Deinterlacing with Motion-Compensated Anisotropic Diffusion

Matthias Ghodstinat, Andrés Bruhn, and Joachim Weickert

Mathematical Image Analysis Group
Faculty of Mathematics and Computer Science, Building E1.1
Saarland University, 66041 Saarbrücken, Germany
{ghodstinat,bruhn,weickert}@mia.uni-saarland.de

Abstract. We present a novel deinterlacing scheme that makes consequent use of discontinuity-preserving partial differential equations (PDEs). It combines the accuracy of recent variational motion estimation techniques with the directional interpolation qualities of anisotropic diffusion filters. Our algorithm proceeds in three steps: First, we interpolate the interlaced images by means of a spatial edge enhancing diffusion process (EED). Then we apply the variational optic flow technique of Brox et al. (2004) in order to obtain a precise interframe registration. Finally we use a spatiotemporal generalisation of EED for motion-compensated inpainting of the missing data in the original sequence. Experiments demonstrate that the proposed method outperforms not only classical deinterlacing schemes, but also a recent PDE-based approach.

1 Introduction

Since the beginning of television more than 70 years ago, interlacing is the predominant sampling technique for recording and transmitting video. Although it was originally developed to increase the frame rate of television sets based on cathode ray tubes, it forms nowadays the basis for all analogue broadcast systems (PAL, SECAM, NTSC). Interlacing is based on a simple idea: Instead of considering the complete image domain of a video sequence, alternatingly only the even-indexed and the odd-indexed lines of an image are stored, respectively. While such a proceeding reduces the vertical resolution, it allows to double the effective frame rate at the same time. This in turn improves the temporal smoothness of the video and thus prevents from large area flickering due to fast motion. Although interlacing proved to be a good compromise between quality and bandwidth consumption, it became obsolete in the era of high definition television (HDTV). Today's digital devices such as plasma screens or LCD panels even require scanline converters to display video streams that are originally interlaced. Thus the problem arises how interlaced video data can be converted to progressive image content, i.e. to video sequences that offer the full vertical resolution. This task that requires the filling-in of missing information at even and odd lines of subsequent video frames is referred to as *deinterlacing* [3, 31, 32]. It is closely related to *image* or *video inpainting* [2, 10, 11] where missing information at arbitrary locations has to be restored and *video superresolution* [13, 21, 31, 36] that aims at improving both spatial and vertical resolution by combing information from several consecutive frames of an image sequence.

Deinterlacing has been researched for about 20 years and various algorithms have been proposed in the literature to tackle this problem. Depending on their strategy, these algorithms can be classified into four types of methods [2, 22]: The simplest methods are *non-adaptive linear techniques*. Such methods fill in missing information by simply repeating or averaging lines in vertical or temporal direction [3, 24, 32]. *Directional interpolation methods* follow a slightly different strategy. In order to allow the preservation of small image details they adapt the interpolation process to the orientation of the local image structure [3, 31, 37]. Thus they succeed to improve the quality in heavily textured image regions, where sufficient information from the spatial neighbourhood is available. A third class of methods is given by so-called *motion adaptive algorithms*. These techniques respect the motion of objects during the interpolation process by locally switching between spatial and temporal information [2, 8, 17, 22, 28]. The type of information that is actually used for filling in the missing information is thereby determined by analyzing the local motion situation. In general, this is done by means of a simple motion detector. The fourth and most advanced class of deinterlacing techniques are so-called *motion compensated approaches*. In contrast to their motion adaptive counterparts these techniques make use of a real motion estimator. Thus they are able to correct the image sequence by the occurring motion either before or during the actual interpolation step [1, 4, 10, 23, 29].

Due to their ability to consider motion information in the deinterlacing process, motion adaptive and motion compensated approaches give the best results out of these four classes [2, 22, 23, 27]. However, both strategies can hardly be compared, since they have quite complementary advantages and shortcomings: While motion adaptive techniques can provide very accurate results, they are known to have severe problems with large displacements [22]. In this case they switch back to pure spatial interpolation and do not exploit the full temporal information. Motion compensated methods, on the other hand, can handle large displacements, but may suffer from inaccuracies of their motion estimators [4].

The goal of this paper is thus to combine the advantages of both strategies within a single algorithm. Close in spirit to the work in [26] that proposes a successive refinement of the deinterlacing results, we propose a three step method that makes consequent use of discontinuity-preserving partial differential equations (PDEs). While the variational optic flow technique of Brox *et al.* [5] is used to provide accurate displacement fields for the motion compensation step, edge-enhancing anisotropic diffusion filters (EED) [34] are applied before and after the correction of the image sequence to perform pure spatial and motion adaptive spatiotemporal interpolation, respectively. Quality benchmarks with four different scenes show a very good performance of the combined method: It not only allows to deinterlace images with small and large motion, it even preserves small details in both cases. Comparisons to classical interlacing methods and a recent PDE-based technique demonstrate the clear superiority of our approach. Please note that our algorithm is intended to serve as an offline tool. Nevertheless, we point out ways to speed up the computation such that real-time performance can be achieved.

Our paper is organised as follows. In Section 2 we briefly present the main idea of our algorithm and motivate the use of its different components. While Section 3 discusses how to apply spatial edge enhancing anisotropic filtering in the context of image inpainting, Section 4 is dedicated to variational methods for motion estimation and motion compensation. In Section 5 the motion adaptive part of our algorithm is explained. Here, the spatial model for edge enhancing anisotropic inpainting is generalised to the spatiotemporal domain. Finally, Section 6 gives a detailed evaluation of our approach as well as comparisons to other methods from the literature. A summary in Section 7 concludes the paper.

