

# Robust Automatic Line Scratch Detection in Films

Alasdair Newson, Andrés Almansa, Yann Gousseau, and Patrick Pérez

**Abstract**—Line scratch detection in old films is a particularly challenging problem due to the variable spatiotemporal characteristics of this defect. Some of the main problems include sensitivity to noise and texture, and false detections due to thin vertical structures belonging to the scene. We propose a robust and automatic algorithm for frame-by-frame line scratch detection in old films, as well as a temporal algorithm for the filtering of false detections. In the frame-by-frame algorithm, we relax some of the hypotheses used in previous algorithms in order to detect a wider variety of scratches. This step's robustness and lack of external parameters is ensured by the combined use of an *a contrario* methodology and local statistical estimation. In this manner, over-detection in textured or cluttered areas is greatly reduced. The temporal filtering algorithm eliminates false detections due to thin vertical structures by exploiting the coherence of their motion with that of the underlying scene. Experiments demonstrate the ability of the resulting detection procedure to deal with difficult situations, in particular in the presence of noise, texture, and slanted or partial scratches. Comparisons show significant advantages over previous work.

**Index Terms**—Film restoration, line scratches, adaptive detection, *a contrario* methods, affine motion estimation.

## I. INTRODUCTION

THE restoration of old films is a subject of primary interest due to the great quantities of old film material present in film archives. Unfortunately, manual digital restoration is extremely time-consuming and labour intensive. For instance, the recent restoration of George Méliès's "Voyage dans la Lune" (1902)<sup>1</sup> took one year (for fifteen minutes of film). It is clear, therefore, that automatic or semi-automatic tools designed for the detection and restoration of defects are highly desirable.

Some of the most common defects in films include dust/dirt, blotches, flicker and line scratches. Here, we consider the last defect, the line scratch, usually caused by an abrasion to the physical film. A good explanation of the physical origins of line scratches may be found at [1]. These line scratches appear as thin bright or dark lines which are roughly

Manuscript received May 30, 2013; revised October 22, 2013 and December 25, 2013; accepted December 28, 2013. Date of publication January 23, 2014; date of current version February 4, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Richard J. Radke.

A. Newson is with Technicolor, Cesson-Sévigné 35570, France, and also with Télécom ParisTech, Paris 75013, France (e-mail: alasdairnewson@gmail.com).

A. Almansa and Y. Gousseau are with Télécom ParisTech, Paris 75013, France (e-mail: andres.almansa@telecom-paristech.fr; yann.gousseau@telecom-paristech.fr).

P. Pérez is with Technicolor, Cesson-Sévigné 35570, France (e-mail: patrick.perez@technicolor.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2014.2300824

<sup>1</sup>Partly restored by Technicolor, presented at the Cannes Film Festival 2011.

straight and vertical. These defects also present the singular characteristic of *temporal persistence*, meaning that they remain in the same or a similar spatial position for several frames. Consequently, line scratch detection algorithms must be specially adapted to this defect.

However, these characteristics are very variable, making line scratch detection and restoration a particularly difficult challenge. For instance, in some cases, the scratch is semi-transparent, so that some of the original image information is still available, whereas in others all the information is removed. Also, scratches are not necessarily completely straight and vertical, and their shape may in fact vary from frame to frame. Finally, although scratches can often be static, they may also move with any type of motion.

We propose a line scratch detection method which is composed of two algorithms: a "spatial" algorithm which provides a pixel-precision detection of line scratches in single frames, and a "temporal" step which rejects false alarms based on information available in the whole image sequence.

The contributions of this paper are as follows. Firstly, we propose a pixel-precision line scratch detection algorithm which is robust to the presence of noise and texture. The algorithm's robustness is due to the use of the *a contrario* methodology [7], previously used for gradient alignment detection in images. In particular, we propose a modification to the methodology which makes the detection robust to texture and clutter with characteristics that vary throughout the image. This drastically reduces the number of false alarms. The spatial algorithm presents good recall (most of the scratches are detected), with very few true scratches being rejected.

Secondly, we propose a temporal filtering step to remove false detections left over from the spatial detection. In contrast to most previous approaches, we *reject* false detections, rather than *validating* true scratches. This is done by using a motion coherence criterion: we consider that detections which move in the same manner as the underlying scene are not true scratches. In particular, we avoid the difficult task of tracking true scratches, whose temporal behaviour is difficult to determine. In order to decide on the rejection of a detection, we also estimate a robust affine scene motion, in contrast to some previous methods [9], [19] which employ less robust motion estimation.

The advantages of our method will be demonstrated in Section V on a series of degraded film sequences and discussed in comparison with previous work.

The paper is structured as follows. In Section II we briefly recall some of the previous works on spatial and temporal line scratch detection. In Section III we present the proposed method for spatial scratch detection. In Section IV, we introduce the temporal, motion-based filtering whose goal is the

removal of false alarms. Finally, experimental validations are presented in Section V. Preliminary versions of our work have appeared in [20], [21].

## II. PRIOR WORK

Line scratch detection can be performed using only spatial information, on a frame-by-frame basis. Another body of work, which we shall call temporal approaches, include motion information to improve the detection. As acknowledged in a recent review [16], both approaches are complementary and benefit from one another’s advantages.

Kokaram [15] was the first to introduce a spatial model for the detection of line scratches. This model is based on the hypothesis that “side-lobes” are visible on either side of a line scratch. The horizontal scratch profile is modelled by a damped sinusoid and Bayesian estimation is used to determine whether an observed profile corresponds to a scratch or not. In [5], Bruni *et al.* provide a physical explanation for this model; the side-lobes are caused by light diffraction during the film scanning process. This model is also used in the work by Bruni *et al.* [4]. This approach is considered to be among the most efficient for line scratch detection (see the recent review in [16]) and we use it for comparison with the proposed approach. In other methods, such as [2] and [19], scratches are detected in the wavelet domain. The Hough transform is used in both [15] and [6] to detect prominent lines. Finally, Kim *et al.* use neural networks in [14] to establish scratch texture characteristics, before applying morphological filtering.

These spatial detection algorithms have several weaknesses. Firstly, the scratch is represented as a straight, vertical line. In practice, this hypothesis is often violated, and as a consequence many true scratches may be missed. With such methods, a slanted or non-straight scratch will be at best partially detected, which is of little use for restoration since the scratch will most likely have to be annotated by hand. Furthermore, experiments show that existing algorithms cope badly in noisy or textured regions. We deal with this important problem explicitly, by considering a locally adaptive detection model, and setting the thresholds accordingly. Finally, line scratch detection algorithms often represent the scratches as covering the entire height of a frame. This sort of detection runs the risk of restoring parts of the image which are not degraded. We relax several of the hypotheses found in other papers, such as the existence of side-lobes, allowing our algorithm to detect a wider variety of scratches.

Temporal approaches may be found in [9], [11]–[13] and [19]. The goal of these algorithms is to validate the detections based on hypotheses concerning the temporal nature of line scratches. Joyeux *et al.* use the hypothesis that scratches have sinusoidal horizontal motion, due to the supposition that they are caused by rotating mechanical parts. The true scratches are then tracked by integrating this hypothesis into a Kalman filter. The drawbacks of this approach are the restrictive nature of the sinusoidal hypothesis, and the lack of any global motion analysis. In [9], Güllü *et al.* propose the local block matching error between frames as a criterion for distinguishing true scratches from false alarms. If the

local block matching error around a scratch is high, then the scratch trajectory is validated as a true scratch. Unfortunately, this criterion is quite sensitive to phenomena such as flickering or other film degradations. This sensitivity degrades the robustness of the validation decision. Finally in [19], Müller *et al.* use motion estimation for line scratch filtering. The algorithm rejects any scratch detections which display similar motion to that of the scene. The scene motion is determined by hierarchical block matching in the left and right neighbourhoods of each scratch, and the scratch’s motion is determined on the spatial location of the scratch itself. While the basic principle of this approach (the motion of a scratch is not coherent with the scene motion) is very reasonable and will also be included in the present work, the approach from [19] suffers from clear disadvantages. Firstly, in practice it is extremely difficult to track thin structures from frame to frame. Second, block matching is not particularly well-adapted to such structures. Finally, the global motion is estimated in a very basic and non-robust manner.

In order to illustrate the benefits of our approach, we shall compare our work to three of the previous algorithms. For spatial detection, we shall use the work of Bruni *et al.* [4], and for the evaluation of our temporal filtering, we shall use the work of Güllü *et al.* [9] and Müller *et al.* [19].

We now proceed to describe our spatial line scratch detection algorithm.

## III. SPATIAL LINE SCRATCH DETECTION ALGORITHM

The proposed algorithm consists of a pixel-by-pixel scratch detection step, followed by the grouping and validation of these detections into visually significant scratch segments. Our grouping algorithm uses the *a contrario* methodology, a generic and automatic approach to setting detection parameters. Furthermore, we propose a modification of the classical methodology, relying on a local estimation of background models, which allows for grouping under spatially varying conditions of noise and clutter. This variant could potentially be used for other tasks.

### A. Pixel-Wise Detection Criteria

We first identify the potential scratch points by relying on a pixel-wise detection criterion. Several other such criteria have been presented in the literature, and are based on operations such as morphological filters [12], [14] or extrema detection in a 1D signal. Our criterion is a close variant of the classical test introduced by Kokaram [15], consisting of a threshold on the difference between the grey-scale image, and a horizontally median filtered version of this image. This test basically detects outliers with respect to a horizontal neighbourhood. In addition, in [15] the image is vertically subsampled before the thresholding, to highlight the scratches. Contrary to this original criterion, we do not take the central pixel into account when determining the median value. We also use a  $3 \times 3$  Gaussian filter with a standard deviation of one pixel to reduce the noise in the image, instead of vertical sub-sampling.

