Detection and Tracking Scheme for Line Scratch Removal in an Image Sequence

Bernard Besserer and Cedric Thiré

Laboratoire Informatique, Image, Interaction (L3i), Université La Rochelle,
17042 La Rochelle cedex 1, France
{bernard.besserer,cedric.thire}@univ-lr.fr
http://www.univ-lr.fr

Abstract. A detection and tracking approach is proposed for line scratch removal in a digital film restoration process. Unlike random impulsive distortions such as dirt spots, line scratch artifacts persist across several frames. Hence, motion compensated methods will fail, as well as single-frame methods if scratches are unsteady or fragmented.
The proposed method uses as input projections of each image of the input sequence. First, a 1D-extrema detector provides candidates. Next, a MHT (Multiple Hypothesis Tracker) uses these candidates to create and keep multiple hypothesis. As the tracking goes further through the sequence, each hypothesis gains or looses evidence. To avoid a combinatorial explosion, the hypothesis tree is sequentially pruned, preserving a list of the best ones. An energy function (quality of the candidates, comparison to a model) is used for the path hypothesis sorting. As hypothesis are set up at each iteration, even if no information is available, a tracked path might cross gaps (missed detection or speckled scratches). At last, the tracking stage feeds the correction process. Since this contribution focus on the detection stage, only tracking results are given.

1 Introduction

Despite of fast-growing use of digital media, the photochemical film is still the storage base in the motion picture industry and several million reels are stored at film archives. Film is a good medium for long term storage, but future mass-migration to digital media is ineluctable and digital processing at this step could ensure the removal of the various, typical film-related damages, see figure 1. Though traditional restoration techniques are necessary (the film should be able to withstand mechanically the digitisation step), digital restoration lets us expect results beyond today's limitations (automated processing, correction of previously photographed artifacts, etc.).

Digital restoration has only very recently been explored [1,2,3]. The main visual defects are dust spots, hairs and dirt, instabilities (both exposition and position) and scratches, some of them are now easily detected and removed, especially if the defect appears only in a single frame. This is not the case for scratches. Scratches are mainly vertical (parallel to the film transport direction),

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3023, pp. 264–275, 2004.

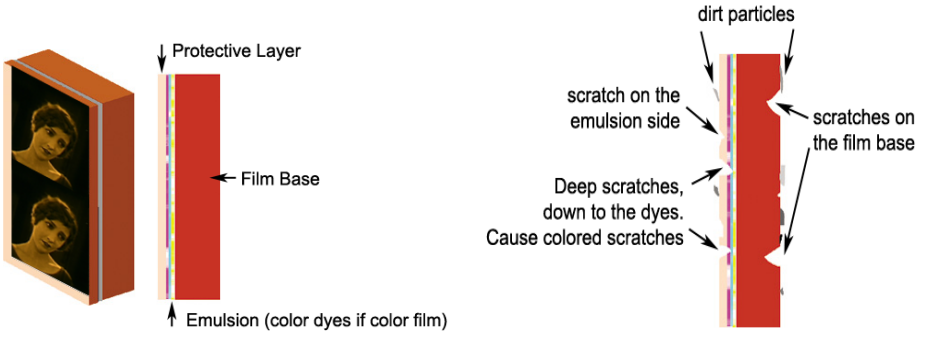

Fig. 1. Film structure and film damage

caused by slippage and abrasion during fast starts, stops and rewinding. Because the scratch is spread over many frames, and appears at the same location during projection, this damage is readily seen by the viewer, and also difficult to detect and correct using image processing.[1]

Early work about digital line scratch removal can be related to Anil C. Kokaram's research activities [4,5,6]. His detection scheme, based on vertical mean, is still used today. Other approaches use vertical projections and local maxima or minima detection. Bretschneider et Al [7,8] suggest a wavelet decomposition using the low frequency image and the vertical components for a fast detection. Some recent work [9,10] improve Kokaram's approach, but most of the techniques are intraframe methods, neglecting the scratch tracking [11].