2 A Novel Three Step Algorithm

Let us consider an interlaced RGB image sequence $\mathbf{f}(\mathbf{x}, t)$, where $\mathbf{f} = (f_1, f_2, f_3)^\top$ stands for the different colour channels, $\mathbf{x} = (x, y)^\top$ denotes the location within a rectangular image domain Ω and $t \geq 0$ denotes time. Let us furthermore specify the subsets of odd-indexed and even-indexed lines in the image domain as Ω_o and Ω_e with $\Omega_\mathrm{o} \cup \Omega_\mathrm{e} = \Omega$. Then, assuming w.l.o.g. that the first frame only contains odd-indexed lines, the deinterlacing problem comes down to recovering $\mathbf{f}(\mathbf{x}, t)$ for $\mathbf{x} \in \Omega_\mathrm{e}$ if t is even and $\mathbf{f}(\mathbf{x}, t)$ for $\mathbf{x} \in \Omega_\mathrm{o}$ if t is odd. In order to solve this problem we propose the following PDE-based strategy that is based on three consecutive steps:

(1) *Spatial Deinterlacing.* In order to allow the estimation of full resolution displacement fields in our motion compensation step (Step 2), we have to deinterlace our input image sequence first. However, since the images are not yet motion compensated, the use of temporal information is not recommendable. Thus, we restrict ourselves in our first step to a pure *spatial* interpolation process that is anisotropic [35]. It is based on the 2-D edge enhancing diffusion (EED) scheme [34] that already proved its favourable performance in the context of image interpolation for image (de)compression [15, 16].

(2) *Motion Estimation and Compensation.* In order to keep displacements small and thus the accuracy high, only blocks of three consecutive frames are considered in our second step. Its goal is to register the first and the last frame of each block onto the central one. Thus, even in the context of large displacements, the temporal information becomes spatially aligned such that it can be easily used for interpolation. For computing the required displacement fields, we make use of the variational optic flow approach of Brox *et al.* [5]. This discontinuity-preserving method is among the most precise techniques for motion estimation in terms of error measures.

(3) *Motion Compensated Anisotropic Diffusion.* In the final step of our algorithm we recompute the originally missing lines of each frame using *spatiotemporal* information. To this end, we consider the temporally aligned blocks from Step 2 and apply a spatiotemporal and thus motion adaptive variant of the EED-based interpolation process from Step 1. Due to the anisotropic nature of EED, this allows to recover details that have not been aligned correctly in the motion compensation step.

After we have explained the basic outline of our new deinterlacing scheme, let us now discuss its three basic steps in more detail.

3 Spatial Deinterlacing

Following our strategy from the previous section, we start by deinterlacing all frames separately. This can be seen as a special instance of our original problem for a fixed time t, where the set of known data points Ω_+ is either given by Ω_e or Ω_o. Our goal is now to find for each frame $\mathbf{f}(\mathbf{x},t)$ a deinterlaced version $\mathbf{u}(\mathbf{x},t)$ that is smooth in $\Omega \setminus \Omega_+$ and identical to $\mathbf{f}$ in Ω_+. In order to solve this task, we propose to use a spatial deinterlacing process based on edge enhancing anisotropic diffusion (EED) [34]. Such schemes have already been applied successfully in the context of image inpainting, where they provided excellent interpolation results [15, 16]. Given an interlaced RGB image at time $t = t_0$, EED computes the corresponding deinterlaced result as steady state of the three coupled nonlinear evolution equations

$$\partial_\tau u_i = \operatorname{div}\big(D(\nabla u_1,\nabla u_2,\nabla u_3)\ \nabla u_i\big) \quad \text{in } \ \Omega \times (0,\infty) \qquad \text{for } \ i = 1,2,3 \tag{1}$$

with Neumann (reflecting) boundary conditions across image boundaries,

$$\partial_n \mathbf{u} = \mathbf{0} \ \text{ on } \ \partial\Omega \ , \tag{2}$$

Dirichlet boundary conditions that prescribe the solution at given lines,

$$\mathbf{u}(\mathbf{x},t_0,\cdot) = \mathbf{f}(\mathbf{x},t_0) \ \text{ in } \ \Omega_+ \ , \tag{3}$$

and the initial condition that the original image is used where lines are available:

$$\mathbf{u}(\mathbf{x},t_0,0) = \begin{cases} \mathbf{f}(\mathbf{x},t_0) & \text{in } \ \Omega_+ \\ \mathbf{0} & \text{in } \ \Omega \setminus \Omega_+ \end{cases} . \tag{4}$$

Here, u_i stands for the different RGB channels, $\nabla u_i = (u_{ix}, u_{iy})^\top$ denotes their spatial gradient, τ serves as evolution time which is a pure numerical parameter, and the 2×2 matrix $D(\nabla u_1,\nabla u_2,\nabla u_3)$ represents a diffusion tensor that couples the different colour channels. This tensor that steers the diffusion process is constructed from the eigenvectors $\mathbf{v}_1$, $\mathbf{v}_2$ and eigenvalues λ_1, λ_2 of the joint structure tensor [12, 14]

$$J_{\rho,\sigma}(u_1, u_2, u_3) = K_\rho * \sum_{i=1}^{3} (\nabla u_i^\sigma)(\nabla u_i^\sigma)^\top \ . \tag{5}$$

