

Detection and removal of video defects using rational-based techniques

Lazhar Khriji^{a,*}, Mahmoud Meribout^a, Moncef Gabbouj^b

^aDepartment of Electrical and Computer Engineering, College of Engineering, Sultan Qaboos University, P.O. Box 33 Muscat 123, Oman

^bInstitute of Signal Processing, Tampere University of Technology, P.O. Box 553, FIN-33101 Tampere, Finland

Available online 7 April 2005

Abstract

This paper presents a Rational and Vector Rational based interpolator methods for reconstruction of missing data in video sequences. The interpolation of missing data is important in many areas of image processing, including the restoration of degraded motion pictures, reconstruction of dropouts in digital video and automatic re-touching of old photographs. Here, a detection technique is investigated for localization of the defects, and then a spatial vector rational interpolator algorithm is proposed to, reconstruct the missing data. This algorithm exhibits desirable properties, such as, edge and details preservation and accurate chromaticity estimation. In such approach, color image pixels are considered as three-component vectors in the color space that is more appropriate for the human visual system. Therefore, the inherent correlation that exists between the different color components is not ignored. This leads to better image quality compared to that obtained by component-wise or marginal processing. The experimental results demonstrate the usefulness of the vector rational interpolator in an application involving the restoration of defects in video sequences. The resulting edges obtained using the proposed interpolator are free from blockiness and jaggedness. The complexity evaluation of the algorithm shows that the implementation of the algorithm on a dedicated IMAP-based parallel hardware architecture can lead to an execution time of 5.7 and 15.6 ms for (256×256) binary and color images, respectively.

© 2005 Elsevier Ltd. All rights reserved.

Keywords: Video sequences; Interpolation; Vector rational interpolator; Missing data; Defect reconstruction; Hardware implementation

1. Introduction

Film and video archives nowadays go through an accelerated process of degradation. Even if films are stored in protective conditions, the degradation process is not stopped, but rather slowed down. Missing data and noise are common forms of degradation in real film and video data. Video data manifests as dropout in the digital stream in the case of digital broadcasting. While there are several chemical and physical treatments that are able to clean, or at least slow down some types of degradations, in many cases it is impossible to do so, or the costs involved get too high. In fact, since the preservation of the cultural heritage plays an important role in our society, motion picture

restoration has drawn a lot of attention lately. As a result, quite a number of digital restoration algorithms using powerful computers are now available [9,12]. Some of these algorithms use spatial information to recover a damaged area, while others use temporal information from consecutive frames, or a combination of them.

In film the degradation problem is caused by abrasion of the film material and the effect is called ‘Dirt and Sparkle’. Numerous work have considered the removal of this artifact as a two stage process, first detect the missing locations [6,11] and then reconstruct the underlying image data using a spatio-temporal image sequence interpolation process. The reconstruction stage may be further specified as a motion reconstruction followed by an image reconstruction stage. The motion interpolation stage is a crucial step in generating useful interpolated data since in the regions of missing data motion estimates are completely unreliable. Indeed, most of techniques in use today for cinematographic film restoration are based on chemical and mechanical manipulations.

* Corresponding author.

E-mail addresses: lazhar@squ.edu.om (L. Khriji), meribout@squ.edu.om (M. Meribout), moncefgabbouj@tut.fi (M. Gabbouj).

In contrast to classical image sequence restoration, an automatic digital film restoration system is further complicated by severe constraints:

- The old films must be scanned at high resolutions in order to preserve the definition and the visual quality of the motion picture images.
- The restoration algorithms must also preserve the visual quality of the films.
- The processing power has to be cheap, processing time as short as possible and the restoration process should be automated, to cut exploitation costs.

This paper addresses the problem of missing data detection and reconstruction for black and white image sequences and then the forward extension to color image sequences. It introduces an alternative efficient adaptive detection algorithms and the reconstruction of missing information in an easy manner. Firstly, we are concerned by the detection of damaged regions. The algorithm is working on spatial domain, which makes it faster than that using motion estimation. Secondly, the interpolation's algorithm is based on rational and vector rational filters which were recently proposed in [7,8]. In fact, the adaptive algorithm using this vector approach yields better interpolated images than those obtained with a number of linear and nonlinear techniques [2,10]. Furthermore, the nonlinearity of the operator can reconstruct the defects, which have been detected by the pre-localization step, without introducing overshoot artifacts in the restored image frames.

The reconstruction algorithm is based on the fact that, although images are highly non-stationary signals, a strong correlation does exist between pixels belonging to the same region. For this reason, the interpolation of a defect should take into account the statistics and properties of the regions that hypothetically contain the defect. Thus, our interpolation algorithm first checks the existence of edges in order to take them into consideration; the edge orientation is estimated and the most convenient data to be used in the reconstruction of the missing pixels are selected.

In our algorithm we assume that:

- The direction of edges does not change inside the defect region.
- The contrast of edges remains unaltered inside the defect region.

This is equivalent to assuming the stationarity of features of edges inside the defect region and permits an easy but effective reconstruction. The reliability of these assumptions depends on the size of the defect; therefore, the smaller the defect is, the more accurate the interpolation is. In addition, a parallel and low cost hardware architecture based on the IMAP hardware is proposed in this paper. The complexity evaluation shows that the algorithm can be

performed in 5.7 and 15.6 ms for binary and color images, respectively (i.e. under frame rate).

