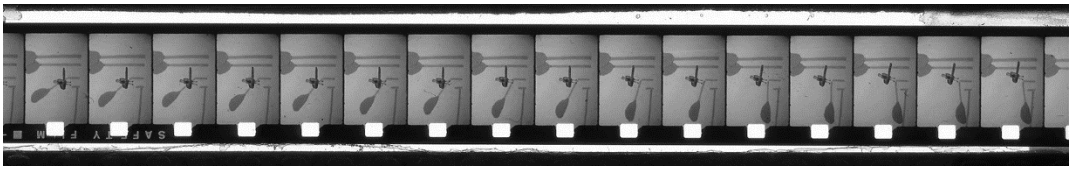

Fig. 8 Raw scanned strip of 16 Super 8 frames

Processing with Photoshop

Straightening the Film. Each 200-foot reel results in about 800 to 900 scan files of 16/17 frames each, for some 13,000 individual frame images. Again, using a photo editor to straighten and cut out each one of these frames manually, and with consistent registration, would be impossibly time-consuming. What makes the project work is Adobe Photoshop's automated tools. These include *actions*, where simple sequences of commands can be recorded into a keypress, and *scripting*, where more extensive instructions can be programmed via a form of JavaScript. Both actions and scripts can be executed manually or automated with a batch menu command.

The first task was to use a batch command to flip the images ninety degrees so they are up-and-down; because I scanned emulsion-down, I also needed to invert the images. The more difficult problem was to make each film strip perfectly perpendicular, and then to locate and cut out each frame as a separate image file. A film strip only 6 mm wide must glide along a rigid path or else the frames will wobble while viewed, and this accuracy is not feasible with cardboard and plastic. What we need Photoshop to do is to find the x and y coordinates of the top and bottom sprocket holes in the strip, calculate how much the bottom sprocket is out of horizontal kilter, and then calculate how much rotation is needed to straighten it. Then these steps can be automated for directories of files through a batch command.

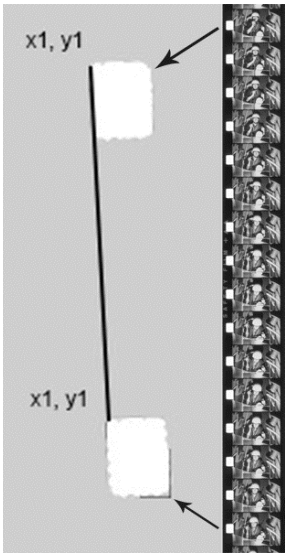

Fig. 9. Film sprocket coordinates

Because sprocket areas are clear and show as white, it is only necessary to click inside them with Photoshop's magic wand tool. Like the English-language expression about horseshoes and hand grenades, the strength of the magic wand tool is that close is good enough; if any area inside the sprocket hole is clicked the tool will find the boundaries and expand the selection to fill it. My first trial script did this; however, it was not only inexact but also failed to determine whether the bottom sprocket drifted left or right—it merely drew a rectangle from top x1 y1 to bottom x2 y2 and estimated the rotation based on the width of the rectangle.

