

Deinterlacing Using Variational Methods

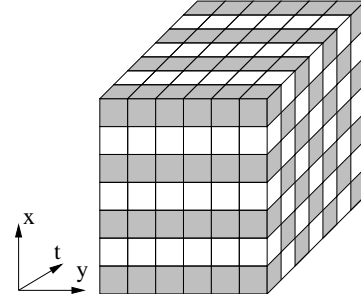
Sune Høgild Keller*, François Lauze and Mads Nielsen

Abstract—We present a variational framework for deinterlacing that was originally used for inpainting and subsequently redeveloped for deinterlacing. From the framework, we derive a motion adaptive (MA) deinterlacer and a motion compensated (MC) deinterlacer and test them together with a selection of known deinterlacers. To illustrate the need for MC deinterlacing, the problem of details in motion (DIM) is introduced. It cannot be solved by MA deinterlacers or any simpler deinterlacers but only by MC deinterlacers. The major problem in MC deinterlacing is computing reliable optical flow (motion estimation, ME) on interlaced video. We discuss a number of strategies for computing optical flows on interlaced video hoping to shed some light on this problem. We produce results on challenging real world video data with our variational MC deinterlacer that in most cases are indistinguishable from the ground truth.

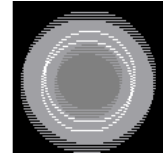
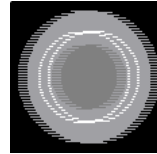
I. INTRODUCTION

Throughout the history of television, interlaced scan has been the dominant scan format for recording, broadcasting, storing and displaying of video and television. Interlacing is cost efficient and has until recently been sufficient to ensure the best possible viewing experience to the human visual system (HVS). For several reasons, this is no longer the case. Firstly, interlacing artifacts that were once not a problem, have started to become visible as screens have grown larger and brighter with higher contrasts. Secondly, progressive scan screens (and tv/video cameras) have become available, offering significantly better viewing quality.

In progressive scan all horizontal lines of a 2D spatial video frame are scanned. In interlaced scan one alters between recording the even or odd horizontal lines of the frames leading to the notation of these half frames as even and odd *fields*. Interlacing scan is illustrated in Fig. 1(a). When the interlaced fields are recorded separated in time, then two neighboring fields (even, odd pair) cannot be merged to one full frame to be displayed on progressive screens without problems, and neither can the illusion of full frames on large and bright interlaced displays. It is a well-known fact that the HVS is more sensitive to flicker in large areas, which is why interlaced scan was thought of in the first place: On smaller screens, no one will spot flickering of single lines. As screens grew in size, brightness and contrast, interlacing artifacts became visible and annoying to the HVS. The flickering of single lines is called line flicker and cannot be illustrated in print, but the two other interlacing artifacts, serration and line crawl, are shown in Fig. 1(b) Details on interlacing artifacts can be



(a)



(b)



(c)

Fig. 1. (a) The interlacing (width, height, time)-volume. (b) Interlacing artifacts shown by a merge of a time t field with its neighboring time $t + 1$ field: firstly, serration due to horizontal motion in the scene is depicted, secondly, line crawl due to vertical motion is depicted. (c) Shows how the simple scene in (b) should look perfectly deinterlaced.

found in the works of Bellers and de Haan [3], Pigeon and Guillotel [27], and Keller *et al.* [19].

The broadcasting and distribution of television and video media is still dominated by interlaced scan. Even with the rise of high-definition television (HDTV) and the known superiority in viewing quality of progressive video over interlaced video, interlacing persists, e.g. as one of the formats used for HDTV in the US and Japan (1080i, 1080×1920 resolution with only 540 lines scanned in each frame). As most plasma and LCD screens are progressive displays and they are dominating the world market of both television sets and computer monitors today, and as devices used for television and video (and music) are integrating with computers into media centers, there is – and will be – a need for conversion between interlaced and progressive formats. Progressive to interlaced conversion is a simple downsampling whereas interlaced to progressive conversion – *deinterlacing* – is, by the definitions of the sampling theory, an ill-posed upscaling (subsampling) problem.

A. The Deinterlacing Problem Taxonomy

In practice, deinterlacing deals with a list of problems, a taxonomy of deinterlacing problems. A first version of this taxonomy was the removal of the three deinterlacing problems as introduced in [27] by Pigeon and Guillotel. To that, we add the problem of odd length vertical motion vector component, which was carefully analyzed by Bellers and de Haan in [3],

S. H. Keller (sunebio@diku.dk) is with the PET Center, Rigshospitalet (Copenhagen University Hospital), Blegdamsvej 9, DK-2100 Copenhagen, Denmark, phone +45 3545 1633, fax: +45 3545 3898. F. Lauze (francois@diku.dk) and M. Nielsen (madsn@diku.dk) are with The Image Group, Department of Computer Science, Faculty of Science, University of Copenhagen, Universitetsparken 1, DK-2100 Copenhagen, Denmark, phone: +45 3532 1400, fax: +45 3532 1401. Manuscript number: TIP-03671-2007.R1.

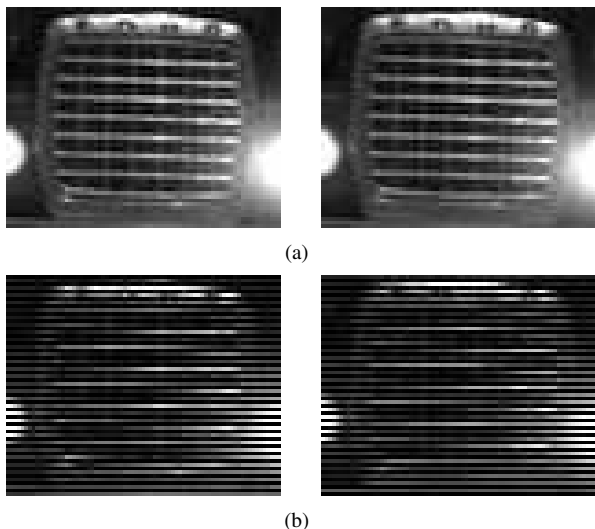


Fig. 2. The details in motion problem. (a) Cutout of frames 6 and 7 of the progressive sequence *Truck* of a truck driving towards the camera. (b) The same two frame cutouts viewed as interlaced fields where almost all information on near horizontal structures is lost, thus making deinterlacing an extremely difficult task.

With this problem, motion vectors in the non-existing lines of the interlaced volume point to other non-existing lines in neighboring fields. This is of course only a problem with motion compensated deinterlacers using motion vectors. As a last problem, we introduce the problem of details in motions (DIM), with an example given in Fig. 2. The details can be any highly detailed (high spatial frequency) parts of moving object (or stationary objects in case of camera motion) from edges to texture. Thus, we have the deinterlacing problem taxonomy:

- Serration
- Line crawl
- Line flickering
- Odd vertical motion vector length
- Details in motion (DIM).

To this list, we can add the problems generally encountered in video and image processing. We will focus on the details in motion problem in this paper.

In this paper, we will present a variational framework for deinterlacing and two deinterlacing algorithms developed from that, both of which do high quality deinterlacing. Let us say we have a representative selection of interlaced video w.r.t. motion and details depicted (high spatial frequency content). Using simple spatial or temporal deinterlacing techniques on this video will yield acceptable (spatial interpolation) to poor (temporal interpolation) results. The poor quality comes from not adapting to or compensating for the presence of motion.

Using motion adaptive deinterlacers will often get rid of most of the interlacing artifacts. They use motion detection to determine whether to use local temporal interpolation – in case of no motion and thus high local temporal coherence – and when to interpolate spatially only – in case of motion. Motion adaptive deinterlacers are, however, unable to solve the problem of details in motion. It is basically the problem of correctly deinterlacing sequences containing details (high spatial frequencies) in motion. ‘Correctly deinterlacing’ meaning

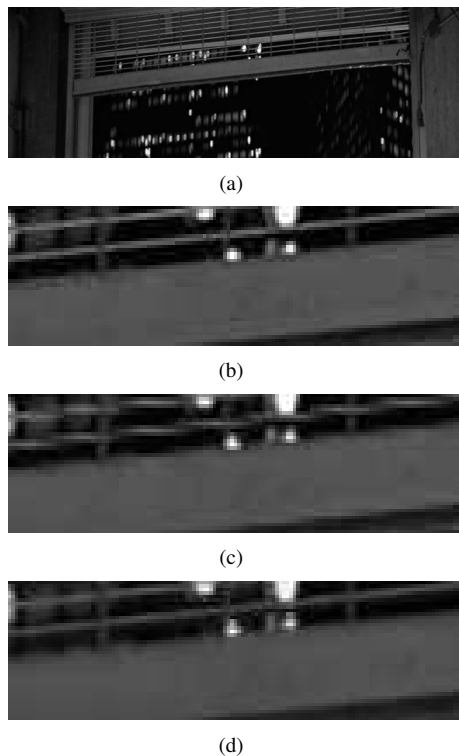


Fig. 3. (a) Part of the progressive sequence *Grille* with closeup zoom, (b). The security roller grille is moving downwards at a speed of app. 2 pixels per frame ($\approx 1/4$ vertical grille space). This sequence is then artificially interlaced. (c) Deinterlaced output using our variational motion adaptive algorithm (the four other motion adaptive deinterlacers tested on *Grille* give similar or worse results). (d) deinterlaced output of our variational motion compensated deinterlacer, the result being almost identical to the original with unbroken grille lines and sharp, straight edges.

that the output appears detailed to the human visual system and does not contain any visible artifacts to annoy it. Since the human visual system is highly sensitive to edges, even a single edge which is poorly deinterlaced can significantly lower the visual quality of the progressive output.

The problem of details in motion can be handled by smoothing the problem region but the inevitable blurring is also an artifact which annoys the viewer. A better choice is to propagate information along the motion trajectory (optical flow) doing motion compensated (MC) deinterlacing, which done right will transport detailed video content recorded in neighboring fields into the lines in the present field where they are missing. From a mere visual inspection of the two frames/fields of the sequence *Truck* given in Fig. 2, it is obvious how difficult it would be to recreate the correct structure using only local temporal and spatial information as all non-MC deinterlacers do. Fig. 3 shows another sequence with details in motion, on which any of the five motion adaptive deinterlacers we have tested fails (as well as all simple deinterlacers tested). As seen in Fig. 3 our variational motion compensated deinterlacer is able to solve the problem of details in motion in this case.