2. Detection of defects in black and white sequences

2.1. Classification

A classification, of motion picture defects, has been produced based on their spatio-temporal characteristics. The main interest of this classification is that each category needs to be treated using different sorts of algorithms:

- Global defects. These defects affect each frame of the sequence in a global way. For example, the brightness of the images may vary along the sequence in a periodic way. Often, these defects are not visible on one single frame; we have to view the sequence in motion in order to see them.
- Local time-correlated defects. These defects are local (i.e. they affect a relatively small part of each frame) and correlated in time (i.e. their position in one frame gives us clues about their position in the next frame). For example, white scratches running vertically across many frames belong to this category. They are produced by abrasion and their position is almost the same in each frame they appear in.
- Local non-time-correlated defects LNTC defects have been defined in the introduction as satisfying two characteristics:
 - (1) They are local: they affect a relatively small part of an image, destroying the data present in this area. They have a more or less well-determined shape.
 - (2) They are not correlated in time. The position in a frame does not give any information about their position in the other frames.

White dots produced by electrostatic electricity at shooting time, crackles of the varnish layer and dark blotches caused by dirt accumulation belong to this category. These defects are very common in old motion pictures. In this paper we develop an algorithm for the restoration of LNTC defects. The restoration of the other kinds of defects will not be treated here.

2.2. Detection technique

The description of the detection algorithm is as follows:

- A horizontal segment of length D_x is applied to the test image in order to detect the width of the scratch. If the detected width was greater than D_x , then the detected horizontal segment would be considered as background. Otherwise, it would be considered as the width of the scratch. Checking each pixel with pixel after it does this. If the difference between the pixel under consideration

and the pixel after it less than the threshold level T , then a connection of two pixels is obtained and then the next pixel is checked. Again if the same scenario to the third pixel happened then a connection of three pixels obtained and so on till the length of D_x is reached. On the other hand, if the difference between the pixel under consideration and the next pixel is greater than the threshold level T , then this pixel would be treated as not defected pixel and this process is restarted again to check the next pixel. Based on that, there are possibilities of having zero connection, a connection of two pixels, a connection of three pixels,..., and a connection of D_x pixels. The result of this step is shown in Fig. 9(b).

- Next, the mean of each column of the image obtained in the previous step is computed. The result will be a one-dimensional image, which is shown in Fig. 9(c). Based on this result, columns, which have a high value in the one-dimensional image, will be considered as the marker of the position of the scratch in the original test image, while columns with a low value will be excluded since these columns do not belong to the damaged columns (scratch area).
- Finally, the resulting one-dimensional image is threshold with the value of $G1$ in order to obtain the columns of the original image, which contain vertical scratches. In fact, some scratches do not cover the whole height of the image. Therefore, a pixel will be considered to belong to a scratch if its column was been marked $G1$ and if its value in the original test image greater than $G2$. Thus, the mask is obtained for the scratch. Fig. 9(d) shows the binary image.

2.3. Reconstruction algorithm

We describe in the sequel the reconstruction method of the defect information after its localization.

2.3.1. Method

According to Fig. 1 and in order to reconstruct the X sample at the central position, the interpolator operates on four consecutive alternate samples of known data, $\mathbf{a}_3, \mathbf{a}_1, \mathbf{b}_1,$ and \mathbf{b}_3 . The nonlinear interpolator has already been proposed in [7], where a vector rational function is used in order to weight the contributions to X of its two adjacent samples (\mathbf{a}_1 , and \mathbf{b}_1) according to the differences ($\mathbf{a}_1 - \mathbf{b}_3$) and ($\mathbf{b}_1 - \mathbf{a}_3$). The resulting operator is able to reconstruct sharp edges more accurately than a linear filter because it is able to roughly detect their position with sub-pixel accuracy. Using the scalar version [7], the interpolated

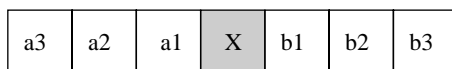


Fig. 1. Samples considered for one-dimensional interpolation. Sample X is to be constructed from samples $\mathbf{a}_3, \mathbf{a}_1, \mathbf{b}_1$ and \mathbf{b}_3 .

pixel X is computed as follows

$$X = \alpha \cdot \mathbf{a}_1 + (1 - \alpha) \mathbf{b}_1 \tag{1}$$

where

$$\alpha = \frac{1 + k(\mathbf{a}_1 - \mathbf{b}_3)^2}{2 + k[(\mathbf{a}_1 - \mathbf{b}_3)^2 + (\mathbf{b}_1 - \mathbf{a}_3)^2]} \tag{2}$$

The parameter k is some positive constant and is used to control the amount of the nonlinear effect. The behavior of the proposed interpolator structure for different positive values of the parameter k is the following:

- $k \approx 0$ the form of the filter is given as a linear filter:

$$X = \frac{\mathbf{a}_1 + \mathbf{b}_1}{2} \tag{3}$$

- For positive values of k , the $(\mathbf{a}_1 - \mathbf{b}_3)^2 + (\mathbf{b}_1 - \mathbf{a}_3)^2$ term perceives the presence of a detail and accordingly reduces the smoothing effect of the operator.