As an English professor and not a programmer I was out of my depth, and requested help at the Adobe forums. One senior user, John Mack, kindly suggested that I use the color sampler tool instead, which makes it easier to programmatically manipulate multiple values (discussion at https://forums.adobe.com/collaboration/119586). The way Mack's script operates is that the user selects any white pixel inside the sprockets of the top and bottom frames, and then a line of code searches left and right to find the limits of that color, giving top x1-x2 and bottom x1-x2. The code then takes the y1-y2 difference between the two sprockets and does an arctangent calculation to render the rotation percentage required to straighten the image.

```
// Save the current preferences
var startRulerUnits = app.preferences.rulerUnits;
// Set Photoshop to use pixels
app.preferences.rulerUnits = Units.PIXELS;
Main();
// Return the app preferences
app.preferences.rulerUnits = startRulerUnits;

function Main() {
if (app.activeDocument.colorSamplers.length!=2) {
alert('two color sampler points are required');
return;
}
```

```
else {
point1 = [];
point2 = [];

var mySampler = app.activeDocument.colorSamplers[0];  //find out where x y coordinates of
first sampler are
var samptx =mySampler.position[0];
var sampty =mySampler.position[1];
var sampx = parseInt(samptx);
var sampy = parseInt(sampty);

for (var s=0,len=app.activeDocument.colorSamplers.length;s<len;s++) {
var colorSamplerRef = app.activeDocument.colorSamplers[s];
MagicWand( colorSamplerRef.position[0].value ,colorSamplerRef.position[1].value);
if (s==0) point1=TopLeft();
else point2=TopLeft();
app.activeDocument.selection.deselect();
  }
  Rwidth = point2[0]-point1[0];
  Rheight = point2[1]-point1[1];
  degrees = Math.atan(Rwidth/Rheight) * 180 / Math.PI;
  activeDocument.activeLayer.isBackgroundLayer=0; // Make it a normal Layer
  app.activeDocument.activeLayer.rotate(degrees);
}

  activeDocument.selection.selectAll();
   var file = app.activeDocument;
   var selec =  file.selection;
   var newRect = [ [sampx-300, sampy -500], [sampx-300 , sampy+12520], [sampx + 1400,
sampy + 12520], [sampx + 1400, sampy - 500] ]; // get selection for cropping so that all images
are same size
   selec.deselect;
   selec.select(newRect);

 }
function TopLeft() {
try{
TL=[];
var SB = app.activeDocument.selection.bounds;
TL[0]=SB[0].value;
TL[1]=SB[1].value;
return TL;
}
catch(e){}
}
```

```
function MagicWand(Xpoint,Ypoint) {
// =======================================================
var idsetd = charIDToTypeID( "setd" );
    var desc121 = new ActionDescriptor();
    var idnull = charIDToTypeID( "null" );
        var ref33 = new ActionReference();
        var idChnl = charIDToTypeID( "Chnl" );
        var idfsel = charIDToTypeID( "fsel" );
        ref33.putProperty( idChnl, idfsel );
    desc121.putReference( idnull, ref33 );
    var idT = charIDToTypeID( "T    " );
        var desc122 = new ActionDescriptor();
        var idHrzn = charIDToTypeID( "Hrzn" );
        var idPxl = charIDToTypeID( "#Pxl" );
        desc122.putUnitDouble( idHrzn, idPxl, Xpoint );
        var idVrtc = charIDToTypeID( "Vrtc" );
        var idPxl = charIDToTypeID( "#Pxl" );
        desc122.putUnitDouble( idVrtc, idPxl, Ypoint );
    var idPnt = charIDToTypeID( "Pnt " );
    desc121.putObject( idT, idPnt, desc122 );
    var idTlrn = charIDToTypeID( "Tlrn" );
    desc121.putInteger( idTlrn, 20 );
    var idAntA = charIDToTypeID( "AntA" );
    desc121.putBoolean( idAntA, true );
executeAction( idsetd, desc121, DialogModes.NO );
}
```

Fig. 10. Straightening script with additions by John Mack. Film examples and the script files are at http://keneckert.com/kenfilms.

A small section is added near the end for insurance. If the script already knows where the first sprocket begins through the first color sampler, this is a good time to situate it at a consistent x and y position so that each successive sprocket will be in roughly the same place for each strip file; thus the script shaves off a few pixels to nudge the strip upward and leftward into position. This adjustment will be important for registration when we cut out the frames—ideally I wanted each first sprocket to start at 100 x, 360 y. Frame separation will be done by selecting the sprocket area and then expanding the selection by a predetermined size outward to include the image, and then saving the selection as a separate file.

I loaded groups of strips into Photoshop and for each strip I manually clicked the two color samplers and executed the straightening script with a function key. To simplify, the steps are:

- Use the magic wand to click inside the top and bottom sprocket holes
- Run the straightening script; save

These steps can also in turn be recorded as an action and run as a batch command, and files can be quickly visually inspected later:

- Open a test strip file and initiate recording of a Photoshop action:
    1. Use the magic wand to click inside the top and bottom sprocket holes
    2. Run the straightening script and save the file
- Stop recording the completed action
- Run a batch command executing this action on each file in the directory.