In this context, $K_\rho *$ denotes convolution with a Gaussian K of standard deviation ρ and ∇u_i^σ indicates that the channel u_i has been presmoothed with a Gaussian of standard deviation σ prior to differentiation. Assuming the eigenvalues to be ordered in a decreasing manner, i.e. $\lambda_1 \geq \lambda_2$, the EED diffusion tensor finally reads

$$D(\nabla u_1, \nabla u_2, \nabla u_3) = (\mathbf{v}_1 \mid \mathbf{v}_2) \begin{pmatrix} \Psi_1(\lambda_1) & 0 \\ 0 & \Psi_2(\lambda_2) \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \end{pmatrix} \tag{6}$$

where the $|$ operator merges neighbouring vectors into a matrix and the functions applied to the first and to the second eigenvalue are given by

$$\Psi_1(s) = \frac{1}{\sqrt{1 + \frac{s}{\epsilon^2}}} \ , \qquad \Psi_2(s) = 1 \ . \tag{7}$$

with ϵ being a contrast parameter. While the Charbonnier diffusivity [9] chosen for $\Psi_1(s)$ inhibits the diffusion process across the most dominant image structure, the second function $\Psi_2(s)$ allows smoothing along it. This in turn gives us the desired anisotropic edge-preserving behaviour that is characteristic for EED.

In order to compute the steady state, i.e. $\tau \to \infty$, of the coupled evolution equations given by (1)-(6), we used an explicit scheme based on finite difference approximations. Typical run times for such an implementation are in the order of about 3 seconds for images of size 300×300. Recent parallelisations of more efficient numerical schemes on the Cell Processor of Sony's Playstation 3 even achieve up to 34 frames per second [25], thereby dealing with an almost twice as large and less regular inpainting domain Ω_-. Taking those differences into account, run times of about 20 milliseconds can be expected for the spatial deinterlacing step, if modern parallel architectures are used.

4 Motion Estimation and Compensation

After we have deinterlaced at least three consecutive frames of the original image, we can continue with our second step: the motion estimation and compensation. To this end, we consider blocks of three consecutive frames, and register the first and the last frame onto the central one:

$$\mathbf{f}(\mathbf{x}, t_0 - 1) \xrightarrow{\mathbf{w}^-} \mathbf{f}(\mathbf{x}, t_0) \xleftarrow{\mathbf{w}^+} \mathbf{f}(\mathbf{x}, t_0 + 1)$$

Although it is possible to use more than three frames in this step, one should be aware that this results in larger displacements which in turn would deteriorate the quality of the motion estimation.

In order to compute the two motion fields $\mathbf{w}^+ = (v^+, w^+)^\top$ and $\mathbf{w}^- = (v^-, w^-)$ we make use of the highly precise variational optic flow technique of Brox *et al.* [5]. For RGB images, this technique computes the displacement field between two consecutive frames as minimiser of the energy functional

$$
\begin{aligned}
E(\mathbf{w}^\pm) = \quad & \int_\Omega \psi\Big(\underbrace{\sum_{i=1}^{3} \big|f_i(\mathbf{x}+\mathbf{w}^\pm, t_0 \pm 1) - f_i(\mathbf{x}, t_0)\big|^2}_{\text{colour constancy}} \\
& \qquad + \gamma \underbrace{\sum_{i=1}^{3} \big|\nabla f_i(\mathbf{x}+\mathbf{w}^\pm, t_0 \pm 1) - \nabla f_i(\mathbf{x}, t_0)\big|^2}_{\text{colour gradient constancy}} \Big)\, d\mathbf{x} \\
& + \alpha \int_\Omega \psi\big(\underbrace{|\nabla v^\pm|^2 + |\nabla w^\pm|^2}_{\text{spatial smoothness}} \big)\, d\mathbf{x}\,.
\end{aligned} \tag{8}
$$

While the first expression in the data term models the assumption that the colour of objects remains constant over time, the second one renders the approach more robust against varying illumination. This is achieved by assuming constancy of the spatial image gradient of the different colour channels given by $\nabla f_i = (f_{ix}, f_{iy})^\top$. The weighting between the two assumptions is realised with a positive scalar γ. In order to allow for a correct estimation of large displacements, both assumptions are used in their original nonlinear form. Please note that this is extremely important in the context of deinterlacing, since the typical motion is in the order of several pixels. Finally, deviations from both the data and the smoothness term are penalised in a non-quadratic way via a robust function ψ. This improves the performance of the approach with respect to outliers and noise in the case of the data term and preserves motion boundaries by modelling a piecewise smooth flow field in the case of the smoothness term. For both purposes the regularised L_1-norm is used, which, for the smoothness term, comes down to the total variation (TV) regulariser [30] given by $\psi(s^2) = \sqrt{s^2 + \epsilon^2}$. The regularisation parameter ϵ is set to 10^{-3}. In this context, one should note that without the preservation of motion boundaries, blurry flow information would lead to a filling-in of wrong temporal data at the corresponding image locations. This in turn would result in a significant deterioration of object boundaries in the deinterlaced image both with respect to quality and localisation.

The minimisation of this energy functional is done via its associated Euler-Lagrange equations. These equations are given by the following coupled system of nonlinear elliptic PDEs:

$$
\begin{aligned}
0 = \psi'(...)\Big(\sum_{i=1}^{3} \gamma_i \, \big(f_i(\mathbf{x}+\mathbf{w}^\pm, t_0 \pm 1) - f_i(\mathbf{x}, t_0)\big) \frac{\partial}{\partial x} f_i(\mathbf{x}+\mathbf{w}^\pm, t_0 \pm 1)\Big) \\
+ \alpha \operatorname{div}\big(\psi'\left(|\nabla u^\pm|^2 + |\nabla v^\pm|^2\right) \nabla u^\pm\big)\ ,
\end{aligned} \tag{9}
$$

