

Spatio-temporal video deinterlacing using control grid interpolation

Ragav Venkatesan,^{a,c} Christine Zwart,^b David Frakes,^{b,c} Baoxin Li^a

^aSchool of Computing Informatics and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA

^bSchool of Biological and Health Systems Engineering, Arizona State University, Tempe, AZ, USA

^cSchool of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, USA

Abstract. With the advent of progressive format display and broadcast technologies, video deinterlacing has become an important video processing technique. Numerous approaches exist in literature to accomplish deinterlacing. While most earlier methods were simple linear filtering-based approaches, the emergence of faster computing technologies and even dedicated video processing hardware in display units has allowed higher quality, but also more computationally intense, deinterlacing algorithms to become practical. Most modern approaches analyze motion and content in video to select different deinterlacing methods for various spatiotemporal regions. In this paper, we introduce a family of deinterlacers that employs spectral residue to choose between and weight control grid interpolation based spatial and temporal deinterlacing methods. The proposed approaches perform better than the prior state-of-the-art based on peak signal-to-noise ratio (PSNR), other visual quality metrics, and simple perception-based subjective evaluations conducted by human viewers. We further study the advantages of using soft and hard decision thresholds on the visual performance.

Keywords: Deinterlacing, saliency, spectral residue, control grid interpolation..

Address all correspondence to: Ragav Venkatesan, Arizona State University. E-mail: ragav.venkatesan@asu.edu

1 Introduction

Deinterlacing is the process of converting an interlaced video format to a progressive video format.

Interlaced video formats can be very useful when bandwidth is limited and are also well-suited for scanning display systems. Interlaced videos are scanned in such a way that in any given frame with N rows, only $N/2$ alternate rows are newly updated from the previous frame. The remaining rows are updated in the next frame, and when the series of interlaced frames are displayed quickly enough, humans are unable to detect the unchanged lines (since the human eye doesn't update quickly enough). Interlaced videos are generally preferred in video broadcast and transmission systems. Interlaced videos are also preferred in high motion videos where vertical frequency is compromised to get a higher frame rate.

Video interlacing motivates many tasks pertaining to international TV broadcasting, format conversion for example. Moreover, many modern display systems operate on progressive video



Fig 1 Example of poor deinterlacing from a high-definition YouTube video.

streams and thus require a deinterlacer. Poor deinterlacing can be observed today in a wide range of consumer products. Figure 1 shows such a product from a recent YouTube video. Even though deinterlacing is a longstanding topic in video processing and numerous approaches have been taken to solve the problem, there is renewed interest due to recent developments in high speed and dedicated video processing hardware in display systems.

Bellers and Haan defined deinterlacing formally as:

$$\hat{F}(i, j, k) = \begin{cases} F(i, j, k), & j \bmod 2 = k \bmod 2 \\ F^I(i, j, k), & \text{otherwise,} \end{cases} \quad (1)$$

where F is the original interlaced video, F^I is an interpolated version of F in progressive format, \hat{F} is the deinterlaced video, k is the frame index, and i and j are the row and column matrix coordinates, respectively, that specify a pixel location within a frame.²⁶ It is the interpolator estimating $F^I(i, j, k)$ that the deinterlacer's quality depends on.

Based on the type of interpolator used to estimate $F^I(i, j, k)$, a deinterlacer can be classified as spatial, temporal, or a combination of both. Spatial interpolators interpolate within a given

frame and are usually preferred when there is a high degree of motion in the video. In such cases, the content of the video changes too quickly for temporal interpolators to perform well. Temporal interpolators work exclusively across frames and work well when there is little motion. Most modern deinterlacers employ method switching algorithms that use different estimates or combinations of different estimates from different interpolators for particular regions of video. Motion in the video is usually the preferred basis for method switching; in a region of video with high motion a near-spatial interpolator is preferred. In this paper, we propose a novel perception-inspired approach to such interpolator selection.

The regions of video that are perceptually salient are those that the human eye fixates upon and are thus effectively updated more often by the human visual system than those regions that are not perceptually salient. Good cinematographers ensure that the region with most activity is always salient.¹ With this understanding, it follows that the salient regions of the video, those that need to be updated more often, are better off interpolated using data from as small a temporal window as possible (preferably from within the same frame). A purely background pixel that doesn't change across two frames, on the other hand, can be fully temporally averaged well. However, many other non-salient regions can be interpolated more effectively using a spatio-temporal approach. This assertion form the foundation of the proposed family of algorithms.

Spectral residue in the context of perception is quite well studied.² The quaternion Fourier implementation of spectral residue was studied first by Zhang *et al.*³ In this paper, we use a similar saliency map and spectral residue in weighting to linearly combine spatial and temporal interpolator contributions. The spatial and temporal interpolators that we use are one-dimensional (1D) control grid interpolation (1DCGI), and two-dimensional (2D) control grid interpolation (2DCGI), respectively^{4,5}. 1DCGI is an intra-frame optical flow-based interpolator that works

like an edge-directed interpolator, and *2DCGI* is a more traditional optical flow-based temporal interpolator. While neither one of these methods alone is best for deinterlacing, a combination of the two yields perceptually beneficial results. In this paper we study such combinations.

The rest of the paper is organized as follows: Section 2 covers related work, Section 3 explains the proposed approaches, Section 4 describes the experiments, Section 5 documents the results, and Section 6 provides concluding remarks.

2 Related work

A straight forward temporal deinterlacer takes the form:

$$\hat{F}^{LA}(i, j, k) = \begin{cases} F(i, j, k), & j \bmod 2 = k \bmod 2 \\ \frac{F(i, j, k-1) + F(i, j, k+1)}{2}, & \text{otherwise.} \end{cases} \quad (2)$$

This method is called the Temporal line average (LA), simply LA, or the *bob* algorithm.²⁶ The algorithm performs well when there is very little motion. Many modern method switching algorithms still incorporate LA as one of the methods when the difference across two frames is lower than a threshold.

A fully spatial non-linear interpolator that works within a small window is the Extended LA or Edge-based LA (ELA)^{6,7}. Figure 2 shows the window of operation of ELA. While interpolating for the point X , three directional differences are estimated as $C1 = |a - f|$, $C2 = |b - e|$, and $C3 = |c - d|$, where a, b, c, d, e , and f are defined as in Figure 2. The minimum difference among $C1$, $C2$, and $C3$ is chosen. The interpolated value for X is then the average of the two points that corresponded to the minimum difference.