A drawback of this criterion is its tendency to detect steep intensity fronts, rather than just “peaks”. We avoid this by

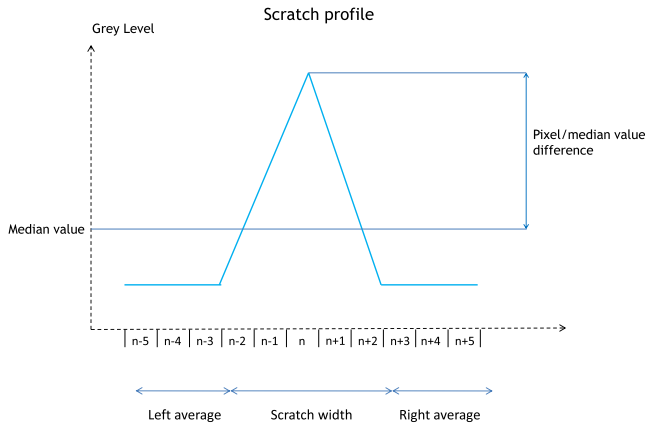


Fig. 1. Line scratch profile and pixel-wise detection criteria.

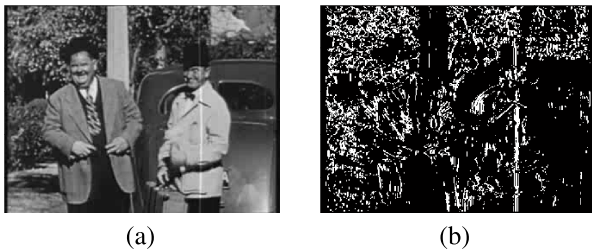


Fig. 2. Binary detection image from “Laurel and Hardy”. White pixels are detected and black pixels are not. (a) Original frame. (b) Binary detection image.

stipulating that the average grey-level values should be similar on either side of the scratch.

Our final pixel-wise detection criteria may be written in the following manner. Let  $I_g(x, y)$  be the Gaussian filtered grey level image. Let  $I_m(x, y)$  denote the median value over a local horizontal neighbourhood of pixel  $(x, y)$ , and  $I_l(x, y)$  and  $I_r(x, y)$  be the left and right horizontal averages, as defined below. The two Boolean criteria are:

$$\begin{aligned} c_1(x, y) &: |I_g(x, y) - I_m(x, y)| \geq s_{med}, \\ c_2(x, y) &: |I_l(x, y) - I_r(x, y)| \leq s_{avg}. \end{aligned} \quad (1)$$

where,  $s_{med}$  and  $s_{avg}$  are grey-level thresholds. We can therefore define a binary image indicating detections as

$$I_B(x, y) = \begin{cases} 1 & \text{if } c_1(x, y) \text{ and } c_2(x, y) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Fig. 1 shows an illustration of our two detection criteria.

For the proposed criteria, we set the width of the median filter to 5 pixels, and the value of  $s_{med}$  to 3 grey levels. These values are the same as in [15] and also appeared to us to be good empirical choices. The left and right averages are each taken over 3 pixels on either side of the 5 central pixels, and  $s_{avg}$  has been experimentally set to 20 grey levels. These parameters were used for all the examples in the experimental section.

As may be seen in Fig. 2, such a pixel-wise detection is bound to produce many false alarms, and also misses some scratch pixels. Therefore, a further *grouping* step is needed in

order to determine the *significant scratch segments* present in the pixel-wise scratch detections.

### B. Scratch Point Grouping and Validation

Because of false detections due to noise and texture (see Fig. 2), an extremely robust approach is needed to group the pixels into segments. One of the most well-known methods of detecting prominent lines in binary images is the Hough transform, and this is used by [6] and [15] for the grouping of scratch detections. Unfortunately, this approach contains thresholds which need to be tuned from sequence to sequence, and does not offer a precise spatial localisation of line segments. In order to group the pixel-wise detections, we turn to a more sophisticated set of methods known as a *contrario* methods, used for alignment detection by Desolneux *et al.* in [8].

In a word, the *a contrario* methodology is a generic way to detect visual objects in digital images. Detection thresholds are set in order to control the number of false detections in a white noise image, or more generally under a *background model*. This model usually relies on an independence assumption on the basic elements to be grouped for the detection. A group is validated as soon as it is very unlikely that this group has been generated by the background model. That is, groups are detected when they are very unlikely under the hypothesis that basic elements are independent. A comprehensive presentation of such approaches may be found in [8].

1) *A Contrario Line Segment Detection*: First of all, we present the *a contrario* approach as it is used to detect line segments in [7]. In this case, the basic elements to be grouped are pixels, and segments are detected as groups of pixels whose gradients are perpendicular to a given direction.

Given a line segment made of  $l$  pixels, a variable  $x_i$  is associated to each pixel. The variable  $x_i$  is equal to 1 if the pixel is aligned with the segment and 0 otherwise. “Aligned” pixels are those whose gradient orientation is orthogonal to the segment orientation, up to some angular precision  $p\pi$  radians, with  $p \in [0, 1]$ . Let  $s = x_1 + \dots + x_l$  be the number of aligned pixels. This is the quantity upon which the detection of segments is based. Larger values of  $s$  are associated to more meaningful line segments.

Now, the detection of segments require thresholds that depend on  $l$  and  $p$  and are therefore non-trivially set. The aim of the *a contrario* approach is precisely to set these thresholds. The detection relies on the probability distribution of  $s$  under some background model.

In the case of line segments as described in [7], the background model specifies that all gradient orientations are independent and follow a uniform distribution in  $[0, 2\pi]$ . This is the case, for example, in a Gaussian white noise image. As above, we consider a segment made of  $l$  pixels. Let  $X_i$  be a random variable associated to the deterministic observation  $x_i$ . Under the background model, each  $X_i$  follows a Bernoulli distribution of parameter  $p$ , so that the random number of aligned points  $S_l = X_1 + \dots + X_l$  follows a binomial law:

$$\Pr(S_l \geq k_0) = \sum_{k=k_0}^l \binom{l}{k} p^k (1-p)^{l-k} =: B(p; k_0, l). \quad (3)$$

Segments of length  $l$  having  $k_0$  aligned pixels are meaningful when  $B(p; k_0, l)$  is small enough. Intuitively, this probability is small when the observed segment has a number of aligned points  $k_0$  which is too large to occur by chance (as specified by the background model). In order to threshold this probability, the total number of tested line segments has to be taken into account. Indeed, even very improbable events can occur if the number of tests is high enough. For this, one considers the *number of false alarms* (NFA), defined in [7] as:

$$\text{NFA}(l, k_0) = N_{\text{tests}} B(p; k_0, l), \quad (4)$$

where  $N_{\text{tests}}$  is the total number of segments to be tested. Since segments are defined by a beginning and end point, it is easily seen that this number may be approximated as  $N_{\text{tests}} = M^2 N^2$ , with  $M$  and  $N$  the linear dimensions of the image. A segment is detected if  $\text{NFA}(l, k_0) \leq \varepsilon$  for some parameter  $\varepsilon$ . In other words, a segment is only meaningful if the number of false alarms under the background model is less than  $\varepsilon$ . It is shown in [8], that such a definition of meaningful segments implies that the expected number of detected segments under the background model is bounded by  $\varepsilon$ .

2) *Locally Adaptive Grouping for Line Scratch Detection*: We now rely on the same principles to group pixels that have been detected by the pixel-wise procedure of Section III-A. We must first define a background model to represent the binary image obtained with Equation (2). In the case of orientation grouping (see Section III-B.1), the background model corresponds to an image where the direction of the gradients are randomly and uniformly distributed. This crucial hypothesis accounts for situations in which we do not wish to detect alignments (homogeneous regions, noisy regions, isotropic texture etc.). In the present case, such a background model would not be satisfactory. The difficulty arises because the pixel-wise detection step produces an amount of false detections that varies greatly across the image. For example, strongly textured or cluttered areas yield many more detections than smooth regions, as seen in Fig. 2. Moreover, the visibility of scratches depends on their local neighbourhood. Given that the power of the *a contrario* methodology hinges on the background model, we shall now consider our background model to be a binary image in which labels are independent and the label probability of each pixel varies spatially. The computation of this probability will be based on a *locally adaptive estimation*.

The label probability for a given pixel is estimated as the maximum detection density on four squares of equal size surrounding the pixel. Four squares are used in order to deal with situations where the pixel is on the border of areas with different background models. Detection density is defined as the proportion of pixels, contained within a square, whose labels equal 1. The size of the sides of these squares is set to the width of the image, divided by a constant. We set this constant to 30 for all experiments in this paper.

