TEMPORAL EXTENSION TO EXEMPLAR-BASED INPAINTING APPLIED TO SCRATCH CORRECTION IN DAMAGED IMAGE SEQUENCES

Guillaume Forbin, Bernard Besserer, Jiří Boldyš, David Tschumperlé
Laboratoire Informatique, Image, Interaction (L3i)
University of La Rochelle
Av. Michel Crépeau, 17042 La Rochelle cedex 1, France
g.forbin@surrey.ac.uk - bbessere@univ-lr.fr - boldys@utia.cas.cz - David.Tschumperle@greyc.ensicaen.fr

Abstract

This paper presents an adaptation of the exemplar-based inpainting process for image sequences. The aimed application is digital restoration of old movies, and particularly scratch removal, which still remains an unsolved problem. Most of the current correction methods act frame by frame and fail to preserve the temporal coherences of the processed movies. Our new scratch concealment algorithm extends the recently published exemplar-based inpainting method by taking into account, for the block matching step, several images from the sequence. Furthermore, the method is improved by explicitely using motion estimation. A few results demonstrating the effect of this innovative extension are presented and compared to static methods.

1 Introduction

Achievements in the field of movie restoration are presented in this paper. Typically, movies undergo several types of degradations, caused by time and frequent usage or by accidents or other unexpected events occurred during production. Our research on movie restoration is a part of a large European project dealing with preservation and restoration of audiovisual materials[1].

Our contribution to this project mainly concerns scratch concealment. For high-quality restoration of images or movies, a two-step method is classically used : artifacts detection (scratch or dust spots for instance), followed by artifacts removal by using specific interpolation techniques. The first step, i.e. the prior detection step for line scratches, has been published in [1] and gives correct results. The proposed method uses projections of each image of the input sequence as input. First, a 1D-extrema detector provides candidates. Next, a MHT (Multiple Hypothesis Tracker) uses these candidates to create and keep multiple hypothesis. As the tracking goes further through the sequence, each hypothesis gains or looses evidence. To avoid a combinatorial explosion, the hypothesis tree is sequentially pruned, preserving a list of the best ones. As hypothesis are set up at each iteration, even if no information is available, a tracked path might cross gaps (missed detection or speckled scratches). Throughout this paper, we consider that we have corrupted movie data as well as scratch positions delivered by our scratch detector (so-called "mask files", see Fig. 1).

Figure 1. Left, a scratched image from the "lost world" sequence (1925, by courtesy of Lobster Film). Right, the result of our scratch detector superimposed over the picture. The detector outputs a binary image (mask files) showing the scratch location and extents.

However, the scratch removal issue has not been satisfactorily solved yet in the literature. Thus, a new algorithm, belonging to the group of the inpainting techniques, is presented in this paper. It is based on an exemplar-based inpainting technique, which is extended and applied on different corrupted sequences.

1.1 Review of existing scratch correction algorithms

Scratches are very current defects in damaged film archives, characterized as bright or dark vertical lines (or colored in case of color films), spread either over several frames or being restricted in size. The temporal aspect of this degradation implies that a well performed restoration needs to be invisible in all individual frames, but also when the entire sequence is played. One major problem of the al-

[1] FP6 IST project PRESTOSPACE : www.prestospace.org

ready proposed correction methods lies in the fact that they are not able to reproduce the original grain of the movie when interpolating scratches, resulting in non-natural looking restored frames. This is the case of the methods based on low-pass or median filtering [2, 3], for instance.

Kokaram [4] used the assumption that the images follow a 2D AR model and then the sequence is generated with a 3D AR model. Despite interesting results the algorithm is, by design, not able to deal with anisotropic textures or single isophotes.

Partial success has been achieved for scratch removal in the particular case where the original texture is preserved to a certain extent. The scratches appear mostly due to abrasion by mechanical parts of the playing / recording devices and it does not always damage all the film layers. The method described in [5] uses Fourier series expansions to separate low and high frequencies of the frame data. The low frequencies (basic image structure) are reconstructed by means of a polynomial interpolation. Afterwards, a specific bayesian approach using the residual information is applied to restore the high frequencies. This algorithm performs very well and keeps the temporal coherence of the restored frames, even if the sequence is shown.

Unfortunately, the restoration is not satisfactory when texture information is entirely lost. More specific techniques belonging to the inpainting methods have to be applied. These methods allow to sequentially fill the areas of missing textures in a way that copies (or at less imitates) textures from the known areas.

