

# NONLINEAR INTERPOLATORS FOR OLD MOVIE RESTORATION

*L. Khriji (1), M. Gabbouj (2), S. Marsi (3), G. Ramponi (3), and E. Decenciere Ferrandiere (4)*

(1) Electrical Engineering Department, E.N.I.M., Tunisia. e-mail: lazhar.kheriji@enim.rnu.tn

(2) Signal Processing Laboratory, TUT, P.O. Box 553 FIN-33101 Tampere, Finland

(3) DEEI, University of Trieste, Italy

(4) Centre de Morphologie Mathematique, Ecole des Mines de Paris, France

## ABSTRACT

A nonlinear interpolator using a rational function filter is applied to the restoration of image sequence frames of digitized old movies. Samples to be interpolated are due to stationary and random defects. The interpolator is preceded by a defect localization algorithm. The performance of the proposed interpolator has been assessed on several sequences and compared with a classical morphological operator. The hardware implementation of the proposed rational interpolator is also considered. Simulations show that the interpolated frames with the proposed operator are free from blockiness and jaggedness which are very difficult to avoid when using linear operators.

**Keywords:** Rational Filters, Interpolation, Restoration, Old Movies.

## 1. INTRODUCTION

The past few years have witnessed increasing efforts in discretizing old motion pictures for the different purposes of restoring the defective parts or of performing content-based retrieval. In general, old movies suffer from two types of defects resulting in partial loss of information. These are stationary defects which are fixed and independent of the time variation and random defects, which vary from frame to frame in the image sequence. A number of automatic or semi-automatic systems have been proposed in the literature for the restoration of these defects, see for instance, [1], [2]. In both stationary and random cases, a detection-estimation scheme is adopted in [1]. The detection mechanism allows the localization of a defect while the estimation process restores the believed to be true gray-level values. Averaging techniques combined with morphological operators are used in the detection phase of local stationary defects; whereas, a novel scheme based on the concept of connected operators is used for the detection phase in the case of local random defects. A spatial morphological opening is used to interpolate the missing values in

the case of stationary defects. On the other hand, both spatial and temporal interpolation schemes are proposed for the case of local random defects.

The focus of this paper is on the interpolation phase of the restoration and its hardware implementation. The detection and localization of damaged regions are briefly described. The proposed interpolator is based on rational filters which were recently proposed in [3], [4] and [5]. A rational function (defined as the ratio of two polynomials) is a universal approximator and a good extrapolator. It can be trained using an adaptive algorithm, and requires lower degree terms than Volterra expansions, [3]. Furthermore, the nonlinearity of the operator can reconstruct the defects effectively, without introducing overshoot artifacts in the restored frames.

Although the frames to be restored may be highly non-stationary, 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 (see Fig.1). It is assumed that edges inside defect regions remain stationary, this of course depends on the size of the defect; therefore, the smaller the defect is, the more accurate the interpolation will be.

The paper is organized as follows. Section 2 introduces the proposed interpolation scheme, while Section 3 exploits its hardware implementation. Section 4 contains the simulation results for both the localization and the restoration phases. Section 5 concludes the paper.

## 2. THE PROPOSED RATIONAL INTERPOLATOR

Assuming a defect has been localized (more about this in Section 4), the following adaptive rational interpolator is applied.

Let us denote the 1-D samples on the left of the missing sample  $X$  (to be interpolated) as  $a = (a_l, a_{l-1}, \dots, a_1)$  and those on the right by  $b = (b_1, b_2, \dots, b_m)$ . The window width of the interpolator is given by  $l + m + 1$ .

The missing sample value  $X$  is estimated as the output of a rational function operating on the sample sets  $a$  and  $b$  as follows:

$$X = \frac{P(a_1, a_2, \dots, a_l, b_1, b_2, \dots, b_m)}{Q(a_1, a_2, \dots, a_l, b_1, b_2, \dots, b_m)} \quad (1)$$

where  $P(\cdot)$  and  $Q(\cdot)$  are polynomials, of fixed order, in  $a$  and  $b$ . The shapes of these polynomials govern the behaviour of the interpolator. When ground truth is available, a training procedure, similar to the one in [3] could be invoked to select the optimal parameters according to a given error criterion. This work is in progress. Alternatively, one can assume an *ad-hoc* structure that would be suitable for the application at hand.

