# Film Restoration

Theo Turner

February 2018

## 1 Introduction

This document details the author's attempt at an algorithm that can be used to restore old films. Over the course of their lifetime, films can deteriorate as a result of environmental hazards, for example chemical instabilities, humidity and dust. The resulting defects can be remedied using a number of techniques, such as correction of global flicker, blotch reparation, correction of vertical artefacts and video stabilisation. The algorithm detailed here, which utilises all of the above, is implemented in MATLAB. The code for the algorithm is attached to this document, which includes step-by-step descriptions of the functions used as code comments. In the figures presented in this document, the original is placed on the left and the corrected version on the right.

## 2 Detection of scene cuts

While scene cuts are not defects, detecting them serves as a good introduction to the programmatic restoration of old films.

Detection of scene cuts can be achieved quite simply by comparing each pair of frames in sequence. Note that each frame is loaded as a matrix representing the intensity of each pixel in it. First, a threshold value for the intensity change of a pixel to be considered significant, as well as a threshold for the proportion of pixels in the entire frame to indicate a scene transition, is set. In this case, an intensity change of 0.1 and a 75% count of changed pixels was used.

For each pair of frames, the algorithm loops through the pixels and measures their intensity. The absolute intensity difference per pixel between the two frames is computed. If this is greater than the pixel threshold, the pixel is considered significantly different between the frames and the count of changed pixels is incremented. Once all pixels have been looped through, the proportion of significantly changed pixels relative to the total number of pixels in the frame is calculated. If this is greater than the scene change threshold, the point in the sequence of the frames is marked as a scene cut. These scene cuts are saved as an array of frame indices.

Text is overlayed on the video to indicate when each scene cut has occurred. After each cut, marked by the scene change array, 30 frames have text placed on them to allow enough time to read.


Figure 1: The text "scene cut occurred" is overlayed after each scene cut is detected.

# 3 Correction of global flicker

Intensity flicker, fluctuation in image brightness that does not originate from the original scene, may result from film aging, copying, and dust among other causes. Affected film does not have a constant global luminosity over time.

Intensity flicker can be corrected by a similar method to detection of scene cuts, albeit in this case flicker affects the entire frame uniformly, so looping through individual pixels is unnecessary. Again considering pairs of frames in sequence, the mean intensity of each frame is computed and the (signed) difference between them is calculated. This intensity difference is then added to the first frame in each pair, thus reducing the inter-frame change in global luminosity.

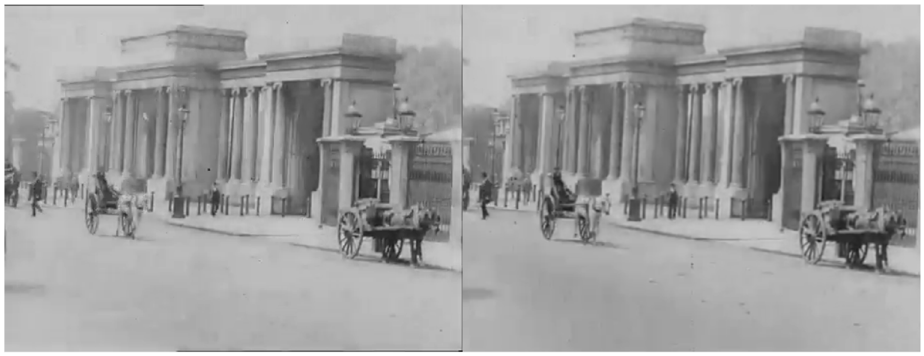

Figure 2: The image increases suddenly in intensity in this frame. The rapid change in luminosity is mitigated, as seen in the right image. Also see Figure 1 above.

# 4 Correction of blotches

Blotches, spots on the image caused by dirt or the loss of the film's gelatin cover (as a result of aging or poor film quality) is a common video artefact. The removal of these involves the filling of 'gaps' in the picture.

