A NEW METHOD FOR A FAST DETECTION AND SEAMLESS RESTORATION OF LINE SCRATCHES IN MOTION PICTURES

S. Milady* and S. Kasaei**
*University of Bremen, **Sharif University of Technology
*Otto-Hahn Alee, NW1, D28359, Bremen, **Azadi St., Tehran
*Germany, **Iran
skasaei@sharif.edu

Abstract

Line scratches are common artefacts in old motion pictures that may produce a very annoying effect on the viewer. In this paper, we will introduce a new method for fast detection and high quality restoration of this kind of degradation. Our detection method relies on the statistical characteristics of the intensity of pixels in the scratched areas of frames. For high quality restoration of the scratches we use a mixed adaptive stochastic AR-Median model to restore the scratched areas. The proposed detection algorithm is fast and more efficient compared to the other available approaches. The main contribution of our restoration algorithm is its ability to jointly adapt the image model to the data and concurrently restore the missing pixels.

Key Words
Line scratch, AR model, median filter, restoration, Bayesian sampling.

1. Introduction

Line scratches are a common type of degradations in archived motion picture films. These kinds of artefacts that are mainly caused by the abrasion of the film by some objects in the projector or the camera's mechanism result in more or less vertical lines with different intensities from the neighbourhood pixels in the same frame and usually exist over much of the image or even over many consecutive frames. According to the film type, these scratches appear in two types: *bright scratches* and *dark scratches*. If the film is of positive type the scratches seem bright and if it is negative they seem dark. In this paper, without loss of generality, we assume that we have dark scratches. The techniques discussed here are simply applicable to bright scratches by inverting the pictures and then applying the techniques to them and finally inverting the restored picture.
In general if we apply a restoration method to all of the pixels in the degraded frame, naturally we will introduce some blur to the undegraded parts of the image. In order to prevent this we must apply the *restoration algorithm* only to the scratched parts of the images. So it is necessary to detect the positions of the scratched pixels by a so-called *detection algorithm*, before applying any restoration algorithm.
The main problem in the detection of line scratches is their similarity to vertical features of the image. In some cases even for a human observer it may be hard to distinguish between a line scratch and a vertical feature (*e.g.*, in Fig. 1 look at the line scratch in the image near the door edge and its similarity to the door frame).
As mentioned above, because of the mechanism of the line scratch production, a line scratch usually remains in a similar positions in the consecutive frames; and covers a large part of the image.

Figure 1. A frame from a motion picture film degraded by line scratches.

Because of these special characteristics of line scratches, the restoration and detection methods will be completely different than the other degradations (*e.g.*, blotches [1]) in motion picture films. Consequently, if we use a relatively low efficient method for restoring the scratched pixels of the image, although we may have good results statically but dynamically (*i.e.*, during the re-projection of the restored image sequence) the restored area may be perceived easily (in contrast with blotches that are also compensated by the previous and the next frames).
Several attempts have been made [1-4] to resolve these problems. In the following we give a short discussion of the major existing techniques.
Morris [2] assumes that the intensity of each pixel within the line scratch area is the result of multiplying a factor

d (that is assumed to be fixed for each line scratch) by the intensity of the corresponding pixel of the undegraded image. He then tries to estimate the number of line scratches in the image and their corresponding multiplicative factors using a *Markov Chain Monte Carlo* (MCMC) method. To restore, the intensities of degraded pixels of each line scratch are multiplied by the inverse of the corresponding d factor. The method needs so many iterations to converge, furthermore because of using a simple assumption for the scratch effect model, there would be false detection alarms at the locations of the scratch-like features of the image. Furthermore, for truly detected scratches the restoration results are not of sufficient quality.

Kokaram [1,2] uses a two-stage method to reduce the large number of iterations in the *Morris* method and also uses a more complicated model (a damped sinusoid) for the effect of the scratches on the image pixel intensities to reduce the false detection alarms. After finding the line scratch positions, he uses a 2D-AR model to restore the scratches. The *Kokaram*'s model for line scratches is so much restricted to match all kinds of the scratches, so in some cases the scratches may not be detected, (*e.g.*, our implementation of this algorithm for the image shown on Fig. 1 was not able to distinguish the edge of the door from the scratch near it).