Not all cases of the details in motion problem can be solved with our variational MC deinterlacer, as estimating the optical flow precisely on the perforated structure of interlaced image volumes is not a simple task, and this why why the

problem of odd length vertical motion is included in the taxonomy above. By deinterlacing a set of fields together iteratively as we have chosen to do, one can hope to find meaningful information also when having odd length vertical motion. Another significant step in the right direction is to use modelling of the human visual system and real world physical constraints on recorded image sequence content as we suggest in our variational deinterlacing framework. Our variational motion adaptive deinterlacer was first introduced in [19] and the variational motion compensated deinterlacer in [18]. In this paper, we give the full and in-depth presentation of variational deinterlacing and use the case study of the details in motion problem to show the superiority of motion compensated deinterlacing in terms of obtainable quality. We also discuss the complexity of motion estimation on interlaced image sequences as compared to 'standard' progressive sequence motion estimation.

This paper is organized as follows: In Section II, we discuss deinterlacing in depth and give an overview of related work in the fields of deinterlacing and inpainting. Then in Section III, we present our probability-based variational framework and our two deinterlacing algorithms derived from it. Finally, we present our experimental work and deinterlacing results in Section IV.

II. BACKGROUND AND RELATED WORK

In this section, we will discuss other work on deinterlacing and present ten deinterlacing algorithms from literature, which we have implemented and tested along with our own two algorithms and the DCDi[®] deinterlacing in Faroudja's DVP-1010 video processor. We also discuss motion adaptive and motion compensated deinterlacing, optical flow estimation on interlaced video and the origin of variational deinterlacing.

A. Simple Deinterlacing Techniques

The simple algorithms given here can be found in video processing text books like [32] and [36] and in the 'must-read' book on deinterlacing by Bellers and de Haan, [3]. Starting with the spatial methods, Line Doubling (LDB) is the simplest interpolating of the missing horizontal line by repetition of the above known line. LDB has been widely used in practice along with Line Averaging (LAV) or 'bob', which is the vertical fifty-fifty average of the above and below lines. In the comprehensive testing of deinterlacers in [3] and in our test as well, LAV performs – given its simplicity – extremely well. The same cannot be said about the simplest temporal deinterlacer, Field Insertion (FI), aka merging or weaving. FI fills in the blank lines with neighboring lines in time, and since its results are generally very similar to the images seen on an interlaced display, ([3] and [36]), it was the algorithm we used to simulate the interlaced views of serration and line crawl in Fig. 1(b). Field averaging (FAV) averages the before and after temporal neighboring lines of each missing line. Vertical Temporal interpolation (VT) is a 50/50 combination of LAV and FAV. Many variations of vertical temporal filters have been suggested, e.g. by Thomas in [33].

Median filtering (Med) which is a real classic, is used for deinterlacing in many variations, see for instance [3], [9], [15], [31], [30] and [32]. For our testing, we have chosen a 3-tap vertical spatiotemporal using the forward temporal neighbor along with the two vertical spatial neighbors also used in LAV.

Spatial Edge Adaptive deinterlacing (EA) has been suggested in several forms, e.g. in [13], [22] and [32]. In all cases, one tries to find dominating edge direction along which to interpolate with a skewed LAV filter (perpendicular to the image gradient). This works relatively well on $\pm 45^\circ$ edges, but it fails miserably on data like the example in Fig. 2. We have chosen to implement a scheme that determines the direction of interpolation by measuring the Summed Absolute Differences (SAD) (see [32]) along a set of five candidate edge directions, 0° , $\pm 45^\circ$ and $\pm 63^\circ$ from vertical.

B. Motion Adaptive Deinterlacing

The concept of motion adaptive (MA) deinterlacing has been sketched in Section I. We have chosen to describe and implement three MA deinterlacers. The first is given by Skarabot *et al.* in [30] and [31]. It does explicit per pixel motion detection and takes advantage of the qualities of simpler schemes under different conditions: FAV when no motion is detected, spatiotemporal median filtering when the motion is detected as being slow and LAV when fast motion is detected. Thresholds classify the motion. We denote it MA1. The second algorithm we have implemented is weighted vertical temporal deinterlacing (MA2), which uses a smooth weighted transition between temporal (FAV) and vertical (LAV) interpolation instead of a hard switching between schemes as in MA1. The weights used are fixed by SADs as described in detail by Kovacevic *et al.* in [22]. Inspired by the approach of using several deinterlacers in [22], we have implemented a third motion adaptive deinterlacer (MA3) where MA2 is given a third component, which is the edge adaptive deinterlacer, EA from Section II-A above. The EA is weighed along with FAV and LAV by using its SADs. Detailed descriptions of state-of-the-art edge adaptive deinterlacing algorithms can be found in [1] by Almog and and Bruckstein and [2] by Ballester *et al.*

We have also benchmarked our own algorithms against the Faroudja DVP-1010 video processor, which is considered the state-of-the-art deinterlacer in the high-end home cinema market today (see e.g. www.hometheaterhifi.com). The deinterlacing of the Faroudja processor known as DCDi[®] (**D**irection **C**orrelated **D**e-interlacing) is basically a motion adaptive algorithm with a motion detector (MD) classifying the amount of motion on a 5-10 step scale. When there is no motion, a median spatio-temporal filter is applied, which is then turned stepwise off as the amount of motion detected increases and the contribution from a purely spatial filter increases. The spatial filter is edge adaptive (aka direction correlated) detecting $\pm 45^\circ$ edges. If no dominant edge orientation can be detected, LAV is used as fall back. Similar advanced MA deinterlacers found in products today are Sony's X-Algorithm and HQV by Teranex (Silicon Optix).

C. Motion Compensated Deinterlacing

In the intensity interpolation part, motion compensated deinterlacing is basically the same as motion adaptive deinterlacing except for one very important difference: One replaces the local temporal information at the same (x, y) -coordinates with temporal information along the flow trajectories. In theory, this gives you access to correct temporal information in all pixels, and high quality results are no longer obtained only in stationary parts of the frame. In practice, there are some factors preventing you from getting perfect results in all cases. We will discuss the three most important factors. First and least important, flows often point to spatially non-grid point in the discrete grid, thus invoking a need for spatial interpolation to get the $t \pm 1$ intensity values. Second, no optical flow method provides accurate and reliable flows in all cases wherefore we need local spatial interpolation – like in motion adaptive deinterlacers – as a fall back option where temporal information is not sufficiently reliable.

Third, and most important, precise and reliable optical flow estimation is more difficult on interlaced video than on progressive video, and most existing optical flow methods are developed for use on progressive video. Using any of these methods on interlaced image sequences is not straightforward. A typical approach to interlaced optical flow estimation is using a simple non-motion compensated deinterlacer to create an initial progressive sequence to be used in the optical flow estimation as done e.g. in [22]. The use of a simple deinterlacer of course influences the accuracy of the obtained flow due to the interlacing artifacts introduced, most significantly where flow accuracy is of the utmost importance: Regions troubled by the details in motion problem.

In [3], Bellers and de Haan thoroughly test 16 deinterlacers, among them 11 motion compensated methods all using flows generated by block matching motion estimation. The testing shows that the presented motion compensated methods perform better on average than the simple methods included using objective measures, e.g. the mean square error (MSE). Objective measures are, however, not the optimal method for evaluating deinterlacing results as it models very poorly how the human visual system judges the quality of deinterlacing. Unfortunately, subjective testing is only given to a limited extent in [3] and does not clearly show superior performance of the presented motion compensated deinterlacers. It is also a problem that no motion adaptive deinterlacers are included in the test since they are known to produce better results than simple deinterlacers. Still, the book [3] is highly recommended reading and the best motion compensated methods presented are (in further developed versions) used for deinterlacing in television and video products from Philips and Trimedia media processors from NXP (earlier Philips Semiconductors).

In [5], a newer example of a motion compensated deinterlacing algorithm is presented. Video versions of the deinterlacing results, that could earlier be downloaded from the authors web site, did not really show the full potential of MC deinterlacing as e.g. the `Apollo` example still had severe serration artifacts.

The work by Biswas *et al.* in [4] is a recent attempt to come up with a performance analysis for MC deinterlacing,

but the focus in [4] is mainly on the effect of errors in the flow vector field, and quality is measured using MSE on output deinterlaced sequences, but no generic, objective benchmark for deinterlacer performance is provided.

D. Variational Inpainting and Deinterlacing

The idea of doing deinterlacing using variational methods was born when comparing the problems of deinterlacing and inpainting. Inpainting is to recreate regions in an image or film lost due to scratches, blotches etc. Deinterlacing is creating new lines in a video sequence, but considering the lines as having been lost during the recording of the scene, the problems are very similar. A presentation of our variational framework for deinterlacing derived from an inpainting framework [24] by Lauze and Nielsen is given in Section III.

The idea of variational motion compensated inpainting was first published by Cocquerez and Chanas in [11]. We have only found two (later) publications presenting work on deinterlacing similar to ours in terms of methodology. The first is the paper [35] where Tschumperlé and Besserer do edge adaptive deinterlacing by using structure tensor based inpainting (spatial only). The second is the paper [2] by Ballester *et al.*, in which an inpainting based 1D (horizontal) deinterlacer is presented. It uses line alignment to continue isophotes in the plane (classic geometric image inpainting) and in temporal neighboring horizontal lines (thus it becomes MA or, at best, 1D MC). The 1D MC is only local as the devised cost function punishes distance. This method is thus potentially better than our variational inpainting based MA deinterlacer, but it only has the potential to solve the problem of details in motion locally and horizontally.

Variational methods for optical flow computations have been widely known since the pioneering work [16] by Horn and Schunck was published in 1981. The most accurate optical flow algorithm to date is presented by Brox *et al.* in [6] and is also variational (see the survey by Bruhn *et al.* in [8]). Both these methods and most other variational methods can be integrated into our variational framework for deinterlacing. Recent work by Bruhn *et al.* in [7] shows that variational optical flow computations can be done in real-time on a standard PC.