2.3.2. Algorithm

In order to apply the above interpolation scheme to defect interpolation, we will assume that defects are convex. This is true in most cases.

After detection and localization of defects the interpolation algorithm will be applied only on the defected regions. Let us consider the one-dimensional interpolator described in Section 2.3.1 using the rational interpolator approach. Assuming that the defect width, $dx=5$, and the available data are \mathbf{a}_i 's and the \mathbf{b}_i 's in the white boxes (Fig. 2). We denote by $R(\mathbf{a}_4, \mathbf{a}_1, \mathbf{b}_1, \mathbf{b}_4)$ the computed vector value of Eq. (1):

Step 1 Compute the value of the sample at the central position X_0 using the four known samples ($\mathbf{a}_4, \mathbf{a}_1, \mathbf{b}_1, \mathbf{b}_4$). $X_0 = R(\mathbf{a}_4, \mathbf{a}_1, \mathbf{b}_1, \mathbf{b}_4)$. In order to avoid the anti-aliasing effect, our interpolator takes into account the Nyquist rate requirement. This is why the periodicity is well respected.

Step 2 Compute temporarily:

- X_{-2} as a linear combination between \mathbf{a}_1 and X_0 : $X_{-2} = (2\mathbf{a}_1 + X_0)/3$
- X_2 as a linear combination between \mathbf{b}_1 and X_0 : $X_2 = (2\mathbf{b}_1 + X_0^3)/3$

Step 3 Compute the value of the sample X_1 by applying (1) using the four samples X_0, X_{-2}, X_2 and \mathbf{b}_2 : $X_1 = R(X_0, X_{-2}, X_2, \mathbf{b}_2)$.

Step 4 Compute the value of the sample X_{-1} by applying (1) using the four samples X_2, X_{-2}, X_0 and X_2 : $X_{-1} = R(\mathbf{a}_2, X_{-2}, X_0, X_2)$.

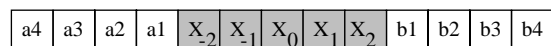


Fig. 2. Example of real detected defects with large $dx=5$. Only the data in the white boxes are available.

Step 5 Recompute the value of the sample X_{-2} by applying (1) over the samples a_3, a_1, X_{-1} and X_1 : $X_{-2} = R(a_3, a_1, X_{-1}, X_1)$.

Step 6 Recompute the value of the sample X_2 by applying (1) over the samples X_{-1}, X_1, b_1 and b_3 : $X_2 = R(X_{-1}, X_1, b_1, b_3)$.

The same algorithm and the same technique can be applied for any defect size dx by using the same procedure with respect to the Nyquist periodicity rule. The price paid is the time needed for finding the best edge orientation.

The above algorithm may also be applied to non-stationary defects. We are currently developing new masks and new interpolators for both random and stationary defects.

3. Detection and removal of scratches in color movies

Recent color films may have the same types of defects as the black and white films. Here, we will concentrate on dark and white scratches, which are caused mechanically by the head of the tape [3]. Fig. 11(b), shows a color picture having such type of defects. In order to use the same detection technique, the RGB color space should be transformed to the luminance–chrominance space $YC_B C_R$. Firstly, the detection technique is applied on the Y -channel, which is the luminance component. Secondly, we will assume that the width of a scratch is not greater than one pixel. Also, as it is clear seen from the color picture in Fig. 11(b), there are dark and white scratches, which are appearing as streaks, and their width does not exceed one pixel.

3.1. Working color space

Notwithstanding the RGB is the most popular color space used conventionally to store, process, display, and analyze color images, the human perception of color cannot be described using the RGB model [4]. Consequently, it is important to use color spaces, which are closely related to the human perceptual characteristics and suitable for defining appropriate measures of perceptual errors between color vectors. A number of such color spaces are used in areas such as multimedia, video communications (e.g. high definition television), motion picture production,

the printing industry, and graphic arts [1]. Among these, perceptually uniform color-spaces are the most appropriate to define simple yet precise measures of perceptual errors [4] luminance–chrominance space $YC_B C_R$. In general the Y -channel describes the luminance, the C_B the blue–yellow and the C_R the red–green chrominance components. The $YC_B C_R$ format is typically used in video coding and (MPEG and JPEG) so the use of the same format will avoid the extra computation required in conversion. Conversion from RGB to $YC_B C_R$ space is given, by the following linear equation

$$\begin{pmatrix} Y \\ C_B \\ C_R \end{pmatrix} = \begin{pmatrix} +0.29900 & +0.58700 & +0.11400 \\ -0.16875 & -0.33126 & +0.5000 \\ +0.5000 & -0.41869 & -0.08131 \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

Without loss of generality, the detection algorithm is applied to scalar values. Indeed, the Y -channel, which is the luminance component, will be used in the detection phase.

3.2. Defect detection and reconstruction

Any defected pixel can be identified by comparison of the brightness values of the analyzed pixel with the ones of the surrounding pixels. Correction of the defected pixel can be considered as a replacement of its value by the value of some function, whose arguments are values taken from a local window around this defected pixel.

The present algorithm is more capable in removing defects than some of the common defect-removal filters, without any speed penalty. This algorithm is based on a pixel identification concept. Rather than processing every pixel in a digital image, this algorithm ‘intelligently’ interrogates a sub-image region to determine which are the ‘defected’ pixels within the sub-image. With this knowledge, only the ‘defected’ pixels are eventually filtered, whereas the ‘non-defected’ pixels are left untouched (i.e. left unchanged). The corresponding diagram is presented in Fig. 3.