In this paper, we follow the latter alternative. The missing sample is estimated here using the following rational function [4], [5]:

$$X = \frac{a_1 + b_1 + k[b_1(b_1 - a_3)^2 + a_1(a_1 - b_3)^2]}{2 + k[(a_1 - b_3)^2 + (b_1 - a_3)^2]} \quad (2)$$

The resulting operator is able to reconstruct sharp edges more accurately than a linear filter because it is able to detect their position with sub-pixel accuracy. Parameter  $k$  is some positive constant and is used to control the amount of the nonlinear effect. Note that when  $k = 0$ , the estimator is a first-order linear interpolator. For larger (positive) values of  $k$ , the quadratic terms in the denominator in the above expression perceive the presence of a detail. The smoothing effect of the operator is thus reduced accordingly.

## 2.1. The reconstruction algorithm

The underlying defects are assumed to be convex. This is true in most cases. The algorithm first detects the best edge orientation in the regions surrounding the defect. Pixels belonging to the regions in that direction (i.e., across the borders of different objects) should be used in the reconstruction of the missing samples (see Fig. 1). Thus, big homogeneous or textured regions will be reconstructed directly avoiding processing delays or computational complexity. On the other hand, the satisfactory reconstruction of smaller homogeneous or textured regions, regions with small details, lines and edges requires a good edge detection. The one-dimensional interpolator of Eq.(2) is adopted over this edge orientation.

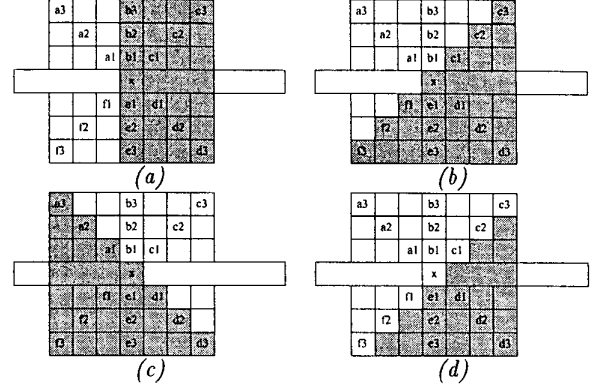


Figure 1: Position of the edge of an object with respect to the interpolation mask. (a) vertical edge, (b) diagonal edge (45 degree), (c) diagonal edge (135 degree), (d) diagonal edge outside the mask center.

If the defect has a width  $dx > 1$ , as depicted in Fig. 2, various steps are required to complete the reconstruction. The known samples are the  $a_i$ 's and the  $b_i$ 's in the white boxes in Fig. 2. Denote by  $R(a_3, a_1, b_1, b_3)$  the intermediate reconstruction values computed using Eq.(2). Then the reconstruction proceeds as follows:

**Step-1:** Compute the missing center sample  $X_0$  using the samples  $(a_4, a_1, b_1, b_4)$ .  $X_0 = R(a_4, a_1, b_1, b_4)$ .

**Step-2:** Compute the following intermediate estimates:

- $X_{-2}$  as a weighted average of  $a_1$  and  $X_0$ :  $X_{-2} = \frac{2a_1 + X_0}{3}$
- $X_2$  as a weighted average of  $b_1$  and  $X_0$ :  $X_2 = \frac{2b_1 + X_0}{3}$

**Step-3:** Compute  $X_1$  by applying Eq.(2) using samples  $X_0, X_{-2}, X_2$  and  $b_2$ :  $X_1 = R(X_0, X_{-2}, X_2, b_2)$ .

**Step-4:** Compute  $X_{-1}$  by applying Eq.(2) using  $a_2, X_{-2}, X_0$  and  $X_2$ :  $X_{-1} = R(a_2, X_{-2}, X_0, X_2)$ .

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

**Step-6:** Finally, recompute  $X_2$  by applying Eq.(2) using  $X_{-1}, X_1, b_1$  and  $b_3$ :  $X_2 = R(X_{-1}, X_1, b_1, b_3)$ .



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

## 3. HARDWARE IMPLEMENTATION OF THE RATIONAL INTERPOLATOR

The hardware implementation of the proposed algorithm is entirely subordinate to the realization of the operator

defined in Eq.(2). In the following we focus our attention only on this part of the structure. Reconsidering this equation it can be noted that the output is evaluated through a weighted sum of two pixels, ( $a_1$  and  $b_1$ ); the two weights  $W_1$  and  $W_2$  are complementary (i.e. their sum is always unitary) and their values are:

$$W_1 = \frac{1 + k(a_1 - b_3)^2}{2 + k[(a_1 - b_3)^2 + (a_3 - b_1)^2]} \quad (3)$$

$$W_2 = \frac{1 + k(a_3 - b_1)^2}{2 + k[(a_1 - b_3)^2 + (a_3 - b_1)^2]} \quad (4)$$

Exploiting this representation a simple but effective hardware implementation has been achieved. Moreover a significant simplification in the realization has been obtained noting that the values of the weights  $W_1$  and  $W_2$  can be coarsely quantized, at the condition that their sum remains strictly unitary. Several tests have demonstrated that a 4-bit fixed point representation for the weights presents results very close to the ones obtained through infinite precision simulations. In such a way the advantage is twofold: the circuit used for the evaluation of the weights can be obviously simplified and moreover the multipliers employed for the evaluation of the final weighted sum can be realized with a slightly reduced complexity.