In our approach, we consider a large number of test sequences (old footage and new shoots). We state that a tracking mechanism considerably increases the detection quality because line scratches can be very unsteady. The x-position of the line scratch can move sideways up to 10 % of the image width (see figure 2). Consequently, the intra-frame shape of the scratch is not perfectly vertical and the corresponding slope might reach 5 degrees. All the methods based on full frame projection or vertical mean fail in this case.

A tracking improves the detection as well, essentially in noisy images. The localisation of the scratch detection is better and therefore its correction as well. At last, since a line scratch has a continuous life over many frames, our method allows an inter-frame tracking in order to assign a unique identifier to the detected scratch for its entire lifetime. This is important for our user interface (selection on a *per-scratch* basis instead of a *per-frame* basis).

The present work deals essentially with persistent line scratches (several consecutive frames). Other methods based on motion compensation and temporal discontinuity in image brightness are more suitable for short line scratches (ap-

[1] Proper film digitisation requires a wet gate process, which dramatically reduces visible scratches. In a wet gate, a liquid (perchloroethylene) fills the gaps, and most of fine scratches are no longer visible. But the wet gate process requires the use of chemicals and is not very compatible with a high digitalisation throughput

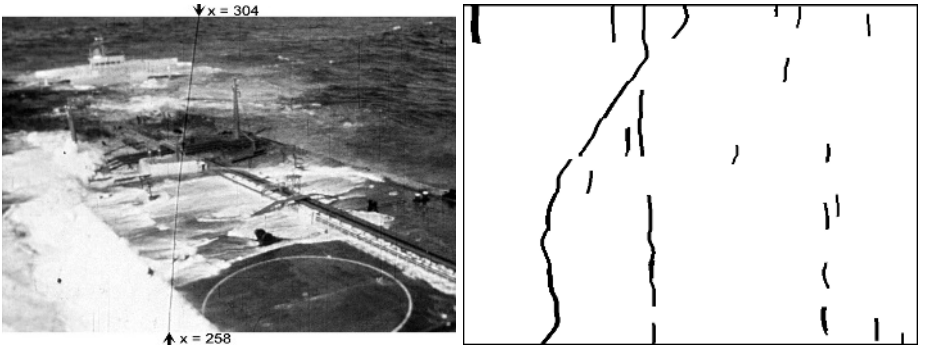

Fig. 2. Image exhibiting a notably slanted scratch, from the "marée noire, colère rouge" documentary and tracked path over 9 consecutive frames. Shown image is the 2nd one.

pearing randomly on a single frame only), as well as dust spots. Those methods fail with persistent scratches, present on the previous/next frame at nearly the same position and consequently matched and labelled as part of the scene.

2 Pre-processing : Image Projection

Though we cannot assume line scratches to be vertical over all an image, this hypothesis is locally true for a few consecutive horizontal lines in the original image I. We consider that scratch abscissa is locally constant over a band of H lines of I. Several advantages direct us to work with an image P, the vertically sub-sampled projection of the original image I. Each line of P is the vertical mean value of H lines of I (we call H the projection height) :

$$P(x,y) = \sum_{i=0}^{H-1} \frac{I(x, y \times H + i)}{H} \ . \tag{1}$$

- The amount of data (and processing time) is reduced by a factor of H.
- Noise is reduced by $\sqrt{(H)}$ if gaussian.
- Line scratches intensity remains unaltered (assumed constant over H lines).

This simple method gives very good results, though more complex projection schemes may be used, for example overlapping bands or a weighted mean. Let us emphase that the H parameter is of primary importance, because it will impact all the remaining processing steps and determine the maximum detectable scratch slope q. Above this maximum slope, scratches become attenuated after projection. H can be determined with respect to q by the following relation : $H = \frac{1}{\tan(q)}$; for q=5 degrees, we have H=12 pixels. According to image size, we use H=8, H=12 or H=16 (exact divisors). Figure 3 illustrates the projection transform.