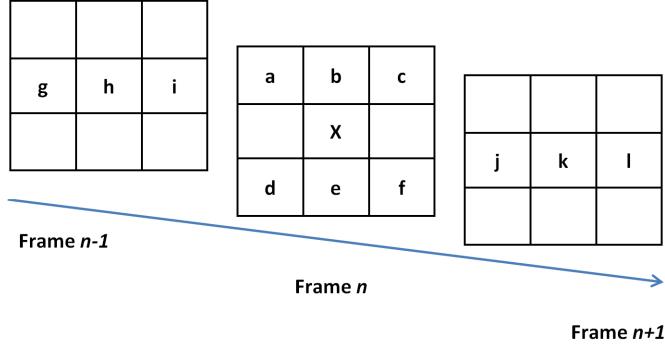


Fig 2 Neighborhoods for STELA and ELA.

Many edge-based interpolators similar to ELA have also been proposed. One efficient ELA implementation (EELA) uses directional spatial correlation instead of angular edge directions.⁸ The low complexity interpolation method for deinterlacing (LCID) uses four directions rather than the three used in ELA.⁹ Instead of estimating edge directions using differences, LCID uses the edges from a sobel filtered image and interpolates along the detected edges.¹⁰

Spatio-temporal edge-based median filtering (STELA) adds a temporal component to an intra-frame deinterlacer like ELA.¹¹ STELA is a two-pronged approach. It divides a video frame into low frequency and high frequency frames. In the low frequency frame, STELA works on a $3 \times 3 \times 3$ neighborhood as shown in Figure 2. It estimates six directional differences unlike ELA, which works with only three. The six directional differences are $C1 = |a - f|, C2 = |b - e|, C3 = |c - d|, C4 = |g - l|, C5 = |h - k|$, and $C6 = |i - j|$. The deinterlaced estimate for any point $X = Med\{A, b, e, h, k\}$, where A is the average value of the two points that yield the minimum directional change among $C1$ through $C6$ and Med , is a median operator. Although A is the preferred value for X , the median filter is added as a backup in case there is noise in the video. Whenever there is noise in the video and that alters the decision to choose A , the median eliminates the noisy pixel and still provides an acceptable result. The high frequency frames are

subject to line doubling or *weaving*. The line doubled version is then added to the processed low frequency frames. STELA showed that spatio-temporal methods work better than purely spatial deinterlacers like ELA when the interlaced video contains both low-motion background regions and fast-changing foreground regions.

A computationally efficient spatio-temporal deinterlacer is the vertical temporal filter (VTF).¹²

VTF is a filtering algorithm and is defined as:

$$\hat{F}^{VTF}(i, j, k) = \begin{cases} F(i, j, k), & j \bmod 2 = k \bmod 2 \\ \sum_l \sum_m F_{k+l}(i, j + m) h_l(m), & \text{otherwise,} \end{cases} \quad (3)$$

where Weston proposed the filter $h_l(m)$ to be:

$$h_l(m) = \begin{cases} \frac{1}{2}, \frac{1}{2} & (m = -1, 1 \& l = 0) \\ -\frac{1}{16}, \frac{1}{8}, -\frac{1}{16} & (m = -2, 0, 2 \& l = -1, 1). \end{cases} \quad (4)$$

VTF is not an adaptive algorithm like STELA or ELA but is still among the most popular deinterlacing algorithms because of its computational efficiency. Adaptations of the algorithm are seen in deinterlacing literature as late as 2013. Content adaptive VTF (CAVTF) and spatially registered VTF or SRVTF are two examples^{13,14}.

CAVTF is a two-step algorithm where each pixel is classified into one of three classes by using a modified adaptive dynamic range encoding. Once each pixel is classified and provided sufficient temporal differences exist, an adaptive version of VTF is implemented wherein filter values depends on the neighborhood pixel values. SRVTF is a VTF algorithm applied not to the

interlaced video but to spatially registered frames. A global motion estimation is performed to estimate motion vectors v_x and v_y as:

$$(v_x^*, v_y^*) = \underset{(v_x, v_y) \in MV}{\operatorname{argmin}} \sum |F(i, j, k - 1) - F(i + v_x, j + v_y, k + 1)|, \quad (5)$$

where the motion vectors MV don't span more than 8 pixels in either direction ($\{(v_x, v_y) | -8 \leq v_x, v_y \leq 8; v_x, v_y \text{ are even}\}$).¹³ After estimating the motion vectors, spatial registration is performed as:

$$F^{SR}(i, j, k - 1) = F(i - v_x^*/2, j - v_y^*/2, k - 1) \quad (6)$$

and

$$F^{SR}(i, j, k + 1) = F(i + v_x^*/2, j + v_y^*/2, k + 1). \quad (7)$$

Traditional VTF is performed on the spatially registered frames F^{SR} to get \hat{F}^{SR} . A frame-difference-like technique is used as a reality check to make sure that the registered frames do perform better than the original VTF. The frame differences are d_1 and d_2 , which are defined as:

$$\begin{aligned} 3d_1 = & |F^{SR}(i, j - 2, k - 1) - F^{SR}(i, j - 2, k + 1)| + |F^{SR}(i, j, k - 1) - F^{SR}(i, j, k + 1)| \\ & + |F^{SR}(i, j + 2, k - 1) - F^{SR}(i, j + 2, k + 1)| \end{aligned} \quad (8)$$

and

$$\begin{aligned}
3d_2 = & |F(i, j - 2, k - 1) - F(i, j - 2, k + 1)| + |F(i, j, k - 1) - F(i, j, k + 1)| \\
& + |F(i, j + 2, k - 1) - F(i, j + 2, k + 1)|.
\end{aligned} \tag{9}$$

Deinterlacing is performed as:

$$\hat{F}^{SR}(i, j, k) = \begin{cases} \sum_l \sum_m F^{SR}(i, j + m, k + l) h_l(m) & if(d_1 < d_2) \\ \hat{F}^{VTF} & otherwise. \end{cases} \tag{10}$$

The reasoning behind registration is that compensation for motion yields more suitable pixel neighbors for VTF to work with. This along with a second level verification using the frame differences, which gives the option to revert back to the original VTF, makes the algorithm robust.

