HIGH QUALITY DEINTERLACING USING INPAINTING AND SHUTTER-MODEL DIRECTED TEMPORAL INTERPOLATION

David Tschumperlé, Bernard Besserer
david_tschumperle@yahoo.fr, bernard.besserer@univ-lr.fr
L3I, University of La Rochelle, Av. Michel Crépeau, 17000 LA ROCHELLE, France.

Abstract The proposed deinterlacing scheme is aimed towards high-quality progressive image output, typically for a videotape-to-film blow-up operation. This operation consists in increasing the resolution of video images, recording these images on 16mm or 35mm film and reducing the typical video artefacts (due to interlaced scanning, electronic noise), eventually adding some film-look features (color correction, motion blur,...). This paper focuses on the *deinterlacing part*. Since video real-time is not relevant for digital blow-up (film recorders are slow), the proposed approach gives more weight to *quality* for both *spatial and temporal characteristics* of the image. Each field from the interlaced source is extended to a full frame by using an accurate in-painting process, outstretching edges and preserving their intensity. Motion estimation is then performed from these two inpainted images, and the computed forward and backward motion vector fields are finally used for temporal integration. During this process, the inter-field motion is converted to a motion blur appearance, similar to the one physically produced by the prolonged exposure time in film cameras. Finally, the resulting image is upsized and enhanced by a warp-filter, which unlike high pass filters does not augment noise.

Keywords: Image deinterlacing, Scan-rate conversion, Optical flow, Inpainting.

Introduction

Interlaced scanning was introduced at the early ages of TV to suit the available CRT display technology and reduce the transmitted bandwidth. The method involves splitting each TV frame into two parts, called 'fields'. Each field contains half of the scan lines of a frame (1^{st} field : *odd lines*, 2^{nd} field : *even lines*). Since the display side (TV set) uses this interlaced scheme, so do the capture side : older tube cameras and CCD sensors mostly work interlaced ; this technical heritage pass the years until a complete digital and progressive chain will be build up in a near future. Yet even modern digital videotape formats use an interlaced scheme, and such video formats are now widely used for shooting. But the huge majority of movie theaters are equipped with film projectors as well as they are in festivals (some 16mm, mostly 35mm), making interlaced to progressive conversion still useful.
When using interlaced cameras, an object, or even the complete scene has often moved between the two fields being captured, resulting in a *'blurred'* image involving a double exposure effect, also called *comb effect* (or *herringbone pattern*) on vertical edges. These flaws are easily seen if

the playback is paused ; while playing the sequence, other artefacts are also rising such as interline twitter and field aliasing.
Scan rate conversion (progressive to interlaced such as 3-2 pulldown and interlaced to progressive) are well explored topics both on scientific and technical aspects [3]. Since real-time is essential for a lot of applications, such as DVD playback on computer display, some very simple techniques are used, such as the classical BOB and WEAVE methods, reinterpolation or mixing odd and even field to a frame, which is accurate for motionless sequences only. More clever approaches laid on signal processing theory (analysis in the Fourier space), other methods use T-shaped spatio-temporal filters and some approaches use motion estimation [4, 5, 9, 8], usually for switching between different deinterlacing methods according to the amount of motion detected in the sequence.
In this paper, we focus on *high quality deinterlacing*. The typical application is digital tape-to-film blow-up. The input data is an *interlaced image sequence* typically shoot by an digital camcorder in SDTV resolution (720×576 pixels for PAL, 25 fps, interlaced). All the post-production is usually done in this format, which is easy to handle with modern technology. However, if the final result has to be shown in film theatres, the post-production company has to resize the pictures from SDTV (Standard TV) format to HDTV (High Definition TV, 1920×1080 pixels, progressive) or 2K resolution (2048×1536 pixels) and delivers them to a film recorder. Film is inherently progressive, and usually offers high resolution. Using low-quality deinterlacing method such as throwing a field away induces an upsizing of nearby factor 5 in vertical direction (288 to 1536 lines). Even with good interpolation methods (Bicubic, Splines), the artefacts are magnified and edges became unsharp.
Our method, as others recently proposed in the literature, needs spatial and temporal interpolation. But as we are not dependent of real-time constraints, we base each of these sub-steps on high-quality inpainting and interpolation algorithms, coming from the state of the art of the computer vision and image processing literature. Moreover, we propose an innovative reconstruction step simulating a real camera acquisition process, leading to physically coherent results.