$$0 = \psi'(...)\Big(\sum_{i=1}^{3}\gamma_i\,\left(f_i(\mathbf{x}{+}\mathbf{w}^{\pm},t_0{\pm}1){-}f_i(\mathbf{x},t_0)\right)\frac{\partial}{\partial y}f_i(\mathbf{x}{+}\mathbf{w}^{\pm},t_0{\pm}1)\Big)$$
$$+\,\alpha\ \mathrm{div}\left(\psi'\left(|\nabla u^{\pm}|^2+|\nabla v^{\pm}|^2\right)\nabla v^{\pm}\right)\,, \tag{10}$$

where $\psi'_D(...)$ abbreviates the factor in front of the data term that actually reads

$$\psi'(...) = \psi'\Big(\sum_{i=1}^{3}\left|f_i(\mathbf{x}+\mathbf{w}^{\pm},t_0\pm 1)-f_i(\mathbf{x},t_0)\right|^2 + \gamma\sum_{i=1}^{3}\left|\nabla f_i(\mathbf{x}+\mathbf{w}^{\pm},t_0\pm 1)-\nabla f_i(\mathbf{x},t_0)\right|^2\Big)\,. \tag{11}$$

As suggested in [5], we implemented a coarse-to-fine warping strategy based on two nested fixed point iterations. This yields typical run times of about 3 seconds for computing both the forward and the backward flow field for images of size 300×300. However, also in this case the computations can be accelerated significantly: Either one can use more sophisticated numerical schemes such as the nonlinear multigrid methods presented in [6, 7] or can consider recent implementations on parallel architectures such as graphics hardware [18, 33] or the Cell processor [19, 20]. While efficient numerics already allow to reduce the computational effort to less than a second, highly parallel implementations promise run times of about 30 millisecond for both flow fields. Thus real-time seems also to be possible for the second step of our deinterlacing algorithm.

After the two displacement fields have been computed, all frames are registered towards the central one. To this end, we use a backward warping strategy based on bilinear interpolation. Alternatively, also higher order splines can be used, but first results did not show significant improvements in terms of quality. From a computational viewpoint, this effort is neglectable.

5 Motion Compensated Anisotropic Diffusion

In the third and last step of our algorithm we apply a spatiotemporal anisotropic diffusion process to *recompute* the originally missing lines of the central frame of each registered image block. To this end, we consider a spatiotemporal generalisation of the EED-based deinterlacing algorithm from Section 3. It is defined on the image domain of the complete block and reads

$$\partial_\tau u_i = \mathrm{div}\big(D(\nabla_t u_1,\nabla_t u_2,\nabla_t u_3)\ \nabla_t u_i\big) \quad \text{in } \ \varOmega\times[t_0-1,t_0+1,]\times(0,\infty) \quad \text{for } \ i=1,2,3\,. \tag{12}$$

Here, $\nabla_t = (u_{ix}, u_{iy}, u_{it})^\top$ denotes the spatiotemporal gradient of u_i which allows to consider temporal information in contrast to the spatial one that we used in Step 1. Consequently, also the joint structure tensor is extended to the spatiotemporal domain

$$J_{\rho,\sigma}(u_1,u_2,u_3) = K_\rho * \sum_{i=1}^{3} (\nabla u_i^\sigma)(\nabla u_i^\sigma)^\top . \tag{13}$$

Finally, the joint diffusion tensor is adapted correspondingly and reads now

$$D(\nabla u_1,\nabla u_2,\nabla u_3) = (\mathbf{v}_1 \mid \mathbf{v}_2 \mid \mathbf{v}_3) \begin{pmatrix} \Psi_1(\lambda_1) & 0 & 0 \\ 0 & \Psi_2(\lambda_2) & 0 \\ 0 & 0 & \Psi_3(\lambda_3) \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^\top \\ \mathbf{v}_2^\top \\ \mathbf{v}_3^\top \end{pmatrix} , \tag{14}$$

where the functions of the three eigenvalues are defined as

$$\Psi_1(s) = \frac{1}{\sqrt{1+\frac{s}{\epsilon^2}}} , \qquad \Psi_2(s) = 1 , \qquad \Psi_3(s) = 1 . \tag{15}$$

This yields an anisotropic diffusion process that smoothes only along the plane perpendicular to the local gradient of the evolving image, but not across it.

Actually, this process is fully *motion adaptive*: By inhibiting the diffusion in gradient direction, it stops the filling-in of information along that direction which is least consistent with respect to the colour value of the current location. This in turn prevents the algorithm from using meaningless temporal information, if the estimated displacement field did locally not allow for a correct interframe registration. Moreover, since the directions are not necessarily axis-aligned, the combination of compensation and adaptation allows to correct small misalignments if the registration did not perfectly match.

As in Step 1, this process is discretised by means of a simple explicit scheme using finite differences and the solution for $\tau \to \infty$ is computed. However, since only the originally missing lines of the central frame are recomputed, it is hardly more expensive from a computational viewpoint than the pure spatial interpolation process. The main difference comes from the additional number of neighbours required for the discretisation of the anisotropic diffusion process in 3-D. Typical run times are in the order of 4 seconds for images of size 300×300. Considering the additional effort compared to the 2-D case, run times of about 40 milliseconds per frame seem to be possible on parallel hardware. Thus also for this third step real-time is well within our computational reach.