While VTF is a fixed range filter, a non-local means filter-based approach was proposed by Wang *et al.* that estimates a missing pixel using an adaptive weighted average of all pixels in a patch-matched neighborhood.¹⁵ By choosing an optimal range for the patch matching algorithm, good performance is achieved without compromising efficiency. Hong *et al.* use a similar distance-based weighting scheme to weight their sinc-based interpolator.¹⁶ An example of a purely motion-based approach is deinterlacing using hierarchical motion analysis.¹⁷ This method uses motion analysis (in four-stages), pixel estimation, and pixel correction procedures to generate a likely pixel estimate. Although this method performs well, it is computationally expensive.

3 Proposed Algorithms

The proposed family of algorithms are method switching approaches that choose either a temporal average or a linearly weighted combination of spatially and temporally interpolated estimates. The spatially interpolated estimate is generated with $1DCGI$ and the temporally interpolated estimate with $2DCGI$.⁵ The choice is based on a threshold frame difference and the linear weights are the normalized spectral residues. At the core of the idea is the use of spectral residue to make a choice between the spatial and temporal interpolators. The link between spectral residue and human perception is studied in.² Spectral residues for color images are estimated using the quaternion Fourier transform approach in.³ The quaternion Fourier transform of an image is studied in.¹⁸ Any color image can be represented using quaternions of the form:

$$q(i, j, k) = Ch_1(i, j, k) + Ch_2(i, j, k)\mu_1 + Ch_3(i, j, k)\mu_2 + Ch_4(i, j, k)\mu_3, \quad (11)$$

where μ_p for $p = 1, 2, 3$ satisfies $\mu_p^2 = -1$, $\mu_1 \perp \mu_2$, $\mu_2 \perp \mu_3$, and $\mu_1 \perp \mu_3$. The three color channels of an image can be allocated to Ch_2 , Ch_3 , and Ch_4 , respectively, while Ch_1 is set to zero.

The quaternion Fourier transform (QFT) of an image is:

$$Q(u, v, k) = \frac{1}{\sqrt{WH}} \sum_{j=0}^{W-1} \sum_{i=0}^{H-1} e^{\mu_1^{2\pi(\frac{jv}{W} + \frac{iu}{H})}} q(i, j, k), \quad (12)$$

and its inverse is:

$$q(i, j, k) = \frac{1}{\sqrt{WH}} \sum_{v=0}^{W-1} \sum_{u=0}^{H-1} e^{\mu_1^{2\pi(\frac{jv}{W} + \frac{iu}{H})}} Q(u, v, k), \quad (13)$$

where $q(i, j, k)$ represents intensity samples in the spatial domain, $Q(u, v, k)$ represents intensity



Fig 3 Mother video (left) and the detected saliency (right) after thresholding by $B=4\%$ of the bit depth.

samples in the frequency domain, and W and H are the width and height of the image in pixels, respectively. The phase spectrum of an image can be extracted by $Q^{phase} = \frac{Q}{\|Q\|}$. An approximation to spectral residue can be obtained by Gaussian smoothing of the inverse QFT, q^{phase} . The L_1 norm of such a smoothed phase is also a measure of the visual saliency of the image.³ Since we use the spectral residue for weighting between spatial and temporal interpolators, we normalize the spectral residue as:

$$S(i, j, k) = \frac{\|g * q^{phase}(i, j, k)\|_1}{\max(\|g * q^{phase}(i, j, k)\|_1)}. \quad (14)$$

An example of the resulting saliency map is shown in Figure 3. Unlike SRVTF that uses motion as a region classifier, we use the spectral residue.

Two kinds of deinterlacers can be formulated in this manner: a hard decision deinterlacer (HDD) that uses the thresholded (by B) saliency map and a soft decision deinterlacer (SDD) that uses the normalized spectral residue. While HDD is a direct method-switching algorithm, SDD is a method-combination algorithm. These proposed approaches are detailed in Section III-C. Since the approaches are built upon two interpolators (*1DCGI* and *2DCGI*) that operate in 1D and 2D, respectively, we first briefly describe the interpolators in Section 3.1 and 3.2.

3.1 1D Control Grid Interpolation

The 1D control grid interpolator $1DCGI$ is based on the optical flow brightness constraint.¹⁹ The brightness constraint assumes that the intensity associated with any discrete location in a source data set is present at some location in a destination data set. Displacement vectors relating the source and destination locations of all corresponding intensities thus define a spatial transform between the two data sets.

Consider a row of pixel intensities from a video frame. Within the $1DCGI$ framework, that row can be defined as the source data set and the neighboring row below it as the destination data set. The brightness constraint can then be posed as:

$$I(i, j) = I(i + 1, j + \beta), \quad (15)$$

where i and j represent row and column matrix index variables, respectively, and β represents the aforementioned displacement vector, which is horizontal or along the column index axis in this formulation. A Taylor series expansion is then applied to form the modified brightness constraint:

$$I(i, j) \approx I(i, j) + \frac{\partial I(i, j)}{\partial i}(1) + \frac{\partial I(i, j)}{\partial j}(\beta), \quad (16)$$

which reduces to:

$$\frac{\partial I(i, j)}{\partial i}(1) + \frac{\partial I(i, j)}{\partial j}(\beta) \approx 0. \quad (17)$$

Direct approaches to solving Equation 17 are sensitive to noise, which typically motivates incorporation of a smoothness constraint that is applied to the collection of displacement vectors. Instead, $1DCGI$ ensures smoothness by defining control point (or node) displacements at regularly spaced

intervals and generating the intermediate displacements (for non-node locations) using linear interpolation. A detailed treatment including solution mechanics is provided in .⁴

The transform comprised of all displacement vectors for a given source row can then be used to accomplish inter-row directional interpolation. That is, each pair of intensities connected by a displacement vector (one intensity residing in the source row and the other in the destination row) is first averaged, and the averaged intensities are then placed midway between the source and destination rows (along the displacement vectors). Finally, a new discrete row of data is regridded at the location between the original two rows. Note that interpolants within rows at arbitrary distances between the source and destination rows are customarily generated using a distance weighted average. Full details of the approach are covered in .¹⁹

In the deinterlacing application, matches are made between the alternating rows of data from an interlaced frame, which correspond to the same point in time:

$$I(i, j) = I(i + 2, j + 2\beta_+), \quad (18)$$

or

$$I(i, j) = I(i - 2, j - 2\beta_-), \quad (19)$$

where the indices j and the rows of pixel values $i - 2, j$ and $i + 2$ are known, and β_+ and β_- define the horizontal (along the column index axis) displacements in each independent equation. After being solved for, the β_+ values for row i and the β_- values for row $i + 2$ can thus be used to generate independent top-down and bottom-up estimates of the row of pixel intensities at row $i + 1$ via directional interpolation. The two interpolated row estimates are typically weighted equally in

forming the final interpolated row of pixel intensities (because the interpolated row is equidistant from the known rows considered in the interpolation).