E. Optical Flow Estimation in Interlaced Video

Known optical flow methods for progressive video need to be redesigned before they can be applied to interlaced video. The typical approach is to use a simple non-motion compensated deinterlacer to create an initial progressive sequence, but this kind of simple preprocessing is likely to introduce errors. Choosing a different approach leaves us with a hen-egg problem: In the new lines, should we compute the flow or the intensities first? In 'solving' the hen-egg problem we do in all cases need some optical flow and/or intensity initializations. We have considered four possible solutions listed here in order of estimated effect on visual output quality (all better than the typical approach given above):

- 1) *The 'half a pixel up half a pixel down' approach:* Use original data only and correct for the (zigzag) misalignment when vertically doubling the optical flow field.
- 2) *The original data only approach:* Use given input data only but process the even fields and the odd fields separately as two independent progressive sequences. This is what we actually chose to do in our variational MC deinterlacer (given in Section III). To merge the two intermediate flow fields and get flow vectors for the new lines, a 'fusion-scaling-smoothing'-scheme is needed. A similar approach was used by Chang *et al.* in [10].
- 3) *The idealistic approach:* Develop an interlaced optical flow estimator which is specifically designed to run on the perforated interlaced data volume (interlaced input data at its true vertical position with empty lines). There is an unknown complexity to this task, e.g. in the obvious idea of allowing for flows to go through not yet deinterlaced lines and point at further away existing lines (flow vectors could have lengths one, two or more in time). As in the above case, some post-interpolation of flows vectors for the new lines is needed. This approach is unlikely to be usable in practice. Methods like the 3D recursive search MC deinterlacing given in [3] can be said to be halfway to the idealistic approach by using the interlaced lines in the forward fields ($t + 1$ and $t + 2$) and the fully MC deinterlaced frame in the backward direction ($t - 1$). The algorithm put forth by Delogne *et al.* in [12] could also be considered a (simplified) ideal approach, but their assumption of uniform translational motion over four consecutive fields places it far from our idea of an ideal approach, which requires almost perfect motion modelling.
- 4) *The simultaneous approach:* Develop a motion compensated deinterlacer which simultaneously calculates intensities and optical flows in the new lines as done in the video super resolution and temporal super resolution algorithms given by Keller in [21] (by alternating between updating the two in an iterative approach). Simultaneous deinterlacing does not exist (yet).

Our discussion here is conceptual and rather broad, but a similar discussion with focus on existing schemes (in literature) is given in [25] by Li and Nguyen.

III. THEORY AND ALGORITHMS

A. Variational Framework

The framework we present was first developed for image and image sequence inpainting by Lauze and Nielsen in [24], but was reformulated by Keller in [21] to be used in general for image sequence restoration and enhancement, e.g. video super resolution (increased spatial resolution of each frame in the video) and temporal super resolution (changing the frame rate in a given sequence). A similar energy formulation of the inpainting problems was given by Coquerez *et al.* in [11].

The derivation of the probabilistic (Bayesian) formulation of our framework leading to the energy expressed below is given in the appendix section of this paper. A similar Bayesian

formulation of the deinterlacing problem but leading to a Markov random field solution can be found in [25] by Li and Nguyen.

Formulating the energy of the deinterlacing (and general image sequence restoration/enhancement), we denote the observed interlaced image sequence u_0 , \vec{v} is the optical flow field of the deinterlaced and optionally de-noised progressive image sequence u . The joint energy of \vec{v} and u knowing u_0 is factored as

$$E(u, \vec{v}) = E_0(u, u_0) + E_1(u_s) + E_2(u_s, u_t, \vec{v}) + E_3(\vec{v}) \quad (1)$$

where u_s is the spatial distribution of intensities, and u_t is the temporal distribution of intensities. E_0 is the data term for the image sequence, E_1 the spatial regularizer (prior) on the image sequence, E_3 the regularizer on the optical flow and E_2 acts as both data term for the optical flow and a temporal regularizer on the image sequence. The optimum solution is then sought for in order to construct the progressive image sequence and recover the flow field by minimizing the energy.

Under mild regularity assumptions, a minimizing pair (u, \vec{v}) must satisfy the condition $\nabla E(u, \vec{v}) = 0$ where ∇ is the gradient and the solution expressed by the coupled system of equations

$$\begin{cases} \frac{\partial E}{\partial u}(u, \vec{v}) = 0 \\ \frac{\partial E}{\partial \vec{v}}(u, \vec{v}) = 0. \end{cases} \quad (2)$$

B. Selecting the Terms of the Energy

Selecting the model to use for each of the E_i -terms in (1) is a trade-off between mathematical tractability and maximum obtainable quality – models describing 'real' image sequence content well are mathematically complex. Here is a short discussion on how each term is selected.

$E_1(u_s)$ should punish too large local spatial variations in intensities and also try to model the true spatial intensity distributions in the new pixel positions. Using total variation on the spatial image gradient allows for smooth regions as well as edges and was first introduced by Rudin *et al.* [29].

The last term, $E_3(\vec{v})$, should enforce a reasonable local smoothness of the flow field. It is assumed that the displacement in the image sequence is caused by objects moving so that each pixel does not necessarily have an individual displacement but will have a displacement similar to that of (some of) its neighbors. Just as in $E_1(u_s)$, we need smooth regions – this time of the flows – with edges in the flow field at boundaries between objects moving differently, thus total variation is also the obvious choice for $E_3(\vec{v})$, but here applied to the local 3D spatiotemporal gradient of the flow field.

The term $E_2(u_s, u_t, \vec{v})$ states that the image content information should stay the same along the flow vectors and penalize large discrepancies – this is the most complex of the three terms. This term operates on both the intensities and the flow. It is a representation of the brightness constancy assumption or the optical flow constraint (OFC), see [6] and [16]. Using total variation on the OFC allows smooth flows but also temporal edges, which occur at occlusions to

help solving the occlusion problem. To improve performance under changing brightness (e.g. from changes in lighting), $E_2(u_s, u_t, \vec{v})$ is in [24] and [6] extended to also include the gradient constancy assumption (GCA).

The term $E_0(u, u_0)$ in (1) is the well-known data (or likelihood) term from energy and probability formulations of diffusion. In deinterlacing, it controls the diffusion (de-noising) of the existing lines.

C. Full Energy Formulation

We define the desired progressive image sequence u to be on a spatiotemporal domain denoted Ω with D being the domain of the missing lines in Ω , that is $D \subset \Omega$. The domain of known lines (the interlaced input) is then $\Omega \setminus D$. Using the energy terms as described above in Section III-B, the energy (1) is instantiated as

$$\begin{aligned} E(u, \vec{v}) &= \lambda_0 \int_{\Omega \setminus D} (u - u_0)^2 dx + \lambda_1 \int_{\Omega} \psi(|\nabla u|^2) dx + \\ &\lambda_2 \int_{\Omega} \psi(|\mathcal{L}_{\vec{V}} u|^2) dx + \\ &\lambda_3 \int_{\Omega} (\psi(|\nabla_3 v_1|^2) + \psi(|\nabla_3 v_2|^2)) dx \end{aligned} \quad (3)$$

where ∇ is the spatial gradient operator, ∇_3 is the local spatiotemporal gradient, the λ_i 's are weighing constants and v_1 and v_2 are the x - and y -components of the flow field, i.e. $\vec{v} = (v_1, v_2)^T$, and $\vec{V} = (\vec{v}^T, 1)^T$. Since the function $|\cdot|$ is not differentiable at the origin, we replace it by the approximation $\psi(s^2) = \sqrt{s^2 + \varepsilon^2}$, with $\varepsilon = 0.01$ in our experiments. The $\mathcal{L}_{\vec{V}} u$ -term is the \vec{V} -directional derivatives of u , also known as the Lie-derivative (see for instance the book on Riemannian geometry by Gallot *et al.* [14]). To set the notation straight, we have

$$\mathcal{L}_{\vec{V}} u = \frac{\partial u}{\partial \vec{V}} = \nabla u \cdot \vec{v} + u_t = \vec{V}^T \nabla_3 u \approx u(\mathbf{x}, t) - u(\mathbf{x} + \vec{v}, t + 1) \quad (4)$$

given that

$$\mathcal{L}_{\vec{V}} f(x) = \lim_{\epsilon \rightarrow 0} (f(x + \epsilon \vec{V}) - f(x)) / \epsilon = \nabla f(x) \cdot \vec{V}$$

and $\nabla_3 u$ is again the spatiotemporal gradient of u . The right hand side of the approximation in (4) is the brightness constancy assumption and the other terms in (4) are all notions for the linearized version of the brightness constancy assumption, that is the optical flow constraint. The energy in (3) is the same as the energy given for inpainting in Chapter 9 of Lauze's thesis [23] and in [11] by Cocquerez *et al.*: The difference lies in the definition of the domain of missing data (D in our notation).

D. Variational Motion Adaptive Deinterlacing

The algorithm we give here is the best of two variational MA deinterlacers presented in [19], in which we neither introduced the overall variational framework nor focused on whether to use a local 3D prior (on the intensities) or to split the prior into a 2D spatial prior and a 1D temporal prior.

In order to get a variational motion adaptive deinterlacer from the presented framework, we assume the flow to be zero, $\vec{v} = 0$, and thus only have to minimize

$$\frac{\partial E}{\partial u}(u, \vec{v} = 0) = \frac{\partial E(u)}{\partial u} = 0$$

as the term E_3 in (1) disappears. Setting $\vec{v} = 0$ (or $\vec{v} \approx 0$) is the basic assumption of all MA deinterlacing and enables the use of local temporal data when the assumption is true, but requires that the task of optical flow estimation (motion estimation) is replaced by motion detection. Using total variation as the distribution for the term E_2 in (1), which is now only a temporal prior on the intensities, we do not need explicit motion detection. Total variation in E_2 allows for temporal edges and thus gives us implicit motion detection: motion creates a local temporal edge that our methods will not diffuse across, and thus we have implicit motion detection and adaptation.