3.2.1. Detection procedure

Denote by $x(i, j)$ the pixel value at location (i, j) . First apply the quadratic operator at location (i, j) to determine the ‘Energy’ of that pixel [5] (this is the central pixel of a square

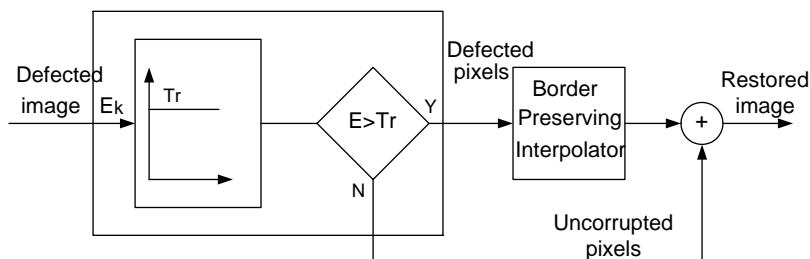


Fig. 3. Conceptual diagram of the defect detector and reconstruction.

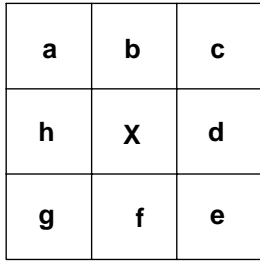


Fig. 4. Element of 3×3 sliding window centered on the pixel x for non-recursive implementation.

window 3×3 or 5×5, see Fig. 4)

$$E(i, j) = \max(E_1(i, j), E_2(i, j), E_3(i, j)) \quad (4)$$

where

$$E_1(i, j) = |2(x(k, l) - \mu)^2 - (x(k + 1, l) - \mu) \cdot (x(k - 1, l) - \mu) - (x(k, l - 1) - \mu) \cdot (x(k, l + 1) - \mu)| \quad (5)$$

$$E_2(i, l) = |2(x(k, l) - \mu)^2 - (x(k - 1, l - 1) - \mu) \cdot (x(k + 1, l + 1) - \mu) - (x(k + 1, l - 1) - \mu) \cdot (x(k - 1, l + 1) - \mu)| \quad (6)$$

$$E_3(i, j) = |4(x(k, l) - \mu)^2 - (x(k - 1, l - 1) - \mu) \cdot (x(k + 1, l + 1) - \mu) - (x(k + 1, l - 1) - \mu) \cdot (x(k - 1, l + 1) - \mu) - (x(k - 1, l) - \mu) \cdot (x(k, l - 1) - \mu) - (x(k, l + 1) - \mu) \cdot (x(k - 1, l) - \mu)| \quad (7)$$

and μ is the local mean of the pixels inside the processing window. This Energy operator produces different values for impulses with the same magnitude located in different backgrounds:

- This ‘Energy’ is then compared to a locally computed threshold Tr defined as

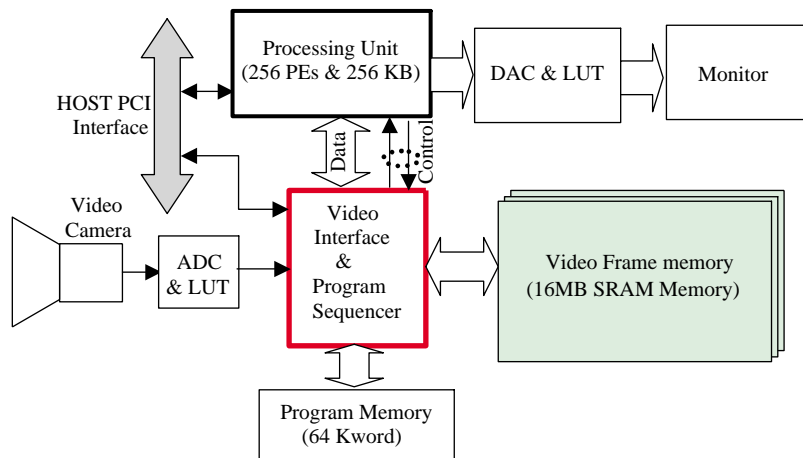


Fig. 5. Overall hardware architecture.

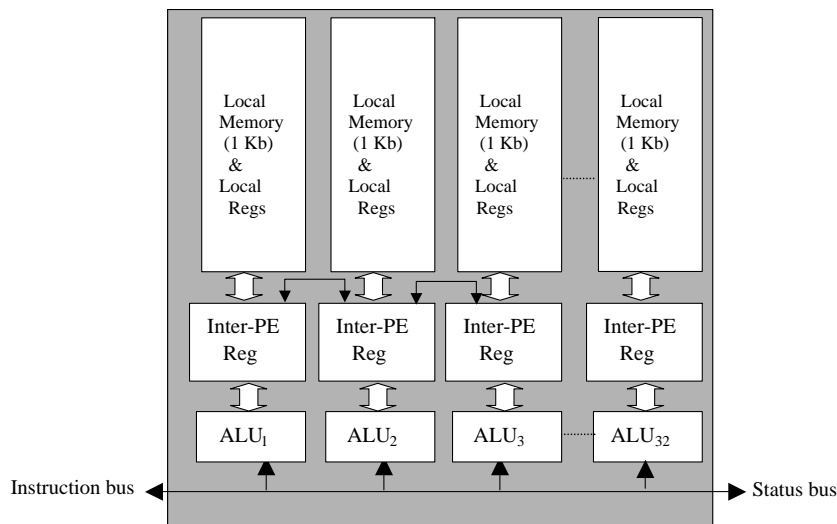


Fig. 6. IMAP LSI architecture.