Cutting out Frames. After straightening the strips, we need to digitally cut out each frame and save it into its own image file so that we can string them together in Virtual Dub or another video editor. This actually turns out to be easier than straightening the strips, for if the sprockets are at reasonably consistent positions after straightening, selecting all 16/17 can be programmed with an action. Again, magic wand only requires that you be close, and to click somewhere inside the sprocket hole. Once magic wand has created a rectangular selection of the sprocket hole, we just need to expand the selection by a predetermined size to capture the entire film area.

Fig. 11. Expansion of the sprocket area to predetermined coordinates

```
#target photoshop
if (documents.length == 0) {
   alert("nothing opened");
} else {
   // start

   //setup
   var file = app.activeDocument;
   var selec =  file.selection;

   //run
   var bnds = selec.bounds; // get the bounds of current selection
   var // save the particular pixel values
      xLeft = bnds[0],
      yTop = bnds[1],
      xRight = bnds[2],
      yBottom = bnds[3];
```

```
   var newRect = [ [xLeft-95,yTop -300], [xLeft-95,yTop+550], [xLeft + 1400,yTop + 550],
[xLeft + 1400,yTop - 300] ]; // set coords for selection, counter-clockwise

   selec.deselect;
   selec.select(newRect);

   // end
}
```

Note: For regular 8mm film, use values: var newRect = [[xLeft-180,yTop +70], [xLeft-180,yTop+870], [xLeft + 1350,yTop + 870], [xLeft + 1350,yTop +70]].

Fig. 12. Script for clipping out frames (selection tolerance set to ten)

Figure 12 shows the cut-out script for Photoshop, which enlarges the selection to include the entire image area for each frame, with the sprocket hole included for visual checking. The excess area can be cropped later as desired. This script can then be iterated 16/17 times as an action and run as a batch command on each scanned strip, with the following programmed steps in the action:

- Select frame one sprocket hole with magic wand
- Execute script to extend selection size to entire frame
- Crop and save
- Undo to restore entire strip
- Select frame two sprocket hole and repeat steps, up to frame 16/17

As this is CPU and hard disk intensive, the batch may take several minutes to process through a series of files. On an Intel i5 8600 PC with a solid state drive I averaged 3 fps. One of Photoshop CS6's annoyances is that it has no simple function for saving a selection, and so the entire file must be cropped down to the selection, saved, and then restored with undo. This not only slows the program, it also causes Photoshop to try to rewrite all 16/17 frames with the same file name. The workaround is to instruct the batch to append an alphabetic letter to each frame save, which will result in an ungainly mélange of letters after each set of 16/17 frames, but will sequence the frames correctly so that File Explorer or another utility can rename them later with a consistent title. Numbers cannot be appended instead of letters, as they will be confused with the numbers referencing the film strips.

In Photoshop's batch menu box:

- Destination: Folder
- File naming: Document Name + Serial Letter (a, b, c . . .) + extension
- After frames are rendered: Select all files; right click on first file; select "rename" from menu and rename all files.

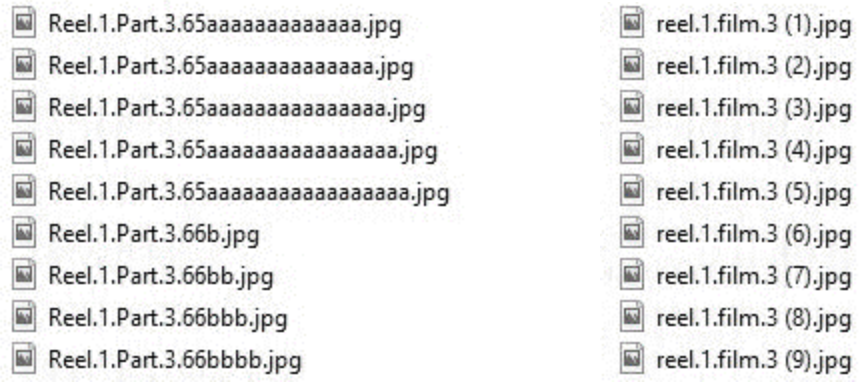

Fig. 13. Raw outputted image files (left) and renamed image files (right)

While this is perhaps not the most elegant code, it functions. Photoshop CS6 for some reason crashes on every ninety-third strip file during batching, and so the program either needs to be periodically restarted or the strips need to be parceled into separate directories. The script can also become confused when it encounters a bad splice or damaged sprocket hole, as well as when it encounters perfectly clear leader, which it will interpret as the same white color as within the sprocket holes—I set selection tolerance to ten, but it can be adjusted. I also resorted to writing smaller mini-actions that re-center mangled frames or add padding on frames with missing and damaged edges—Virtual Dub will later require that each frame be the same pixel dimensions in order to create a video file. Some very bad frames simply need to be manually cut out and saved. Yet overall the script gave an error rate of under 1 percent, and because 8mm sprocket holes are larger, 8mm often gives only a few bungled frames over an entire reel.