The data term on the intensities, E_0 in (1) is chosen to be a hard constraint as it simply leaves the input lines (the domain Ω/D) unchanged, which is the standard approach in the world of deinterlacing but not in the world of variational approaches. In e.g. variational de-noising, the data term primarily acts as a reaction to the diffusion action of the remaining terms controlling the degree of de-noising. Our data term is $u = u_0|_{\Omega/D}$ and plugging in the total variation terms, the general energy formulation can be rewritten to

$$E(u) = \int_D (\alpha_s \psi(|\nabla u|^2) + \alpha_t \psi(|\partial_t u|^2)) dx, \quad u = u_0|_{\Omega/D} \quad (5)$$

where ∇ is again the spatial image gradient, ∂_t the local temporal gradient and the constants α_s and α_t the weights of the spatial and temporal priors respectively. The spatial gradient term is thus a variation of the 'true' total variation term introduced by Rudin *et al.* in [29]. In order to do the actual minimization, we use the fact that $\psi'(s)/s = 1/\psi(s)$ and from using calculus of variation, we obtain the Euler-Lagrange equation

$$-\alpha_s \text{div}_2 \left(\frac{\nabla u}{\psi(|\nabla u|^2)} \right) - \alpha_t \partial_t \left(\frac{\partial_t u}{\psi(|\partial_t u|^2)} \right) = 0, \quad u = u_0|_{\Omega/D} \quad (6)$$

where div_2 is the divergence operator. We have chosen to solve this system using gradient descent, and the gradient descent equation corresponding to (5) is

$$\partial_\tau u = \alpha_s \text{div}_2 \left(\frac{\nabla u}{\psi(|\nabla u|^2)} \right) + \alpha_t \partial_t \left(\frac{\partial_t u}{\psi(|\partial_t u|^2)} \right), \quad u = u_0|_{\Omega/D} \quad (7)$$

where τ is the evolution time of this iterative solver. The numerics of the solution is given in [21]. In (6) and (7) it can be seen clearly that a large gradient (temporally or spatially) will shut down the diffusion contribution of its own term, thus preserving edges spatially as well as temporally, the latter being the implicit motion detection and adaption of variational motion adaptive deinterlacing (VMA).

E. Variational Motion Compensated Deinterlacing

We turn now to the full motion compensated deinterlacing done by splitting the energy in (3) into $E(u)$ and $E(\vec{v})$ and

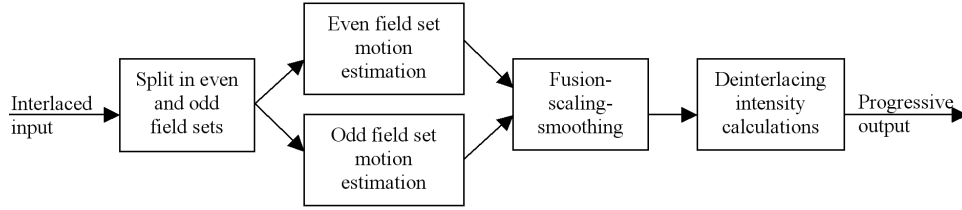


Fig. 4. Flow chart of our algorithm. First we split the even and odd fields apart, estimate the motion on each by minimizing the energy in (8), then we combine the two flows by minimizing the energy in (11) to finally compute the intensity output by minimizing (12).

minimizing them according to (2). A flow chart of the full algorithm is given in Fig. 4.

1) *Algorithm Part I, Variational Optical Flow Estimation on Interlaced Video:* Solving the system according to (2), the optical flow energy $E(\vec{v})$ loses the E_0 - and E_1 -terms and is

$$E(\vec{v}) = \int_{\Omega} \psi(|\mathcal{L}_{\vec{v}}u|^2)dx + \lambda_3 \int_{\Omega} (\psi(|\nabla_3 v_1|^2) + \psi(|\nabla_3 v_2|^2))dx$$

leaving out λ_2 for obvious reasons. To get the most accurate flow field possible, we wish to use the variational optical flow algorithm proposed by Brox *et al.* in [6] as it was reported to be the most accurate in the optical flow survey in [8]). To do so, we have to add the gradient constancy assumption to the flow data term E_2 as it was suggested in Section III-B. The GCA was also added to the intensity energy of the inpainting in [24] to help data transport into the large unknown regions, but with an accurate flow field on narrow regions of missing data (the missing lines) we leave it out of $E(u)$ in deinterlacing, as it will make the algorithm much more complex (see [24]). Inpainting results presented by Lauze in [23] both with and without the gradient constancy assumption in $E(u)$ indicate that for small regions of missing data, the difference in output quality between the two is likely to be small.

The flow energy we will minimize is then

$$E(\vec{v}) = \int_{\Omega} \psi\left(|\mathcal{L}_{\vec{v}}u|^2 + \gamma|\mathcal{L}_{\vec{v}}\nabla u|^2\right)dx + \lambda_3 \int_{\Omega} (\psi(|\nabla_3 v_1|^2) + \psi(|\nabla_3 v_2|^2))dx. \quad (8)$$

where γ is a weighing constant. As with the $\mathcal{L}_{\vec{v}}u$ in (1) we have that

$$\mathcal{L}_{\vec{v}}\nabla u = \frac{\partial \nabla u}{\partial \vec{V}} \approx \nabla u(\mathbf{x}, t) - \nabla u(\mathbf{x} + \vec{v}, t + 1) \quad (9)$$

where the right hand side of the approximation is the gradient constancy assumption and the two other terms are its linearized version.

The resulting Euler-Lagrange equation to be discretized is

$$\begin{aligned} \frac{\partial E}{\partial \vec{V}} &= 2\psi' \left(|\mathcal{L}_{\vec{v}}u|^2 + \gamma|\mathcal{L}_{\vec{v}}\nabla u|^2 \right) \cdot \\ &\quad \left[(\mathcal{L}_{\vec{v}}u) \cdot \nabla u(x + \vec{V}) + \gamma \mathcal{H}(u(x + \vec{V}))(\mathcal{L}_{\vec{v}}\nabla u) \right] \\ &\quad + \lambda_3 \left(\begin{array}{c} \text{div}_3 \left(\frac{\nabla_3 v_1}{\psi(|\nabla_3 v_1|^2)} \right) \\ \text{div}_3 \left(\frac{\nabla_3 v_2}{\psi(|\nabla_3 v_2|^2)} \right) \end{array} \right) = 0 \end{aligned} \quad (10)$$

where div_3 is the 3D local spatiotemporal divergence operator, and \mathcal{H} is the spatial hessian. As discussed in Section II-E, we use 'the original data only' approach to calculate the optical flow. First, we extract the odd field sequence, i.e. the sequence of the known lines for odd fields, and 'compress' it vertically by removing the unknown lines. In the same way, we produce the even field sequence (yielding a field distance of twice of the input field distance). We then run this algorithm:

- 1) Compute the flow \vec{v}_o for the odd field sequence and the flow \vec{v}_e for the even field sequence applying the implementation of (10).
- 2) Join \vec{v}_o and \vec{v}_e in order to produce a candidate flow field for the full progressive sequence using *fusion-scaling-smoothing*.
- 3) Repeat 1 and 2 on the time-reversed sequences to calculate backward flows as well.

To join the two flow fields \vec{v}_o and \vec{v}_e both containing flows from fields at time t to time $t+2$, we have to *fuse* – interlace them in time – then we have to *scale* twice, first the vertical component of the flow is doubled to accommodate the full frame height, then the flow vectors are halved in length to be from time t to time $t+1$. Finally, we have to *smooth* the new flow field to get the two parts to be consistent with each other and to fill in good flow estimates in the empty lines of the full progressive sequence. This all ends up being *fusion-scaling-smoothing* and can be expressed mathematically as the minimization of the energy functional

$$E(\vec{v}) = \lambda \int_{\Omega/D} (\vec{v} - \vec{v}_{o/e})^2 dx + \int_{\Omega} (\psi(|\nabla_3 v_1|^2) + \psi(|\nabla_3 v_2|^2)) dx. \quad (11)$$

Composing $\vec{v}_{o/e}$ in (11) is the fusion-scaling part, and the full minimization is the smoothing. The function ψ is defined as before. The fusion is done by just interleaving the two flow fields, and the initial flow values in the empty lines are found by line averaging. The scaling is elementary as we assume

linearity of the flow from time t to time $t + 2$ and just halve the vectors. It is then up to the smoothing step to complete the process of producing one optimally merged and consistent flow field. For the Euler-Lagrange equation of (11), the first term is – after composing $\vec{v}_{o/e}$ – the same as E_0 in (12) and just acting on the flow now. The second term is the same as E_3 in (8), which ensures coherence and gives a feeling of integration between the two steps of the full flow calculation. In our testing, we skipped the de-noising of the already computed flow, that is we set $\lambda = 0$, as we had high confidence in the already computed flows. Solvers are discussed below in Section III-F.

2) *Algorithm Part II, Variational Motion Compensated Deinterlacing*: The energy we minimize for the intensities, $E(u)$, when the forward and backward flows have been calculated is

$$E(u) = \lambda_0 \int_{\Omega/D} (u - u_0)^2 dx + \lambda_1 \int_{\Omega} \psi(|\nabla u|^2) dx + \lambda_2 \int_{\Omega} \psi(|\mathcal{L}_{\vec{v}} u|^2) dx. \quad (12)$$

The Euler Lagrange equation derived from this energy still using $\psi(s^2) = \sqrt{s^2 + \varepsilon^2}$ is

$$\frac{\partial E}{\partial u} = \lambda_0 \chi(u - u_0) - \lambda_1 \operatorname{div}_2 \left(\psi'(|\nabla u|^2) \nabla u \right) - \lambda_2 \operatorname{div}_3 \left(\psi'(|\mathcal{L}_{\vec{v}} u|^2) (\mathcal{L}_{\vec{v}} u) \vec{v} \right) = 0 \quad (13)$$

where χ is the characteristic function taking on the value 1 in Ω/D (known lines) and 0 in D (new lines), and the discretization of the last term suggested by the approximation given in (4). The weight λ_0 can be set to zero to do pure deinterlacing without de-noising, leaving all original pixels (u_0) untouched in the output similar to what almost all other deinterlacers do. Setting $\lambda_0 = 1$ includes de-noising (any other value of λ_0 is just similar to scaling λ_1 and λ_2 with the inverse of the same value). As argued by Bellers and de Haan in [3], de-noising can help remove some temporal flickering noise. We agree on this, but wish to emphasize the importance of carefully controlling the de-noising to avoid over-smoothing and loss of valuable details.

Now we know how to obtain backward and forward flow and can use them in solving (13) to minimize the intensity energy $E(u)$.

The algorithm for the intensity calculations is very similar to the one used for motion compensated sequence inpainting by Cocquerez and Chanas in [11] and by Lauze in [23]. The optical flow algorithm we use is also fairly well-known from [23] and [6], so the part of the overall scheme predicted most likely to cause trouble is the full task of doing optical flow estimation on interlaced sequences including the fusion-scaling-smoothing algorithm.