1. Description of the deinterlacing scheme

(a) General idea.
Let us consider an interlaced color image $I_{inter} : [w, h] \rightarrow [0, 255]^3$. The general idea of our algorithm consists in the estimation of an accurate motion between the spatially interpolated odd and even fields I_{odd} and $I_{even} : [w, h/2] \rightarrow [0, 255]^3$ of an interlaced image. Then one uses this information to integrate this motion over time, according to a model related to a rotating camera shutter. This yields the process physically plausible, and results in a high quality deinterlaced image, based on all the pixel informations of the initial interlaced image. This process is repeated for each interlaced image of the input sequence.
As the two consecutive fields are spatially unleveled, we must extend both of them to full frames first, using an interpolation process allowing to double their lines. Then, the motion estimation is performed and the estimated motion vector field is used to build the output image by a temporal integration. This deinterlaced output is eventually upsized with an Image-Dependent Warping filter [1], enhancing the edges without raising video-typical electronic noise.

These successive steps are detailled in the followings (Fig.1).

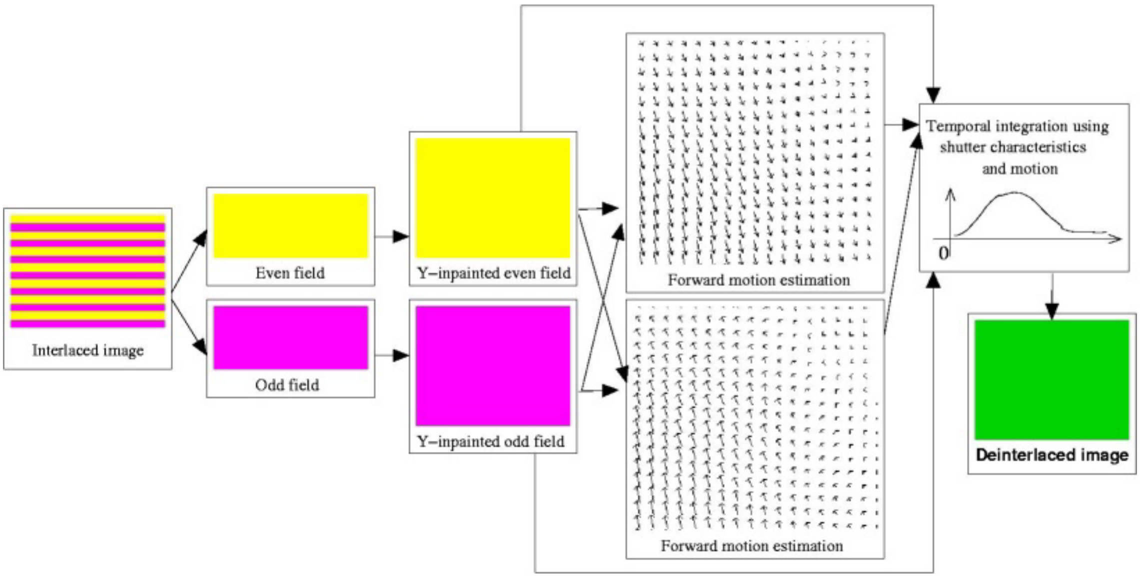


Figure 1. General flow of our deinterlacing algorithm.

(b) Image interpolation using inpainting techniques.
Doubling the lines of the even and odd fields I_{even} and I_{odd} needs spatial interpolation. Many schemes are possible and have been already handled in the literature, such as bloc, linear, bicubic or B-splines interpolation. Unfortunately, these methods do not take care of the image discontinuities and often produce blur artefacts, by smoothing the edges. We propose here a very precise interpolation scheme that respects image discontinuities. It is based on a technique named *Image Inpainting*, which has been recently published in the literature. Basically, it consists in filling-in holes in images by interpolating neighborhood pixels in a non-linear way, in order to reconstruct the isophotes inside the holes. Usually, these kind of methods are based on PDE and variational tools [2, 10], block matching or tensor voting [7], and are really time consuming algorithms. Note that using inpainting methods can also be used in the context of video restoration in order to remove scratches from old films (this is a part of the PrestoSpace european project where we are involved in). In our case, the images are more simple to handle, since the ‘holes’ we want to repaint are 1-pixel-wide lines regularly sampled along the Y-axis. We propose the following scheme, which is a simplification of our previous work on image inpainting [10]. We applied it separatly on the even and odd images I_{even} and I_{odd}.