Under this background model, the probability for a given segment to have at least  $k_0$  pixels with a label value of 1 is no longer given by a binomial distribution. It is easy to see that it is now given by a *Poisson binomial distribution* and is

equal to:

$$\Pr(S_l \geq k_0) = \sum_{k=k_0}^l \sum_{\substack{x_1, \dots, x_l \\ \sum x_i = k}} \prod_{i=1}^l p_i^{x_i} (1 - p_i)^{1-x_i}, \quad (5)$$

where  $p_1, \dots, p_l$  are the local detection probability at each pixel, the definition of which will be given below. This expression is quite costly to estimate and an approximation is therefore needed. In [8], Desolneux *et al.* suggest the use of Hoeffding's inequality [10], which provides an upper bound on the probability that the sum of some independently distributed random variables exceeds a certain value. In the present case, the interest of this approximation is that it still holds when the variables are independent but not identically distributed [10]. Therefore, it provides us with an approximation of  $\Pr(S_l \geq k_0)$ , where again  $S_l$  is the number of pixels having a label value of 1 along a segment of length  $l$ . The approximation is the following:

$$\Pr(S_l \geq k_0) \leq H(l, k_0) := e^{-l(r \log \frac{r}{\langle p \rangle} + (1-r) \log \frac{1-r}{1-\langle p \rangle})}, \quad (6)$$

where  $\langle p \rangle = l^{-1} \sum p_i$  is the average detection probability along the segment,  $r = \frac{k_0}{l}$ , and  $\langle p \rangle l < k_0 < l$ . We therefore define the number of false alarms of a segment as

$$\text{NFA}(l, k_0) = N_{\text{tests}} H(l, k_0). \quad (7)$$

A segment is **detected** if its NFA is **smaller** than  $\varepsilon$ : such a segment is said to be " $\varepsilon$ -meaningful". Thanks to Inequality 6, the expected number of detected segments under the background model is smaller than  $\varepsilon$ .

In all experiments, we use the parameter  $\varepsilon = 1$ , as in [7]. This choice is reasonable, since  $\varepsilon$  is a bound on the expected number of false detections under the background model. However, it may be further tuned to fit the user's needs, depending on whether the importance should be put on recall or precision. Furthermore, as explained in [7], detection results vary slowly with respect to  $\varepsilon$ , making it an easy parameter to tune, if so desired. The impact of this parameter will be empirically tested in Section V-D.

Since scratches are roughly vertical, we test all segments with a maximum deviation from the vertical direction of  $\pm 10$  degrees. We discretise these angles by 0.5 degrees. The  $N_{\text{tests}}$  parameter is therefore set to  $M^2 N \Theta$ , where  $\Theta$  is the number of angles tested. With the aforementioned parameters, we have  $\Theta = 40$ .

3) *Maximality*: With the previous detection procedure, many redundant segments are detected. This is because a very meaningful segment often contains, and is contained by other segments which are  $\varepsilon$ -meaningful. In order to keep only the best detection for such cases, we use the notion of *maximality*, as introduced in [7]. A segment is *maximal meaningful* if it neither contains, nor is contained, by a segment which is more meaningful (that is, a segment with a smaller NFA). Therefore, we only accept segments which possess this property.

When using the usual NFA definition, Equation (4), it may be shown [8] that maximal meaningful segments start with a detected pixel preceded by an undetected pixel, and end similarly. Knowing this, the number of segments tested, and

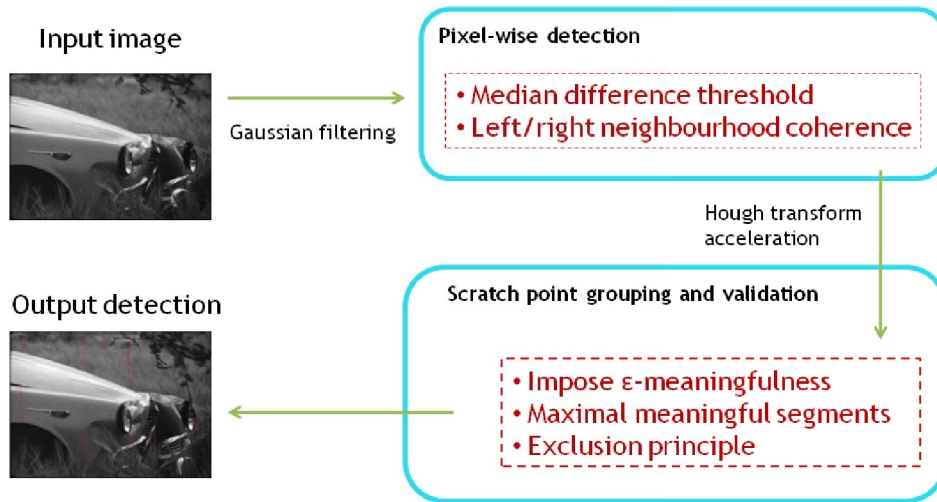


Fig. 3. Summary diagram of the proposed spatial detection algorithm.

therefore the computational cost, is *greatly* reduced. This result relies on the properties of the binomial law.

Now, our approach relies on the use of the Hoeffding approximation for the NFA definition, Equation (6). A natural question is whether maximal segments according to this new definition have the same property. In the Appendix, it is shown that this is indeed the case. Maximality of segments may therefore be tested in the same way as with the usual definition, which also yields the same crucial computational acceleration of the detection.

4) *Exclusion Principle*: Since scratches have a width of several pixels, different segments may correspond to the same scratch. Since for restoration purposes we would like as precise a representation of the scratches as possible, we use an *exclusion principle* as defined in [8], which states that a pixel may belong to one scratch only. If a pixel  $s$  is contained by several segments, then the most meaningful segment retains  $s$ . All other segments which contain  $s$  have this pixel removed. The NFAs of the modified segments are then recalculated, and those that are no longer  $\epsilon$ -meaningful are thrown away. This principle can be applied not only to pixels which belong to several segments, but also to those which are at a distance of  $\tau_x$  from more than one segment. In our experiments, we set  $\tau_x$  to three pixels. The entire spatial detection algorithm is represented in Fig. 3.

5) *Algorithm Speed-Up*: In order to speed-up the procedure, we apply a pre-selection of scratches candidates. For this, we apply a very permissive Hough transform to  $I_B$ , and only analyse the lines which correspond to local maxima. Tests show that no or very few real scratches are lost by this pre-processing. Note that this sort of speed-up has been used previously by other authors in [6], [15]. In fact, a parallel may be drawn with the work of Kokaram [15] in that the latter consists of a permissive Hough transform, followed by a statistical parameter estimation method. However, there are two main differences between our work and the latter approach. Firstly, our statistical (*a contrario*) step does not require an explicit scratch profile model. Secondly, our approach looks for the

best *sub-segments* in a line, rather than validating the *whole* line. For these reasons, our approach appears more powerful and robust than that of [15].

In the next section, we shall explore the temporal aspects of line scratch detection and present our temporal filtering algorithm.

#### IV. TEMPORAL FILTERING ALGORITHM

Although the previous algorithm detects line scratches with good spatial precision and is robust to noise and texture, it does not deal with the problem of false alarms due to thin vertical structures that are part of the captured scene. On a frame-by-frame basis, these closely resemble line scratches. In some situations, it is practically impossible to differentiate the two without prior knowledge concerning the scene structure.

Unfortunately this sort of knowledge is difficult to obtain and use. One other way to distinguish between true and false scratches is to use temporal information contained in the image sequence. Since scratches are caused by physical damage to the actual film, their motion is completely independent of that of the scene. Therefore, any detections displaying motion which is coherent with the scene should correspond to false detections.

Consequently, we reject any scratch detections having a trajectory which conforms to the dominant scene motion. We shall refer to this criterion as the *motion coherence* criterion.

This criterion does not deal with scratches which move with the scene, or are completely still in a static scene. However, such situations are impossible to resolve without prior knowledge on the nature of scratches.

##### A. Motion Coherence

The major challenge when using motion information for scratch filtering is determining the trajectories of true and/or false detections. This is a very difficult task in the case of true scratches due to the generic nature of the trajectories of line scratches. Instead, we shall try to determine the trajectories of *false detections*.



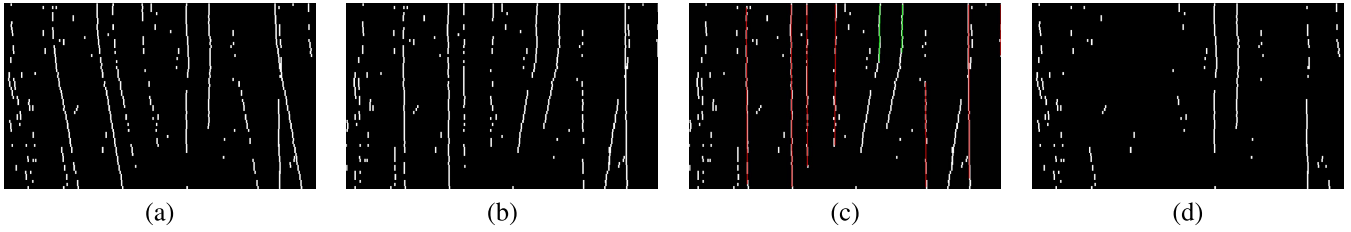


Fig. 4. Different  $x-t$  binary maps, for the “Afrunden 2” sequence. The horizontal axis represents the average horizontal position of a scratch, and the vertical axis represents the frame number  $t$  which the scratch belongs to. Each white point corresponds to a detected scratch segment in a given frame. The original video may be viewed at: <http://www.enst.fr/~gousseau/scratches>. (a) Original detections ( $I_T$ ). (b) Realigned detections ( $I'_T$ ). (c) Detected trajectories. In red false alarms, in green true scratches. (d) Final filtered detections.

In all that follows, we suppose that a scratch detector provides the initial detections. In our work, we naturally use the detection scheme presented in Section III, but it should be noted that it is possible to use any spatial detection scheme in the literature, although the results may be worse if pixel-precision detections are not given.

For temporal filtering purposes, we shall represent a scratch detection by its average column position and the index of the frame in which it is found. Let  $I_T$  be the  $x-t$  binary detection map of the detected scratches. For an example of this representation, see Fig. 4 (a). Let  $S$  represent an initial scratch detection. We will refer to this as a *segment*. Let  $\tilde{x}(S)$  and  $\tilde{y}(S)$  be the average column and row indices of the segment, and  $\mathbf{t}(S)$  denote the frame in which the segment was detected.