Fig. 3. Image from the "lost world" movie and projected image $P(x, y)$ for 7 consecutive frames. The projection height H is 16. The scratches (dark one on the left side, bright one and dark one in the middle) are still visible.

3 Line Scratch Candidates Selection

The next step is the extraction of candidates which are used as input in the tracking process. The typical spatial signature for a scratch is a local extremum of the intensity curve along the x axis. So pixels candidate should be local maxima or minima horizontally, to find bright or dark scratches respectively. Many different methods exist in the literature to achieve this detection, and we experimented several ones [12]. For this work, we want our candidate extractor to meet the following requirements :

- Generate signed output : positive for bright scratches, negative for dark ones.
- Give a quality measure for each candidate, not only a simple binary output.
- Normalise quality measure between some known bounds, typically $\{-1, +1\}$.

The method we use relies on greyscale morphology [13,14]. Candidates for a line scratch are extracted by computing the difference between the original image and its opening or closing with a structuring element B_w. The opening will

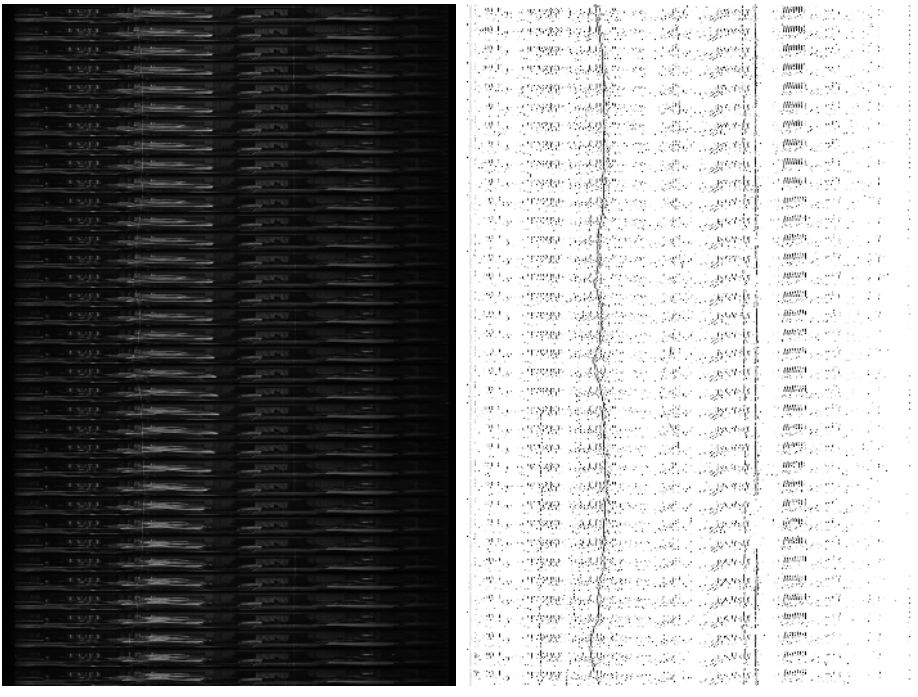

Fig. 4. The image P(x,y) (left side) is computed from a 36 frames (1.5 second) sequence. The output Q(x,y) (right side) is here shown as greyscale image ; real output values are signed so the tracker cannot confuse bright scratch candidates and dark ones.

remove thin structures brighter than the background, while closing will remove thin structures darker than the background. This way, to extract bright candidates, we subtract from P its opening with B_w, and symmetrically candidates for dark scratches are defined as the difference between the pixel values in P and their closing with B_w:

$$D^+(x,y) = P(x,y) - ((P(x,y) \ominus B_w) \oplus B_w) \; . \tag{2}$$

$$D^-(x,y) = ((P(x,y) \oplus B_w) \ominus B_w) - P(x,y) \; . \tag{3}$$

- $D^{\pm}(x,y)$ is the difference between the greyscale pixel value being considered, and a spatial neighbourhood of width w.
- $\oplus$ stands for morphological dilatation, $\ominus$ for morphological erosion and B_w is an unconstrained 1-D structuring element, of width w.