Let us denote by $U : [w, h] \rightarrow [0, 255]^3$ the color image we want to resize ($U = I_{even}$ or I_{odd}). We first compute the *smoothed structure tensor field* of U, denoted by $G_\sigma : [w, h] \rightarrow \mathcal{P}(2)$,

$$G_\sigma = G * \text{Gauss}(\sigma) \quad \text{where} \quad \forall x, y \in [w, h], \quad G(x, y) = \sum_{i=1}^{3} \nabla U_i \, \nabla U_i^T$$

Gauss(σ) is a 2D gaussian kernel with a variance σ. Note that each point of the field G_σ is a *structure tensor,* which is a symmetric and demi-positive definite 2×2 matrix, whose spectral values give informations on the local geometry of the color image U (more informations about the use of structure tensors to define the image geometry can be found in [10, 11]). Then, we construct the doubled image $U' : [w, 2h] \rightarrow [0, 255]^3$ by the following way :

$$\forall x, y \in [w, h], \quad \begin{cases} U'(x, 2y) & = & U(x, y) & \\ U'(x, 2y+1) & = & \frac{1}{2}\left(U(x + \frac{\Theta_y}{\Theta_x}, y) + U(x - \frac{\Theta_y}{\Theta_x}, y+1)\right) & (\text{if } \Theta_x \neq 0) \\ U'(x, 2y+1) & = & \frac{1}{2}\left(U(x, y) + U(x, y+1)\right) & (\text{if } \Theta_x = 0) \end{cases}$$

where $\Theta = (\Theta_x, \Theta_y)^T$ is the principal eigenvector of $G_\sigma(x, y)$.
The idea behind this filling equation is to average each set of two lines *in the direction of the estimated color isophotes* $\Theta(x, y)^\perp$ for each point of the interpolated line. Instead of classical inpainting techniques, our method is quite fast to compute (needs only one iteration per pixel) and accurate as well. The figure below illustrates the differences between classical interpolations and our proposed inpainting-based scheme.

(a) Original interlaced image

(b) Bloc interpolation of the odd field

(c) Linear interpolation

(d) Our inpainting method

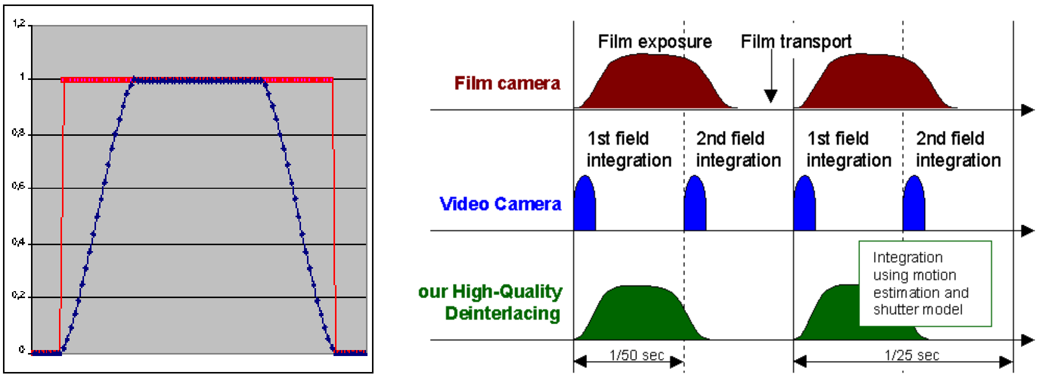
Figure 2. Comparisons of interpolation algorithms to double the y-axis of half-fields.

Note how the aliasing effects disapperar with our method (visible on the roof), thanks to *the respect of the image isophotes* at each point of the image. In the followings, we denote by I'_{even} and I'_{odd} the inpaint-resized versions of the original even and odd frames.

(c) Motion estimation and temporal integration.