The first step of the proposed temporal filtering is to determine the trajectories of the false detections, a non-trivial task given the irregularity of the detection map. Instead of determining the trajectories in  $I_T$ , we will create another binary map in which the positions of the segments are *realigned* with respect to an estimated global scene motion. Let us call this new detection map  $I'_T$ . Before explicitly defining the new detection map, let us observe that the positions of *false detections* will appear as straight vertical lines, due to the motion coherence hypothesis. For an example of  $I'_T$ , see Fig. 4(b). The problem of detecting straight, vertical lines in an image is much more constrained than following generic trajectories, and therefore easier to solve.

In order to realign the segments, we need to obtain an estimation of the scene’s global motion. To do this, we use the algorithm from Odobez *et al.* [22] to estimate an affine approximation of the dominant motion in a robust manner. At a pixel position  $q = (x_q, y_q)$ , the motion vector  $(u(q), v(q))$  is expressed as:

$$\begin{cases} u(q) = c_1 + a_1x_q + a_2y_q \\ v(q) = c_2 + a_3x_q + a_4y_q \end{cases} \quad (8)$$

where  $c_1$  and  $c_2$  are the parameters describing the constant motion components, and  $a_1 \dots a_4$  are the parameters associated with the spatially varying components of the motion.

Let  $(x, y)$  be a pixel in a frame  $t$  and  $(x', y')$  be the corresponding position in frame  $t + 1$ . We have the following relationship:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 + 1 & a_2 & c_1 \\ a_3 & a_4 + 1 & c_2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} := \tilde{A}_{t,t+1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (9)$$

This motion estimation is carried out between each consecutive pair of frames throughout the image sequence.

Let  $x$  and  $y$  represent the spatial coordinates of a pixel in frame  $t$ . It is possible to find its corresponding coordinates,  $(x^r, y^r)$ , in a frame  $r$  with the following relationship:

$$\begin{bmatrix} x^r \\ y^r \\ 1 \end{bmatrix} = \tilde{A}_{r,t}^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (10)$$

with

$$\tilde{A}_{r,t} := \tilde{A}_{t-1,t} \tilde{A}_{t-2,t-1} \dots \tilde{A}_{r,r+1}. \quad (11)$$

This provides us with the necessary tools with which to realign the segments. The binary detection map  $I'_T$  resulting from this realignment is defined as:

$$I'_T(x, t) = \begin{cases} 1 & \text{if } \exists S \mid \tilde{x}^r(S) = x, \mathbf{t}(S) = t \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

where  $\tilde{x}^r(S)$  is the average column index of the segment warped to the reference frame. It is important to note that it is the original detection segments which are realigned with respect to the global motion, and not the detection map  $I_T$  itself. This is due to the fact that several segments may correspond to the same spatio-temporal point in  $I_T$ . An example of a realigned detection map may be seen in Fig. 4(b).

Once the segments have been realigned, we need to detect vertical line segments in  $I'_T$ , and thereby determine the trajectories of the false detections. Interestingly, the *a contrario* step of the algorithm presented in Section III may be used for this task, since we wish to detect line segments in a new binary image (the detection map). The detection also guarantees a precise and unique description of the trajectories, using the maximality and exclusion principles defined in Section III-B. The maximality principle implies that trajectories with temporal holes may be grouped together, and also that the beginning and end points of the trajectories can be detected very robustly. By using the exclusion principle, only the best representation within a certain neighbourhood will be chosen, which avoids having to make difficult decisions if trajectories are too close to each other. A trajectory detected in this manner will simply be a list of  $x-t$  positions in  $I'_T$ .

In terms of implementation details, we allow a maximum slope of 5 degrees in comparison to the vertical when detecting the trajectories. This corresponds, roughly, to allowing a trajectory to deviate one pixel every eleven frames with respect to the underlying scene motion. Before detecting the

trajectories, we perform a horizontal morphological dilation of one pixel on  $I'_T(x, t)$ . This is necessary, since the spatio-temporal trajectories shown in Fig. 4(c) are discretised, and therefore the trajectories may not be precisely detected without dilation.

We now define a *trajectory set*  $T$  as the set of segments whose  $x$ - $t$  positions are within a horizontal distance of one pixel of the  $x$ - $t$  positions of a given detected trajectory.

Up to this point in the paper, it has been assumed that all the trajectories detected as vertical lines in  $I'_T$  correspond to false detections. Unfortunately, true scratches may be coherent with the global scene motion when the scene is static. Therefore, scratches may be totally or partially represented as straight vertical lines in  $I'_T$ . Such a situation may be seen in Fig. 4(c). Two short trajectories are present (in green) which correspond to the partial trajectories of true scratches. Since these scratches happen to be static, their trajectories are detected as vertical segments in  $I'_T$  when the scene does not move, but naturally are lost when the scene moves (they become significantly slanted after that point). The scratches should not be rejected as there is no significant motion of the scene over the time interval during which they are detected.

We determine whether the scene has moved sufficiently during a trajectory's timespan by inspecting the maximum *horizontal* motion of the scene. For this purpose, it is sufficient to look at the original positions of the segments in the trajectory set. Since these have been identified as conforming to the underlying scene motion, their original positions reflect this motion. This obviously holds true in the case of a static scratch and no scene motion. Therefore, we reject a trajectory set if there exist two segments  $Q$  and  $R$  belonging to this set which verify the following inequality:

$$|\tilde{x}(Q) - \tilde{x}(R)| \geq \tau_m, \quad (13)$$

where  $\tau_m$  is a motion threshold. This corresponds to the *absolute distance* that the scene has moved between the frames  $\mathbf{t}(Q)$  and  $\mathbf{t}(R)$ . It is important to note that this corresponds to the scene motion *locally* in the area of the scratches. This can be crucial for situations such as zooming, in which case the scene presents different motion at different positions in the image. In all of our evaluations, we set the parameter  $\tau_m$  to 10 pixels.

### B. Further Filtering Criteria

Apart from motion coherence, there are other criteria which are reasonable to use for scratch filtering, and which are much more easily implemented than the motion coherence. One example, which we have used in our evaluations, is a scene cut criterion. This stipulates that an entire trajectory set is rejected if its beginning and ending frame indices are within a temporal distance  $\tau_c$  from a scene cut. In our experiments, we set  $\tau_i$  to 5 pixels and  $\tau_c$  to 4 frames. We used a simple Edge Change Ratio based scene cut detection algorithm (see [18]) for the detection of the scene cuts, in which the edge detection was done using the Sobel operator. This can obviously be replaced by other scene cut detection algorithms, if necessary. This is a

minor step of the proposed procedure. Indeed, in all of our experiments, it was used only for the ‘‘Laurel and Hardy’’ sequence.

## V. RESULTS

In this section, we present quantitative results of our algorithm. Its performance is compared with other approaches with respect to three criteria: recall, precision and the  $F1$ -score. Table I shows this quantitative evaluation. Recall is defined as the number of true detections divided by the total number of true scratches present in an image. Precision is defined as the number of true detections divided by the total number of detections. Basically, recall determines what percentage of the line scratches are detected, and precision shows what percentage of our detections were correct detections. The  $F1$ -score is a reflection of both criteria, and defined as  $2 \frac{\text{recall} * \text{precision}}{\text{recall} + \text{precision}}$ . We evaluate both the spatial detection step and the temporal filtering algorithm. These two contributions are compared with three other algorithms: the spatial method of Bruni *et al.* [4] and the temporal algorithms of Güllü *et al.* [9], and Müller *et al.* [19]. While another more recent spatial method [14] exists, it is a supervised algorithm (contrary to Bruni's and ours, which are automatic) and contains several parameters which are not given in the paper, such as the number of nodes in the input and hidden layers of the neural network, making implementation impossible without testing a series of architectures.

In our experiments, all the parameters are set to the values given throughout the paper. In particular, the detection threshold  $\varepsilon$  is set to 1, the exclusion parameter  $\tau_x$  is set to three pixels, and the motion threshold  $\tau_m$  to 10 pixels. Note that we impose a minimum scratch length of one tenth of the image height for all of our spatial detections. The parameters for Bruni's algorithm are those given in [3] and [4], apart for the scratch colour parameter (black or white), which was set manually for each sequence. Three parameters are required by Güllü's algorithm. As it relies on Bruni's algorithm for the initial spatial detection step, the scratch colour must be specified (black or white). The second parameter is a maximum search distance for the block matching algorithm. For this, we chose a maximum distance of 7 pixels either side, which corresponds to the maximum motion we expect in the sequence. Finally, we need a maximum mean absolute difference (MAD) threshold which identifies the presence of a scratch. We chose 15 grey levels for this threshold. These last two thresholds are not specified in [9]. Finally, Müller *et al.* [19], propose both a spatial approach and a temporal filtering step. Unfortunately, their spatial detection algorithm is not fully detailed, and cannot be reimplemented. However, we can compare the temporal step of the present approach with the temporal step from [19], which has some similarities with ours. Therefore, we use our spatial detections as inputs and filter these detections with the temporal part of the approach from [19]. This algorithm requires the setting of a neighbourhood size on the left and right hand sides of the scratch. This parameter is not specified in [19], and we chose a horizontal neighbourhood of 5 pixels. The other necessary thresholds are given in [19].