Because line scratches can be poorly contrasted relatively to their background, whereas natural image structures generally show a much stronger response, we locally normalise the result, to consider the significance of the extremum with respect to its spatial neighbourhood, using the following formula :

$$\text{if} \quad (((P(x,y) \oplus B_w) - (P(x,y) \ominus B_w)) > s)$$
$$Q(x,y) = A \times \frac{D^+(x,y) - D^-(x,y)}{(P(x,y) \oplus B_w) - (P(x,y) \ominus B_w)}$$
$$\text{else} \quad Q(x,y) = 0$$

- $Q(x,y)$ stands for the output image of this detector. This image is signed ; positive values standing for local maxima and negative values for local minima. The tracking stage will use this image as input.
- $(P(x,y) \oplus B_w) - (P(x,y) \ominus B_w)$ is the local contrast.
- s is a threshold (see below).
- A is a scaling factor, which determines output values range $[-A... + A]$. We typically use $A = 127$, to store $Q(x,y)$ as an 8-bit greyscale image

The major tuning parameters are w and s. w defines the maximum scratch width, and the size of the neighbourhood used to normalise output. It strongly depends on the input image resolution. We obtained satisfactory results with $5 \leq w \leq 9$ at video resolution (720×576), and $9 \leq w \leq 13$ at high resolution (2048×1536). The threshold s has two goals : reduce the amount of false alarms, and inhibit candidate extraction in smooth areas. It controls the sensitivity and is usually set to some low value, but still required to eliminate spurious candidates. Figure 4 shows a result of a candidate detection.

4 Tracking Problem Formulation

4.1 General Background

After the pre-processing and detection stage, we still have to track the scratch candidates over the sequence. A human observer will easily locate the most visible scratches in figure 4, but the visual localisation of incomplete ones requires more concentration. An automated tracking system should be fooled by false alarms too, especially if vertical structures are present in the image.

The proposed tracking scheme should be able to distinguish real scratches from false alarms, to close the gaps caused by detection failures or discontinuous scratches, to find the optimum path through candidates and also uniquely identify the scratches (the detection process will assign an unique ID to each scratch, ranging from the frame where it appears to the frame where it vanishes.

The input of our tracking scheme is the image $Q(x,y)$. The lines are read sequentially, from the top to the bottom, so that the temporal axis matches the y axis of this image. Each line is a data set Z_t for the tracker (observation). In fact, such a representation (figure 4, right side) is very similar to a radar (or sonar) plot (echoes vs. time). Tracking schemes for such applications are common in the literature, and several approaches exist : Kalman filtering, AR methods, probabilistic data association filter (PDAF), multiple hypothesis trackers (MHT), Monte-Carlo and particle filters, ... ; see an overview in [15]. Our problem is even simpler, since only one parameter should be estimated :

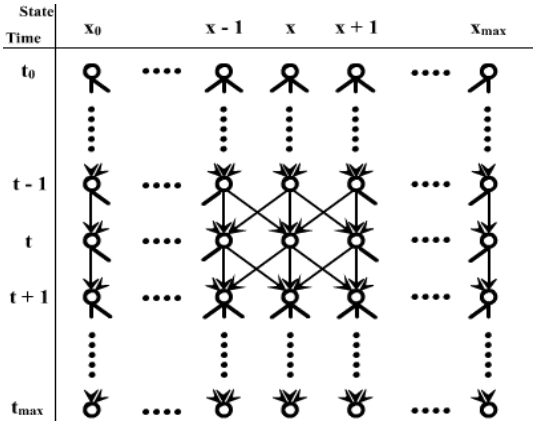

Fig. 5. Basis structure of trellis diagram built for each track. Since the representation space is matched against the state space, the possible transitions from state X_{t-1} to X_t are linked to a one-pixel deviation from the x position attached to state X_{t-1}.

the scratch localisation on the x-axis. Therefore, the state space and the world model (or representation space) are tightly matched.