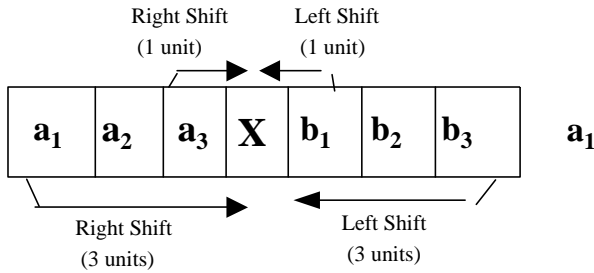


Fig. 7. Shift operations required to compute Eq. (2) for the pixel X .

$$\text{Tr} = \frac{\alpha}{N} \sum_{n=1}^N E_n \quad (8)$$

where E_n is the ‘Energy’ of the n th pixel in the surrounding neighborhood containing N pixels, and α is a constant determined empirically.

Those pixels whose ‘Energy’ exceeds the threshold value Tr are considered as a defected pixels, and then replaced at the output by the bidirectional vector rational interpolator of the pixels inside the filter window. Otherwise, the pixel value is retained.

This technique is based on two hypotheses. The first one is that if we can identify the ‘defected’ pixels, then we no longer need to process every pixel in the image. By not processing the ‘non-defected’ pixels, useful information is preserved. The second hypothesis is if only a selected subset of pixels is processed, considerable saving in computation would be possible.

3.2.2. Reconstruction: border-preserving interpolator

With reference to Fig. 4, the nonlinear interpolator operates on height samples of multivariate data, \mathbf{a} , \mathbf{b} , \mathbf{c} , \mathbf{d} , \mathbf{e} ,

\mathbf{f} , \mathbf{g} and \mathbf{h} (mask 3×3) to reconstruct the defected sample \mathbf{x} in the central position. In order to weight the contributions to \mathbf{x} of its height neighborhood samples, a vector rational function is used. The normalized weights verify two conditions. First, each weight is a positive number. Second, in a flat area (where we assume that the height neighbors of the pixel under consideration have the same value), the sum of the weights is equal to one, which ensures that the output is unbiased. The interpolator output is as follows

$$\begin{aligned} \mathbf{x} &= \frac{\sum_{\{u \neq v; u, v \in \{a, b, c, d, e, f, g, h\}\}} w_{uv} \cdot (\mathbf{u} + \mathbf{v})}{2 \sum_{\{u \neq v; u, v \in \{a, b, c, d, e, f, g, h\}\}} w_{uv}} \\ &= \frac{1}{W} \left(\frac{w_{ab} + w_{ac} + w_{da}}{2} \mathbf{a} + \frac{w_{ab} + w_{bc} + w_{bd}}{2} \mathbf{b} \right. \\ &\quad + \frac{w_{ac} + w_{bc} + w_{cd}}{2} \mathbf{c} + \frac{w_{bd} + w_{cd} + w_{da}}{2} \mathbf{d} \\ &\quad + \frac{w_{ef} + w_{eg} + w_{eh}}{2} \mathbf{e} + \frac{w_{ef} + w_{fg} + w_{fh}}{2} \mathbf{f} \\ &\quad \left. + \frac{w_{eg} + w_{fg} + w_{gh}}{2} \mathbf{g} + \frac{w_{fh} + w_{gh} + w_{he}}{2} \mathbf{h} \right) \quad (9) \end{aligned}$$

where $W = w_{ab} + w_{bc} + w_{cd} + w_{da} + w_{ef} + w_{fg} + w_{gh} + w_{he}$. The weights are computed as follows

$$w_{uv} = \frac{1}{8 + k \|\mathbf{v} - \mathbf{u}\|_p} \quad (10)$$

with $\mathbf{u}, \mathbf{v} \in \{\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{f}, \mathbf{g}, \mathbf{h}\}$ and $\|\cdot\|_p$ denotes l_1 - or l_2 -norm.

3.3. Hardware implementation of the algorithm

Fig. 5 shows the overall block diagram of the hardware architecture. Its modular and comprises:

- A core Processing Unit comprising 8 reprogrammable IMAP VLSI Processors (Fig. 5). Each IMAP circuit

