

MULTIDIMENSIONAL ADAPTIVE NON-LINEAR FILTERS FOR CONCEALMENT OF INTERLACED VIDEO

Till Halbach and Tor A. Ramstad

Department of Telecommunications
Norwegian University of Science and Technology (NTNU)
Email: {halbach,tor}@tele.ntnu.no

ABSTRACT

The application of multidimensional adaptive non-linear filters for interpolation of interlaced imagery is investigated. Moreover, a new scheme for combination of several data estimates by weighting based on one-sample correlation is proposed. This technique achieves superior performance when combining vertical and temporal signals at a moderate complexity increase. In the spatial domain, one-dimensional non-adaptive Lagrange interpolators are found to provide sufficient objective and subjective image quality as compared to two-dimensional linear and non-linear filters, and median-type filters.

1. INTRODUCTION AND OUTLINE

Natural digital video comes mostly in interlaced form. Interlacing a single picture/frame of a video means to split the lines of samples/pixels into two fields. Odd-indexed lines are usually assigned to the top field, and lines of even index are counted belonging to the bottom field [1, 2]; see also Fig. 1(a).

When top and bottom fields are independently transmitted – e.g. with IP or ATM packet-based networks, by putting each field into a single packet – their high correlation can be exploited for error concealment in case of packet loss. The task is then to estimate/interpolate the lost field(s) such that the quality of the reconstructed picture is closest possible to its corresponding original. The worst-case scenario is when information of both the current frame’s two fields is completely lost due to packet deletion. Then, temporal concealment is the only possibility for estimation of lost data. In case only one field is lost but the other one is preserved (see Fig. 1(b)), the signal of the lost field is estimated from the correctly received field by smart filtering. Several methods are developed in the subsequent sections and compared to each other with respect to objective and subjective performance.

The paper is organized as follows. First, various one-dimensional interpolation methods are discussed. It is then

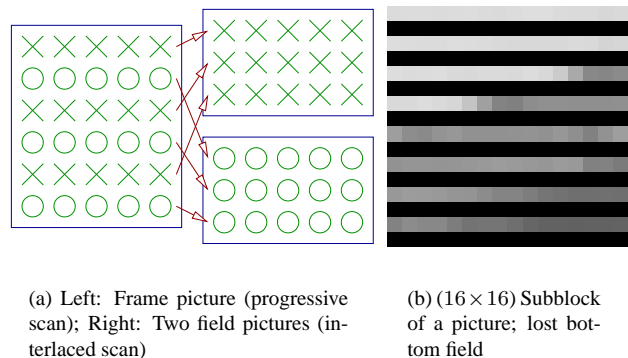


Fig. 1. Interlaced picture/block

investigated whether and how these schemes can be extended to two- and three-dimensional filters. Finally, the developed techniques are compared in performance to median-type filters.

2. INTERPOLATION BY MULTIDIMENSIONAL FILTERS

Fig. 1(b) shows a block of a picture in interlaced coding mode after complete loss of the bottom field. All even numbered lines belong to the lost bottom field and are zero, while odd numbered lines belong to the correctly received and decoded top field. This scenario is – not only for a single block but the whole picture – considered to evaluate different approaches to efficient error concealment. Comparisons of different concealment schemes are exemplified by means of a subset of luminance pictures from the *Stefan* sequence. The PSNR values quantify the error made by the respective concealment method with respect to the original video sequence. These methods depend strongly on the assumed underlying signal model.

2.1. 1-D spatial filtering

A simple concealment approach is to assume vertical pixel correlation. Then, concealment is identical to upsampling the columns of pixels of the top field by a factor of 2. A

trivial but efficient solution is nearest-neighbor interpolation (pixel line replication/copy). This achieves a PSNR of 23.08 dB for the test picture.

A better but also somewhat more complex solution is the use of Lagrange interpolators [3]. In finite-length Lagrange interpolation, a $(N - 1)$ th-order polynomial through known N data points is defined, which allows to obtain arbitrary points on the curve defined by this polynomial. Assuming discrete-time uniform samples, a digital interpolator for the signal $x(i)$ is accomplished as

$$y(n) = \sum_{i=1}^N \left[\prod_{\substack{j=1 \\ j \neq i}}^N \frac{n-j}{i-j} \right] x(i) = \sum_{i=1}^N \phi_i^{(L_{N-1})}(n) x(i),$$

where ϕ_i is the set of linearly independent basis functions, here chosen to be discrete Lagrange interpolation functions (symbolized by the superscript L).