Joyeux et al. [4] use a Kalman filter to detect line scratches. They use a spatio-temporal method to find and track the pixels that may belong to a scratch. At first by a preprocessing algorithm the points that may belong to a line scratch are produced and then used as the input to the Kalman filter. If these points match the sinusoid model for the scratch trajectories along consecutive frames of the scratched sequence, the Kalman filter tracks the line scratch, otherwise it is diverged and the candidate line scratch is rejected. After the detection stage, the restoration is performed in two low and high frequency stages [4]. This method produces good results compared with the two previous one, but the tracking algorithm that is used for detection is a relatively complicated algorithm and in cases that the line scratch undergoes an intermittent disappearing from a portion of the frame may lead to divergence. Although the restoration algorithm performs relatively good but the restored area in some cases is again too smooth and can be perceived.

There are several other techniques [5] which are relatively simpler than the methods stated above. And they are just suitable for estimation of the low frequency components of the corrupted parts of the image sequences.

In the following sections we introduce our proposed methods for detecting and interpolating the line scratches.

2. Line Scratch Detection

We have examined over 1300 scanned frames (with resolution 576×720) of different types of real degraded movies and have considered the characteristics of the line scratches within them. Here, we summarize the result of this statistical investigation and introduce 4 criteria which are as follows:

(I) The slopes of the majority of line scratches fall within the interval $[-0.01, +0.01]$.

(II) The width of line scratches are less than 5 pixels.

(III) The rate of changes of the intensity of the pixels within the line scratches along the horizontal direction is usually much faster and oscillatory than that of other pixels within vertical image edges.

(IV) In the center of line scratches, the intensity of the pixels shows a little fluctuation and remains almost constant. The level of the intensity values for line scratches is higher than that of image edges.

According to these 4 criteria, we now formulate our proposed method for line scratch detection .The justification of each step of the proposed algorithm is shown as a number referring to one or more of the 4 criteria in parenthesis after each step.

1. Apply a 5-tap median filter in the horizontal direction on the degraded image , to compute the resulting image M (for boundary pixels modify the number of taps), as:

$$M(i,j) = med\{\ G(i,j-l)\ \big|\ l \in \{-2,-1,0,+1,+2\}\ \} \quad (1)$$

where i and j represent the row and column indices of the image and $med\{.\}$ denotes the median of its arguments (I, II , IV).

2. Define the binary image B as follows:

$$B(i,j) = \begin{cases} 1 & if \quad G(i,j) - M(i,j) \geq T1 \\ 0 & if \quad G(i,j) - M(i,j) < T1 \end{cases} \quad (2)$$

where $T1$ is a threshold value and is chosen low enough to detect all the edge points and scratches (I).

3. Apply a morphological opening with a vertical 1×3 points structuring element on B to reduce the non-vertical edge points. Store the result in B^{open} and define the following vector:

$$N(j) = \sum_{i=1}^{row} B^{open}(i,j), \qquad j = 1,2,\ldots,col \quad (3)$$

where row and col are the number of rows and columns of the image, respectively. In fact $N(j)$ represents the number of 1's in each column of the B^{open} . Then, define a new binary vector d as:

$$d(j) = \begin{cases} 1 & if \quad N(j) \geq \mu + \sigma \\ 0 & if \quad N(j) < \mu + \sigma \end{cases}, \quad j = 1,2,\ldots,col \quad (4)$$

where μ and ς represent the mean and standard deviation of the N 's component, respectively (I, IV).

4. After computing the d vector, as mentioned above, if the non-zero components of d fall within 3 neighbouring columns we set all of those component equal to zero except the component for which the corresponding number of non-zero pixels in the column (*i.e.*, the corresponding component of N) has the greatest value (II).

5. Define another vector k as follows:

$$k(j) = \frac{1}{row} \sum_{i=1}^{row} (G(i,j) - \mu_G(j))^4 \, , \, j = 1,2,..., col \tag{5}$$

where $\mu_G(j)$ is the average of the intensities of j th column of the G image. $k(j)$ is the well-known statistical measure, so called *kurtosis.* We use (5) to measure the correlation of the intensity of pixels along the vertical direction. It is worth to mention that, instead of using this measure we could use the variance (like [5]), but experimentally we found out that better results are obtained using the kurtosis measure (III, IV).