F. Numerical Solvers

To solve all three Euler-Lagrange equations in variational MC deinterlacing – (10), (13) and the one derived from the fusion-scaling-smoothing energy in (11) – we have used a fixed point scheme with Gauss-Seidel solvers. A detailed

implementation description for those interested is given in [21]. A fixed point solver sets the nonlinear part of a system to be constant for a number of iterations to enable the use of a linear solver on the now linear system. This speeds up convergence.

A opposed to other deinterlacers that use one-step solutions, the fact that we use iterative solvers in combination with a temporal aperture spanning both backwards and forwards in time is an advantage as it gives us an increased spatial and temporal coherence when calculating either intensities or flow fields. By iterating, information from further away than just the nearest neighbors (filter taps) are propagated to the current pixel – if it is coherent with our modelling.

In the inpainting algorithm in [24], the flow and intensity energies are minimized simultaneously in a multiresolution setting (on each level in the multiresolution pyramid first the flow, then the intensity energy is minimized). In inpainting, there is larger regions of intensity data missing but in deinterlacing, all missing regions (the empty lines) will be gone as soon as we go a scale factor two up in the pyramid. Thus multiresolution is only used in the flow energy minimization. In a possible future deinterlacer, one could introduce simultaneousness by alternating between intensity and flow energy updates at the finest pyramid level (as with video super resolution in [21]).

IV. EXPERIMENTS

We present our results in two main parts. In the first part, we look at results of variational motion adaptive deinterlacing and benchmark it against other deinterlacers. We then compare our variational motion compensated deinterlacer with the best methods from the first part of the test to show the superiority of motion compensated deinterlacing – it solves the problem of details in motion, which will be the focus problem of the tests. But first, we give some information on our test setup.

A. Quality Measurements

We use both objective and subjective quality evaluation of the deinterlacers in test, but give focus to the subjective evaluation. The example given in Fig. 3 of the sequence *Grille* suffering from the details in motion problem illustrates why: In spite of the subjectively large difference in quality between variational motion adaptive and variational motion compensated deinterlacing on *Grille*, the objective difference measured using the mean square error (MSE) is insignificant (and in favor of the MA deinterlacer!) as only a small percentage of the pixels in each frame is affected by the problems caused by the moving grille. The MSE and most other objective measures are global. Thus, the local problems, which are very disturbing to the human observer, are not given significant weight in the measure. Objective measure like the MSE could be applied more locally, but then the output MSEs become a large data set itself and the evaluation of it tedious and possibly also subjective (when sorting and selecting what to give as results).

The goal of our evaluation is to measure how pleased the human visual system is with our deinterlacing results and thus, subjective evaluation is the best measure, which is also

acknowledged in industry, e.g. at the consumer electronics producer Bang & Olufsen [34] and Philips [28]. In industry, one follows specific norms to validate subjective testing, e.g. the ITU recommendation [17]. In this work, we have used a test panel of 4-7 persons and rely to a large extent on our expert knowledge and scrutinizing inspections of the results in the subjective evaluation. From the results presented in this section, we hope the reader will agree on our conclusions.

The exact form of the MSE used for objective evaluation is

$$\text{MSE} = \frac{1}{N} \sum_D (u - u_{gt})^2 \quad (14)$$

where u_{gt} is the ground truth, and we sum over the missing lines D only, making N equal to half the number of pixels in the sequence. The MSE cannot be used on image sequences that only exist as interlaced video but only on artificial interlaced video which is produced by removing every other line of a progressive sequence, the progressive sequence providing the ground truth. Some measures enabling objective evaluation of deinterlacing of true interlaced material are given in [3] but rely heavily on the optical flow computed for a sequence. Further discussion on objective vs. subjective evaluation can be found e.g. in [3].

B. Test Material and Online Video Results

In testing, we used both true interlaced video and interlaced material made from removing lines in progressive video. We have chosen sequences challenging the deinterlacers, and several sequences are plagued by the problem of details in motion. All tests were conducted on the 8 bit luminance channel. Online video results of running variational motion adaptive and compensated deinterlacing on the test sequences *Building*, *Copenhagen Pan*, *Truck* and *Credits* are given at <http://image.diku.dk/sunbio/VDI/DI.zip> [20] along with their progressive/interlaced originals. For both motion adaptive and motion compensated variational deinterlacing, line averaging was used as initialization of the new lines.

C. Simple and Motion Adaptive Results

The ten simple and motion adaptive deinterlacers mentioned in Section II have been thoroughly tested along with our variational motion adaptive deinterlacer (VMA) and as reported in [19], the VMA was judged the best deinterlacer in terms of output quality. In this paper, we give additional test results for these eleven deinterlacers to show the advantages of VMA but also the limitations of simple and motion adaptive deinterlacers when faced with the problem of details in motion. We have added the Faroudja DVP-1010 video processor with DCDi[®] deinterlacing. It prides itself of being very good at edge adaptation and thus should be the best motion and edge adaptive deinterlacer there is. It is a 'closed box' system and due to Macrovision and HDCP copy protections, we have not been able to do any objective evaluation of its deinterlacing performance, and all subjective evaluations have to be done in real-time. We have conducted A/B real time comparison tests with the Faroudja on different high quality displays (a plasma, several LCDs and a DLP projector) by streaming the

TABLE I
OBJECTIVE RESULTS FOR DEINTERLACING GIVEN AS MEAN SQUARE ERRORS (MSE) AND SUBJECTIVE RANKINGS IN PARENTHESES.

Scheme	Grille	Building	Credits
Line Doubling (LDB)	151.7 (13)	92.8 (7)	516.7 (6)
Line Averaging (LAV)	63.7 (8)	27.4 (4)	159.8 (4)
Field Insertion (FI)	247.2 (4)	414.6 (7)	1639.0 (12)
Field Averaging (FAV)	162.4 (4)	253.7 (7)	908.5 (12)
Vertical Temporal (VT)	99.0 (4)	98.5 (7)	405.6 (7)
Median (Med)	127.8 (8)	89.2 (7)	509.8 (7)
Motion Adaptive 1 (MA1)	63.1 (8)	27.9 (6)	158.9 (4)
Motion Adaptive 2 (MA2)	110.5 (4)	83.3 (7)	337.5 (7)
Edge Adaptive (EA)	87.1 (8)	58.5 (2)	511.2 (7)
Motion Adaptive 3 (MA3)	95.7 (8)	61.2 (7)	342.5 (7)
Faroudja DVP-1010	– (3)	– (4)	– (3)
Var. Motion Adap. (VMA)	70.2 (2)	35.8 (3)	165.7 (2)
Var. Motion Comp. (VMC)	80.3 (1)	44.7 (1)	103.7 (1)

other results recorded on DVD through the Faroudja to make the deinterlacing the only difference in the evaluation.

The parameter settings for VMA are the same as in [19], the main point being that the temporal to spatial weighing is $\alpha_s = \alpha_t = 1$, which makes the motion detection completely implicit.

Objective results and subjective rankings for the sequences *Grille* (Fig. 3), *Building* (Fig. 5) and *Credits* (Fig. 6) are given in Table I (VMC results will be discussed in Section IV-D). As can be seen from the table, there is no clear correlation between objective and subjective rankings, which was also the case in [19]. Taking into account that neighbors in the subjective ranking of a sequence can be both marginally and very different from each other in terms of visual quality, the correlation becomes even more difficult to find.

On *Grille*, LDB is subjectively the worst. The remaining methods are not quite as bad with our VMA being the least bad of all. Even though EA scores significantly worse MSEs than the four deinterlacers in front of it, it is subjectively evaluated to be a lot better than all other deinterlacers in test (except for VMC) as it should be clear from inspecting Fig. 5. Besides the good subjective result on *Building*, EA deinterlacing in general performs horribly as it mostly creates serious salt and pepper like artifacts.

On *Credits*, variational MA is ranked fourth by MSE but subjectively a clear number one (except VMC). The VMA result is shown in Fig. 6 along with the original progressive sequence (and the VMC result). Judging overall from the three sequences in Table I, the Faroudja DVP-1010 is ranked just below VMA in terms of subjective quality, and the same is the case when extending the evaluation to a larger set of sequences (10+).

In Fig. 7 of the sequence *Copenhagen Pan*, which is true interlaced so that no MSE can be measured, it is clearly seen how our variational MA deinterlacer outperforms the two other motion adaptive methods shown, MA1 and MA2. In our earlier tests in [19], MA1 and MA2 are the two methods that come closest to our variational MA deinterlacer in both subjective and objective quality, but they are still significantly worse. The bad performance of MA3 is most likely caused

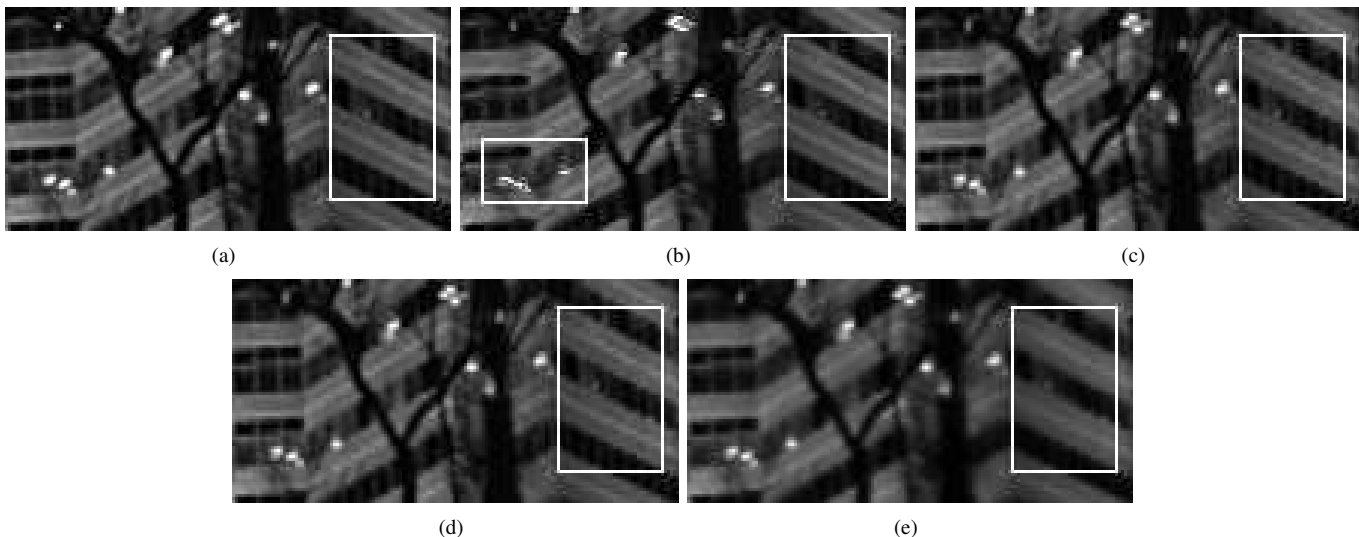


Fig. 5. The sequence *Building* with tilting camera motion, 75×150 pixels cutout. (a) Original progressive, deinterlaced with (b) EA, (c) LAV, (d) variational MA, and (e) variational MC, which is very close to the original (a bit smoother due to de-noising). The white rectangle marks a region with very noticeable differences. EA is also close to the original but the small rectangle marks a region with serious artifacts, and there are also very noticeable salt and pepper artifacts during playback.