By this method, a symmetric FIR filter is derived, indicating linear phase. To achieve continuity, N must be even, which gives odd filter length. The interpolated sample is always obtained from the middle of the polynomial fit due to the curve's highest accuracy around this point. A realization of a sliding Lagrange filter is depicted in Fig. 2. The filter coefficients for a second-order Lagrange interpolation filter are $h^{(L_1)}(n) = (0.5, 1, 0.5)$. Such a filter is very simple to realize in FIR structures by bit shifts. However, the filter's frequency response $|H(e^{j\omega})|$ is far from being an ideal low-pass filter; it should thus only be used if the variation from one vertical sample to the other is small or, in other words, if the signal's bandwidth is small compared to the sampling frequency. However, its performance is with a 27.68 dB PSNR quite fair. The signal has been mirrored at its borders beforehand to minimize edge effects.

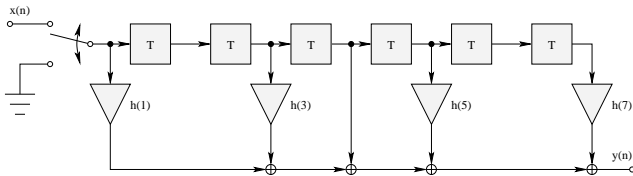
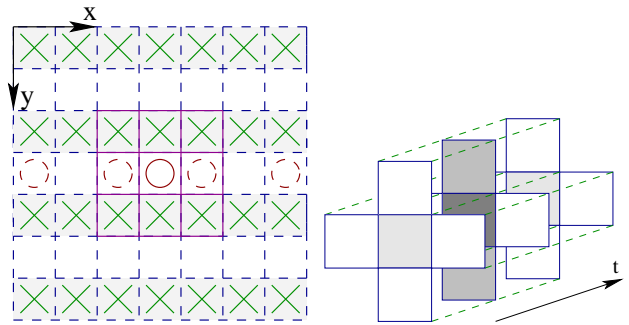


Fig. 2. Moving-window sixth-order FIR filter

A discrete sixth-order Lagrange interpolation filter has the impulse response $h^{(L_3)}(n) = (-0.0625, 0, 0.5625, 1, \dots)$. All coefficient multiplications (except one and zero) are easily implemented by 4 bit right shifts and additional fixed-point multiplications by 9 for $h^{(L_3)}(3, 5)$. This results in a PSNR of 28.09 dB. Our simulations show that the sixth-order filter performs generally best with the test images (listed in Sec. 2.3). It is further noted that bit shifts cannot be employed for the coefficients $(0.0117, 0, -0.0977, 0, 0.5859, 1, \dots)$ of the tenth-order filter.

2.2. 2-D spatial filtering

In this section, the simple 1-D model is extended to a spatial 2-D model to exploit horizontal pixel correlation as well. According to a widely accepted spatial-correlation model, a natural image can be well modeled as a first-order AR process in both directions, with the nearest-sample correlation coefficient $\rho \approx 0.95$. The AR(1) model advocates that a second-order filter provide sufficient gain. The filter is applied vertically first, according to the 1-D case, which results in a vertical estimate for each missing row; see Fig. 3(a). Then, a horizontal Lagrange filter is employed to these rows, providing a second estimate. Assuming equal pixel correlation in both picture dimensions, both estimates are finally averaged arithmetically.



(a) Spatial interpolation by second-order (solid-line rectangle) and sixth-order (dashed-line rectangle) Lagrange filters

(b) Vertical and temporal second-order interpolation; dark-shaded: pixel to be estimated; light-shaded: used for estimation

Fig. 3. 1-D, 2-D and 3-D signal models; the location of the pixel to estimate is illustrated by a solid-line circle

The approach of horizontal and vertical filtering results in a PSNR of 27.12 dB, which is worse than the result obtained for the same filter and a vertical correlation model. The reason for this is that, e.g. at edges and in regions of high picture activity, low correlation 'overrides' high correlation. It is therefore a better solution for computation of the desired estimate to weight the models of vertical and horizontal correlation differently according to their actual amounts. The assumption of vertical and horizontal correlation mirrors hereby quite nicely the statistics of real-world video material, i.e. vertical and horizontal edges. The correlation in both dimensions is in a simple but efficient manner approximated by the absolute difference of neighboring field samples.