6. Define an oscillation measure OS for the kurtosis vector as follows:

$$OS(j) = \sum_{l=-2}^{2} \sum_{\substack{m=-2 \\ m \neq l}}^{2} \frac{|k(j+l) - k(j+m)|}{(l-m)^2} \, , \, j = 1,2,..., col \tag{6}$$

In fact this step constructs the main part of the detection algorithm.

7. In the columns of the image where the corresponding components of the d vector is 1, we say that we have detected line scratches if at those places the OS value exceeds a threshold $T2$ that is determined experimentally.

3. Restoration of Line Scratches

3.1 The image model

In this section we propose a new stochastic model for images. This model is able to model both the local smoothness of the images and other nonlinear characteristics of them (like the edges) in an adaptive manner.

Our image model is a combination of two well known models for images: the AR linear model and the *median* non-linear model.

The AR model is able to model the local smoothness of the images and the median model is able to model the abrupt changes and local fluctuations of the pixel intensities, which are seen in almost all films because of the structure of the film material.

We assume that the intensity of a pixel in an undegraded image I at the position $x = [i\,, j]$ can be computed by the following equation:

$$I(x) = \beta [\sum_{i=1}^{3} \alpha_i \, I(x + q_i)] + (1-\beta) \, med \, \{I(x + q_x) \, | \, i = 1,2,3\} + e(x) \tag{7}$$

where α_i 's are the AR model coefficient (see [2]), q_i 's are three offset vectors, representing the three nearest causal neighborhood pixels of the pixel at location x and are equal to $q_1 = [0,-1]$, $q_2 = [-1,0]$, and $q_1 = [-1,-1]$, respectively. The $med\{.\}$, as before, indicates the median value of its arguments and here is equal to the median of the intensities of the three causal neighborhood pixels of the AR model, and finally $e(x)$ is an additive *i.i.d* Gaussian noise with zero mean and variance σ_e^2 .

The role of the β parameter in this model is very critical, in fact it is this parameter which adjust the model from being a purely AR model ($\beta = 1$), or being purely nonlinear median filter ($\beta = 0$).

It is evident that the α_i coefficients, β , σ_e^2 must be determined in a suitable manner. For simplicity, we consider all of these parameters as a single vector $\mathbf{P}$.

Also in our case there are some unknown pixels which we (following the *Kokaram*'s notation) designate them by the vector $\mathbf{i}_u$ and the other remaining known pixels are represented by $\mathbf{i}_k$.

Assuming the model of (7) we compute the joint probability distribution of the vector $\mathbf{i}_u$ and $\mathbf{P}$ given the vector $\mathbf{i}_k$ using the *Bayes* rule as follows.

$$p(\mathbf{i}_u, \mathbf{P} | \mathbf{i}_k) \propto p(\mathbf{i}_k | \mathbf{P}, \mathbf{i}_u) \, p(\mathbf{P}) \, p(\mathbf{i}_u) \tag{8}$$

where we have assumed that the $\mathbf{P}$ and $\mathbf{i}_u$ random vectors are independent. We also assume uniform priors for $\mathbf{i}_u$ and all of the components of the $\mathbf{P}$ vector except for the variance σ_e^2 , that we assume a *Jeffery*'s [2] prior $p(\sigma_e^2) = 1/\sigma_e^2$.

3.2 Restoration

In the restoration process, we try to find the *Maximum a priory* (MAP) estimation for $\mathbf{i}_u$ and $\mathbf{P}$ given the known pixels of the image (8). If we were able to do so, we could estimate the intensity of the unknown pixels using a model which has been adapted to the known image data. However, because of the nonlinear nature of (7), computing the probability density function of (8) analytically is not possible. So we use the *Gibbs* sampler [1, 2, 3], which is a numerical method for getting samples from the probability density functions, and after sufficient iterations we can compute the MAP solution numerically. The sampler iteratively samples from the following distributions.

$$p(I(x)\left|\mathbf{i}_{-x}, \mathbf{P}\right) \qquad \forall x \in Scratched\ Area \tag{9}$$

$$p(\alpha_i \left|\mathbf{i}, \mathbf{P}_{-\alpha_i}\right) \qquad i = 1,2,3 \tag{10}$$

$$p(\beta \left|\mathbf{i}, \mathbf{P}_{-\beta}\right) \tag{11}$$

$$p(\sigma_e^2 \left|\mathbf{i}, \mathbf{P}_{-\sigma_e^2}\right) \tag{12}$$

where $\mathbf{i}$ is the vector containing all the information about the known and unknown pixels of the image, $\mathbf{i}_{-x}$ represents all of the pixels except the pixel at position x and other notations have similar meanings.