1.2 Review of existing inpainting algorithms

One of the main approaches tries to prolong image contours (isophotes) from the vicinity of the missing textures [6]. It is related to the diffusion process in physics and PDE's (partial differential equations) are typically used. The results of these methods are convincing when removing small defects. However, diffusion PDE's cannot reconstruct textures and some blur appears on scratched and textured regions. This is particularly noticeable when dealing with old grainy films.

In [7], isophotes prolongations with PDE's and texture synthesis are combined. The image is decomposed in two sub-images, containing separately texture information and main image structure. Reconstructing separately these two images and mixing them allows to perform a textured inpainting. Nevertheless, textures in the vicinity of the isophotes still remain blurred, and macro-textures cannot be reconstructed by such a method.

Advantages of the previous approaches are combined in [8, 9]. The missing area is filled in by an isophote-driven image sampling process. The observation and the assumption have been made that exemplar-based texture synthesis is sufficient for propagating linear isophotes within the missing data area. Very interesting results are obtained by this recent inpainting algorithm, although, so far, this algorithm has only been applied to still images. In this paper, we propose to extend it to image sequences. We investigate two ways to perform this temporal extension : the straightforward frame-by-frame application and using the estimation of motion between successive frames.

In the next section, we describe the original exemplar-based inpainting algorithm in detail and we apply it for our problem of scratch removal. Then, different temporal extensions are discussed and applied and some results are presented[2].

2 Exemplar-based inpainting applied to scratch removal

2.1 Presentation of the algorithm

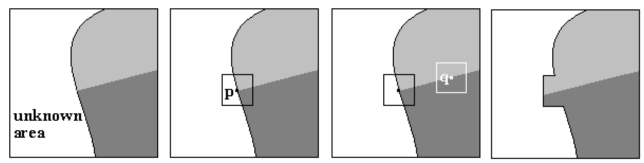

Figure 2. The exemplar-based inpainting process.

Briefly, the algorithm developed in [9] starts the filling process at the location where the isophotes hit the boundary of the unknown area. Thus, these isophotes are prolongated first : each pixel of the boundary is characterized by a recovering priority value. The neighborhood (pattern) of the pixel with the maximum priority is then compared to all patterns of the known image area using a block-matching technique, in order to find the most similar pattern (in the sense of the L_2-norm). Afterwards, the unknown pixels of the first pattern are replaced with the corresponding pixels from the second pattern. This process is repeated until all the unknown region is filled (Fig. 2).

In this example the maximum priority is given to **p** and the first pattern centered on this pixel is determinated (black square). Only the known pixels of this pattern (gray part in Fig. 2) are considered for the block-matching step, which compares the first pattern to all the patterns entirely included in the known area of the image. The most similar pattern is found (white square, centered on **q**) and the unknown pixels of the first pattern (white area of the black square) are attributed the corresponding part of the second pattern values. Fig. 3 shows a result for such a single image (intra-frame) processing.

2.2 Modification of the exemplar-based algorithm

In order to make the algorithm more specific to our purpose of scratch removal, we propose some modifications.

[2] Images used for illustrating our work by courtesy of Centrimage (www.centrimage.com) and HS-ART (www.hs-art.com)

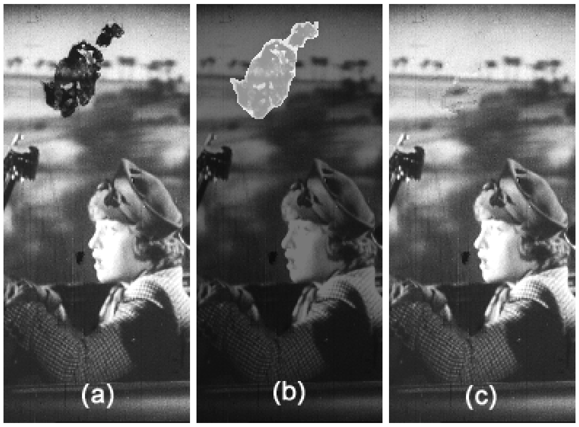

Figure 3. Results of the algorithm in a single image. **(a)** shows an image with a large dirt blotch, **(b)** the mask which confines the unknown pixels and **(c)** the result.

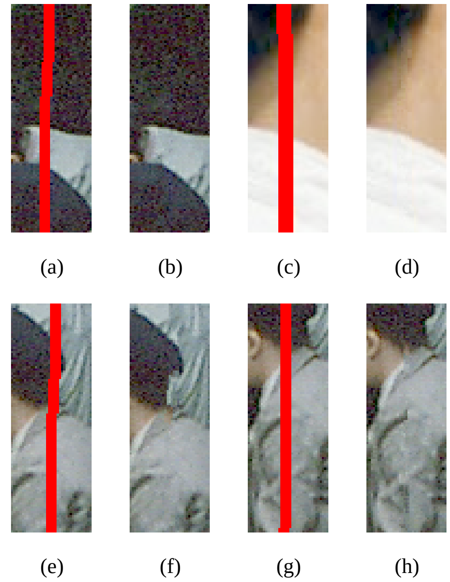

Figure 4. Scratch removal using exemplar-based inpainting.