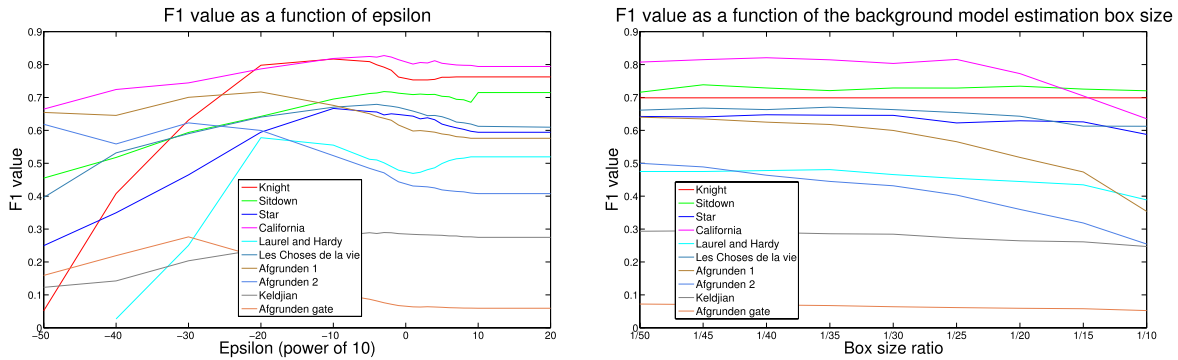


Fig. 5. Variation of the performance ( $f_1$ -score) of the spatial detection algorithm with respect to the values of  $\epsilon$  and the box size used for the background model estimation. The box size is a fraction of the image dimension for each sequence.

TABLE I  
RECALL, PRECISION AND F1 VALUES COMPARISON, IN PERCENTAGE. WE COMPARE OUR SPATIAL AND TEMPORAL RESULTS WITH THOSE OF BRUNI *et al.*, GÜLLÜ *et al.* AND MÜLLER *et al.* IN SMALLER FONT ARE THE RESULTS OF THE PIXEL-WISE EVALUATION OF PRECISION FOR BRUNI’S AND GÜLLÜ’S ALGORITHMS

Evaluation	Algorithm	Film									
		Knight	Sitdown	Star	California	Laurel-Hardy	Choses Vie	Afgrunden 1	Afgrunden 2	Keldjian	Gate
Recall	Bruni	<b>100.00</b>	<b>80.93</b>	<b>95.00</b>	<b>82.07</b>	41.87	43.43	75.11	59.43	12.63	49.81
	Güllü	<b>100.00</b>	<b>80.93</b>	<b>95.00</b>	35.56	38.74	29.99	68.75	52.98	11.48	31.29
	Müller	09.57	17.26	41.20	54.07	38.65	16.31	50.65	42.34	14.06	47.12
	Spatial	79.34	73.72	82.21	81.10	<b>59.35</b>	<b>64.88</b>	<b>86.66</b>	<b>94.01</b>	<b>77.35</b>	<b>89.03</b>
	Temporal	79.34	72.63	79.68	80.82	59.26	60.11	86.39	93.71	77.16	<b>89.03</b>
Precision	Bruni	29.54	56.47	56.87	10.79	08.84	25.06	09.35	07.85	07.11	02.45
	Pixel-wise evaluation	23.93	28.54	13.67	06.62	07.88	05.67	03.20	01.85	01.75	00.48
	Güllü	29.54	56.47	<b>57.32</b>	11.91	12.01	25.45	10.27	07.43	05.98	02.27
	Pixel-wise evaluation	23.93	28.54	13.78	06.67	10.37	03.47	03.56	01.74	01.47	00.44
	Müller	<b>79.86</b>	51.62	46.57	<b>81.22</b>	31.89	<b>75.48</b>	48.17	35.28	29.08	03.06
	Spatial	71.70	<b>72.07</b>	53.15	79.60	38.31	67.80	45.85	28.01	17.43	03.32
	Temporal	73.97	71.85	56.25	79.89	<b>45.29</b>	69.19	<b>67.76</b>	<b>50.92</b>	<b>38.06</b>	<b>11.93</b>
F1-score	Bruni	45.61	66.52	71.15	19.08	14.60	31.78	16.62	13.86	09.10	04.68
	Güllü	45.61	66.52	<b>71.50</b>	17.84	18.33	27.53	17.86	13.03	07.86	04.23
	Müller	17.09	25.87	43.72	64.92	34.95	26.82	49.38	38.49	18.96	05.74
	Spatial	75.33	<b>72.88</b>	64.56	80.34	46.56	<b>66.31</b>	59.97	43.16	28.45	06.40
	Temporal	<b>76.56</b>	72.24	65.95	<b>80.35</b>	<b>51.34</b>	64.33	<b>75.94</b>	<b>65.99</b>	<b>50.98</b>	<b>21.04</b>
Execution time (s)	Bruni	0.03	0.05	0.08	0.03	0.03	0.12	0.08	0.08	0.06	0.06
	Spatial	0.49	106.22	98.67	17.13	1.85	2.52	5.33	4.65	2.24	12.63

We are presented with a difficulty when comparing our algorithm, which produces a precise description of the line scratches, with methods that suppose that line scratches cover the entire height of the image. Our approach has an advantage in terms of precision, whereas the second type has an advantage with respect to recall. In terms of recall, we shall evaluate all the algorithms on a pixel-wise basis, in other words the number of annotated scratch pixels detected divided by the total number of scratch pixels. For the algorithms of Bruni *et al.* and Güllü *et al.* we shall consider that all the pixels in a detected column are detected. On the other hand, a fair and meaningful comparison of precision is more difficult to achieve. Naturally, we should evaluate our algorithm and that of Müller *et al.* on a pixel-wise basis, that is to say the number of annotated pixels detected divided by the total number of pixels detected. However, if we do this for the algorithms in [4] and [9], we shall bias the precision of their algorithms, especially when short scratches are present. Therefore, for these algorithms, we shall consider that a detection is “correct” if it touches at least one annotated pixel. This obviously confers a considerable advantage on Bruni’s

and Güllü’s algorithms, but it would be unfair to evaluate them otherwise. For comparison, we have also included the pixel-wise precision evaluation for Bruni’s and Güllü’s algorithms in Table I, which are written in smaller font below the main evaluation. For all of these evaluations, we allow a spatial detection error of 2 pixels.

Tests were carried out on ten film sequences of varying characteristics. The first three (“Knight”, “Sitdown” and “Star”) are commonly found in the line scratch literature, and are found in Kokaram’s book [17]. “California” and “Laurel and Hardy”, contain straight, vertical scratches, similar to the first three examples. “Les Choses de la Vie” displays scratches which are more difficult to detect (not completely straight, slanted and/or faint). While the first six sequences are useful for the evaluation of our spatial line scratch detection algorithm, the temporal filtering step is of little use in these cases, since the sequences are either very short, or contain no false alarms which may be rejected using temporal aspects. The last four sequences are longer and illustrate the improvement on precision we are able to obtain by using temporal filtering.



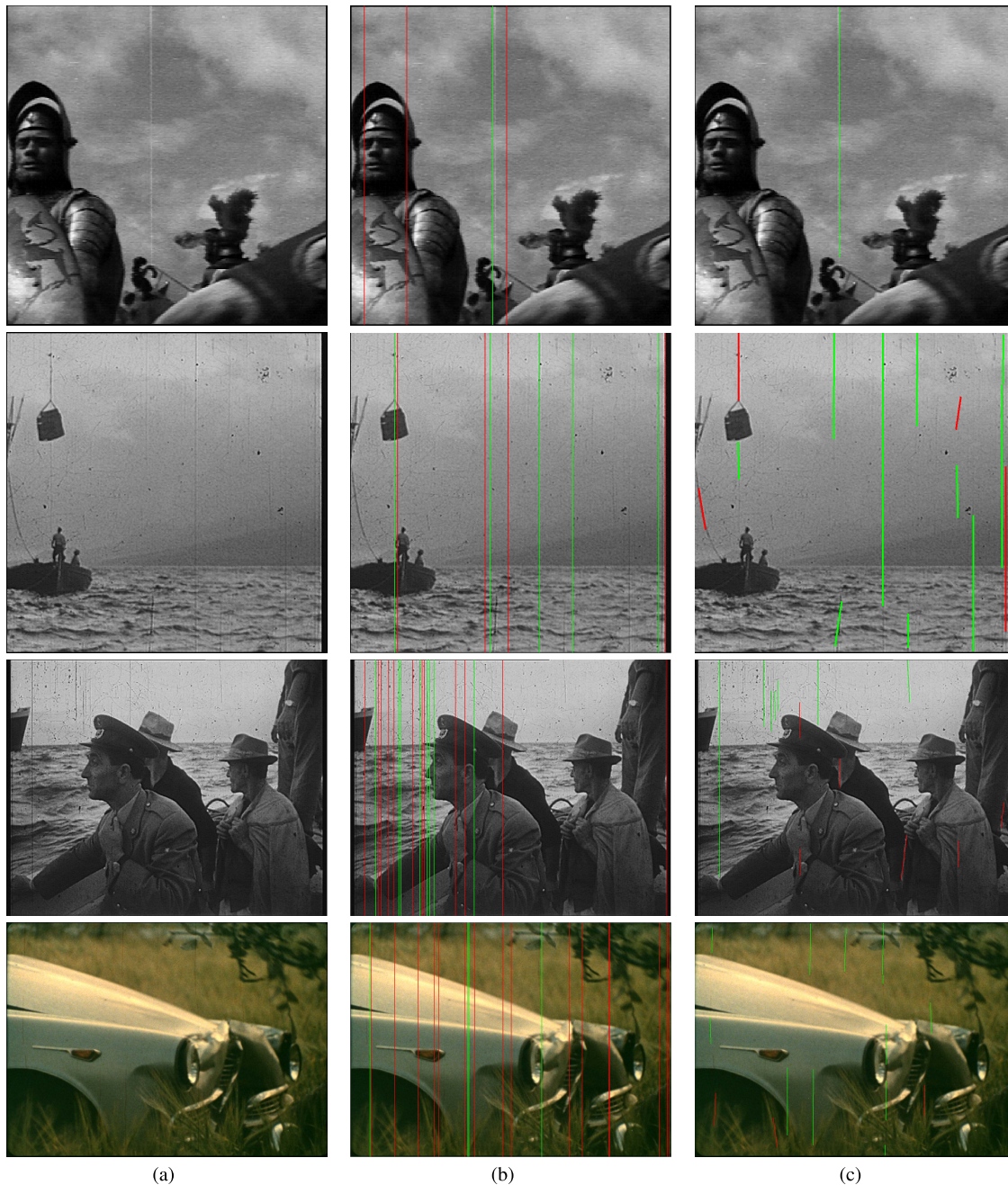


Fig. 6. Spatial scratch detections. From top to bottom: “Knight,” “Sitdown,” “Star” and “Les Choses de la Vie.” Correct detections are shown in green, false alarms in red. Please note that we enforce a minimum scratch height of one tenth of the height of the image for our algorithm. (a) Original frame. (b) Detection from [4]. (c) Our spatial detection.