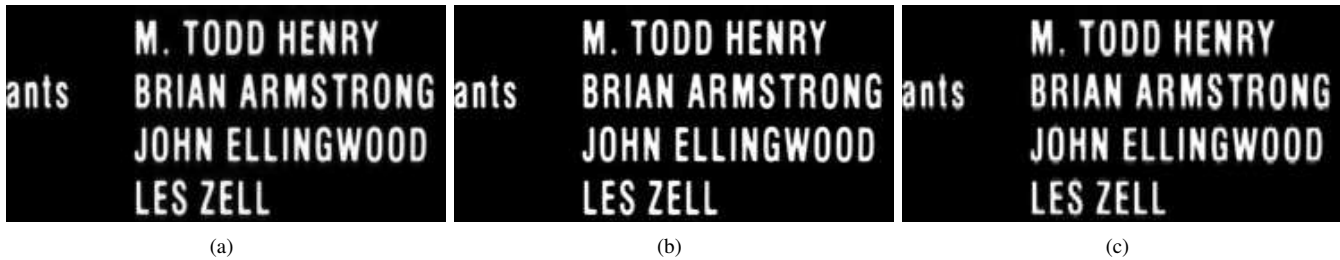


Fig. 6. The sequence *Credits* has rolling end titles and the output of our variational MC deinterlacer in (a) is practically indistinguishable from the progressive original in (b) when viewed as video but appears a bit smoother as still (due to de-noising). The result of variational MA deinterlacing shown in (c) is also the second best performing deinterlacer on *Credits*, and it does look quite good as still but has severe flickering when played.

by our implementation of edge adaption (part of MA3, see Section II-B) which is not as robust as it could be.

None of the simple or motion adaptive deinterlacers produce outputs that are even close to the progressive original (ground truth) on sequences containing areas with details in motion.

Our VMA deinterlacer is on average best, and the result shown in Fig. 7(c) is one of the best, but still we do not show the full frame of *Copenhagen Pan*, because there are areas with details in motion in the sequence where all MA deinterlacers perform poorly.

D. Variational Motion Compensated Deinterlacing Results

The general limitation of MA deinterlacers – the inability to solve the problem of details in motion – is why (variational) motion compensated deinterlacing is so interesting in spite of its increased complexity.

1) *Parameter Testing and Intermediate Flow Results*: There are more parameters to tune in the VMC than in the VMA deinterlacer. In Table II, an overview is given of what we found to be the optimal setting after extensive testing.

For the motion estimation step, we used an 80 level multiresolution pyramid with a scale factor of 1.04 between the levels, which we found to be optimal for our PAL resolution

TABLE II
RECOMMENDED SETTINGS FOR VARIATIONAL MOTION COMPENSATED DEINTERLACING.

	fixed point iterations	relaxation iterations	weights
Optical flow estimation, Eq. (8)	5	20	$\gamma = 100$ $\lambda_3 = 70$
Fusion-scaling-smoothing, Eq. (11)	10	100	$\lambda = 1$
Deinterlacing, Eq. (12)	5	50	$\lambda_0 = 1$ $\lambda_1 = 0.1$ $\lambda_2 = 5$

test material. Lowering the number of levels does affect the flow mildly, but any differences disappear almost completely when the flow is further processed in the fusion-scaling-smoothing step. We suggest setting the weight of the flow field regularization to $\lambda_3 = 70$. Lowering λ_3 will over-segment the flow, and higher values will provide an unnecessary smoothing of the flow as some regularization inevitable occurs in the fusion-scaling-smoothing step. Setting the weight of the gradient constancy assumption to $\gamma = 100$ is a good choice. Higher values will possibly provide additional details to the flow, but these details as well as any over-segmentation are

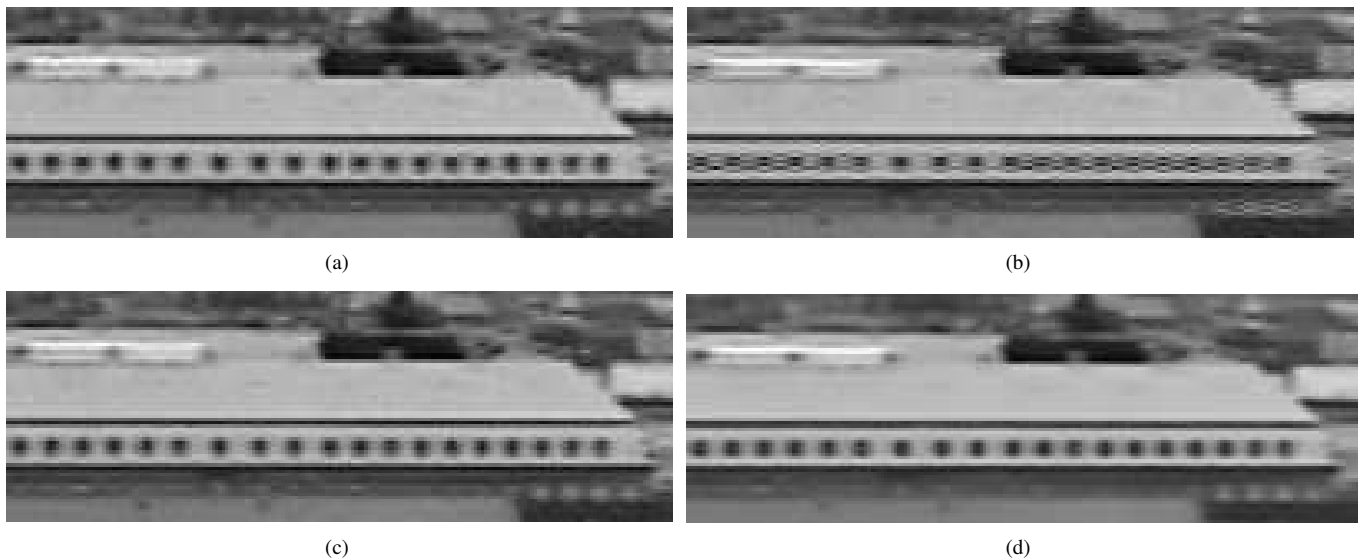


Fig. 7. Cutout of *Copenhagen Pan* deinterlaced with (a) MA1, (b) MA2, (c) our variational MA, and (d) our variational MC. The motion in *Copenhagen Pan* is a app. 5 pixel/frame camera pan to the left. MA1 gives median 'sparks' around the windows, MA2 serrates heavily, variational MA is almost perfect and variational MC just a bit better, mainly due to de-noising.

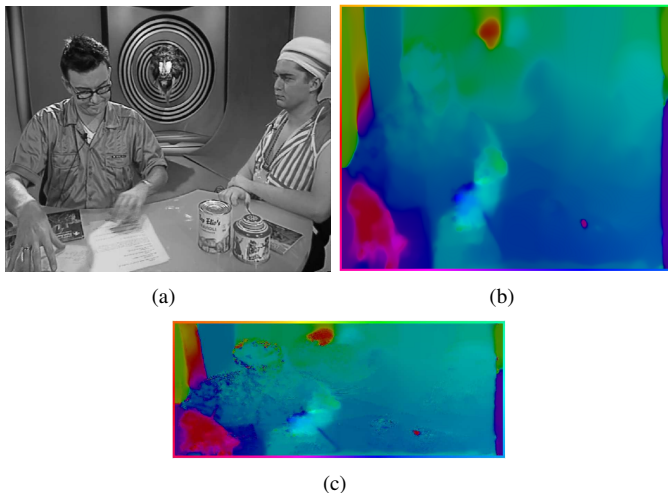


Fig. 8. Typical flow results. The sequence (a) is a zoom-in / left pan. (c) shows how the flow is typically over-segmented before fusion-scaling-smoothing. (b) shows how the fusion-scaling-smoothing not only upscales the flow, but also homogenizes it, removing incorrect flows (the over-segmentation) in the process. The border shows the flow directions coded by the hue, while the intensity codes the magnitude of the flow normalized in $[0.5,1]$.

smoothed out in the fusion-scaling-smoothing step.

In the fusion-scaling-smoothing setting, the flow field regularization to $\lambda = 1$ (and up to 3) is appropriate: Higher values will produce a smoother flow, which can both help get rid of over-segmentation but also remove fine details in the flow. We ran 10 outer fixed point and 100 inner iterations. In Fig. 8, it can be seen how typical flow outputs look before and after fusion-scaling-smoothing.

In the deinterlacing step optimizing the solution of (13), setting the data term weight $\lambda_0 = 0$ and thus shutting off de-noising in original pixels results in more temporal flicker in moving detailed regions. The temporal regularization weight λ_2 should be at least 1, but also values up to 10 yielded good

results, although they do introduce some smoothing in moving regions. From our setting, $\lambda_2 = 5$ is optimal. We recommend and have generally used $\lambda_2 = 5$ with spatial regularization weight $\lambda_1 = 0.1$. The chosen setting of λ_1 minimizes temporal flicker in the output.

Generally our three-step algorithm has – within limits – proven to be quite robust to changes in parameter settings, but further parameter testing might still improve performance. Especially, the flow calculations can give very detailed (or controlled poorly, over-segmented) flows, but they are smoothed out by the fusion-scaling-smoothing. It would be optimal to make sure that the most accurate and detailed flow possible reaches the intensity calculations. Flow estimations on interlaced video is, as discussed earlier in this paper, a difficult problem and work on improved optical flow estimation for deinterlacing will most likely be the source of the biggest improvements of (variational) MC deinterlacing.