Kalman filtering or PDAF approaches combine the different hypothesis at each step, while the MHT multiple hypothesis scheme keeps multiples hypothesis alive [16]. The idea is that by getting more observations Z_t, received at time t, and matching these to the hypotheses, the hypothesis corresponding to a real scratch path will gain more evidence, making it distinguishable from false ones. Besides, the hypothesis for a not perfectly continuous scratch will not disappear too quickly. Another advantage of such an approach is to unify in one concept path initialisation, path tracking and path decay.

Of course, the challenge is to maintain a reasonable number of hypothesis, according to available memory and computing power. Rejecting unprobable hypothesis or keeping the best one at each stage are possible approaches. Since the whole sequence could be digitised prior to the processing, an exhaustive search of the optimum path for each scratch is also possible (although not reasonable) ; but our implementation heads to process data *on the fly* and therefore can be used in near-real time systems, gathering the images as outputted by a telecine.

4.2 Path Hypothesis Generation

The path hypothesis generation consists in building a trellis diagram in the state space. In this trellis diagram, we find possible states X_t at time t, and branches representing transitions from one state at time t to the next at time $t + 1$. The tracking process can follow simultaneously multiple scratches, but to keep the algorithm simple we will use a different trellis for each track, and consider simultaneous tracks as independent (accordingly, track merging is impossible).

A particular path hypothesis through the trellis is defined as a set of sequential states $X_{t0}, X_{t1}, ..., X_{t-1}, X_t$, $t0$ being the starting time (initialisation) for the track we consider. Sequentially, as shown on figure 5, each state at time t is linked with only 3 states at time $t-1$, and conversely. This is due to the fact that we tolerate only, after projection, a horizontal displacement of one pixel between two consecutive lines for a track. As a consequence, the total number of possible paths for a particular track, at time t, is t^3.

A new trellis for a new track (holding one state $X_{t0} = x$ and one path hypothesis) is generated if, for a given data set Z_t (a line taken from the image $Q(x, y)$), an unmatched, isolated but relevant candidate is found.

4.3 Path Hypothesis Update

As stated earlier, the challenge is to prune the hypothesis tree, which grows up even if the new data set does not contain high detection values. Since the path hypothesis are represented as a trellis diagram in the state space, a practicable approach consist in weighting each transition from a state at stage $m-1$ to a state at stage m. The well-known Viterbi algorithm can be used to sequentially prune the paths. All paths kept at the previous state are extended to the possible states at the next stage, and the best path leading to each state is selected. Other paths are eliminated after further consideration. The Viterbi algorithm can be used in its standard form when the transition costs between states depend only of the previous state and the current measurements. If this condition is not met (non-markovian) then the kept path might be sub-optimum.

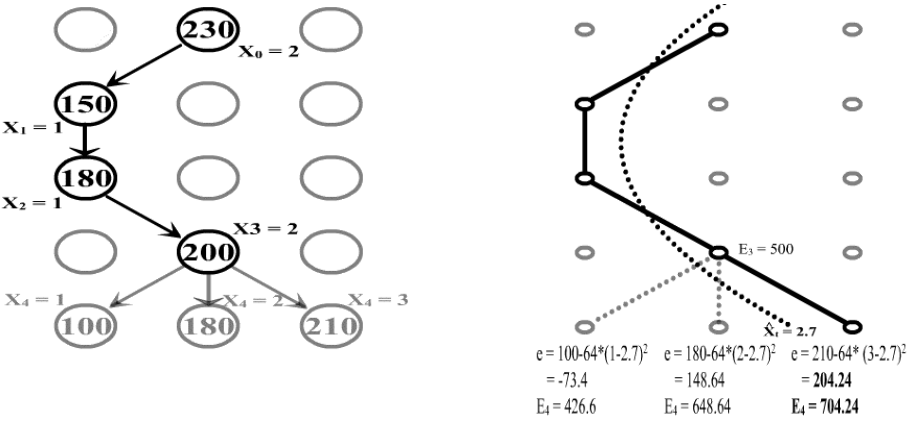

Fig. 6. These figures illustrate a possible path trough the state space, and therefore the representation space $Q(x, y)$ (see text). Each line of $Q(x, y)$ is used as observation Z_t for the tracker. To prune the paths and keep the L-best ones, the likelihood of a track is measured by a cost or energy function, based on both the quality of candidates (left figure) and closeness to an estimated path using short term history for parameter estimation (dotted line in the right figure). The cost computation is done for each path in the L-list, and for each possible new state.