Fig. 6 shows some examples of our spatial detection results, compared with those of the algorithm presented in [3]. Table I presents the results of our two algorithms, in comparison with the results of the algorithms of Bruni, Güllü and Müller. We have also shown the average execution times per frame of our spatial algorithm with that of Bruni *et al.*, for each sequence. The execution times of the temporal algorithms were not compared since they do not constitute a significant bottleneck of the detection process. The computer architecture is an Intel Core i5 CPU (2.67 GHz), and the code was written in Matlab, with mex functions for certain parts of the algorithm.

Our execution times are slower than those of Bruni *et al.*, due to the fact that we test many segments in the image. However, it is clearly possible in the present case to use acceleration techniques as found in [23].

The annotation of the sequences was done by manually noting the beginning and end points of each scratch segment. In the case of scratches which were not completely straight, several consecutive segments were annotated. We performed this annotation task because, to the best of our knowledge, no standard database exists for scratch detection. The complete annotated sequences, as well as the

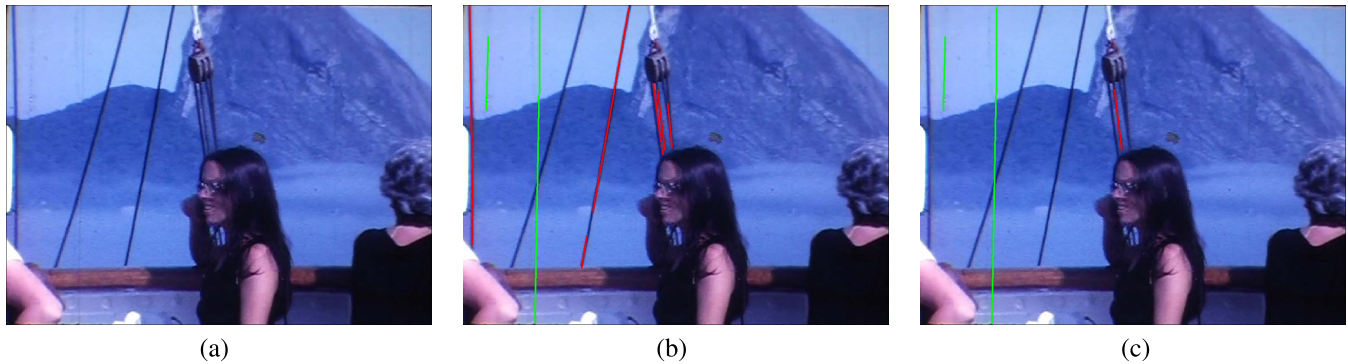


Fig. 7. Detections from a frame of “Keldjian” with false alarms due to thin vertical objects removed with temporal filtering. Correct detections are shown in green, false alarms in red. (a) Original frame. (b) Spatial detections. (c) Detections after temporal filtering with the proposed method.

detection results can be downloaded from the following address: <http://www.enst.fr/~gousseau/scratches>.

#### A. Recall

In the first four sequences, Bruni’s and Güllü’s algorithms produce better recall than ours. This is due to the fact that our evaluation gives the benefit of the doubt to these algorithms by considering that all of the pixels in a detected column are detected. Our spatial algorithm, on the other hand, must determine the beginning and end points of the scratch with high spatial precision. In the “Knight” sequence, for example, we are also able to detect the correct column indices of the scratches 100 percent of the time, but sometimes miss certain parts of a scratch.

In the remaining sequences, we see our spatial algorithm’s strong points: it is able to detect scratches with varying characteristics. This may be explained by our algorithm’s ability to detect and represent slanted and disjointed scratches as a collection of segments with varying length and angle. It is also able to detect faintly contrasted scratches, even in highly textured areas (as in the extract from “Les Choses de la Vie” example). Contrary to the other tested methods, our recall is high for all sequences.

It can be seen that the algorithm of Müller *et al.* produces relatively poor recall on all of the sequences. This is because their corresponding rejection criterion is often verified by true scratches. Indeed, due to the lack of an efficient tracking scheme, a large number of correct detections are rejected as long as the local motion is large enough (more than 0.2 pixels per frame, in absolute value). Furthermore, in practice many “temporal holes” are observed in the resulting scratch detections. This means that a restoration process using these detections is likely to produce flickering scratches, which may be a very undesirable result.

It should be noted that algorithms which filter the detections according to a temporal criterion (ours as well as those of Güllü *et al.* and Müller *et al.*) can only decrease recall (since no new detections are produced). Therefore, an important property of the temporal filtering stage is that it should not deteriorate recall. It may be observed in Table I that our algorithm induces very little loss of recall, with a maximum loss of 4.77 percent in the “Les Choses de la Vie” sequence.

Müller’s algorithm, on the other hand, decreases recall by a maximum of 69.77 percent (“Knight”).

#### B. Precision

As stated earlier, our evaluation procedure confers a strong advantage on Bruni’s and Güllü’s algorithms in terms of precision. In spite of this advantage, our spatial algorithm is able to outperform these algorithms in nine out of ten of the sequences. This performance is due to the *a contrario* grouping and validation process, which limits the number of false detections in noisy situations as well as in textured areas.

As previously mentioned, the first six sequences do not present any interesting situations in terms of temporal filtering, which explains why the precision is practically the same for our spatial and temporal algorithms for these sequences. In the last four sequences, however, our temporal algorithm significantly improves the precision of our spatial algorithm, with a maximum increase of 22.91 percent (“Afgrunden 2”). In Fig. 7, a visual example of the benefits of the temporal filtering step may be seen. The temporal filtering step increases precision in all of the sequences apart from “Sitdown”, which decreases by 0.22 percent only.

Müller’s algorithm presents good precision on most of the videos. In three cases, this algorithm outperforms our temporal approach. Unfortunately, this precision comes with very low recall, which is of little use for restoration purposes. This is reflected in the *F1*-scores, which are generally quite low.

Güllü’s algorithm fails to significantly increase precision because the MAD threshold introduced in [9] is not robust enough. It is sufficient that one MAD value be quite high for an entire trajectory to be validated as a true scratch. In practice, this happens often even though we allow a very tolerant MAD threshold. Conversely, our algorithm robustly determines a complete trajectory, so that a better-informed decision can be made.

#### C. *F1*-Score

The *F1*-score is defined as the harmonic mean of the recall and precision:

$$F = 2 \frac{\text{recall} * \text{precision}}{\text{recall} + \text{precision}} \quad (14)$$



Fig. 8. An illustration of our spatial method's robustness to texture and noise on two highly textured test images. In the first example, our locally adaptive method produces only one false detection, whereas with a global background model estimation, the algorithm fails. The second example illustrates this principle on an image with real scratches. (a) Original frame. (b) Binary detection. (c) A *contrario* detection with a global background model estimation. (d) A *contrario* detection with a local background model estimation. (e) Original frame. (f) Binary detection image. (g) A *contrario* detection with a global background model estimation. (h) A *contrario* detection with a local background model estimation.

This score illustrates the performances of the methods more clearly than either the recall or precision alone. The results show that our spatial algorithm retains a good  $F1$ -score for all of the sequences, and outperforms Bruni's, Güllü's and Müller's algorithms in nine out of ten sequences. Furthermore, our temporal filtering step improves the  $F1$ -score of our spatial algorithm in all of the sequences apart from "Sitdown" where it decreases by 0.64 percent, and "Les Choses de la Vie" where it decreases by two percent. This is an important point, since it implies that this temporal filtering step may be used on any type of sequence (short, long, with or without motion) with practically no deterioration in the resulting detection.

To resume, both the spatial and temporal algorithms introduced in this paper provide a significant improvement on previous methods. The same set of parameters was used for all the sequences. We do not have to specify to our algorithm whether black or white scratches are being detected, which is a significant advantage over other methods. Furthermore, our algorithm produces pixel-precision detections, which can be crucial for avoiding restoring non-degraded parts of images.

#### D. Algorithm Parameters

In this section, we will have a short discussion about several of the more important parameters in our algorithm.