The first step in correcting blotches is detecting them. This is rudimentary: intensity disparity compared to the previous frame is measured as before. This is done by taking the absolute difference between the frames. However, this produces false positives due to objects moving in the frame. As such, a movement mask is applied to the frame to detect moving objects, and pixels covered by this mask do not have the blotch correction algorithm applied to them. The movement mask is computed on a per-neighbourhood-of-pixels basis, with moving objects marked using the absolute frame difference in each neighbourhood. Finally, in order to overwrite artefacts produced at the edges of the mask, it is dilated and a mean filter is applied.

In order to actually correct blotches, the missing frame data needs to be overwritten. The simplest approach giving a natural-looking result is to overwrite the blotches with content from the preceding or following frame. However, note that some blotches last for more than a single frame. Therefore, an average of the two preceding and two following frames have been taken instead to overwrite the missing data.

Figure 3: This large blotch is made near-invisible by overwriting it with the average of the surrounding frames.

Figure 4: Several small blotches are corrected in this frame.

## 5 Correction of vertical artefacts

Vertical artefacts are unwanted vertical stripes appearing across the entire frame. Due to difficulty in detecting them directly, a frame-wide filter is required to correct them.

By separating the image into rows, an intensity curve across the image can be produced. By applying a one-dimensional median filter to each of these, which smooths the curve while keeping the important edges, large changes in intensity between adjacent pixels is reduced, thereby masking the artefacts. This has the undesired effect of making the image appear less clear, so it is sharpened using a Laplacian filter. The Laplacian filter also brightens the image, so it's intensity is adjusted down at the end of the function.

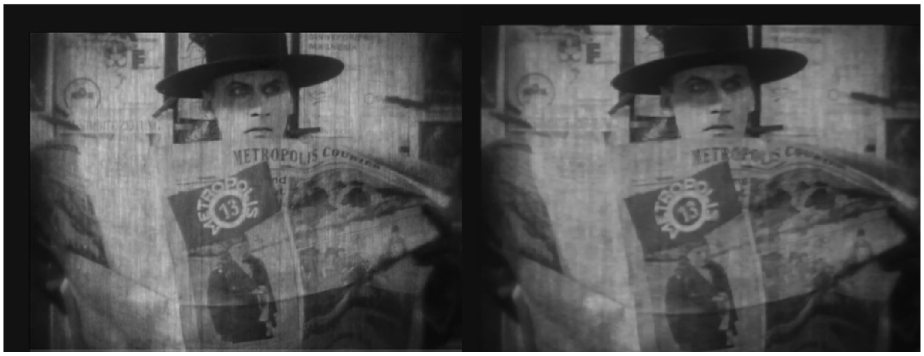


Figure 5: The vertical artefacts in the third scene are reduced using a median filter.

# 6 Correction of camera shake

Camera shake is unstable video. In order to correct it, video frames should be aligned together in order to remove the unwanted motion.

Point feature matching is used to stabilise the video. In each frame, salient points are collected using corner detection. Mappings between these points are computed using fast retina keypoint descriptors, which are binary, allowing Hamming distance to serve as a simple way to calculate matching cost. The inter-frame distortion is approximated through MATLAB's RANSAC variant function, with the affine transform represented by a scale-rotation-translation matrix. The video stabilisation transform used in this function is based on publications from Boston university [1] and the National Taiwan University [2].


Figure 6: The first frame in a pair where there is significant camera shake.


Figure 7: The second frame in a pair where there is significant camera shake. Notice that the scene in the corrected version stays in place.

## References

[1] A. Litvin, J. Konrad, and W. C. Karl, “Probabilistic video stabilization using kalman filtering and mosaicing,” in *Image and Video Communications and Processing 2003*, vol. 5022. International Society for Optics and Photonics, 2003, pp. 663–675.

[2] K.-Y. Lee, Y.-Y. Chuang, B.-Y. Chen, and M. Ouhyoung, “Video stabilization using robust feature trajectories,” in *Computer Vision, 2009 IEEE 12th International Conference on.* IEEE, 2009, pp. 1397–1404.