In practice, $1DCGI$ is a straight forward line-to-line edge directed interpolator that is comparable in style to ELA. In both cases, interpolation is carried out between pairs of pixels in neighboring rows (one from each row) that are registered with displacements so as to minimize the difference in intensity between pixels in a pair. However, in contrast to $1DCGI$, ELA is limited to integer displacements, which significantly reduces the angular resolution of the transform used to directionally interpolate.

3.2 2D Control Grid Interpolation

2D control grid interpolation ($2DCGI$) extends $1DCGI$ into two dimensions.²⁰ Like $1DCGI$, $2DCGI$ is based on the optical flow brightness constraint, which is traditionally posed for a time series of 2D data (e.g., a video) as:

$$I(i, j, k) = I(i + \alpha, j + \beta, k + \delta k), \quad (20)$$

where i and j again represent row and column matrix index variables, respectively, k represents time, α represents a vertical (along the row index axis) displacement vector, β again represents a horizontal (along the column index axis) displacement vector, and δk represents some change in time. Applying a partial Taylor series expansion (ignoring higher order terms) yields:

$$I(i, j, k) \approx I(i, j, k + \delta k) + \frac{\partial I(i, j, k + \delta k)}{\partial i}(\alpha) + \frac{\partial I(i, j, k + \delta k)}{\partial j}(\beta), \quad (21)$$

which can be reformatted as:

$$\frac{\partial I(i, j, k + \delta k)}{\partial i}(\alpha) + \frac{\partial I(i, j, k + \delta k)}{\partial j}(\beta) \approx I(i, j, k) - I(i, j, k + \delta k), \quad (22)$$

where the right hand terms of the equation simply represent frames of a video sequence separated in time by δk .

Like the $1DCGI$ case, $2DCGI$ uses control points to ensure smoothness across displacement vectors, which also facilitates optical flow-based motion estimation with very high efficiency. However, unlike the $1DCGI$ case, $2DCGI$ orients the control points in a 2D grid structure rather than a 1D vector. Intermediate displacements that originate from in between nodes on the control point grid are generated with bilinear interpolation. Note that like $1DCGI$, $2DCGI$ does not simply calculate control point displacements and then fill in other displacements via interpolation after the fact. Rather, the whole collection of displacements (those at nodes and otherwise) is solved for concurrently through the use of basis functions (linear and bilinear for $1DCGI$ and $2DCGI$, respectively).⁵ In both cases the resulting collection of displacement vectors that defines the transform for interpolation is piecewise smooth and globally continuous. Once forward and backward (in time) displacement vectors have been solved for between frames k and $k + \delta k$ with $2DCGI$, as top-down and bottom-up (in space) displacement fields were solved for between rows with $1DCGI$, a new frame of data between the original two can be generated by combining the forward and backward inter-frame estimates in a distance weighted average. Full details of the approach including solution mechanics are covered in .²⁰

3.3 Switching Schemes

The HDD can be formulated by using *1DCGI* for salient regions of the video and *VTF* for other regions of the video, provided there are sufficient differences among pixel values across frames. Such an approach was first discussed by Venkatesan et al. and is described by the following equations:²¹

$$\hat{F}_n^{HDD}(i, j) = \begin{cases} F_n(i, j), & j \bmod 2 = n \bmod 2 \\ \frac{F_{n-1}(i, j) + F_{n+1}(i, j)}{2}, & D_n(i, j) < T \\ \sum_m \sum_k F_{n+m}(i, j+k) h_m(k), & S_n(i, j) < B; D_n(i, j) \geq T \\ 1D_n(i, j), & \text{else,} \end{cases} \quad (23)$$

where $h_m(k)$ is Weston's VTF, $D_n(i, j)$ is frame difference, $S_n(i, j)$ is the spectral residue, and $1D_n(i, j)$ is the *1DCGI* estimate.

The novel method, SDD can be obtained by linearly weighting *1DCGI* and *2DCGI* estimates. The spatially-salient regions in a video are those particular regions that the human eye localizes first and that therefore demand the sharpness of a spatial interpolator. The non-salient regions take time for the human eye to register and are therefore handled sufficiently well by a more smoothing temporal interpolator. Thus, the linear choice is made as $1D_n(i, j)S_n(i, j) + 2D_n(i, j)(1 - S_n(i, j))$. Whenever the frame difference $D_n(i, j)$ across two frames is lower than the threshold T , two

intensity units for example, a frame average is performed. The SDD is formulated as:

$$\hat{F}_n^{SDD}(i, j) = \begin{cases} F_n(i, j), & j \bmod 2 = n \bmod 2 \\ \frac{F_{n-1}(i, j) + F_{n+1}(i, j)}{2}, & D_n(i, j) < T \\ 1D_n(i, j)S_n(i, j) + 2D_n(i, j)(1 - S_n(i, j)), & D_n(i, j) \geq T, \end{cases} \quad (24)$$

where $1D_n(i, j)$ is the 1DCGI estimate of the n^{th} frame and $2D_n(i, j)$ is the corresponding 2DCGI estimate.

This method avoids the ambiguity of spatio-temporal interpolators like VTF and instead uses a straightforward combination of a purely spatial interpolator and a purely temporal interpolator that is based on the spectral residue. The more salient the region is, the higher the spectral residue and the more weight the spatial estimate gets, and vice-versa. The result is a smoother transition between the spatial and the temporal and therefore a higher quality deinterlaced video. This is particularly noticeable in videos that have a low temporal gradient or small transitions locally, such as in HD videos. The computational complexity of any method switching algorithm depends on the complexity of the original methods being switched. The authors would like to refer the reader to the original papers for details on computational analysis^{5,20}. Switching methods may be accompanied by increases in computational complexity due to decision making. In case of our method that increase comes from the overhead of calculating the saliency using the spectral residue.³

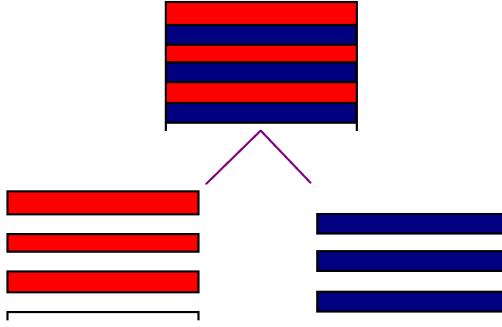


Fig 4 Interlace type 1: one frame is split into two fields.