As seen above, the 2-D case produces vertical and horizontal estimates for lost samples $y_{i,j}$ from the received field samples $x_{i,j}$, denoted by e_v and e_h , respectively. A new estimate e for one sample y is now defined as the weighted sum of estimates $e = w_v \cdot e_v + w_h \cdot e_h$, with the weight-

2-D filtering method	2-D Gain	3-D filtering method	3-D Gain
Vert. near.-neighbor	16.24	Temp. near.-neighbor	20.31
Vert. 2nd-o. Lagr.	20.25	Temp. 2nd-o. Lagr.	23.91
Vert. 6th-o. Lagr.	20.45	Weig. temp. 2nd-o. + vert. 2nd-o. Lagr.	25.60
Vert. 10th-o. Lagr.	20.42	Sw. temp. 2nd-o.+ vert. 2nd-o. Lagr.	25.03
Vert. + hor. 2nd-o. Lagr.	19.47	Weig. temp. 2nd-o.+ vert. 6th-o. Lagr.	25.83
Weig. vert.+ hor. 2nd-o. Lagr.	19.56	Sw. temp. 2nd-o.+ vert. 6th-o. Lagr.	25.15
Switched vert.+ hor. 2nd-o. Lagr.	19.31	Weig. temp. 2nd-o. + vert. 10th-o. Lagr.	25.86
MED1	19.36	Sw. temp. 2nd-o.+ vert. 10th-o. Lagr.	25.13
MED3	20.08		

Table 1. Gain of interpolation techniques averaged over all test images

ing factors $w_i = 0, \dots, 1$. Vertical and horizontal distance variables d_i are further defined as $d_v = |x_{1,1} - x_{2,1}|$ and $d_h = |x_{1,1} - x_{1,2}|$, and according to the distorted image

$$\begin{bmatrix} x_{1,1} & x_{1,2} \\ y_{1,1} & y_{1,2} \\ x_{2,1} & \cdot \end{bmatrix},$$

treating a one-sample distance along a row and a two-sample distance along a column equally. By means of these definitions, d is inversely proportional to the actual correlation. $d_v = 0$ means a high vertical correlation. Hence, e has to be chosen close to e_v , $e = e_v$, which in turn requires $w_v = 1$ and $w_h = 0$. $d_h = 0$ means a high horizontal correlation, $e = e_h$, and $w_h = 1$ as well as $w_v = 0$. The relationship between the two estimates is therefore $w_v = 1 - w_h$. The only way to satisfy these given requirements is to set $w_v = d_h \frac{1}{d_v + d_h}$ and $w_h = d_v \frac{1}{d_v + d_h}$. In order to evaluate d_i and hereby w_i , the original field matrix has in practice to be extended (mirrored) by one row at the bottom and one column at the right side, hereby minimizing edge effects. There is the special case where both $d_h = 0$ and $d_v = 0$. Then one estimate (naturally the vertical one) is chosen to be best by assigning the maximum possible distance to the horizontal estimate, e.g. 255 for 8-bit pixel representation. Applied to the reference picture, a PSNR of 27.04 dB is yielded. The reason for this is that the horizontal estimates are based on the vertical estimates, which in turn means a more inaccurate result. As a last strategy, estimate weighting is replaced by simply switching between the vertical and horizontal estimates according to which distance variable is smaller. The PSNR here computes to 26.83 dB.

The multiple-estimate model can easily be further extended to any desired number K of directions, e.g. utilizing diagonal estimates/filtering additionally, by defining the sum $D = \sum_{k=1}^K d_k$ and the weighting factors $w_j = \frac{D - d_j}{D}$. It is stressed that the filtering process for both weighting and switching is adaptive and thus non-linear.

Multiple estimation means a higher complexity than e.g. plain vertical filtering. For every added direction, a distance

function has to be calculated (one addition per pixel). Besides normal filtering in each dimension K , the weighting factors require additionally K additions and one multiplication per pixel. Finally, the total estimate demands K additions and K multiplications per pixel.