Table 1

Overall performance of the algorithm for a frame

Frame type	Algorithm steps	Algorithm sequences	Operations	#Clock cycles/ execution time (ms)	
Binary frame	Step 1	Download $\mathbf{a}_1, \mathbf{a}_3, \mathbf{b}_1, \mathbf{b}_3$ parameters (Eq. (2))	4 Shift operations	$8/200 \times 10^{-6}$	
		Compute numerator of Eq. (2)	1 subtraction, 1 addition	$2/50 \times 10^{-6}$	
		Compute denominator of Eq. (2)	2 subtractions, 2 additions	$4/100 \times 10^{-6}$	
		Compute of Eq. (2)	1 division	$8/200 \times 10^{-6}$	
		Compute X of Eq. (1)	1 subtraction, 1 addition	$2/50 \times 10^{-6}$	
		Step 2	Computer \mathbf{X}_{-2}	1 addition, 1 division	$9/225 \times 10^{-6}$
		Step 3	Computer \mathbf{X}_2	1 addition, 1 division	$9/225 \times 10^{-6}$
		Step 4	Same as Step 1 above	Same as Step 1 above	$24/600 \times 10^{-6}$
		Step 5	Same as Step 1 above	Same as Step 1 above	$24/600 \times 10^{-6}$
		Step 6	Same as Step 1 above	Same as Step 1 above	$24/600 \times 10^{-6}$
	Total (1 row)			$90/2225 \times 10^{-6}$	
	Total frame (256 × 256)			23040/5.7 ms	
Color frame	Calculate Eq. (9)	Download $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}, \mathbf{f}, \mathbf{g}, \mathbf{h}$, parameters (Eq. (9))	6 shift operations	$6/150 \times 10^{-6}$	
		Compute w_{uv} (Eq. (10))	8 addition, 8 multiply, 8 division	$136/3400 \times 10^{-6}$	
		Compute W	7 additions	$7/175 \times 10^{-6}$	
		Compute X	23 additions, 8 multiply, 1 division	$95/2375 \times 10^{-6}$	
		Total (1 row)			$244/6100 \times 10^{-6}$
	Total frame (256 × 256)			62464/15.61	

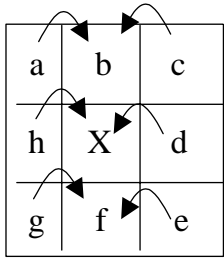


Fig. 8. Shift operations required to compute Eq. (10) for the pixel X.

integrates 32 VLIW processors elements (Fig. 6). The off-chip Program Sequencer module provides the single instruction stream for the processor array. This leads to an overall peak performance of 1.28 Gops, which is, with a proper hardware mapping, quite enough to achieve our algorithm in real-time.

- A Video Interface and Program Sequencer module to store the input video frame into the frame grabber and to coordinate the execution of the program in the Processing Unit in a Single Instruction Multiple Data (SIMD) manner.

- A Program memory for storing the program code and a video frame memory, based on SDRAM technology to store the video frames captured from the video camera. This provides a bandwidth between the onchip and the external memory of 1.28 GB/s which enables the transfer of 644 frames in 33.3 ms.
- A Host PC interface to reconfigure both the processing unit and the video interface (video frame size, video standard, etc.), as well as to initialize the program sequencer.

3.3.1. Black and white images

The hardware algorithm proceeds to store the actual frame into the internal memory of the Processing unit, in such a way that all pixels of each of its column are stored into a same local memory of the corresponding Processing Element (PE). Next, each processing element collects the coefficients of Eq. (2). This can be achieved by shifting the neighboring pixels in the left and right directions using the on-chip inter-PE Reg. As shown in Fig. 7, two shift operations by one and three units are done in the left

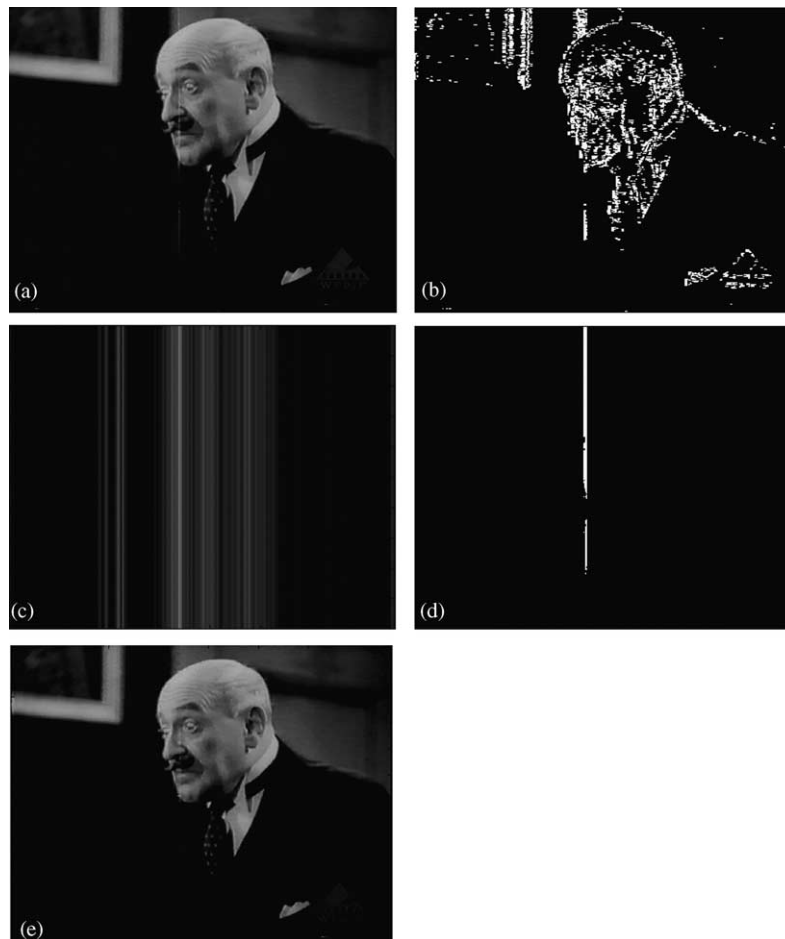


Fig. 9. (a) Original defected image, (b) horizontal segment of length D_x applied to the defected image, (c) columns mean of (b), (d) resulting detection mask, (e) restored image.

and right directions, respectively. This requires 8 clock cycles. This is followed by 16 clock cycles to compute the new value of the pixel X (Eq. (1)). For a frame, this leads to a total execution time of 5.76 m (Table 1), which is much less than the frame speed.