6 Experiments

In order to evaluate the quality of our approach, we have created four interlaced test sequences with increasing difficulty: The *House*, the *Duck*, the *City* and the *Run* sequence (see Figure 1). These sequences have been cropped from video sequences of the European Broadcasting Union (EBU) [1] and then interlaced

[1] `ftp://ftp.ebu.ch/en/technical/hdtv/test_sequences.php`

afterwards. This offers the advantage that the ground truth is known and the deinterlacing quality can be determined in terms of error measures. In our case this was done via the *average absolute colour error*

$$E_{\mathrm{AACE}} = \frac{1}{3N} \sum_{i=1}^{3} \sum_{p=1}^{N} |f_{i,p}^{\mathrm{truth}} - f_{i,p}^{\mathrm{deinter}}| \ , \tag{16}$$

where N is the number of pixels and f_c^{truth} and f_c^{deinter} denote the RGB channels of the ground truth and the deinterlaced image, respectively. For comparing the performance of our method, we have implemented three classical techniques: field doubling, line averaging, and spatiotemporal median filtering. Additionally we have considered a recent PDE-based approach based on motion adaptive total variation that is among the best techniques for deinterlacing [22]. Here, the original implementation of the authors was used. Finally, we also included the results from pure spatial edge-enhancing anisotropic diffusion (2-D EED) that serves as first step in our algorithm. In all cases the parameters have been

Fig. 1. Cropped and interlaced test image sequences created from sequences of the European Broadcasting Union (300 × 300 pixels). *Top Left:* House. *Top Right:* Duck. *Bottom Left:* City. *Bottom Right:* Run.

Table 1. Parameter settings for the spatial and the spatiotemporal deinterlacing step.

Sequence	2-D EED (Step 1)			MCEED (Step 3)		
	σ	ϵ	ρ	σ	ϵ	ρ
House	0.6	1.8	0.0	0.5	0.7	4.0
Duck	0.6	1.6	0.0	0.7	0.8	5.0
City	0.6	1.7	0.0	0.5	0.7	4.0
Run	0.7	2.1	0.0	0.5	0.8	4.0

optimised with respect to the AACE. While the parameters for the optical flow computation were set fixed to $\alpha = 20$ and $\gamma = 5$, the parameters for the spatial and spatiotemporal inpainting steps are listed in Table 1. As one can see, they only vary slightly throughout all sequences. Typical runtimes for our algorithm are in the order of 10 seconds for images of size 300×300. However, as pointed out before, efficient implementations on parallel hardware such as graphics cards or the Cell processor promise frame rates of about 10 deinterlaced frames per second (adding up the expected time for all three steps).

Figure 2 shows by the example of the two sequences with largest displacements (*City, Run*) that our method gives excellent deinterlacing results. As one can see from the corresponding error values in Table 2, our method thereby clearly outperforms all other techniques. In particular for the *Run* scene, where the largest motion is present, the superiority of our method becomes obvious: While pure spatial interpolation methods (line averaging, 2-D EED) perform relatively good – at least they do not introduce wrong temporal information – motion adaptive techniques such as median filtering or the total variation approach show some problems. Also field doubling as pure temporal interpolation method gives very bad results. Only our motion compensated edge-enhancing diffusion technique (MCEED) is capable of improving the spatial results by additionally considering temporal information. Also for images with smaller motion, our method performs favourably. Although, in this situation, the motion adaptive algorithms are better than the spatial ones, they cannot compete with our results that show again the lowest errors.

In order to allow for a visual comparison of the deinterlacing quality of all methods, we show a representative detail from the results for the *Run* scene in Figure 3. It depicts a running man and requires the preservation of many small scale features. How challenging the deinterlacing of this image actually is can be seen from the classical field doubling method that simply merges two consecutive half-frames (fields). Since it uses the temporal information in the worst possible way, its poor results are a good indicator for the difficulty of the scene. Evidently, our motion compensated approach gives the sharpest results of all methods: Features such as the slightly open mouth, the structure of the beard and the numerous details at the shoulders are recovered with a much higher quality than by any other method. Obviously, this is a direct consequence of the use of discontinuity-preserving concepts in *all* parts of our algorithm. It makes explicit that selecting the components carefully and adjusting them to each other is an important task when designing a combined algorithm.

Table 2. Comparison to interlacing methods from the literature. Deviations from the ground truth are measured by the average absolute colour error. RGB values are in the range $[0, 255]$. Superscripts next to error values denote the rank of a method.

Technique	Increasing Motion →			
	House	Duck	City	Run
Field Doubling [31]	2.91^3	4.05^2	6.89^6	13.23^6
Line Averaging [31]	4.22^6	5.92^6	3.70^3	5.61^3
Our Method (2-D EED)	4.06^5	5.87^5	3.66^2	5.59^2
Spatiotemporal Median [3]	2.99^4	4.38^4	4.27^5	7.01^5
Total Variation [22]	2.89^2	4.15^3	3.77^4	5.85^4
Our Method (MCEED)	$\mathbf{2.58}^1$	$\mathbf{3.48}^1$	$\mathbf{3.29}^1$	$\mathbf{4.48}^1$