To overcome this behaviour, we use the list-Viterbi algorithm (L-Viterbi) keeping a list of the L most valuable paths at each state and for each stage [17, 18] We will sequentially prune paths which are unlikely, and choose L so that no valid path is eliminated. The risk that an optimum path is rejected is alleviated as L increases. Like in the Viterbi algorithm, a value is computed along a path (cost function if the value should be minimised, else energy function). While the real Viterbi algorithm use plausibility value to score the transitions from state to state, we give below details on our implementation.

The set of possible paths at time t is noted C_t, with $C_{t,i}$ the i^{th} possible path at this time. At each branch in the trellis, we assign an cost or energy function $E_{t,i}$ which depends on the path $C_{t,i}$ being considered, and for each path $C_{t,i}$ we defined $E(C_{t,i})$ as the cumulative energy of its path branches :

$$E(C_{t,i}) = \sum_{n=0}^{t} E_{n,i} \ . \tag{4}$$

Our tracking problem can now be summarized as finding the L-optimal paths through this trellis, maximising the energy function E. For the path $C_{t,i}$ the energy assigned to the branch linking the state $X_{t-1,i}$ to $X_{t,i}$ is :

$$E_{t,i} = |Q(X_{t,i})| - W * (X_{t,i} - \hat{X}_{t,i})^2 \ . \tag{5}$$

The first term $Q(X_{t,i})$ is the quality criteria of the candidate associated with the state $X_{t,i}$. Using it as part of the cost function is quite obvious, since a line scratch is defined as set of sequential local extrema (candidates) extracted from $P(x, y)$. So a path should maximise the amount of candidates holding a strong quality criteria. The sign of $Q(x, y)$ is used as toggle to prevent mixing "bright" candidates and "dark" candidates, but $|Q(x, y)|$ is used in the energy function.

The second term $(X_{t,i} - \hat{X}_{t,i})^2$ is the squared difference between $X_{t,i}$ and $\hat{X}_{t,i}$, an estimate using the state history on path $C_{t,i}$. This is a tension constraint, a basic line scratch model, which will prevent paths which are not rigid enough to be chosen. The physical behaviour (inertia) of line scratches is reflected by this model. This constraint will prevent a path from locking on isolated extrema, especially when no more valid candidates seems available. The model used for the $\hat{X}(t, i)$ estimate is a 2-order polynomial, which is enough for our requirements :

$$\hat{x}(t) = \sum_{i=0}^{2} (a_i t^i) \ . \tag{6}$$

We estimate the polynomial coefficients using a least square method, on N previous states : $X_{t-n,i}, X_{t-n+1,i}, ..., X_{t-1,i}$. N must be high enough to prevent model divergence, and low enough to fit well the local trajectory. Kalman filter was taken in consideration for this task in earlier work [19].

Finally, W is a scaling factor, used to control the respective influence of both contributions (and so the rigidity of estimated tracks). It is strongly dependant of the parameter A used in the pre-processing stage. We obtained good results with $W = \frac{1}{4}A$, and choosing N according to the projected image height.

4.4 Track Ending Condition

As the update process handles the hypothesis tree pruning, the tracking mechanism is kept running until the track's ending condition is reached. If we introduce predominant history-related factors in the ending condition computation, long scratches will be *kept alive* while the ending condition is quickly reached for short ones. So, only the quality values of candidates are used. The mean value of the quality values $Q(x, y)$ associated to the states $X_t \dots X_{t-N}$ for the best path will be computed. The tracking is suspended if this value falls below a threshold ; we suppose the end of the scratch is reached. The path is stored in an intermediate data file for the subsequent removal processing, and the trellis representation is cleared from memory. This ending condition induces the tracking to overshoot the real scratch end. It could be improved by searching the strongest negative variation of $Q(x, t)$ along the best path if the ending condition is met.