For computing the probability density functions of equation (8) we have used the pseudo-likelihood approximation the results after computations are as follows:

$I(x)$ in (8) has a normal distribution with variance σ_e^2 and mean μ_x that is computed from the following equation.

$$\mu_x = \beta\left(\sum_{i=1}^{3} \alpha_i I(x+q_i)\right) + (1-\beta) med\left\{I(x+q_i)\middle| i = 1,2,3\right\} \tag{13}$$

α_i in (8) has also a normal distribution with the following parameters.

$$\mu_{\alpha_i} = \frac{\sum_x I(x+q_i)R_i(x)}{\beta \sum_x I^2(x+q_i)}, \; \sigma_{\alpha_i}^2 = \frac{\sigma_e^2}{\beta^2 \sum_x I^2(x+q_i)} \tag{14}$$

where

$$R_i(x) = I(x) - \beta \sum_{\substack{j=1 \\ j \neq i}}^{3} \alpha_j I(x+q_i) - (1-\beta)\, med\left\{I(x+q_i)\middle| i = 1,2,3\right. \tag{15}$$

β is also normally distributed with the following parameters.

$$\mu_B = \frac{\sum_x D(x)D'(x)}{\sum_x D^2(x)}, \sigma_B^2 = \frac{\sigma_e^2}{\sum_x D^2(x)} \tag{16}$$

where

$$D(x) = \sum_{i=1}^{3} \alpha_i I(x-q_i) - med\left\{I(x-q_i)\middle| i = 1,2,3\right\} \tag{17}$$

and

$$D'(x) = I(x) - med\left\{I(x+q_i)\middle| i = 1,2,3\right\} \tag{18}$$

The inverse squared of σ_e^2 (*i.e.*, $z = 1/\sigma_e{}^2$) has a Gamma distribution[1] with the following parameters.

$$a = (N - 1)/2, \; b = 1/\left(\sum_x \frac{(I(x)-\mu_x)^2}{2}\right) \tag{19}$$

where N is the number of all the pixels in the image, μ_x is defined by Eq. (13) and the summation is over all of the pixels in the image.

For faster convergence of the algorithm we used the initial state.

4. Experimental Results

We have implemented our algorithms in MATLAB 6.5 on a Pentium IV, 2GHz, 512MB RAM, running Windows NT.

We compared our detection algorithm with the *Kokaram*'s detection method [1] that is one of the most effective algorithms for the detection of the line scratches. The time required for rejecting or accepting each proposed line in our archived images using the *Kokaram*'s method running on our system is about 10 seconds and increases proportionally with the line scratch candidates of the image. However, our algorithm runs in about 3 seconds and this time is independent of the type of images.

The result of implementation of our restoration method on the degraded image in Fig.1 is shown in Fig.2. It is worth to mention that we have implemented the restoration algorithms on several non-overlapping blocks of size 10, which is equal to the width of the line scratch plus the neighboring pixels (2 for our causal 3-tap model). For comparison purposes we applied *Joyeux* [4] and *Kokaram* [1, 2] methods and also our restoration algorithm on more than 20 images of our archive and used a subjective scaling method based on criteria offered in [6]. A group of students compared the restored images and gave points between 1 (for the poorest quality) and 7 (for the best quality). The results are shown in Table 1. For exact comparison of the restored images using different methods, the magnified part of a portion of Fig. 2 (that is specified with a rectangle) are shown in Fig. 3. We have chosen this part of the image because there is an edge (the edge between the hand of the actor and the background that is corrupted with a scratch). This figure clearly shows the remaining of the line scratch in the image restored by *Kokaram*'s method (Fig. 3(a)). The result of the *Joeyux*'s is much better than the *Kokaram*'s method but it still smoothes the edge (Fig. 3(b)). As we mentioned above, although the smoothed edge may be hard to be recognized in a still image, the remaining of the scratched regions of the restored image can be perceived during the reprojection of the film; because of the almost constant position of the line scratch. The result of our restoration algorithm is shown in Fig. 3(c), here it can be seen that the edge is not over smoothed and also there is no maintenance of the scratch. That is because the algorithm uses an adaptive method and in the regions that there is

---

[1] The PDF of a Gamma distribution with parameters a and b is proportional to $x^{a-1} e^{-\frac{x}{b}}$.

abrupt changes (edges), the weight of the median filter $(1-\beta)$ would be higher.