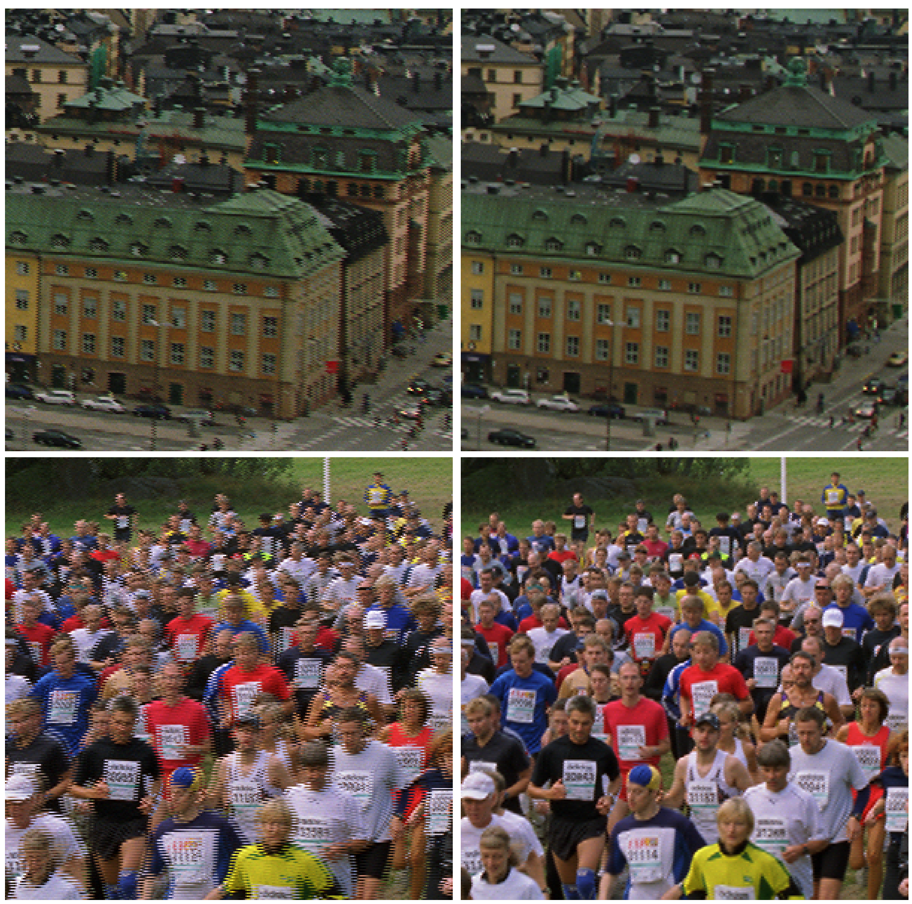

Fig. 2. Results for the *City* and the *Run* sequence. *Left Column:* Interlaced Images. *Right Column:* Deinterlaced images with MCEED (our method).

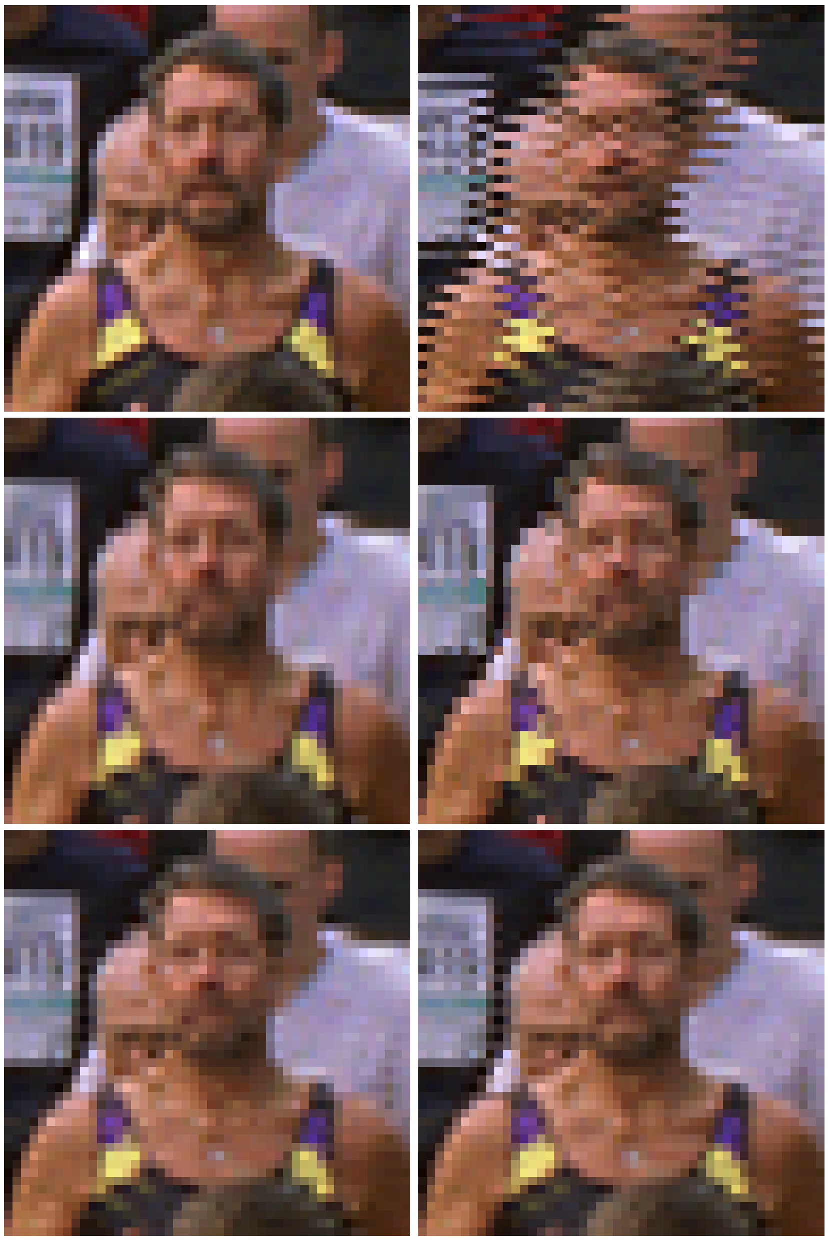

Fig. 3. Detail comparison of the *Run* sequence for different deinterlacing methods (48×48 pixels). *Top Left:* Truth. *Top Right:* Field doubling. *Center Left:* Line averaging. *Center Right:* Median filtering. *Bottom Left:* Motion adaptive total variation [22]. *Bottom Right:* MCEED (our method).