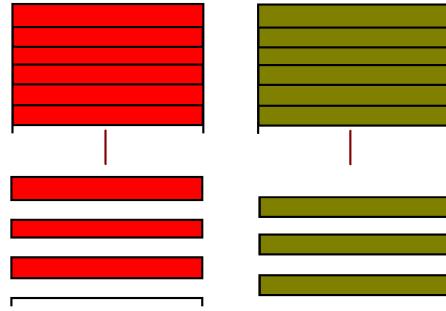


Fig 5 Interlace type 2: each frame gets interlaced into its own respective field.

4 Experiments

4.1 Computational Experiments

The proposed family algorithms were all implemented in MATLAB along with SRVTF. The test video set comprised of 13 commonly used CIF videos from the trace video library²² and high definition videos from the consumer digital video library.²³ These videos were manually and deliberately interlaced, and then deinterlaced using different algorithms. It is reasonable to conclude from deinterlacing literature that when interlacing a video manually, videos can be considered to be interlaced in either of two ways:

1. Fields $n - 1$ and n are split from the same frame. A deinterlaced frame is to be reconstructed into full resolution from the two interlaced fields. Two fields map to one deinterlaced frame and no data is lost while interlacing. This method of manual interlacing is shown in Figure 4.

2. Fields $n - 1$ and n are down-sampled from two unique frames (frames $n - 1$ and n , respectively). One unique de-interlaced frame is to be reconstructed for every field. One field maps to only one frame and half the data is simply thrown away while interlacing. This method of manual interlacing is shown in Figure 5.

We interlaced the videos by using the second method. This enabled us to maintain the number of frames, facilitating the use of reference-based computational metrics for evaluation. We calculated the following computational metrics for each of the methods:

1. Peak signal-to-noise ratio (PSNR).
2. Visual signal-to-noise ratio (VSNR).²⁴
3. Visual information fidelity (VIF).²⁵

To facilitate fair comparisons with the methods in literature without bias from implementation details, we made use of statistical relevance. That is, we used VTF (which is a fairly straightforward method to implement) as common ground. We compared the results of our methods with our VTF results, and the results of other authors with their VTF results reported in the literature. If MSE_{VTF} is the mean squared error from VTF and MSE_{new} is the mean squared error of any new algorithm, then the statistical relevance r (R-value) is defined as:

$$r = 100 * [1 - \frac{MSE_{VTF}}{MSE_{new}}]. \quad (25)$$

The R-value is used to compare our methods with methods like CAVTF and SRCAVTF that were difficult to implement fairly by ourselves.

4.2 Subjective Experiments

Evaluating the perception-inspired philosophy of using saliency in any algorithm is difficult based on computational metrics alone. Hence, we used a subjective evaluation to further support our reasoning. Note that we are using this experiment to only test the effects of bad deinterlacing on saliency and not to compare our methods as we did with computational evaluation schemes.

Specifically, we used the subjective evaluation to verify the following:

1. Poor deinterlacing in salient regions affects viewing experience more than poor deinterlacing in non-salient regions.
2. Temporal deinterlacing in salient regions affects viewing more than spatial deinterlacing does.

The subjective experiments were performed on 11 subjects all of whom had moderate to proficient, technical video processing knowledge. Each subject was shown a play list of videos with the first video being the unaltered video for reference. The altered videos shown (in a randomized order) were videos with:

1. Temporally deinterlaced non-salient regions.
2. Spatially deinterlaced non-salient regions.
3. Temporally deinterlaced salient regions.
4. Spatially deinterlaced salient regions.

The spatial and temporal interpolators were spatial weave and temporal weave algorithms, respectively.²⁶ Between each video was a four second black screen for eye-adjustment. The subjects

were informed before the experiment that the first video was standard and that they should rate the following four videos on a five-point scale in comparison to the first one. The subjects were not informed on particular regions or frames that were altered. While the subjects observed and rated the videos, their eyes were tracked to understand eye-fixation when artifacts were present in the video. The first experiment used the foreman video. We chose the logo and its immediate surroundings in the top left corner of the video as the non-salient region, while the salient region was chosen to be the face. The other regions were left unaltered. The second experiment used the akiyo video, the third used the highway video, and the fourth used the crew video, with similar regions chosen in each as in foreman. These regions agree with the saliency model we used.

The eye-tracker used was the VT2MINI eye-tracking system by EyeTech™ Digital Systems, Mesa, AZ, USA. The typical viewing distance was 25 inches. The tracker provided an accuracy of 0.5 degrees at 60+ frames per second. The ambient illuminance while the experiments were conducted was slightly below regular office lighting in the range of 200 to 275 lux.

5 Results

5.1 Computational Results

Table 1 compares the PSNRs of different deinterlacing algorithms and shows that HDD and SDD outperform the other algorithms. It is also clear from table 2 that the R-values echo the PSNR results from table 1. SDD performs particularly well on videos containing relatively clearly defined saliency, which also agrees with the saliency model we used. Although a study of various computational saliency models and their effects on region-selection for various deinterlacing meth-

Table 1 Table of PSNR. All of the methods in this table were implemented by the authors. Care was taken to ensure that the methods were implemented to the finest detail provided in the respective source papers.