In the physical realization of the circuit all the various elementary operators like adders, multipliers, dividers, have been developed in a combinatorial form using carry look ahead and Wallace tree architectures [6] and then employed in the realization of the complete circuit. The entire circuit has been implemented in a sequential form with an extensive use of flip-flops. The proposed architecture presents a pipeline with a latency of 20 clock cycles and an initiation interval of 4 cycles. The clock cycle has been kept under  $18ns$  to meet the CCIR 601-2 specifics; in such a way it is possible to use the proposed chip to process in real time the movie sequence after its conversion into a standard TV digital sequence. Actually in the CCIR 601-2 standard the samples come with a cadence of  $75ns$ , thus the entire  $72ns$  initial interval of the pipeline can be consumed before a new input sample appears.

Moreover the use of the pipeline yields a significant reduction in the number of elementary operators used for the realization of the algorithm. Considering that all the various operators have been realized with a delay lower than the clock cycle, it is evident that each elementary operation can be performed in a single clock cycle. Using a memory cell to store the result of each operation, the single operator can then be re-employed during other clock cycles to perform other similar tasks. Taking advantage from this sharing of resources it is possible to have a significant reduction in the number of the used operators. The entire set of the required operations can be executed using only the following six operators:

- one  $8 \times 8$ -bit multiplier: used both in the evaluation of the square power and in the multiplication with the control parameter  $k$ ;
- one 8-bit general purpose adder: used to perform various additions in the algorithm;
- one 8-bit subtracter: used to perform the three differences requested;
- one cell to increment by one the numerator of the ratio; this kind of cell is much simpler with respect to a complete adder;
- one  $4 \times 4$ -bit divider for the evaluation of the final weight;
- one  $4 \times 8$ -bit multiplier: used for the evaluation of the two final weighted sum addends.

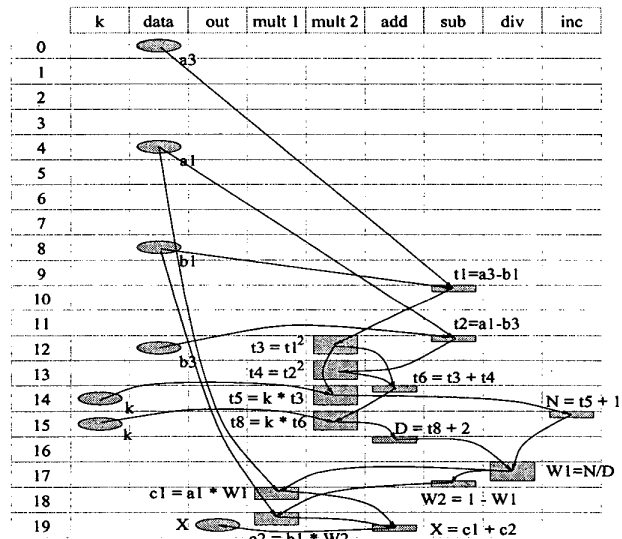


Figure 3: Scheduling of the operator

A scheduling of the operations is represented in Fig.(3). The rows represent the 20 cycles of the pipeline architecture, while the columns show the various operations performed in a single cycle. The four input samples are acquired from the same data bus according to the initiation interval of the pipeline. In such a way the circuit acquires a new sample and produces an output value with the cadence defined in the CCIR 601 standard. For a good versatility of the circuit a separate input has been realized for the control parameter  $k$  that we assume to be always present on this input. After a sample has been acquired it is stored in a memory cell (not represented in the figure) and used by the appropriate operator; the result is stored in another memory cell and so on. It can be noted in Fig.(3) that every operator works for no more than four cycles during the entire latency of 20 cycles.

This characteristic is necessary to be able to keep the circuit ready to process a new input sample after four clock cycles. Actually also the location of the various operations inside the 20-cycle process has been chosen according to the constraint of releasing an operator in due course for the processing of the new data.