The algorithm presented in [9] can be used to fill an area of any shape. Thus, the computational load is very high, since the priority values for every pixel of the target area boundary are re-evaluated at each iteration. This is not necessary in our case : scratches often have the same geometric characteristics and the distance between points from opposite sides of the corrupted area is small, so a concentric filling of the unknown region is sufficient. All the priority values of the boundary pixels are thus calculated only once, speeding up the entire process.

We also propose another modification, related to the way the pixels are copied into the target area. In [9] the filling is made by blocks, resulting in the apparition of block artifacts (similar to JPEG compression). This can be avoided by copying only the central pixel of the estimated neighborhood after the pattern search.

The results presented in Figs. 4(b) and 4(d) are convincing, the algorithm performs well if the problem is to extend linear structure. Unfortunately, the results are less interesting when the isophotes exhibit some curves and discontinuities (presence on an extremity on one side of the missing data area, but not on the other, see Figs. 4(f) and 4(h)). Indeed, the filling process in done by copying data from the image into the missing area. If there are no correct blocks that match the region, many visible artifacts can appear. Favorably, in our case, a sequence of images is available. In the next section, we propose to use the temporal information in order to overcome this problem.

3 Temporal extension

3.1 Hypothesis concerning the scratched area

A natural way to fill in the scratched area is to find correct blocks on the neighbor images when valid blocks are unavailable in the current frame. Indeed, a scratched object in a frame may be unscratched in another frame. This particularly happens when **the scratch is moving and the scene is motionless** or when **the scene is moving and the scratch is motionless**. Localization of the scratched objects in the neighbor frames should be thus the next step.

3.2 Inpainting algorithm based on motion estimation

Let us suppose that an image at the time *t* contains a scratch that we would like to restore, using pixels coming from the *reference images t-1* and *t+1*. For these images, we obtain a mask describing the position of the scratches using [1].

The first step consists in estimating the motion between *t* and *t-1* (backward motion), and between *t* and *t+1* (forward motion). Afterwards the scratched pixel **p** with the maximum priority on the image *t* is found. Here are the possible cases and our proposed solutions :

1. Pixel **p** has an unscratched correspondent pixel either in the image *t-1* or *t+1*. → The same block-matching as in [9] is performed on *t-1* or *t+1*, but on a restricted area (the search is not exhaustive) centered on the corresponding pixel. This pattern search is performed to deal with the inaccuracy due to the motion estimation.

2. **p** has an unscratched correspondent pixel in both (previous, next) frames. → The search for the best block is fulfilled on the two neighbor images, and the chosen block is simply the most similar to the pattern centered in **p**.

3. **p** has no unscratched correspondent pixel. → The search is performed on the current frame, like in the original algorithm.

(a) (b) (c) (d) (e)

Figure 5. Interpolation of the scratch using neighbor images. **(a)** the scratch. **(b)** filling using the images t-1 and t+1, **(c)** images from t-2 to t+2 and **(d)** from t-3 to t+3. **(e)** the remaining pixels without inter-frame correspondents are filled using the intra-frame algorithm.

In the next example, only two neighbor images have been used. In general, the motion between two frames is rather small, thus only a few pixels have an unscratched correspondent pixel (see Fig. 5(b)). To overcome this problem a number *n* of reference frames is chosen and the motion between these reference frames and the current frame is estimated. The bigger the number of the reference frames, the more scratched pixels have a correspondent pixel and the solution 1 and 2 can be applied. **Our algorithm performs really well if a major part of the edges within the scratch are reconstructed using these solutions**. Conversely, a high number of reference frames can also be a source of errors because of the motion. At the moment, n is chosen manually. Here is an algorithm summarizing what has been explained :

```
while unknown pixels exist do
  compute the priority values of the boundary pixels
  for all these pixels sorted by priorities do
    for i = 1 to n/2 do
      Find a correspondent pix. in the image t − i or/and t + i
      if a correspondent pix. exists (unscratched) then
        use the solution 1 or 2
      end if
    end for
    if the pix. isn't corrected yet then
      use the solution 3
    end if
  end for
end while
```

Fig. 5 illustrates the filling process. As can be seen the correction is really convincing. In addition, the restoration is invisible when the sequences are played thanks to the temporal extension, as can be seen in the Figs. 6 and 7. In the first example, the scene is moving and the scratch is motionless. The other example presents the opposite case.