2) *Running Times*: We have not focused on optimizing our code for speed (yet). The code is written in C++ using the CImg library (<http://cimg.sourceforge.net/>) interfacing with Matlab. Using the settings given in Table II, we are a factor 1500 from realtime at 50 frames per second on a standard PC. The flow calculations take up 84% of the total computation time, but we know that simpler variational flow algorithms run in real-time on standard PCs (although on web camera resolution video) [7].

3) *Results*: Four experts and two laymen have judged the output quality of our variational motion compensated deinterlacer (VMC) in an A/B comparison with the Faroudja DVP-1010. On seven different test sequences all suffering from the details in motion problem, our algorithm clearly outperformed the Faroudja DCDi[®] according to all six test persons. The output from our variational MC deinterlacer on each of the three test sequences having a progressive original was judged almost indistinguishable from their original (again in an A/B comparison). As mentioned in Section IV-C, we are unable to

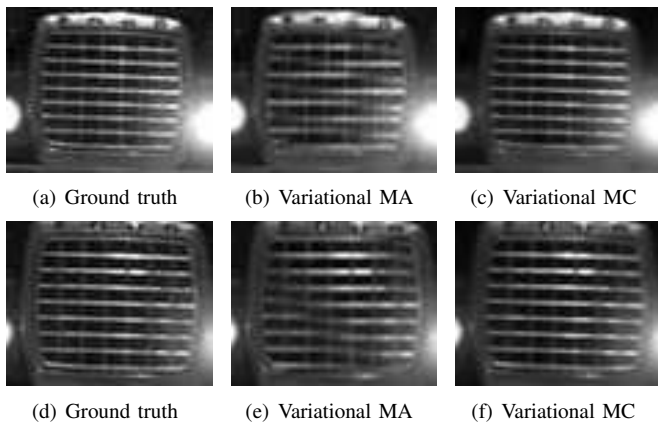


Fig. 9. Deinterlacing of *Truck* shown in Fig. 2. (a)-(c) frame 5, (d)-(f) frame 17. The results of variational motion compensated deinterlacing in (c) and (f) is almost indistinguishable from the progressive ground truth in (a) and (d) while the best of the rest, variational motion adaptive deinterlacing in (b) and (e), creates some horrible artifacts.

produce stills or record video of the output from the Faroudja processor due to copy protection.

None of the eleven other simple and motion adaptive deinterlacers come close to the quality obtained with variational motion compensated deinterlacing when there is details in motion in the test sequence.

On *Grille*, VMC is only ranked fourth by MSE (Table I) but as seen in Fig. 3 is far better than number two subjectively, our VMA. On *Credits*, variational MC deinterlacing is subjectively a lot better than any other deinterlacer in test, and although it is ranked first by MSE as well, the difference down to number two does not justify the difference in MSE (even though the gap in MSEs is relatively large). Fig. 5 shows how variational MC deinterlacing also produces the best result on *Building*, although slightly smoothed it is clearly better than the second best, EA deinterlacing, as observed during playback. Although VMC is best subjectively on *Building*, it is only ranked fourth by MSE. The sequence *Credits* in Fig. 6 has rolling titles, a type of content causing most deinterlacers to perform very bad, but the output of our variational MC deinterlacer is practically indistinguishable from the progressive original during playback.

On the sequence *Truck*, which was used to illustrate the details in motion problem in Fig. 2, we are able to (re)create the correct structure of the front grille whereas all other deinterlacers we have tested, MA and simple, fail at this and create some dark, diagonal lines across the front grille. Comparing the result from our MC deinterlacer with the progressive ground truth in Fig. 9 shows hardly any differences between the two.

On the *Copenhagen Pan* cutout in Fig. 7, the variational MC deinterlacer is not much better than variational MA except for a bit of de-noising. But all the MA deinterlacers we have tested including the Faroudja produce serious flicker on other buildings in the sequence. The Faroudja also produces quite a lot of serration on the roof windows of the castle (the cutout in Fig. 7) thus performing clearly worse than variational MA. The variational MC deinterlaced output has no flicker or any

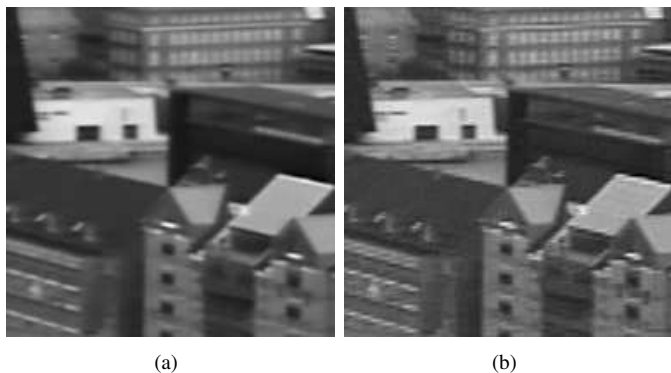


Fig. 10. Another cutout of *Copenhagen Pan* (same frame as in Fig. 7) showing the superiority of variational MC deinterlacing in (a) over variational MA deinterlacing in (b) as it manages to solve the problem of details in motion. Notice the stair step effect on almost horizontal and diagonal lines in the MA output, which also causes severe flickering during playback.

other artifacts to annoy the viewer, and Figures 10(a) and 10(b) show how much better it is on another part of *Copenhagen Pan* compared to the best MA deinterlacer, our VMA.

So far, we have presented only good results from our variational motion compensated deinterlacer, but we also encountered one bad result. On the sequence *Keyboard* with very complex motion (close-up of fingers typing fast on a keyboard using a hand held camera), the motion was not recovered correctly and results in serration clearly observed when inspecting the frames as stills. Of the six persons in the panel, only one (expert) observed the serration in *Keyboard* during playback. The serration in *Keyboard* shows that we could improve robustness against unreliable flows in our method, but improved flow calculations on interlaced video would probably help even more (robustness fixes a problem whereas more reliable and precise flows better the overall quality). The temporal coherence gained by calculating flows on sequences as a set and not only as frame pairs is a major strength, but on dynamic motion with large accelerations, it becomes a weakness.

The last step in proving the quality of our VMC deinterlacer would be testing it against other MC deinterlacers. A survey of motion compensated deinterlacers like the optical flow survey in [8] would be highly appreciated, but the lack of a common set of test data and a good quality measure – like the *Yosemite* sequence and the angular error measure used in the optical flow community – is a problem. Doing a comparison would require a gathering of motion compensated deinterlacing algorithms. Recently, Gerard de Haan kindly granted us access to test an implementation of the adaptive recursive (AR) MC deinterlacer (the best one tested in [3]), but it is only a basic version used for teaching purposes. Alternatively, we could implement the AR MC and other MC deinterlacers ourselves but getting the implementations right to make the test fair is not straightforward: Most quality MC deinterlacers are highly complex algorithms. The implementations of the AR MC deinterlacer in the comparison tests of MC deinterlacers in [25] and [2] are not necessarily correct and thus the tests are not necessarily fair. In [2], the block matching ME uses 21×21 blocks (to get a computational complexity

equal to that of the algorithm of Ballester *et al.*) whereas it reported by Bellers and de Haan [3] to use 8×8 or even 2×2 for final corrections. In [25], very little details are given on versions and implementations of the 'other' MC algorithms in test, and also two different ME algorithms are used for the MC algorithms tested (even making the implementation of Li and Nguyen's own algorithm difficult to others).

Thus, we strongly encourage any work on a fair MC deinterlacing survey.

V. CONCLUSION AND FUTURE WORK

We have shown that using variational methods for deinterlacing is advantageous. The variational motion adaptive deinterlacer yields the highest output quality among simple and MA deinterlacers just outperforming the Faroudja DVP-1010 video processor with DCDi[®] considered to be the state-of-the-art in home cinema deinterlacing. But no MA deinterlacer, including our variational MA deinterlacing, solves the problem of details in motion. This problem is solved by our variational motion compensated deinterlacer, which yields high quality results close to the ground truth even in regions plagued by the problem of details in motion. To fully prove its quality, a benchmarking against other MC deinterlacers is needed. Our MC deinterlacer still has flaws caused by unprecise and unreliable flows. The flow problem is mainly due to the problems of working on interlaced video, but there are also limitations of flow algorithms per se (even though we used the best there is, see [8]). Improvements are to be expected if the flow and intensity calculations are integrated in a simultaneous scheme. In terms of running time, we are far from realtime in our current software version, but the realtime computations of variational optical flows presented by Bruhn *et al.* in [7] show that realtime variational MA and MC deinterlacing is close at hand.

APPENDIX BAYESIAN FRAMEWORK

The inference machine we have used to define a framework for the general image sequence inpainting and upscaling problem to impose some regularization on the ill-posedness of the problem is the Bayesian inference

$$p(u|u_0) = \frac{p(u_0|u)p(u)}{p(u_0)} \quad (15)$$

where $p(u|u_0)$ is the a posteriori, $p(u_0|u)$ is the likelihood, $p(u)$ the prior on (or simply the probability of) u and $p(u_0)$ is the evidence of u_0 .

When it comes to the need of (re)creating information never sampled or missing due to a degradation of the image sequence data, de-noising, inpainting, deinterlacing and other upscaling are very similar problems, and as we will show here, we can define a joint framework of the problems using Bayes inference. In accordance with the definitions used elsewhere in this paper, u in (15) is a desired output image (de-noised, upsampled or inpainted), and u_0 is the input image (e.g. noisy, of low resolution, interlaced or scratched). For clarity, we start by looking at single images. The likelihood $p(u_0|u)$ is also

known as the data (fidelity) term in variational formulations as it tells us how the input relates to the output, e.g. how to place the existing lines of an interlaced signal u in the deinterlaced, progressive output u_0 . $p(u)$ is the spatial prior on u , which decides what image content we allow in our images, how we model images. Maximizing the a posteriori, which is equal to maximizing the right hand side in (15), optimizes the probability of u knowing u_0 . The process is called maximum a posteriori (MAP) and to do MAP, we do not need the evidence $p(u_0)$, which is just a normalization factor (the probability of the input is fixed since it is known), thus we have

$$p(u|u_0) \propto p(u_0|u)p(u). \quad (16)$$

The locus of missing data D which is always known in upscaling, can be harder to find when doing inpainting. When we include it as known in our model, we get a new a posteriori, $p(u|u_0, D)$, and from that a new likelihood term $p(u_0|u, D)$ and a new prior $p(u|D)$. Our choice of prior on u is independent of the locus of missing data, as it is a model of ideal, undamaged image sequences. Thus $p(u|D) = p(u)$ and we have

$$p(u|u_0, D) \propto p(u_0|u, D)p(u).$$

In (pure) de-nosing, D will just cover the whole image and thus can be excluded. The same goes for video super resolution, but in temporal super resolution and deinterlacing, D tells us where data is missing.