The complete scheduling in the physical circuit has been obtained using a finite-state machine with 20 states that provides all the control signals used for the synchronization of the various parts.

The circuit proposed in this paper, realized with a standard cell  $0.6\mu$  CMOS technology, is represented in Fig.(4). The size of the final circuit is  $2.22 \times 2.08 \text{ mm}^2$  with a total area of  $4.62 \text{ mm}^2$  and an aspect ratio of 1.07. The core covers an area of  $2.4 \text{ mm}^2$  and it is composed by 2914 standard cells instances. The simulated power consumption is approximately  $64 \text{ mW}$ .

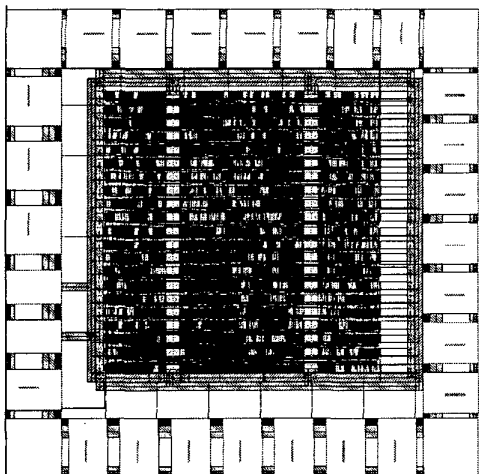


Figure 4: Chip layout

## 4. EXPERIMENTAL RESULT

### 4.1. Defect localization

The defect detection method used in this work is briefly summarised, see details in [1]. We call local random defects those defects that are relatively small (with respect to the image size) and that do not appear at the same position in consecutive frames. In addition they are local maxima or minima. In order to detect light local random defects in a given frame  $I_t$ , we build with the neighbouring frames  $I_{t-1}$  and  $I_{t+1}$  a 3D volume where the third axis corresponds to the temporal dimension. Then we apply a morphological opening by reconstruction using as structuring element a segment of length 1 parallel to the time axis. The use of the reconstruction operator allows to implicitly compensate for displacements that are smaller than the size of the objects. The difference between the original image  $I_t$  and the result

of the opening by reconstruction gives a grey level image  $J_t$  where high grey levels indicate more probable defects. The binarization of this image using a hysteresis threshold gives the defect mask. The procedure to detect dark local defects is the same except that a closing is used instead of an opening.

This method is fast, and gives good results as long as motion in the scene is not too strong. Moreover, it can be adapted to particular local random defects (like for example dark horizontal crackles) by including geometrical criteria in the binarization of image  $J_t$ .

### 4.2. Interpolation results

Several frames from an old movie have been used to test the performance of our nonlinear operator for the reconstruction of defects detected using the method described above (as an example see Fig.5). The interpolated images are presented for visual comparison since in many cases they are the best qualitative measure of performance. Fig. 5(a) displays the original frame of the test sequence which is damaged with defects of a different shape and size. Fig. 5(b) shows the location of the defects. Finally, Fig.5(d) shows the interpolated image by the proposed rational operator. The results obtained using the morphological interpolator are shown in Fig.5(c). Here, a spatial morphological dilation is used to fill the defects. The structuring element depends on the kind of defect: normally it is a square, but in some particular cases it can be different. The method is very fast, but the resulting texture is smooth.

As shown in Fig.5(d), the quality of the processed images is significantly better than that of the original ones. Furthermore, the proposed technique performs slightly better than the one based on morphological interpolation, see Fig.5(c). 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 interpolator requires only few operations. One drawback might prove to be the defects which are located close to the image borders, since they cannot be reconstructed perfectly and remain jagged.

## 5. CONCLUSIONS

Frames from an old movie have been restored using nonlinear operators based on rational functions. The results are compared with a morphological interpolator. Defects have been pre-localised using a simple morphological technique. A hardware implementation of the interpolator has been proposed.

Some processed images are presented for subjective comparison. Our spatial technique achieves line, edge and smooth texture continuation with high probability.

In the case of stationary defects, the performance of the proposed algorithm is better than the one presented in [1] with an increase in computational complexity. Equal performance has been obtained in the case of random defects without resorting to expensive motion estimation. The proposed scheme is hence much faster and suitable for real time implementation.