Video	STELA	VTF	SRVTF	HDD	SDD
Akiyo (CIF)	41.237	41.117	41.364	47.301	49.212
Bowing (CIF)	37.013	40.962	40.726	46.122	42.659
Bridge Far (CIF)	38.788	33.689	34.308	42.423	37.833
Container (CIF)	35.479	31.055	32.821	46.417	46.394
Deadline (CIF)	35.662	33.152	33.009	42.814	39.154
Foreman (CIF)	31.467	32.202	33.802	36.957	37.183
Galleon (CIF)	31.609	27.058	27.163	42.048	41.758
Hall Monitor (CIF)	36.942	32.023	35.027	41.892	38.578
Mother (CIF)	42.599	38.058	41.635	45.635	44.813
News (CIF)	36.855	39.088	38.045	44.597	41.539
Students (CIF)	37.086	33.436	33.954	45.173	42.887
Paris (CIF)	30.943	28.934	29.010	33.799	35.344
Sign Irene (CIF)	36.181	36.401	37.413	40.108	38.381
SVT (HD)	31.078	32.256	32.569	32.435	32.927
Intel Bottles (HD)	42.403	46.160	47.437	43.502	47.131
NTIA Lion (HD)	32.880	33.579	33.082	33.679	37.324
NTIA FoxBird (HD)	36.881	39.656	39.138	36.144	39.789
LA Clouds (HD)	45.791	46.476	44.223	45.826	46.708
LA Underboat (HD)	34.008	36.187	35.530	35.144	36.400
LA Turtles (HD)	25.994	26.040	26.034	26.958	26.047

Table 2 Table of r-values using reported results.

Video	HDD	SDD	SRVTF	SRCAVTF
Akiyo	75.92	84.49	0	40.77
Container	97.09	97.07	-0.09	71.63
Foreman	66.54	68.23	11.16	45.82
Hall Monitor	89.69	77.89	-1.07	33.61
Mother	82.52	78.88	-1.59	25.45
News	71.87	43.13	0	49.77
Paris	67.37	77.14	0.04	74.30

ods is outside the scope of this article, it is noteworthy that with more accurate saliency models the visual quality of the proposed methods should be better.

HDD is a hard-choice algorithm that uses one or another estimate and has a higher PSNR on average. However, the PSNR performance of HDD doesn't necessarily prove its performance in

Table 3 Table of VSNR values. All of the methods in this table were implemented by the authors. Care was taken to ensure that the methods were implemented to the finest detail provided in the respective source papers.

Videos	VTF	SRVTF	HDD	SDD
Akiyo	43.21	43.15	46.81	47.21
Bowing	36.97	36.87	47.27	47.34
Bridge Far	31.77	30.96	41.80	41.81
Container	30.12	29.94	44.16	44.21
Foreman	30.93	30.37	37.70	37.78
Galleon	26.39	26.37	45.44	45.46
Hall	32.52	31.95	43.54	43.51
News	41.02	40.85	47.36	47.69
Paris	26.93	26.94	40.16	39.21
Sign Irene	33.01	32.85	35.19	35.98
Students	28.92	29.02	42.40	42.59
LA Turtle	15.28	15.41	15.77	15.45
LA Clouds	42.13	38.27	43.60	44.26
LA Underboat	28.86	29.46	30.16	32.01
SVT	31.47	31.81	32.99	33.94
NTIA Lion	28.99	28.87	34.60	34.63
NTIA Foxbird	35.62	35.29	34.11	38.12
Intel Bottles	25.40	24.21	24.18	26.87

Table 4 Table of VIF values. All of the methods in this table were implemented by the authors. Care was taken to ensure that the methods were implemented to the finest detail provided in the respective source papers.

Videos	VTF	SRVTF	HDD	SDD
Akiyo	0.9245	0.9244	0.9776	0.9771
Bowing	0.8551	0.8548	0.9324	0.9325
Bridge Far	0.6330	0.6056	0.7845	0.7845
Container	0.7095	0.7080	0.9558	0.9557
Foreman	0.7554	0.7509	0.8220	0.8448
Galleon	0.5261	0.5250	0.9132	0.9132
Hall	0.8183	0.8168	0.8564	0.8560
News	0.8562	0.8547	0.9331	0.9445
Paris	0.6142	0.6132	0.8812	0.8734
Sign Irene	0.7990	0.7979	0.8524	0.8545
Students	0.7503	0.7496	0.9618	0.9599

terms of visual quality. VSNR and VIF are used to compare the methods for visual quality^{24,25}.

Tables 3 and 4 show the results for VSNR and VIF, respectively. Based on these metrics, SDD