4.5 General Algorithm

```
for each observation Z_t
  for each candidate in Z_t with non-zero quality value Q(x,y)
    initialise a new track
  end for
  for each track
    for each path hypothesis (from the L-list at t-1) related to this track
      estimate model parameter using N states along this path
      extend model for time t
      for each new state X_t reachable from state X_t-1
        compute cost for the transition X_t-1 to X_t
        add new transition cost to overall path cost
      end for
    end for
    sort and keep L-best paths, clear other paths from memory
    if end condition for the best path is met
      store track and clear memory
    end if
  end for
end for
```

5 Evaluation and Conclusion

Theis contribution is focused on the detection side of the scratch removal process. The described detection and tracking scheme feeds the subsequent correction process with scratch trajectories ; this later process could rely on several approaches : interpolation, in-painting, ... Showing a result for the complete removal process on the basis of a single image is irrelevant in printed form, since the restoration quality could only be assessed by dynamic rendering[2]. Generally, we do not have film samples before degradation, and working with synthetic data

[2] See video clips attached to the electronic version of this paper

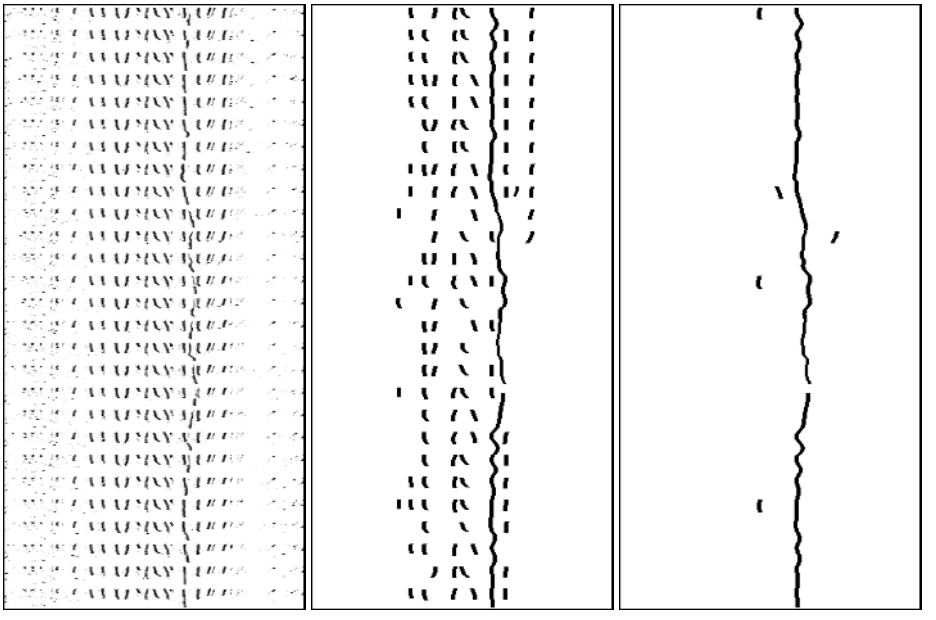

Fig. 7. Left : the image $Q(x, y)$ for 27 consecutive frames. This sequence shows many vertical structures (vertical curtain folds in the background of the scene) beside a real scratch. middle : Tracking results. Right : Tracking result keeping the longest paths

is nonsense (simulated scratches : what model to use), so a objective efficiency measurement is difficult. And because the goal of restoration is to improve the visual quality of degraded film sequences, the appropriate evaluation method is by subjective evaluation.

Even if we still find really weird images (for ex. with a lot of vertical structures, similar to the figure 7) overthrowing our algorithm, the overall efficiency of this detection scheme has been proved, and performs better than the previous ones or other known methods, for jittering scratches as well as steady ones. At present, we are improving the whole scratch removal process, especially the correction step by limiting the repetitive over-corrections..