Table 1. Comparison of the restoration methods.

Methods	Comparison Results
Joeyux Method	6.2
AR Kokaram method	5.0
Proposed method	5.5

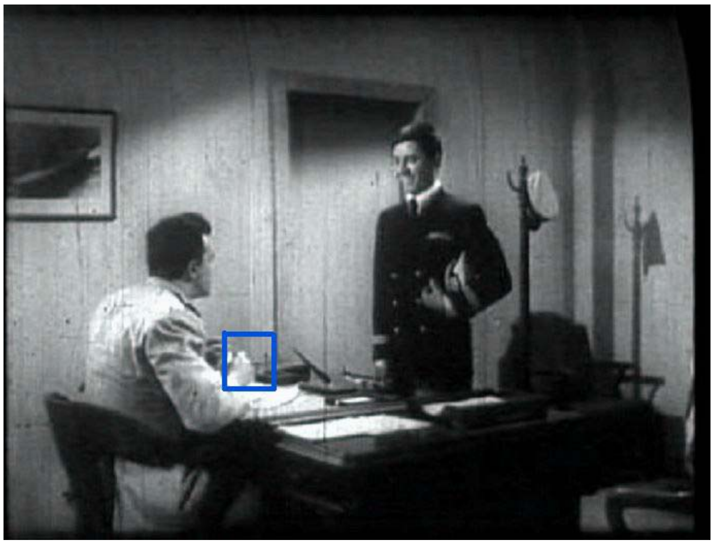

Figure 2. Restored image.

(a) (b) (c)

Figure 3. A restored portion of the degraded frame using

(a) *Kokaram* AR method; (b) *Joeyux* method; (c) Proposed method.

5. Conclusion

In this paper, we introduced a method for detecting and restoring the line scratches in degraded motion picture films. Our detection method is a simple and fast method based on the statistics of the intensity of the pixels within the scratched areas. One of the main advantages of this method, when compared with the current algorithms, is the independence of the running time of the algorithm with respect to the type of images. Our restoration method is an adaptive method that according to the information in the neighbourhood pixels of the scratched portion of the image is able to reach a compromise between a linear AR model and a nonlinear median model for image. The adaptive use of the median and AR model results in a better quality in restored images (especially in preserving the edges) and this is the main contribution of this work. In spite of this, the number of iteration for the restoration algorithm to converge is in the order of the other sampling methods such as AR and MRF models.

References

[1] A.C. Kokaram, Detection and removal of line scratches in degraded motion picture sequences: *Proc. Signal Processing VIII - 8th European Signal Processing Conference (EUSIPCO'96),* vol. 1, Trieste, Sept. 1996, 5-8.

[2] A.C. Kokaram, Removal of line artefacts for digital dissemination of archived film and video: *Proc IEEE Int. Conf. on Multimedia Computing and Systems (ICMCS'99)*, vol. 2, Florence, Italy, 1999, 245-249.

[3] R.D. Morris, W.J. Fitzgerald, and A.C. Kokaram, A sampling based approach to line scratch removal frames: *IEEE International Conference on Image Processing 1*, 1996, 801-804.

[4] L. Joyeux, S. Boukir, and B. Besserer, Film line scratch removal using Kalman filtering and Bayesian restoration: *IEEE Workshop on the Application of Computer Vision(WACV2000),* Palm Springs, California, 2000.

[5] D. Tegolo, et al., Scratch detection and removal from static images using simple statistics and genetic algorithms: *Proc. IEEE International Conference on Image Processing (ICIP2001)*, vol. 1, Thessaloniki, Greece, 2001, 265-268.

[6] A.K. Jain, *Fundamental of digital image processing*, (Englewood Cliffs, NJ: Prentice-Hall, 1989).