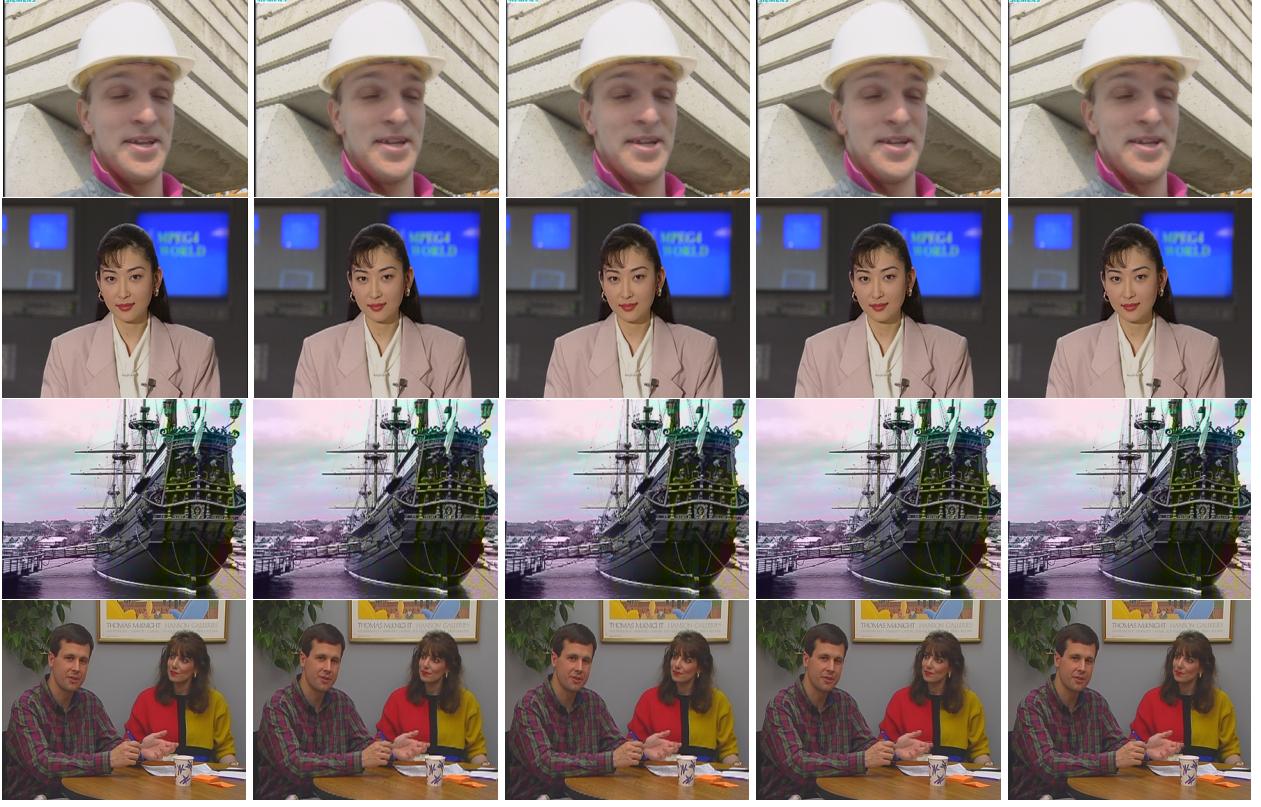


Fig 6 Video screenshots corresponding to different algorithms. From left to right are original, VTF, SRVTF, HDD, and SDD. From top to bottom are original and deinterlaced versions of frame 2 from foreman, akiyo, galleon, and students videos. The performance of the proposed approaches can be best appreciated on the edges of the wall and within the Siemens logo in the foreman video (top), and on the edges of the table in the students video (bottom).

keeps up with and often outperforms HDD. We achieve this result through linear weighting, which provides smoother deinterlacing than hard choices. The reason this work particularly well on HD videos is that the spatial gradients (and temporal gradients in case of SDD) involved in the optical flow foundation of CGI are more accurate in HD cases given the more highly resolved original data.

Figure 6 shows the deinterlaced output for one frame of some of the test videos. In the students video, while regions like the edge of the table were deinterlaced smoothly by the proposed methods, the other methods produce jagged edges. In the same video, the hand (which is a non-salient region) was affected by motion artifacts even using the proposed methods. This is because the hand, being a non-salient region, was interpolated with more weight for the temporal than for the



Fig 7 Spatially deinterlaced non-salient region of video noticed by different subjects. The heat map (hotter the region, longer the gaze) shows the gaze locations of the subjects, for the first 33 frames after the deinterlacing is introduced. This experiment illustrated the fact that any non-salient region of the video that are spatially deinterlaced leads to viewer discomfort. This observation is also supported by the MOS scores.



Fig 8 Temporally deinterlaced non-salient regions of video (such as the wall in the background) missed by different subjects. The heat map (hotter the region, longer the gaze) shows the gaze locations of the subjects, for the first 33 frames after the deinterlacing is introduced. This experiment illustrated the fact that any non-salient region of the video that are temporally deinterlaced doesn't affect viewing. This observation is also supported by the MOS scores.

Table 5 Table of Mean Opinion Scores (MOS). NS - Non-Salient; S-Salient. For the Crew video there was no temporal deinterlacing for non-salient regions.

Videos	NS-Spatial	NS-Temporal	S-Spatial	S-Temporal
Foreman	3.8	3.9	3.59	2.81
Crew	3.95	N/A	3.68	1.95
Highway	3.77	3.4	2.59	2.13
Akiyo	2.81	3.36	3.72	3.68

spatial interpolator. In the foreman video, the diagonal edges in the wall and the Siemens logo, which are non-salient regions, were more smoothly deinterlaced with the proposed methods than with SRVTI.

5.2 Subjective Results

While the computational results quantify the results of both the methods, we used subjective evaluation experiments to validate the philosophy of method switching. Note that these results are not meant to compare two methodologies. Table 5 shows the mean opinion scores (MOS) provided by the subjects. From the MOS results we can observe that videos that contain poorly deinterlaced salient regions consistently received a lower score than those that contained poorly deinterlaced non-salient regions. This is because the subjects seldom noticed (and in most cases did not notice) errors in non-salient regions. Figure 8 shows examples from the foreman video where the subjects did not notice the non-salient region around the Siemens logo that was temporally deinterlaced. However, the subjects did notice the same region when it was spatially deinterlaced, which indicates that the perceptual saliency of a video is not affected by temporal de-interlacing. Spatial deinterlacing in non-salient regions, however affects the perceptual saliency as shown in Figure 7. This is because purely spatial deinterlacing alters the temporal frequency of the video and thus leads to flicker artifacts that affect perceptual saliency. Accordingly, the non-salient regions of a video should be temporally deinterlaced. This experiment affirms the method switching philosophy based on which deinterlacing is performed in this paper.

6 Conclusions

In this paper, we propose a perception-inspired saliency-based approach to spatio-temporal deinterlacing. We use spectral residue in weighting the $1DCGI$ and $2DCGI$ interpolators, which are spatial and temporal in nature, respectively. The deinterlacing approach was validated using a simple subjective evaluation. Specifically, by tracking the eyes of subjects, we observed that the viewing remained unaffected in a video wherein the non-salient regions were temporally deinter-

laced, as compared to a video wherein the non-salient regions were spatially deinterlaced. This observation was further supported by the mean opinion score.

The proposed family of methods were also compared against the state-of-the-art a traditional computational metric (i.e., PSNR) as well as more progressive visual quality metrics (i.e., VSNR and VIF). Whenever possible, a statistical relevance metric was also used to compare with the state-of-the-art. The proposed methods outperformed the state-of-the-art based on each of the metrics considered, which indicates that saliency-based approaches built upon the 1DCGI and 2DCGI interpolators may represent a favorable alternative for video deinterlacing applications.

References

- 1 L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **20**(11), 1254–1259 (1998).
- 2 X. Hou and L. Zhang, “Saliency detection: A spectral residual approach,” in *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*, 1–8, IEEE (2007).
- 3 C. Guo, Q. Ma, and L. Zhang, “Spatio-temporal saliency detection using phase spectrum of quaternion fourier transform,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 1–8, IEEE (2008).
- 4 C. M. Zwart and D. H. Frakes, “One-dimensional control grid interpolation-based demosaicing and color image interpolation,” in *Proc. SPIE*, **8296**, 82960E (2012).
- 5 D. H. Frakes, L. P. Dasi, K. Pekkan, H. D. Kitajima, K. Sundareswaran, A. P. Yoganathan, and M. J. Smith, “A new method for registration-based medical image interpolation,” *Medical Imaging, IEEE Transactions on* **27**(3), 370–377 (2008).