At last, this tracking concept is used in our restoration software suite called RETOUCHE, used by the French post-production group **Centrimage** and by the **CNC** (French national film archives). RETOUCHE has been used for the digital restoration of 3 full-length features in 2K resolution and one video documentary (ca. 500000 frames) with convincing results. The algorithm and its implementation are also fast (less than a second per frame for $2K$ images).

Acknowledgements. Images from "The Lost World" (1925) by courtesy of Lobster Films, images from "Marée noire, colère rouge" (1978) by courtesy of the Cinémathèque de Bretagne.

References

1. Decencière, E., Serra, J.: Detection of local defects in old motion pictures. In: VII National Symposium on Pattern Recognition and Image Analysis, Barcelona, Spain (1997) 145–150
2. Joyeux, L., Boukir, S., Besserer, B., Buisson, O.: Reconstruction of degraded image sequences. application to film restoration. Image and Vision Computing **19** (2001) 503
3. Takahiro, S., Takashi, K., Toshiaki, O., Takamasa, S.: Image processing for restoration of heavily-corrupted old film sequences. In Society, I.C., ed.: 15th. International Conference on Pattern Recognition (ICPR'00). Volume 3., Barcelona, Spain (2000) 17–20
4. Kokaram, A., Morris, R., Fitzgerald, W., Rayner, P.: Detection of missing data in image sequences. i3etip **4** (1995) 1496–1508
5. Kokaram, A.: Detection and removal of line scratches in degraded motion picture sequences. In: Proceedings of EUSIPCO'96, Trieste, Italy (1996)
6. Kokaram, A.: Motion picture restoration. Springer-Verlag (1998)
7. Bretschneider, T., Kao, O.: Detection and removal of scratches in digitised film sequences. In: International Conference on Imaging Science, Systems, and Technology. (2001) 393–399
8. Bretschneider, T., Miller, C., Kao, O.: Interpolation of scratches in motion picture films. In: IEEE Conference on Acoustics, Speech, and Signal Processing (ICASSP). Volume 3. (2001) 1873 –1876
9. Vitulano, D., Bruni, V., Ciarlini, P.: Line scratch detection on digital images: An energy based model. In: International Conference in Central Europe on Computer Graphics and Visualization (WSCG). Volume 10. (2002) 477
10. Maddalena, L.: Efficient methods for scratch removal in image sequences. In: 11th International Conference on Image Analysis and Processing (ICIAP2001), IEEE Computer Society (2001) 547–552
11. Tegolo, D., Isgro, F.: Scratch detection and removal from static images using simple statistics and genetic algorithms. In: International Conference on Image Analysis and Processing. (2001) 507–511
12. Joyeux, L., Buisson, O., Besserer, B., Boukir, S.: Detection and removal of line scratches in motion picture films. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition, Fort Collins, Colorado, USA (1999) 548–553
13. Serra, J.: Image Analysis and Mathematical Morphology. Volume 1. Academic Press, London, England (1982)
14. Serra, J.: Image Analysis and Mathematical Morphology: Theoretical Advances. Volume 2. Academic Press, London, England (1988)
15. Cox, I.J.: A review of statistical data association techniques for motion correspondence. International Journal of Computer Vision **10** (1993) 53–66
16. Cox, I.J., Hingorani, S.L.: An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. In: IEEE Trans. on PAMI. Volume 18. (1996) 138–150
17. Perry, R., Vaddiraju, A., Buckley, K.: Trellis structure approach to multitarget tracking. In: Proc. Sixth Annual Workshop on ASAP. (1999)
18. Bradley, J., Buckley, K., Perry, R.: Time-recursive number-of-tracks estimation for mht. In: Signal and Data Processing of Small Targets, Orlando, FL (2000)
19. Joyeux, L., Boukir, S., Besserer, B.: Tracking and map reconstruction of line scratches in degraded motion pictures. Machine Vision and Applications **Volume 13, Number 3** (2002) 119–128