Extending our formulation to image sequences instead of images, u_0 then denotes an input image sequence and u a desired output image sequence. We now also wish to compute the flow, \vec{v} of u , which gives us a new a posteriori, $p(u, \vec{v}|u_0, D)$ and following from that, a new likelihood term, $p(u_0|u, \vec{v}, D)$ and a new prior, $p(u, \vec{v}) = p(u|\vec{v})p(\vec{v})$. Since the degradation or subsampling of u_0 is independent of the optical flow \vec{v} , the likelihood is $p(u_0|u, \vec{v}, D) = p(u_0|u, D)$. Image sequences are not just 3D volumes but spatiotemporal 2+1D volumes with space and time being separate dimensions, thus we assume that the $p(u|\vec{v})$ part of the prior is $p(u_s, u_t|\vec{v})$ and factor it as

$$p(u_s, u_t|\vec{v}) = p(u_t|u_s, \vec{v})p(u_s|\vec{v}).$$

The term $p(u_t|u_s, \vec{v})$ models the temporal coherence of an image sequence, and $p(u_s|\vec{v})$ gives us a change to model motion blur, but as we consider motion blur a desired artistic expression and for the sake of simplicity, we assume independence of u_s and \vec{v} , thus $p(u_s|\vec{v}) = p(u_s)$.

Gathering the new terms for image sequences, we end up at the final formulation of our Bayesian framework for joint image sequence processing (inpainting, deinterlacing etc.) and motion recovery

$$p(u, \vec{v}|u_0, D) \propto \underbrace{p(u_0|u, D)}_{P_0} \underbrace{p(u_s)}_{P_1} \underbrace{p(u_t|u_s, \vec{v})}_{P_2} \underbrace{p(\vec{v})}_{P_3}. \quad (17)$$

The right hand side terms are: P_0 the image sequence likelihood, P_1 the spatial prior on image sequences, P_3 the prior on motion fields and P_2 a term that acts both as spatiotemporal prior on the image sequence and as likelihood term for the

motion field. To transform the Bayesian formulation into a variational one, we use Mumford's Bayesian to variational rationale from [26], $E(x) = -\log p(x)$. We are then given the continuous minimization problem in (1).

REFERENCES

- [1] A. Almog, A. Levi, and A. Bruckstein, "Spatial de-interlacing using dynamic time warping," in *Proc. IEEE International Conference on Image Processing*. ieeexplore.ieee.org, 2005, pp. 1010–1013.
- [2] C. Ballester, B. Bertalmio, V. Caselles, L. Garrido, A. Marques, and F. Ranchin, "An inpainting-based deinterlacing method," *IEEE Transactions on Image Processing*, vol. 16, no. 10, pp. 2476–2491, 2007.
- [3] E. Bellers and G. de Haan, *De-interlacing. A Key Technology for Scan Rate Conversion*. Elsevier Sciences Publishers, Amsterdam, 2000.
- [4] M. Biswas, S. Kumar, and T. Nguyen, "Performance analysis of motion-compensated de-interlacing systems," *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2596–2609, 2006.
- [5] M. Biswas and T. Nguyen, "A novel de-interlacing technique based on phase plane correlation motion estimation," *International Symposium on Circuits and Systems, ISCAS*, vol. 2, pp. 604–607, 2003.
- [6] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High Accuracy Optical Flow Estimation Based on a Theory for Warping," in *Proceedings of the 8th European Conference on Computer Vision*, T. Pajdla and J. Matas, Eds., vol. 4. Prague, Czech Republic: Springer-Verlag, 2004, pp. 25–36.
- [7] A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger, and C. Schnörr, "Variational Optic Flow Computation in Real-Time," *IEEE Trans. on Image Processing*, vol. 14, no. 5, pp. 608–615, 2005.
- [8] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/Kanade Meets Horn/Schunck: Combining Local and Global Optic Flow Methods," *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005.
- [9] L. Capodiffero, "Interlaced to progressive conversion by median filtering," in *Signal Processing of HDTV, II*, L. Chiariglione, Ed. Elsevier Sciences Publishers, Amsterdam, 1990, pp. 677–684.
- [10] Y.-L. Chang, P.-H. Wu, S.-F. Lin, and L.-G. Chen, "Four field local motion compensated de-interlacing," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. . . .*, vol. 5. ieeexplore.ieee.org, 2004, pp. 253–256.
- [11] J. Cocquerez, L. Chanas, and J. Blanc-Talon, "Simultaneous Inpainting and Motion Estimation of Highly Degraded Video-Sequences," in *Scandinavian Conference on Image Analysis*. Springer-Verlag, 2003, pp. 523–530, INCS, 2749.
- [12] P. Delogne, L. Cuvelier, B. Maison, B. V. Caillie, and L. Vandendorpe, "Improved interpolation, motion estimation, and compensation for interlaced pictures," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 482–491, 1994.
- [13] T. Doyle and M. Looymans, "Progressive scan conversion using edge information," in *Signal Processing of HDTV, II*, L. Chiariglione, Ed. Elsevier Sciences Publishers, Amsterdam, 1990, pp. 711–721.
- [14] S. Gallot, D. Hulin, and J. Lafontaine, *Riemannian Geometry*. Springer-Verlag, 1990.
- [15] P. Haavisto, J. Juhola, and Y. Neuvo, "Scan rate up-conversion using adaptive weighted median filtering," in *Signal Processing of HDTV, II*, L. Chiariglione, Ed. Elsevier Sciences Publishers, Amsterdam, 1990, pp. 703–710.
- [16] B. Horn and B. Schunck, "Determining Optical Flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.
- [17] ITU, "ITU-R recommendation BT.500-11: Methodology for the subjective assessment of the quality of television pictures," Geneva, Switzerland, 6 2002.
- [18] S. Keller, F. Lauze, and M. Nielsen, "Variational motion compensated deinterlacing," in *Proceedings of Statistical Methods in Multi-Image and Video Processing, ECCV 2006 Workshop*, 2006, pp. 49–56.
- [19] —, "A total variation motion adaptive deinterlacing scheme," in *Scale Space and PDE Methods in Computer Vision, 5th International Conference, Scale-Space 2005, Proceedings*, ser. LNCS, R. Kimmel, N. Sochen, and J. Weickert, Eds., vol. 3459. Berlin: Springer, 2005, pp. 408–419.
- [20] S. H. Keller. (2007) Selected electronic results of deinterlacing experiments. [Online]. Available: <http://image.diku.dk/sunebio/VDI/DI.zip>
- [21] —, "Video Upscaling Using Variational Methods," Ph.D. dissertation, Faculty of Science, University of Copenhagen, 2007. [Online]. Available: <http://image.diku.dk/sunebio/Afh/SuneKeller.pdf>
- [22] J. Kovacevic, R. Safranek, and E. Yeh, "Deinterlacing by successive approximations," *IEEE Transactions on Image Processing*, vol. 6, no. 2, pp. 339–344, 1997.
- [23] F. Lauze, "Computational methods for motion recovery, motion compensated inpainting and applications," Ph.D. dissertation, IT University of Copenhagen, 2004.
- [24] F. Lauze and M. Nielsen, "A Variational Algorithm for Motion Compensated Inpainting," in *British Machine Vision Conference*, S. B. A. Hoppe and T. Ellis, Eds., vol. 2. BMVA, 2004, pp. 777–787.
- [25] M. Li and T. Nguyen, "A de-interlacing algorithm using markov random field model," *IEEE Transactions on Image Processing*, vol. 16, no. 11, pp. 2633–2648, 2007.
- [26] D. Mumford, "Bayesian rationale for the variational formulation," in *Geometry-Driven Diffusion In Computer Vision*, B. M. ter Haar Romeny, Ed. Kluwer Academic Publishers, 1994, pp. 135–146.
- [27] S. Pigeon and P. Guillotel, "Advantages and drawbacks of interlaced and progressive scanning formats," *CEC RACE/HAMLET*, 1995, deliverable no R2110/WP2/DS/R/004/b1.
- [28] J. Puttonstein, I. Heynderickx, and G. de Haan, "Evaluation of objective quality measures for noise reduction in tv-systems," *Signal Processing: Image Communication*, vol. 19, no. 2, pp. 109–119, February 2004.
- [29] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.
- [30] A. Skarbot, G. Ramponi, and L. Buriola, "FPGA architecture for a videowall image processor," in *Proceedings of the SPIE International Symposium on Electronic Imaging*, vol. 4304, 2001, pp. 75–84.
- [31] A. Skarbot, G. Ramponi, and D. Toffoli, "Image sequence processing for videowall visualization," in *Proceedings of the International Society for Optical Engineering*, vol. 3961, 2000, pp. 138–147. [Online]. Available: citeseer.ist.psu.edu/skarbot00image.html
- [32] A. Tekalp, *Digital Video Processing*. Prentice-Hall, 1995.
- [33] G. Thomas, "A comparison of motion-compensated interlace-to-progressive conversion methods," *Signal Processing, Image Commun.*, vol. 12, no. 3, pp. 209–229, 1998.
- [34] M. K. Thomsen, "Ekspertene, måleinstrumenterne og de naive," *Ingeniøren*, vol. 17, no. 3, pp. 2–3, April 2007, in Danish.
- [35] D. Tschumperlé and B. Besserer, "High quality deinterlacing using inpainting and shutter-model directed temporal interpolation," in *Proc. of ICCVG International Conference on Computer Vision and Graphics*. Kluwer, 2004, pp. 301–307.
- [36] Y. Wang, J. Ostermann, and Y. Zhang, *Video Processing and Communications*. Prentice-Hall, 2002.