The preceding part allows to compute a very accurate spatial interpolation of the input data. But we need *temporal interpolation* as well. Indeed, a deinterlaced frame contains pixel informations from two images having two different positions t and $t + 1$ in the time axis. In order to reconstruct the deinterlaced image, one must understand precisely *what motion has been performed* between

these two time positions. In our case, motion estimation is performed with the Horn & Schunk optical flow algorithm [6], in order to estimate the *forward motion* $m_f : [w, h] \rightarrow \mathbb{R}^2$ (i.e the motion from I'_{odd} and I'_{even}) as well as the *backward motion* $m_b : [w, h] \rightarrow \mathbb{R}^2$ (from I'_{even} to I'_{odd}). Note that m_f and m_b are not exactly opposed vector fields, since the motion estimation may be perturbed by occlusions or noise. Computing these two motion fields allows us to have precise estimations *of the local pixel motion* during the two time intervals $[t, t + 0.5]$ and $[t + 0.5, t + 1]$, between the two consecutive inpainted images I'_{odd} and I'_{even}.
For the majority of video cameras, the image is field-integrated by the CCD array (see Fig.3(b)). But for a film camera, exposure (corresponding to integration) is full-frame. Generally, a rotating shutter uncovers the sensitive film area during exposure time and covers the film during transport. Assuming a pinhole model for the camera and a rotating shutter close to the focal point, then the exposure function over time can be seen as the red line (Fig.3(a)). But if the aperture is greater or if the shutter is distant to the focal point, light throughput grows as the shutter uncovers the light path (and respectively decrease). A simple model (area of a circular segment (a portion of a disk whose upper boundary is an arc and whose lower boundary is a chord making a central angle) while displacing the chord) for the covering/uncovering leads to the curve dotted in blue (Fig.3)



(a) Shutter integration function. (b) Camera acquisition process.

Figure 3. Shutter principle for video acquisition.

According to the hypothesis that the film camera speed is also 25 fps (and not 24fps ; temporal resampling will be examined in a future work), our deinterlacing scheme uses two successive fields I'_{odd}, I'_{even} and the computed motion vector fields m_f,m_b to integrate the output image according to the shutter model :

$$\forall x, y \in [w, h], \quad I_{desinterlaced}(x, y) = \int_{t=0}^{\frac{1}{2}} S(t)\; I'_{odd}(x - tm_f(x, y)_x, y - tm_f(x, y)_y)\; dt$$

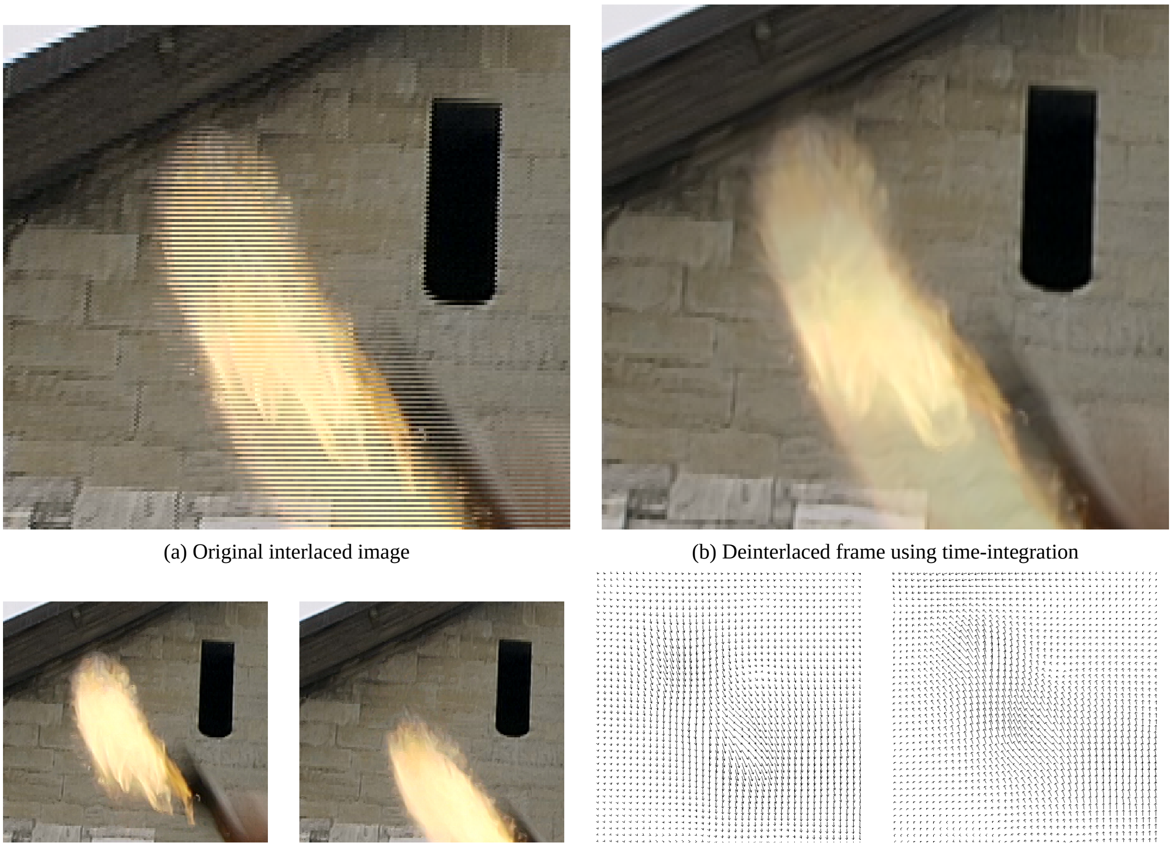
$$+ \int_{t=\frac{1}{2}}^{1} S(1 - t)\; I'_{even}(x - tm_b(x, y)_x, y - tm_b(x, y)_y)\; dt$$