3.3.2. Color images

Similarly to the black and white images, the hardware algorithm proceeds to store the actual color frame into the internal memory of the Processing unit, in such a way that all pixels of each of its column are stored into a same local memory of the corresponding Processing Element (PE). Next, each processing element collects the coefficients of the Eq. (9). This is achieved by shifting the neighboring pixels in the left and right directions using the on-chip inter-PE Reg. As shown in Fig. 8, three shift operations are done in both the left and right directions. This requires 6 clock cycles. For a frame, this leads to a total execution time of 15.61 ms (Table 1), which is much less than the frame speed.

4. Experimental result

Several frames from old video sequences have been employed to assess and to test the performance of our nonlinear operators for the detection and the reconstruction of the defects.

The processed images are presented for visual assessment since in many cases they are the best qualitative measure of performance. Fig. 9(a)–(e) show, the black and white defected frame from the Polish film *Manewry Milosne*, the horizontal segment of length D_x , applied to the defected image, columns mean of the image in Fig. 9(b), and the resulting detection mask, respectively. The proposed vector rational interpolator is reduced to scalar case.

An additional sample processing results are presented in Fig. 10(a)–(c). Fig. 10(a) displays the original frame of the test sequence that is damaged with defects of a different shape and size. Fig. 10(b)–(c) represent the results of the detection in binary form and the reconstruction, respectively.

As for color video sequences, Fig. 11(a) displays the original frame of the test video sequence provided by the *BBC Research and Development Department*, which is damaged, with defects of a different shape and size (see Fig. 11(b)). The sequences consists of 1 in. VTR sequences which have a continuous linear scratch, running along the length of the tape, caused by mechanical damage to a tape guide. However, since the tapes are helically scanned the VTR head sweeps over the scratch periodically as it rotates. The effect of this is to generate a sequence of blips at intervals as the television picture is scanned. Fig. 11(c) shows the result of the detection of both bright and dark scratches in binary form (binary image), and Fig. 11(d)

presents the reconstructed (interpolated) frame by the proposed vector rational operator.

As shown in Figs. 9–11, the quality of the processed images is excellent. Edges and lines are well reconstructed and the texture continuation is smoother. The processing time remains small since the operator is applied only in the lost parts. Moreover, the rational and vector rational interpolators require only few operations. One drawback might prove to be the defects that are located close to the image borders, since they cannot be reconstructed perfectly and remain jagged.

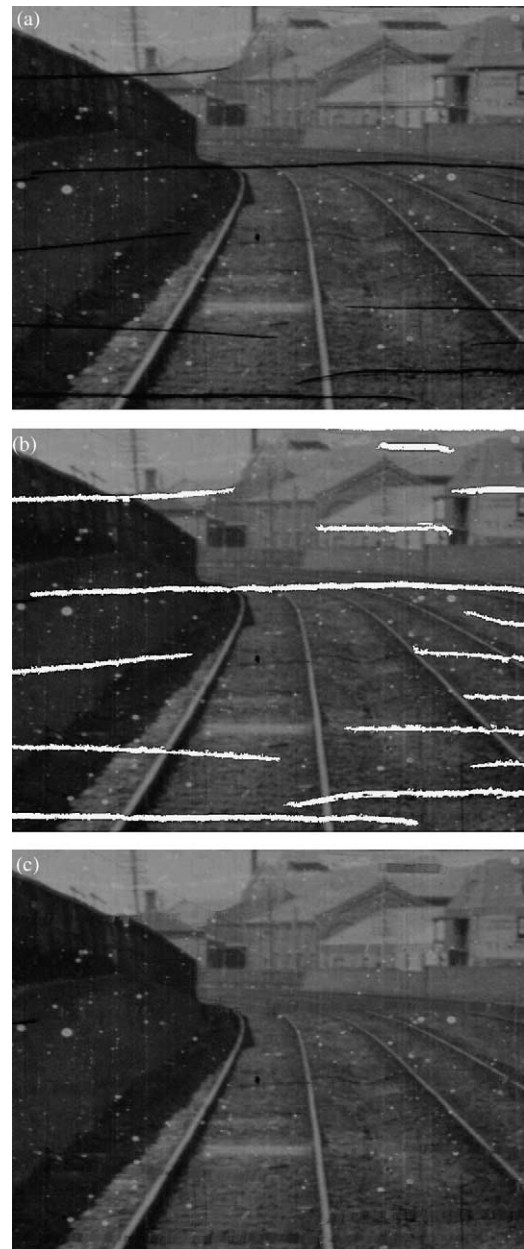


Fig. 10. Original frame of the test sequence (a), result of the detection of crackles and dark random defects (b), restored frame using the proposed rational interpolator (c).