- 6 T. Doyle, “Interlaced to sequential conversion for edtv applications,” in *Proc. 2nd int. workshop signal processing of HDTV*, 412–430 (1990).
- 7 C. J. Kuo, C. Liao, and C. C. Lin, “Adaptive interpolation technique for scanning rate conversion,” *Circuits and Systems for Video Technology, IEEE Transactions on* **6**(3), 317–321 (1996).
- 8 T. Chen, H. R. Wu, and Z. H. Yu, “Efficient deinterlacing algorithm using edge-based line average interpolation,” *Optical Engineering* **39**(8), 2101–2105 (2000).
- 9 C. Pei-Yin and L. Yao-Hsien, “A low-complexity interpolation method for deinterlacing,” *IEICE transactions on information and systems* **90**(2), 606–608 (2007).
- 10 H. Yoo and J. Jeong, “Direction-oriented interpolation and its application to de-interlacing,” *Consumer Electronics, IEEE Transactions on* **48**(4), 954–962 (2002).
- 11 H.-S. Oh, Y. Kim, Y.-Y. Jung, A. W. Morales, and S.-J. Ko, “Spatio-temporal edge-based median filtering for deinterlacing,” in *Consumer Electronics, 2000. ICCE. 2000 Digest of Technical Papers. International Conference on*, 52–53, IEEE (2000).
- 12 M. Weston, “Interpolating lines of video signals,” *US-patent 4,789,893* (December 1988).
- 13 K. Lee and C. Lee, “High quality deinterlacing using content adaptive vertical temporal filtering,” *Consumer Electronics, IEEE Transactions on* **56**(4), 2469–2474 (2010).
- 14 K. Lee and C. Lee, “High quality spatially registered vertical temporal filtering for deinterlacing,” *Consumer Electronics, IEEE Transactions on* **59**(1), 182–190 (2013).
- 15 J. Wang, G. Jeon, and J. Jeong, “Deinterlacing algorithm with an advanced non-local means filter,” *Optical Engineering* **51**(4), 047009–1 (2012).

- 16 S.-M. Hong, S.-J. Park, J. Jang, and J. Jeong, “Deinterlacing algorithm using fixed directional interpolation filter and adaptive distance weighting scheme,” *Optical Engineering* **50**(6), 067008–067008 (2011).
- 17 Q. Huang, D. Zhao, S. Ma, W. Gao, and H. Sun, “Deinterlacing using hierarchical motion analysis,” *Circuits and Systems for Video Technology, IEEE Transactions on* **20**(5), 673–686 (2010).
- 18 T. A. Ell and S. J. Sangwine, “Hypercomplex fourier transforms of color images,” *Image Processing, IEEE Transactions on* **16**(1), 22–35 (2007).
- 19 C. M. Zwart, R. Venkatesan, and D. H. Frakes, “Decomposed multidimensional control grid interpolation for common consumer electronic image processing applications,” *Journal of Electronic Imaging* **21**(4), 043012–043012 (2012).
- 20 D. H. Frakes, C. P. Conrad, T. M. Healy, J. W. Monaco, M. A. Fogel, S. Sharma, M. J. Smith, and A. P. Yoganathan, “Application of an adaptive control grid interpolation technique to morphological vascular reconstruction,” *IEEE Transactions on Biomedical Engineering* **50**, 197–206 (2003).
- 21 R. Venkatesan, C. M. Zwart, and D. H. Frakes, “Video deinterlacing with control grid interpolation,” in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 861–864, IEEE (2012).
- 22 P. Seeling, F. H. Fitzek, and M. Reisslein, *Video traces for network performance evaluation: a comprehensive overview and guide on video traces and their utilization in networking research*, Springer (2007).

- 23 M. Pinson, “The consumer digital video library [best of the web],” *Signal Processing Magazine, IEEE* **30**(4), 172–174 (2013).
- 24 D. M. Chandler and S. S. Hemami, “Vsnr: A wavelet-based visual signal-to-noise ratio for natural images,” *Image Processing, IEEE Transactions on* **16**(9), 2284–2298 (2007).
- 25 H. R. Sheikh, A. C. Bovik, and G. De Veciana, “An information fidelity criterion for image quality assessment using natural scene statistics,” *Image Processing, IEEE Transactions on* **14**(12), 2117–2128 (2005).
- 26 G. De Haan and E. B. Bellers, “Deinterlacing—an overview,” *Proceedings of the IEEE* **86**(9), 1839–1857 (1998).

List of Figures

- 1 Example of poor deinterlacing from a high-definition YouTube video.
- 2 Neighborhoods for STELA and ELA.
- 3 Mother video (left) and the detected saliency (right) after thresholding by $B=4\%$ of the bit depth.
- 4 Interlace type 1: one frame is split into two fields.
- 5 Interlace type 2: each frame gets interlaced into its own respective field.

- 6 Video screenshots corresponding to different algorithms. From left to right are original, VTF, SRVTF, HDD, and SDD. From top to bottom are original and deinterlaced versions of frame 2 from foreman, akiyo, galleon, and students videos. The performance of the proposed approaches can be best appreciated on the edges of the wall and within the Siemens logo in the foreman video (top), and on the edges of the table in the students video (bottom).
- 7 Spatially deinterlaced non-salient region of video noticed by different subjects. The heat map (hotter the region, longer the gaze) shows the gaze locations of the subjects, for the first 33 frames after the deinterlacing is introduced. This experiment illustrated the fact that any non-salient region of the video that are spatially deinterlaced leads to viewer discomfort. This observation is also supported by the MOS scores.
- 8 Temporally deinterlaced non-salient regions of video (such as the wall in the background) missed by different subjects. The heat map (hotter the region, longer the gaze) shows the gaze locations of the subjects, for the first 33 frames after the deinterlacing is introduced. This experiment illustrated the fact that any non-salient region of the video that are temporally deinterlaced doesn't affect viewing. This observation is also supported by the MOS scores.

List of Tables

- 1 Table of PSNR. All of the methods in this table were implemented by the authors. Care was taken to ensure that the methods were implemented to the finest detail provided in the respective source papers.

- 2 Table of r-values using reported results.
- 3 Table of VSNR values. All of the methods in this table were implemented by the authors. Care was taken to ensure that the methods were implemented to the finest detail provided in the respective source papers.
- 4 Table of VIF values. All of the methods in this table were implemented by the authors. Care was taken to ensure that the methods were implemented to the finest detail provided in the respective source papers.
- 5 Table of Mean Opinion Scores (MOS). NS - Non-Salient; S-Salient. For the Crew video there was no temporal deinterlacing for non-salient regions.