where $S(t)$ represents the shutter characteristic function, and $m_f(x, y) = (m_f(x, y)_x, m_f(x, y)_y)^T$, the displacement vector of the forward motion at the point (x, y).

This temporal integration *physically simulates* the acquisition process that would be done with a progressive camera. In our experiments, we choose the function $S(t) = \operatorname{acos}(\frac{R-t}{R})R^2 - \sqrt{2Rt - t^2}\,(R-t)$, where R represents the radius of the supposed camera apperture.

2. Experimental results

Fig.4 and 5 illustrate some deinterlacing results obtained with our algorithm on a large motion sequence, as well as the benefit of our temporal integration.



(a) Original interlaced image
(b) Deinterlaced frame using time-integration
(c) Inpainted odd frame
(d) Inpainted even frame
(e) Estimated forward motion
(f) Estimated backward motion

Figure 4. Deinterlacing an image with a high motion.

Conclusion

Using high quality algorithms for spatial and temporal interpolation steps in our deinterlacing scheme is one of the key point of our algorithm. The second one, which is simulating a part of the physical process of the camera acquisition, completes the deinterlacting process and leads to high quality deinterlaced images that look spatially and temporaly coherent. Sequence re-timing (from 25 to 24fps) is the next step to get a full conversion process from interlaced video to cinema format. This is an outgoing work.

(a) Original interlaced image

(b) Averaging odd and even frame (linearly interpolated), without motion estimation.

(c) Using our deinterlacing method.

Figure 5. Inter-frames integration to create a deinterlaced image.

References

[1] N. Arad and C. Gotsman. *Enhancement by Image-Dependent Warping.* In IEEE Transactions on Image Processing, Vol. 8, No. 8, August 1999.

[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. *Image inpainting.* In Kurt Akeley, editor, Proceedings of the SIGGRAPH, pages 417–424. ACM Press, Addison Wesley Longman, 2000.

[3] E.B. Bellers, G. De Haan. *De-interlacing: A Key Technology for Scan Rate Conversion*. 01 September, 2000 ISBN: 0444505946.

[4] J. Deame. *Motion Compensated Deinterlacing: The Key to the Digital Video Transition*. SMPTE 141st Technical Conference in NY, November 19-22, 1999.

[5] D. Hargreaves and J. Vaisey. *Bayesian motion estimation and interpolation in interlaced video séquences.* IEEE transactions on image processing, vol.6, nř5, 765-769, 1997.

[6] B.K Horn and B.G Schunk. *Determining optical flow.* in Artficial Intelligence, Vol 17. pp 185-204, 1999.

[7] J. Jia and C.K. Tang. *Image Repairing: Robust Image Synthesis by Adaptive ND Tensor Voting*. in IEEE Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin (United States), June 2003.

[8] S.F. Lin, Y.L. Chang, and L.G. Chang. *Motion adaptive interpolation with morphological operation and 3:2 pulldowned recovery for deinterlacing.* In Proc. of 2002 IEEE International Conference on Multimedia and Expo (ICME 2002), Lausanne, Switzerland, August 2002.

[9] K. Sugiyama and N. Neubert, *A picture format converter using motion compensated deinterlacing*. In Smpte Journal,vol.19, nř11,859-865, 2000.

[10] D. Tschumperlé and R. Deriche. *Vector-valued image regularization with PDE's : A common framework for different applications.* In IEEE Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin (United States), June 2003.

[11] S. Di Zenzo. A note on the gradient of a multi-image. Computer Vision, Graphics, and Image Processing, 33:116–125, 1986.