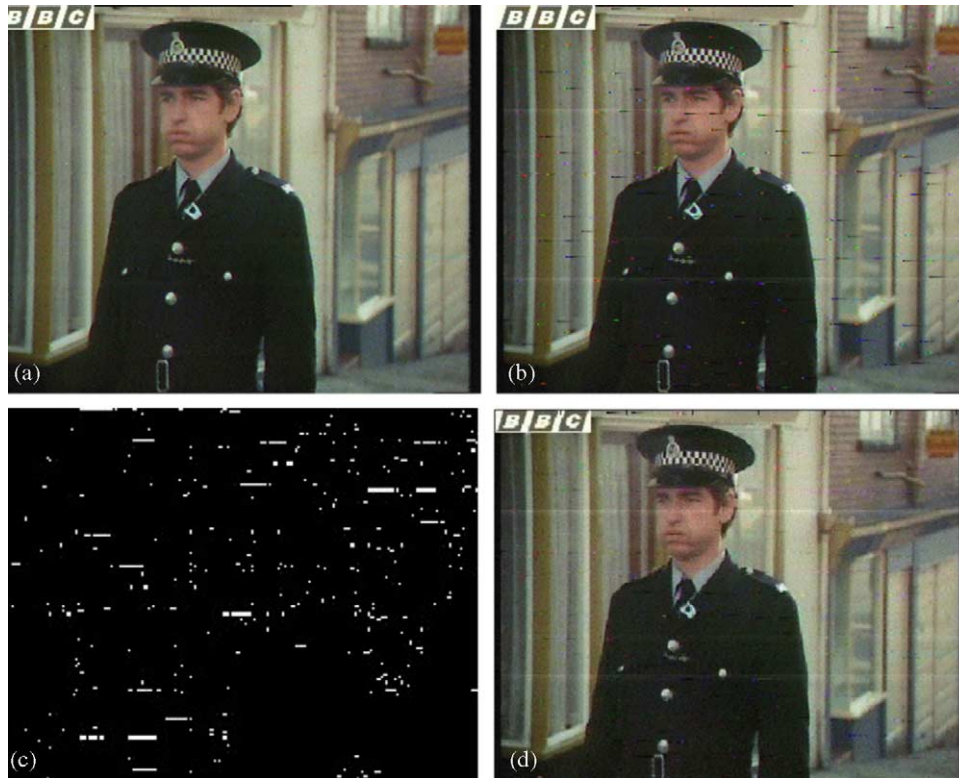


Fig. 11. Original frame of the test sequence (a), corrupted frame (b), defects detection (c), and restored frame using the proposed vector rational interpolator (d).

5. Conclusions

The detection and the reconstruction techniques of defected video sequences (old black and white, and color movies) using a nonlinear operator based on Rational and vector rational filters have been described in this paper. Some processed images are presented in Figs. 9–11 for subjective assessment. Our spatial technique achieves line, edge and smooth texture continuation with high probability. These algorithms do not use any motion estimation, making the present scheme much faster and suitable for real time implementations. The complexity evaluation of the algorithm has shown that the implementation of the algorithm on a dedicated IMAP-based parallel hardware architecture can lead to an execution time of 5.7 and 15.6 ms for (256×256) binary and color images, respectively.

References

- [1] Ballester C, Bertalmio M. Filling in by joint interpolation of vector fields and gray levels. *IEEE Trans Image Process* 2001;10(8).
- [2] Carrato S, Ramponi G, Marsi SA. A simple edge-sensitive image interpolation filter. *Proceedings of third IEEE international conference on image processing, ICIP-96, Lausanne (CH); September 16–19.*
- [3] Decenciere Ferrandiere E. An automatic system for restoring image sequences. *Noblesse workshop on non-linear model based image analysis, Glasgow, UK; July 1998.*
- [4] Gonzales RC, Woods RE. *Digital image processing, 2nd ed.* Englewood Cliffs, NJ: Prentice-Hall; 2002.
- [5] Hamila R, Astola J, Cheikh FA, Gabbouj M, Renfors M. Teager energy and the ambiguity function. *IEEE Trans Signal Process* 1999; 47:260–2.
- [6] Kokaram A, Morris R, Fitzgerald W, Rayner P. Detection of missing data in image sequences. *Image processing*, p. 1496–508 [November].
- [7] Khriji L, Cheikh FA, Gabbouj M. High resolution digital resampling using vector rational filters. *J Opt Eng (Special Issue Sampled Imaging Syst)* 1999;38(5):893–901.
- [8] Khriji L, Gabbouj M. Nonlinear interpolators for old movie restoration. *Proceedings sixth IEEE international conference, on image processing, ICIP-99, Kobe, Japan; October 25–28.*
- [9] Kokaram A. *Motion picture restoration: digital algorithms for artifact suppression in degraded motion picture film and video.* Berlin: Springer; 1998. ISBN 3-540-76040-7.
- [10] Leung H, Haykin S. Detection and estimation using an adaptive rational function filters. *IEEE Trans Signal Process* 1994;42(12): 3365–76.
- [11] Soille P. *Morphological image analysis.* Berlin: Springer; 1999.
- [12] van Roosmalen PMB. *Restoration of archived image and video,* PhD thesis, Delft University of Technology, Delft, The Netherlands; 1999.