Post-Processing. I shot some footage in Kailua-Kona in 1992 on my Brownie 8mm camera, which had all-manual settings, and chose far too high an f-stop. Another useful feature in Photoshop is the ability to salvage badly overexposed film. This is done by choosing the yin-yang-looking moon icon under layers, choosing *levels* as the option, and then changing from *normal* to *multiply* in the settings next to the layer image. The level of darkening can be slid, and the repair process can be batch executed as an action on a directory of overexposed strips. Underexposed film can also be lightened by choosing *screen* instead of *normal*. Free or inexpensive plug-ins can also be found for Virtual Dub to find and clean frame dust and scratches or other detritus, or manual cleaning can be employed with Photoshop's filters or its clone or spot repair tools. This is particularly necessary after darkening through the multiply command, which tends to magnify dust. Color filtering can additionally be applied or automated where filming was done with tungsten lamps.

Checksums on frames can be done by dividing the frame image file number size by 16 or 17 to verify against the number of strip files. In this case, the final product was two terabytes of raw scan strip TIFs, straightened and cleaned strip TIFs, and JPEG frames. These were stored twice on two external drives, and a complete third set was compressed into JPEGs and backed up onto DVD-R. Long-term storage will vary of course depending on personal or institutional resources.

Conclusion

In 1983 I found at an Edmonton, Canada garage sale a 1930s-era hand-cranked Moviegraph home 16mm projector, with three short 1930s reels of Charlie Chaplin and a Mickey McGuire comedy on Agfa safety film. As the film was in good condition, but in places was no longer

projectable because of ripped sprockets, it was a good candidate for flatbed scanning. The digitization process described here worked the same for silent 16mm, except that the reels are wider and on an 8.2 cm scanning space only eight 16mm frames per scan may be saved. The scripts were also easily edited to accommodate larger frame sizes, with the added benefit that larger-gauge film stocks allow more margin of error. With easily damaged (or highly flammable nitrate) film, avoiding travel of rare or sentimentally valuable reels might be an issue in developing an in-house film scanning system. As can be seen in the YouTube video of the transfer (YouTube search string: "Mickey McGuire Farm Comedy Excerpt"), flatbed scanning salvaged damaged film that might otherwise be unusable.

This described transfer process for small-gauge movie film will hopefully encourage further workflow and coding improvements. It is unfortunately not a free solution, as it requires a flatbed scanner with a transparency adapter, a computer, and Photoshop. It also offers no accommodation for Super 8 sound film, unless the sound can be recorded separately and then resynchronized. Its additional downside is that it is highly labor intensive; scanning, straightening, and frame division of a single 200-foot reel of film might easily take fifteen to twenty hours. As I have stressed, its workflow needs to be time-efficient because of the number of frames and steps involved, and at some point good enough has to be sufficient. Yet the process allows high-quality film digitization for non-technically expert individuals or for organizations on minimal budgets, and may be of especial use where it is desired that film materials remain secured.

Attempting perfection is also somehow perverse for such a rough-and-tumble format as 8mm/Super 8. During post-processing it is easy to cross an ethical boundary between restoration of the filmed image and improvement on it, for the limitations of the cameras with their typically imprecise focus and exposure and loose sprocket registration are soon noticed. Dirt in the film may have originated in the *camera* gate and be photographed onto it. Eight millimeter/Super 8 movie film will never look as clean as digital video will; but at some level, this is a feature. Part of its charm is its gritty, earthy feel, and some people download smartphone filters to add this scratchy, filmic look to high definition video. With 8mm/Super 8, such a look is already embedded, and whether it is depicting a president's death, a film school project, or someone's childhood or wedding from decades earlier, its preservation deserves respect.