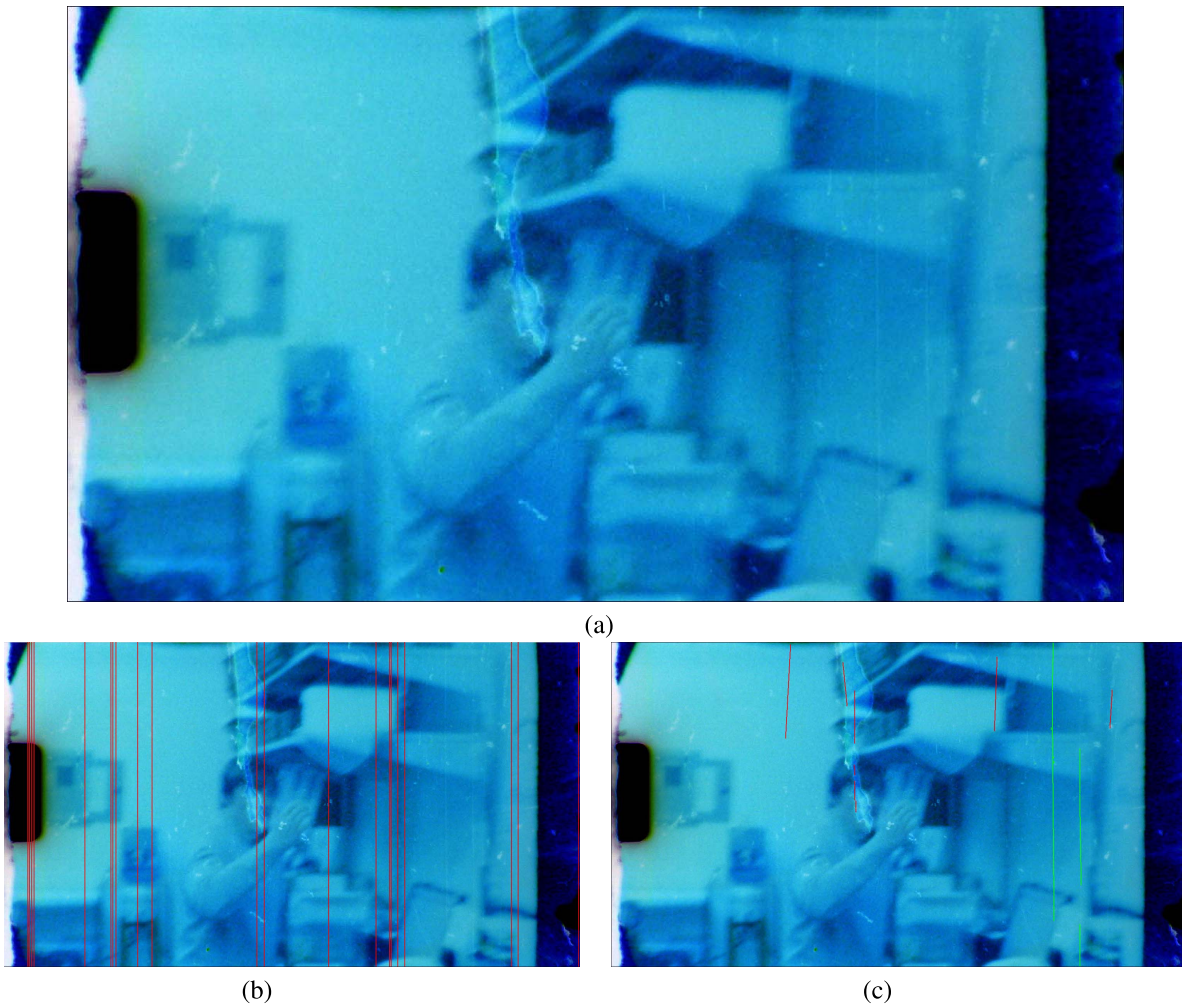


Fig. 9. A high definition scratched film ( $1074 \times 1920$ ). As in all other experiments, a minimum scratch length of one tenth of the image height was imposed. Correct detections are shown in green, false alarms in red. (a) Original frame. (b) Detection from [4]. (c) Our spatial detection.

Although the same parameters were used for all of our experiments, we would like to illustrate the evolution of the algorithm's performance with respect to certain key parameters. In particular, we inspect the influence of the box size used for local estimation and the parameter  $\varepsilon$ . These two parameters are of great importance, since they determine our background model and line scratch detection threshold. The results of this analysis may be seen in Fig. 5.

We can see that the  $F1$  score is relatively stable around  $\varepsilon = 1$ , which is to be expected given the log-dependence of the NFA on  $\varepsilon$  (see [7]). It may be seen that in some sequences, the maximum value of the  $F1$  score is not centred on  $\varepsilon = 1$ . However, this does not mean that we have chosen the incorrect value of  $\varepsilon$ . Since the spatial algorithm is incapable of distinguishing between true scratches and thin vertical structures, lowering  $\varepsilon$  does not imply an increase of the  $F1$  score; we need the temporal filtering step for this. We may also see that the  $F1$  score is stable with respect to the box size used for empirically estimating the local background model. We have shown of range of values from  $\frac{1}{10}$  to  $\frac{1}{50}$  of the image dimension. This means that it is a reasonable choice to make this parameter dependant on the image size.

The thresholds on our scratch model (width of the median filter, the parameters  $s_{med}$  and  $s_{avg}$ ) were empirically determined for the scratches which were found in our sequences. However, they may need to be changed for higher resolution images, in which the scratch may cover more pixels.

Another parameter which may be discussed is  $\tau_m$ , the motion threshold which flags a trajectory as being a series of false detections. The most important aspect of this threshold is that it should not be set too low otherwise all the true scratches in still frames will become flagged as false alarms; we need to be very sure that the scene has moved significantly before taking any action. Ideally, we would like to determine the maximum error of the global motion estimator of [22] and set  $\tau_m$  to a value greater than this. However, this may be too close an analysis for such a task. The parameter was set to a conservative value (10 pixels) and was sufficient to deal with all the sequences in our experiments.

#### E. Robustness to Noise and Texture

One of the major assets of a *contrario* detection procedures is their ability to control the number of false detections in noisy or cluttered images. Often, such methods rely on statistics that

are learned globally from the considered image. In our case, we found it necessary to estimate such statistics *locally*, in order to be resistant to textured regions.

In this section, we illustrate the ability of the approach to control the number of false detections and the importance of performing local statistical evaluation.

This can be observed in the images in Fig. 8. Due to the highly textured nature of the images, false detections are produced when we try to use our algorithm with no local estimation. When we introduce local estimation, detections are limited in areas with high detection density, which illustrates one of the strong points of our algorithm. In the image of the monkey, we produce one false alarm with local estimation, which is coherent with the chosen value for  $\epsilon$ .

### F. Scratch Detection in High Definition Images

In Fig. 9, an example of a high definition image containing scratches is given. This example is interesting since, as was mentioned at the beginning of the paper, the restoration of films is being done for formats of high resolution. It can be seen that the algorithm from [3] is unable to detect the faint, white scratches present on the right hand side of the image, whereas the proposed method locates them with a high degree of spatial precision. In the example in Fig. 9, it may be seen that certain false alarms are present which are not due to vertical scene structures. This is likely due to the preliminary Gaussian filtering stage of our spatial detection algorithm. This filtering may not be sufficient for dealing with noise, since the standard deviation is fixed. Setting this parameter adaptively could help performance.

## VI. FUTURE WORK

There are several aspects of the current work which could be developed further. Firstly, the spatially varying *a contrario* model presented here could potentially be applied to other detection problems, such as the detection of parametric shapes in preprocessed images. Secondly, the global motion model estimates one affine motion only. This is obviously a relatively simple model, and several motions could potentially be estimated to allow for more complex situations. However, the estimation of more complex motions would decrease the robustness of the temporal filtering step. This decrease in robustness must be compared with the robustness of actually tracking the true scratches (as in [9]), and a compromise or a mixture of these methods could therefore be of great interest. Finally, although we have carried out quantitative testing of our detection procedure, the true evaluation of defect detection lies in the final restored sequence. In future, therefore, we could also evaluate the detection qualitatively by inspecting the resulting restoration.

## VII. CONCLUSION

In this paper we have presented a precise spatial line scratch detection algorithm and a temporal filtering step which eliminates false alarms. The spatial algorithm uses an *a contrario* validation step to determine if the detected segments are visually significant or not. Our algorithm provides a

precise description of the detected scratches, which is not given by any other fully automatic algorithm. Furthermore, it has similar performance to the state-of-the-art in simple cases, and outperforms the latter considerably in more difficult situations. The temporal filtering step eliminates false alarms which are caused by thin vertical structures belonging to the scene, by identifying scratch detections which are coherent with the scene's motion or which stop at a scene cut. Our experiments and evaluations were carried out without any sequence-dependant tuning, which illustrates the robustness of the algorithms.

## APPENDIX

### MAXIMALITY PROPERTY AND Hoeffding's APPROXIMATION

In this section, we prove two properties of the meaningful segments defined using the NFA relying on Hoeffding's approximation, Formula (7). These properties are necessary for speeding up the search for maximal segments, as explained in Section III-B.3. These properties are as follows:

- If one appends a 0 (non-detected pixel) to the segment, its meaningfulness decreases (its NFA increases)
- If one appends a 1 (detected pixel) to the segment, its meaningfulness increases (its NFA decreases)

Using Formula (7), these properties reduce to:

$$H[l, k] \leq H[l + 1, k], \quad (15)$$

and

$$H[l, k] \geq H[l + 1, k + 1], \quad (16)$$

where  $H$  is defined as

$$H(l, k) := \exp(-k \log \frac{k}{lp} - (l - k) \log \frac{l - k}{l(1 - p)}), \quad (17)$$

where  $lp < k < l$ . Since the exponential function is strictly increasing, we need to study the following function:

$$f(k, l) = -k \log \frac{k}{lp} - (l - k) \log \frac{l - k}{l(1 - p)}. \quad (18)$$

Now, let us prove Equations (15) and (16).