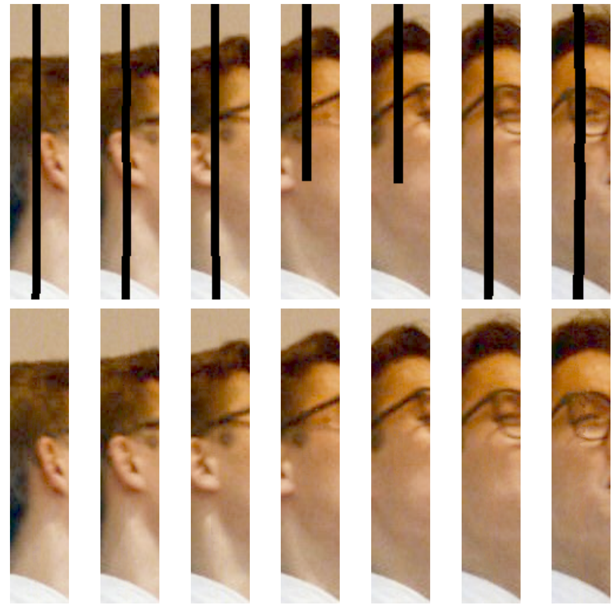

Figure 6. Correction of a motionless scratch in a moving scene.

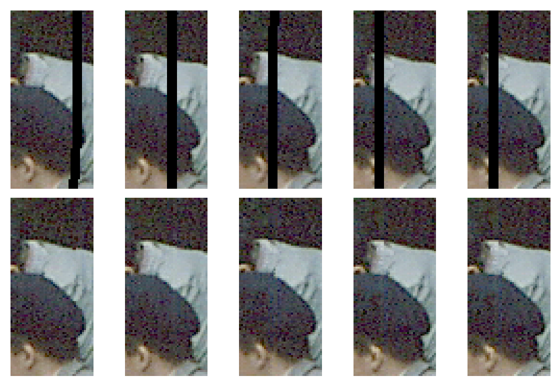

Figure 7. Correction of a moving scratch in a motionless scene.

4 Conclusion

In this paper, an exemplar-based inpainting method has been applied to correct scratches in damaged movie archives. First, the method has been applied straightforwardly. Some defects in the restoration were extremely visible in some cases. Thus, the algorithm has been adapted to deal with and to take the advantage of images sequences, through a temporal extension of the previous algorithm. At the moment, results are convincing when the scene or the scratch are moving and when the scratched objects in a frame are unscratched in at least one neighbor frames. We are currently working on the other case, i.e. when the scene and the scratches are motionless within the entire shot.

Acknowledgements J. Boldys was partially supported by the Czech Ministry of Education under the project No. 1M6798555601 (Research Center DAR).

References

[1] B. Besserer and C. Thiré, "Detection and tracking scheme for line scratch removal in an image sequence," in *Proc. 8th European Conference on Computer Vision (ECCV)*, 2004, vol. III, pp. 264–275.

[2] S. Geman, D.E. McClure, and D. Geman, "A nonlinear filter for film restoration and other problems in image restoration," in *Graphical Models and Image Processing*, 1992, pp. 281–299.

[3] A.C. Kokaram, *Motion Picture Restoration*, Springer-Verlag, 1998.

[4] A.C. Kokaram, "Detection and removal of line scratches in degraded motion picture sequences," in *Proc. of EUSIPCO'96*, Sept. 1996.

[5] L. Joyeux, S. Boukir, and B. Besserer, "Tracking and map reconstruction of line scratches in degraded motion pictures," *Machine Vision and Applications*, vol. Volume 13, Number 3, pp. 119–128, 2002.

[6] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Siggraph 2000, Computer Graphics Proceedings*, Kurt Akeley, Ed. 2000, pp. 417–424, ACM Press / ACM SIGGRAPH / Addison Wesley Longman.

[7] M. Bertalmio, L.A. Vese, G. Sapiro, and S.J. Osher, "Simultaneous structure and texture image inpainting," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2003, vol. II, pp. 707–712.

[8] R. Bornard, E. Lecan, L. Laborelli, and J.H. Chenot, "Missing data correction in still images and image sequences," in *Proc. 10th ACM International Conference on Multimedia*, 2002, pp. 355–361.

[9] A. Criminisi, P. Pérez, and K. Toyama, "Object removal by exemplar-based inpainting," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2003, vol. II, pp. 721–728.