## 6. REFERENCES

- [1] E. Decenciere Ferrandiere, "Restauration Automatique de Films Anciens," *PhD Thesis*, Ecole Nationale Supérieure des Mines de Paris, Centre de Morphologie Mathématique, Fontainebleau, France, 1997.
- [2] A.G. Setos, "The fox movietone news preservation project: an introduction," *SMPTE Journal*, 105(9): 532-536, 1996.
- [3] H. Leung, S. Haykin, "Detection and Estimation Using an Adaptive Rational Function Filters," *IEEE Trans. on Signal Processing*, vol.42, no. 12, pp. 3365-3376, Dec. 1994.
- [4] S. Carrato, G. Ramponi and S. Marsi, "A Simple Edge-Sensitive Image Interpolation Filter," *Proc. Third IEEE Intern. Conf. on Image Processing, ICIP-96*, Lausanne (CH), Sept. 16-19, 1996.
- [5] L. Khriji, F. Alaya Cheikh and M. Gabbouj, "High Resolution Digital Resampling Using Vector Rational Filters," *Optical Engineering Journal*, Special Issue on Sampled Imaging Systems, to appear (May 1999).
- [6] M.R. Zargham, "Computer Architecture - Single and parallel system" *Prentice-Hall*, ISBN: 0-13-529497-5.

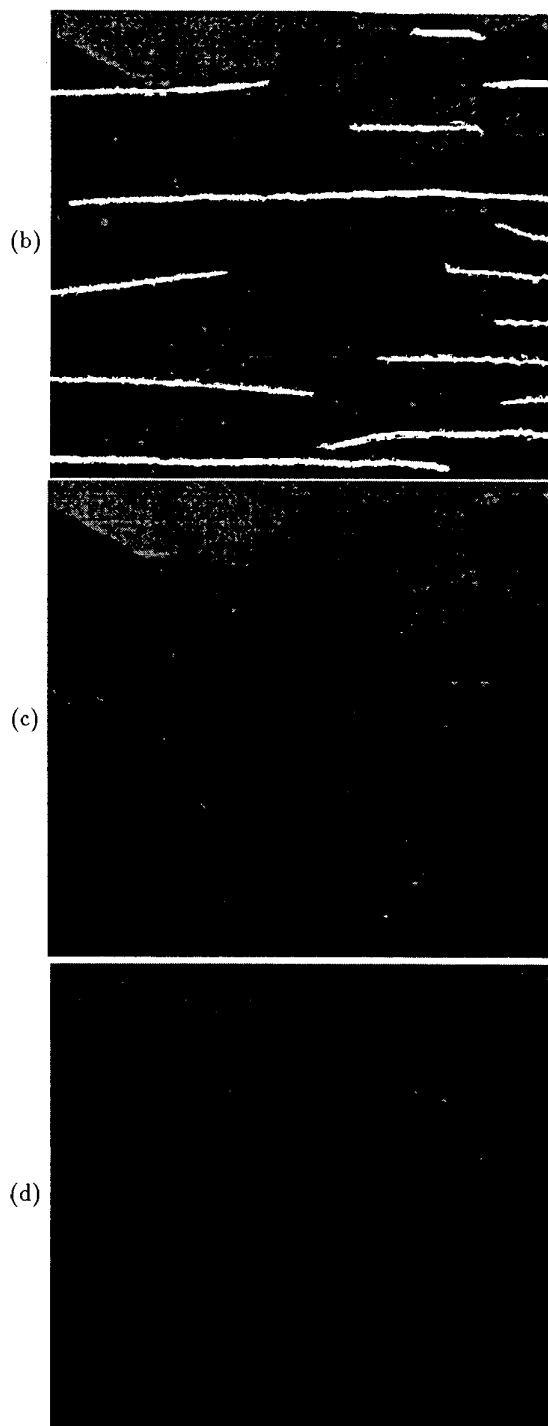
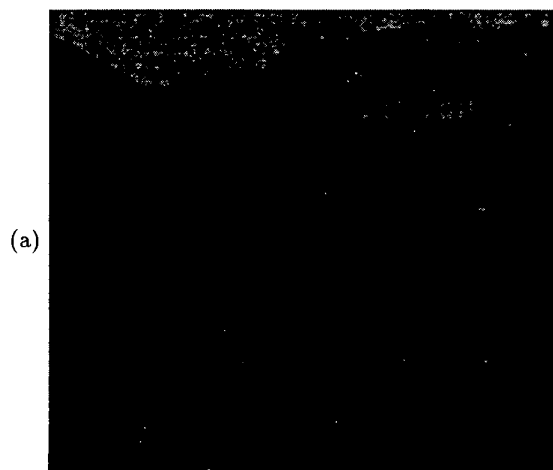


Figure 5: Original frame of the test sequence (a), result of the detection of crackles and dark random defects (b), restored image using the morphological interpolator (c), restored image using the proposed rational interpolator (d).