For Equation (15) to be true, we need the partial derivative of  $f$  with respect to  $l$  to be positive:

$$\frac{\partial f(k, l)}{\partial l} = k \frac{1}{l} - \left[ \log \frac{l - k}{l(1 - p)} + (l - k) \frac{\partial}{\partial l} \log \frac{l - k}{l(1 - p)} \right]. \quad (19)$$

We have the partial result:

$$\frac{\partial}{\partial l} \log \frac{l - k}{l(1 - p)} = \frac{k}{l(l - k)}.$$

Therefore,

$$\begin{aligned} \frac{\partial f(k, l)}{\partial l} &= \frac{k}{l} - \log \frac{l - k}{l(1 - p)} - (l - k) \frac{k}{l(l - k)} \\ &= \log \frac{l - lp}{l - k} \end{aligned}$$

We know that  $l - lp > l - k$ , because  $lp < k$  (the condition for the Hoeffding approximation to hold true). Therefore, the right hand term of Equation (19) is strictly positive, so

that  $f(k, l)$  increases strictly with  $l$ . This means that when a 0 is appended to a segment, its meaningfulness decreases (since its probability increases). This shows the first inequality (Equation (15)) in the case of Hoeffding's approximation.

Now we prove Equation (16). This case is slightly more complicated, since two variables ( $k$  and  $l$ ) vary at the same time. However, since we add the same quantity to both these variables, it is enough to study the partial derivative of  $f(k+t, l+t)$  with respect to  $t$ .

We have

$$f(k+t, l+t) = -(k+t) \log \frac{k+t}{(l+t)p} - (l-k) \log \frac{l-k}{(l+t)(1-p)},$$

so that:

$$\begin{aligned} \frac{\partial f(k+t, l+t)}{\partial t} &= -\log \frac{k+t}{(l+t)p} - (k+t) \frac{\partial}{\partial t} \log \frac{k+t}{(l+t)p} \\ &\quad - (l-k) \frac{\partial}{\partial t} \log \frac{l-k}{(l+t)(1-p)}. \end{aligned}$$

Now

$$\frac{\partial}{\partial t} \log \frac{k+t}{(l+t)p} = \frac{l-k}{(k+t)(l+t)},$$

and

$$\frac{\partial}{\partial t} \log \frac{l-k}{(l+t)(1-p)} = -\frac{1}{l+t},$$

so that

$$\begin{aligned} \frac{\partial f(k+t, l+t)}{\partial t} &= -\log \frac{k+t}{(l+t)p} - (k+t) \frac{l-k}{(k+t)(l+t)} \\ &\quad + (l-k) \frac{1}{l+t} \\ &= \log \frac{lp+tp}{k+t}. \end{aligned}$$

This quantity is strictly negative, since  $lp < k$  and  $tp < t$ . Therefore,  $f(k, l)$  decreases when  $k$  and  $l$  increase from the same quantity. Therefore, we have in particular, that  $H(l+1, k+1) < H(l, k)$ , meaning that meaningfulness increases if a 1 is appended to a segment. Thus, the second inequality (Equation 16) holds true.

We have proven the two necessary properties for a segment to be maximal meaningful in the case of Hoeffding's approximation, thus we can safely prune the search for maximal meaningful segments as explained in Section III-B.3.

#### ACKNOWLEDGMENT

The authors would like to express their gratitude to Vittoria Bruni and Domenico Vitulano for having made an implementation of their method available and for having answered several questions about their line scratch detection. The authors thank Anil Kokaram for having made his image sequences available, which may be found in [17]. The authors also thank Jesse Crowder of Djifi, Studio Canal and Alvaro Pardo of the Universidad Católica del Uruguay for their permission to use their video content. Finally, the authors acknowledge the internet archive <http://www.archive.org> for the video material which is available online.

#### REFERENCES

- [1] (2013, Sep. 27). *Physical Damage, National Film and Sound Archive* [Online]. Available: <http://www.nfsa.gov.au/preservation/handbook/damage-films/physical-damage/>
- [2] T. Bretschneider, O. Kao, and P. J. Bones, "Removal of vertical scratches in digitised historical film sequences using wavelet decomposition," in *Proc. Image Vis. Comput. New Zealand*, Nov. 2000, pp. 38–43.
- [3] V. Bruni, P. Ferrara, and D. Vitulano, "Color scratches removal using human perception," in *Proc. 5th Int. Conf. Image Anal. Recognit.*, Jun. 2008, pp. 33–42.
- [4] V. Bruni and D. Vitulano, "A generalized model for scratch detection," *IEEE Trans. Image Process.*, vol. 13, no. 1, pp. 44–50, Jan. 2004.
- [5] V. Bruni, D. Vitulano, and A. Kokaram, "Line scratches detection and restoration via light diffraction," in *Proc. 3rd Int. Symp. Image Signal Process. Anal.*, vol. 1. 2003, pp. 5–10.
- [6] K. Chishima and K. Arakawa, "A method of scratch removal from old movie film using variant window by Hough transform," in *Proc. 9th Int. Symp. Commun. Inf. Technol.*, 2009, pp. 1559–1563.
- [7] A. Desolneux, L. Moisan, and J. M. Morel, "Meaningful alignments," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 7–23, Oct. 2000.
- [8] A. Desolneux, L. Moisan, and J. M. Morel, *From Gestalt Theory to Image Analysis: A Probabilistic Approach*, 1st ed. New York, NY, USA: Springer-Verlag, 2008.
- [9] M. K. Gullu, O. Urhan, and S. Erturk, "Scratch detection via temporal coherency analysis and removal using edge priority based interpolation," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2006, pp. 4591–4594.
- [10] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Amer. Statist. Association*, vol. 58, no. 301, pp. 13–30, Mar. 1963.
- [11] L. Joyeux, S. Boukir, and B. Besserer, "Film line scratch removal using Kalman filtering and Bayesian restoration," in *Proc. 15th IEEE Workshop Appl. Comput. Vis.*, Dec. 2000, pp. 8–13.
- [12] L. Joyeux, S. Boukir, and B. Besserer, "Tracking and MAP reconstruction of line scratches in degraded motion pictures," *Mach. Vis. Appl.*, vol. 13, no. 3, pp. 119–128, 2002.
- [13] L. Joyeux, O. Buisson, B. Besserer, and S. Boukir, "Detection and removal of line scratches in motion picture films," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1999, pp. 548–553.
- [14] K.-T. Kim and E. Y. Kim, "Film line scratch detection using neural network and morphological filter," in *Proc. IEEE Conf. Cybern. Intell. Syst.*, Sep. 2008, pp. 1007–1011.
- [15] A. Kokaram, "Detection and removal of line scratches in degraded motion picture sequences," *Signal Process.*, vol. 1, no. 9, pp. 5–8, Sep. 1996.
- [16] A. Kokaram, F. Pitie, D. Corrigan, D. Vitulano, V. Bruni, and A. Crawford, *Advances in Automated Restoration of Archived Material*. Boca Raton, FL, USA: CRC Press, 2011.
- [17] A. C. Kokaram, *Motion Picture Restoration: Digital Algorithms for Artefact Suppression in Degraded Motion Picture Film and Video*, 1st ed. London, U.K.: Springer-Verlag, 1998.
- [18] R. Lienhart, "Reliable transition detection in videos: A survey and practitioners guide," *Int. J. Image Graph.*, vol. 1, no. 13, pp. 469–486, 2001.
- [19] S. Müller, J. Bühler, S. Weitbruch, C. Thebault, I. Doser, and O. Neisse, "Scratch detection supported by coherency analysis of motion vector fields," in *Proc. 16th IEEE Int. Conf. Image Process.*, Nov. 2009, pp. 89–92.
- [20] A. Newson, A. Almansa, Y. Gousseau, and P. Perez, "Temporal filtering of line scratch detections in degraded films," in *Proc. 20th IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 1–5.
- [21] A. Newson, P. Pérez, A. Almansa, and Y. Gousseau, "Adaptive line scratch detection in degraded films," in *Proc. 9th Eur. CVMP*, 2012, pp. 66–74.
- [22] J. M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *J. Vis. Commun. Image Representation*, vol. 6, no. 4, pp. 348–365, 1995.
- [23] R. G. von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, Apr. 2010.





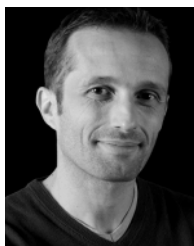
**Alasdair Newson** received the M.Sc./Engineering degree in computer science from Université de Technologie de Compiègne, France, in 2010. He is currently pursuing the Ph.D. degree with Technicolor and Télécom Paris-Tech, France. His research interests include image and video restoration, editing, and inpainting.



**Yann Gousseau** received the Engineering degree from Ecole Centrale de Paris, France, in 1995, and the Ph.D. degree in applied mathematics from the University of Paris-Dauphine in 2000. He is currently a Professor with Télécom Paris-Tech. His research interests include the mathematical modeling of natural images and textures, stochastic geometry, image analysis, computer vision, and image processing.



**Andrés Almansa** received the H.D.R., Ph.D., and M.Sc./Engineering degrees in applied mathematics and computer science from Université Paris-Descartes, ENS Cachan, France, and Universidad de la Republica, Uruguay, where he has been an Associate Professor since 2004. His current interests include image restoration and analysis, sub-pixel stereovision and applications to earth observation, high quality digital photography, and film restoration.



**Patrick Pérez** received the Ph.D. degree from the University of Rennes in 1993 and joined INRIA as a Full Time Researcher in 1994. From 2000 to 2004, he was with Microsoft Research Cambridge. In 2009, he joined Technicolor as a Senior Researcher. He is currently a member of the editorial board of the *International Journal of Computer Vision*. His research interests include image description, search and analysis, and as well as photo and video editing.