Fig. 4. Detail comparison for the main building in the *House* sequence (60×30 pixels). *Left:* Truth. *Center:* Motion adaptive total variation [22]. *Right:* MCEED (our method).

Two more examples for the discontinuity-preserving deinterlacing property of our method are given in Figures 4 and 5. Figure 4 shows a detail of the deinterlaced *House* image and compares it to the result of the motion adaptive total variation technique. Since the motion in this scene is rather low, one would expect good results from both methods. However, only our technique allows to recover the small horizontal structures due to the excellent interpolation properties of the spatiotemporal edge-enhancing diffusion filter. A similar observation can be made in Figure 5, where a detail of the deinterlaced *City* image is depicted. Here, we compare our method against a pure spatial edge-enhancing diffusion filtering that also offers a very good interpolation quality. Although the 2-D EED algorithm gives the second best results for this scene, it cannot reconstruct the horizontal lines properly. Once again, only our approach allows to fill in the necessary information. In contrast to the previous example, where only the edge-preservation was important, the motion compensation plays an equally important role for this scene. By correctly aligning the subsequent images, the missing lines can be recovered correctly from the temporal information. Using spatial information alone this problem could not have been solved. This shows that high quality deinterlacing is possible for scenes with large displacements if motion adaptive and motion compensated approaches are combined.

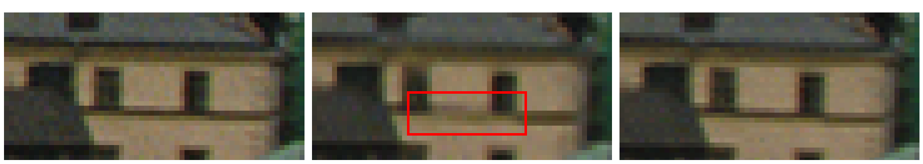

Fig. 5. Detail comparison for a building in the upper left corner of the *City* sequence (60×30 pixels). *Left:* Truth. *Center:* Spatial EED (our method, only Step 1). *Right:* MCEED (our method).

7 Summary and Conclusions

In this paper we have demonstrated that motion compensated and motion adaptive approaches can benefit from each other. To this end, we developed a combined three strep strategy that makes use of recent PDE-based approaches for motion estimation and image inpainting. Thereby we focused on such techniques that are capable of preserving discontinuities in the computational process. In the experimental section, the advantages of our new method became explicit: In contrast to other approaches that performed well either for scenes with large or for scenes with small motion, it was able to achieve high quality results in both cases. Moreover, due to the discontinuity-preserving nature of its single components, it allowed to recover small details in the deinterlacing process that could not be restored by other schemes. This shows that both contributions are equally important for the success of our algorithm: Without combining motion compensated and motion adaptive techniques, our method would not work for small and large motion, while without preserving discontinuities, important features would get lost during the deinterlacing process.

Our ongoing work addresses the implementation of efficient numerical schemes on recent parallel hardware. Moreover, it focuses on the application of such schemes to images in full HDTV resolution (1920×1280). Evidently, this requires additional speedups that have to be achieved both on the algorithmic as well as on the parallelisation side. In this context, also simplified models may help to bridge the gap towards real-time HDTV deinterlacing. This, however, is topic of our future work.

Acknowledgements

We thank Sune Keller, Francois Lauze and Mads Nielsen for providing their deinterlacing algorithm for testing.

References

1. A. Almog, A. Levi, and A. M. Bruckstein. Spatial de-interlacing using dynamic time warping. In *Proc. 2005 IEEE International Conference on Image Processing*, volume 2, pages 1010–1013, Genova, Italy, Sept. 2005.
2. C. Ballester, B. Bertalmio, V. Caselles, L. Garrido, A. Marques, and F. Ranchin. An inpainting-based deinterlacing method. *IEEE Transactions on Image Processing*, 16(10):2476–2491, Oct. 2007.
3. E. B. Bellers and G. de Haan. *A Key Technology for Scanline Conversion*. Elsevier, Amsterdam, 2000.
4. M. Biswas, S. Kumar, and T. Nguyen. Performance analysis of motion-compensated de-interlacing systems. *IEEE Transactions on Image Processing*, 15(9):2596–2609, Aug. 2006.
5. T. Brox, A. Bruhn, N. Papenberg, and J. Weickertw. High accuracy optic flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Computer Vision – ECCV 2004*, volume 3024 of *Lecture Notes in Computer Science*, pages 25–36. Springer, Berlin, 2004.
6. A. Bruhn and J. Weickert. Towards ultimate motion estimation: Combining highest accuracy with real-time performance. In *Proc. Tenth International Conference on Computer Vision*, volume 1, pages 749–755, Beijing, China, June 2005. IEEE Computer Society Press.

7. A. Bruhn, J. Weickert, T. Kohlberger, and C. Schnörr. A multigrid platform for real-time motion computation with discontinuity-preserving variational methods. *International Journal of Computer Vision*, 70(3):257–277, Dec. 2006.
8. L. Capodiferro. Interlaced to progressive conversion by median filtering. In L. Chiarglione, editor, *Proc. 3rd International Workshop on HDTV*, volume 3, pages 677–684, 1989.
9. P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *Proc. 1994 IEEE International Conference on Image Processing*, volume 2, pages 168–172, Austin, TX, Nov. 1994. IEEE Computer Society Press.
10. J. P. Cocquerez, L. Chanas, and J. Blanc-Talon. Simultaneous inpainting and motion estimation of highly degraded video-sequences. In J. Bigün and J. Gustavsson, editors, *Image Analysis*, volume 2749 of *Lecture Notes in Computer Science*, pages 685–692, June 2003.
11. L. D'Amore, L. Marcellino, and A. Murli. Image sequence inpainting: a numerical approach for blotch detection and removal via motion estimation. *Journal of Computational and Applied Mathematics*, 2:396–413, 2007.
12. S. Di Zenzo. A note on the gradient of a multi-image. *Computer Vision, Graphics and Image Processing*, 33:116–125, 1986.
13. M. Elad and A. Feuer. Superresolution restoration of an image sequence: adpative filering approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:817–834, 1999.
14. W. Förstner and E. Gülch. A fast operator for detection and precise location of distinct points, corners and centres of circular features. In *Proc. ISPRS Intercommission Conference on Fast Processing of Photogrammetric Data*, pages 281–305, Interlaken, Switzerland, June 1987.
15. I. Galić, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, and H.-P. Seidel. Towards PDE-based image compression. In N. Paragios, O. Faugeras, T. Chan, and C. Schnörr, editors, *Variational, Geometric and Level Set Methods in Computer Vision*, volume 3752 of *Lecture Notes in Computer Science*, pages 37–48. Springer, Berlin, 2005.
16. I. Galić, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, and H.-P. Seidel. Image compressions with anisotropic diffusion. *Journal of Mathematical Imaging and Vision*, 31:255–269, 2008.
17. D. Gillies, M. Plantholt, and D. Westerkamp. Motion adaptive field rate upconversion algorithms for 900 lines/100 hz/2:1 displays. *IEEE Transactions on Consumer Electronics*, 36:149–160, May 1990.
18. H. Grossauer and P. Thoman. GPU-based multigrid: Real-time performance in high resolution nonlinear image processing. In A. Gasteratos, M. Vincze, and J. K. Tsotsos, editors, *Computer Vision Systems*, volume 5008 of *Lecture Notes in Computer Science*, pages 141–150. Springer, Berlin, 2008.
19. P. Gwosdek, A. Bruhn, and J. Weickert. High performance parallel optical flow algorithms on the Sony Playstation 3. In O. Deussen, D. Keim, and D. Saupe, editors, *VMV '08: Proc. Vision, Modeling, and Visualization*, pages 253–262, Konstanz, Germany, October 2008. AKA.
20. P. Gwosdek, A. Bruhn, and J. Weickert. Variational optic flow on the Sony Playstation 3 – acurate dense flow fields for real-time applications. Technical Report 233, Department of Mathematics, Saarland University, Germany, April, 2009.
21. H. He and L. P. Kondi. A regularization framework for joint blur estimation and super-resolution of video sequences. In *Proc. 2005 IEEE International Conference on Image Processing*, volume 3, pages 329–332, Genova, Italy, Sept. 2005.

22. S. Keller, F. Lauze, and M. Nielsen. A total variation motion adaptive deinterlacing scheme. In R. Kimmel, N. Sochen, and J. Weickert, editors, *Scale-Space and PDE Methods in Computer Vision*, volume 3459 of *Lecture Notes in Computer Science*, pages 408–415. Springer, Berlin, 2005.
23. S. Keller, F. Lauze, and M. Nielsen. Deinterlacing using variational methods. *IEEE Transactions on Image Processing*, 17(11):2015–2028, Nov. 2008.
24. A. Kokaram. *Motion Picture Restoration*. Springer, London, 1998.
25. H. Köstler, M. Stürmer, C. Freundl, and U. Rüde. PDE based video compression in real time. Technical Report 07-11, Institut für Informatik, Univ. Erlangen-Nürnberg, Germany, 2007.
26. J. Kovačević, R. J. Safranek, and E. M. Yeh. Deinterlacing by succesive approximation. *IEEE Transactions on Image Processing*, 6(2):339–344, Oct. 1997.
27. M. Li and T. Nguyen. A de-intelacing algorithm using Markov random field model. *IEEE Transactions on Image Processing*, 16(11):2633–2648, Nov. 2007.
28. A. Nguyen and E. Dubois. Spatio-temporal adaptive interlaced-to-progressive conversion. In E. Dubois and L. Chiarglione, editors, *Proc. 4th International Workshop on HDTV*, pages 749–756, Kawasaki, Japan, 1992.
29. A. J. Patti, A. M. Tekalp, and M. I. Sezan. Image sequence restoration and deinterlacing by motion compensated kalman filtering. In *Image and Video Processsing*, volume 1903 of *Proceedings of SPIE*, pages 59–70. SPIE Press, Bellingham, 1993.
30. L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
31. M. Tekalp. *Digital Video Processing*. Prentice Hall, Englewood Cliffs, 1995.
32. Y. Wang, J. Ostermann, and Y. Zhang. *Video Processing and Communications*. Prentice Hall, Upper Saddle River, NJ, 2002.
33. A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers. An improved algorithm for TV-L^1 optical flow. In B. Rosenhahn, D. Cremers, and A. Yuille, editors, *XXXXXX*, volume xxxx of *Lecture Notes in Computer Science*, pages xxx–xxx. Springer, Berlin, 2009.
34. J. Weickert. Theoretical foundations of anisotropic diffusion in image processing. *Computing Suppl.*, 11:221–236, 1996.
35. J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart, 1998.
36. N. A. Woods and N. P. Galatsanos. Non-stationary approximate Bayesian superresolution using a hierarchical prior model. In *Proc. 2005 IEEE International Conference on Image Processing*, volume 1, pages 37–40, Genova, Italy, Sept. 2005.
37. H. Yoo and J. Jeong. Direction-oriented interpolation and its application to deinterlacing. *IEEE Transactions on Consumer Electronics*, 48(4):954–962, 20.