Finally, we compare the performance of vertical interpolation to two median-type filters [4]. The filters are similar to the ones in [5] but differ in that they account for the interlaced subsampling lattice. The first one is a pure median filter with $y_{i,j}^{(\text{MED1})} = \text{MEDIAN}(x_{i-1, \{j-1, j, j+1\}}, x_{i+1, \{j-1, j, j+1\}})$ (based on 6 samples), and the second one is a recursive linear median hybrid filter defined by $y_{i,j}^{(\text{MED3})} = \text{MEDIAN}(x_{i-1, \{j-1, j, j+1\}}, x_{i+1, \{j-1, j, j+1\}}, x_{i,j}^{\text{FIR}}, \bar{x}_V)$, $\bar{x}_V = (x_{i-1, j} + x_{i+1, j})/2$, and where $x_{i,j}^{\text{FIR}}$ is the sample obtained by filtering $x(n)$ with $[h_1, h_2, h_2, h_1]$, $h_1 = [0, -1/8, -1/8, 0]^T$, and $h_2 = [-1/8, 1/2, 1/2, -1/8]^T$. Both median-type filters show, though more complex, subjectively the same quality as tenth- or even second-order vertical Lagrange interpolators. Objectively, $y^{(\text{MED1})}$ performs, averaged over all test images as specified in Sec. 2.3, 1.06 dB worse, and $y^{(\text{MED3})}$ 0.34 dB worse than vertical tenth-order filters.

2.3. 3-D filtering (Spatial and temporal filtering)

In natural video material with high picture rates, temporal correlation is often higher than spatial correlation. This means that, in addition to the 2-D case, also information in temporally preceding and/or succeeding pictures can be used for error concealment. For this, the *Stefan* still image example is extended to a short sequence of three pictures. The first simple strategy is temporal nearest-neighbor interpolation. The PSNR value of 24.15 dB shows that this method performs better than the vertical nearest-neighbor scheme. However, sequences with much motion are hereby poorly predicted. Temporal Lagrange interpolation of second order, based on the pixels of the adjacent two frames at the same spatial position, achieves 28.17 dB in PSNR. Larger number of frames, as e.g. necessary for sixth- and

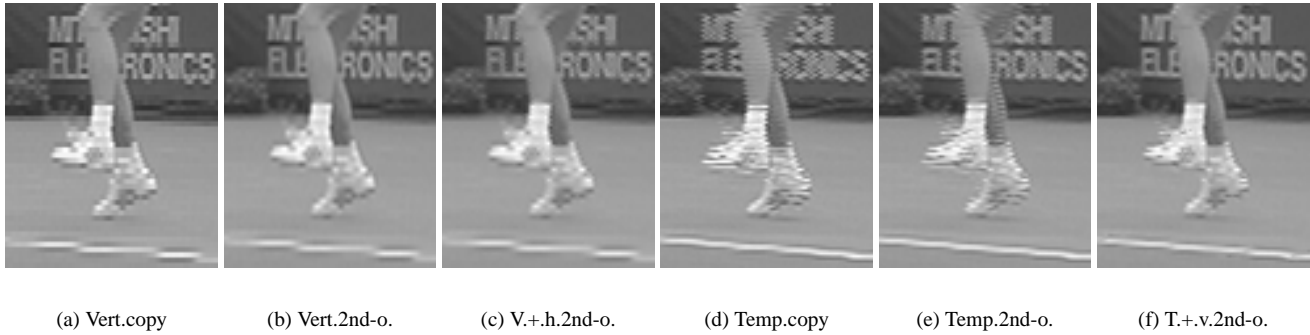


Fig. 4. Visual comparison of various concealment methods

tenth-order Lagrange filtering are not investigated due to the inherent latency of the algorithm and possible delay constraints.

The computational complexity can be increased further by allowing two estimates, a vertical and a temporal one, which are weighted or switched to form the total estimate as described above in Sec. 2.2. For illustration, see Fig. 3(b). In the special case $d_v = 0$ and $d_t = 0$, the temporal estimate is chosen to be optimum. The vertical second-, sixth-, and tenth-order Lagrange interpolators lead (with weighted estimates) to the PSNRs 30.41 dB, 30.74 dB, and 30.83 dB, respectively. Switching does not improve the PSNR of the reconstructed picture. Tab. 1 lists the filtering gain/difference (in dB) with regard to the unfiltered picture, averaged over the 30-fps CIF subsequences (with picture index) *Foreman* (40), *Mobile* (99), *Container* (10), *News* (10), *Tempete* (150), *Bus* (51), *Flower* (4), *Hall Monitor* (26), *Salesman* (11), *Silent* (51), and *Stefan* (106). In the 3-D case, both previous and next picture are used additionally, considering indices 39 – 41 for instance for *Foreman*.

The visual evaluation (Fig. 4) shows that low-motion objects in the picture (e.g. the background) are better temporally interpolated than spatially. Fast moving objects, on the other hand, have a higher quality when solely spatially estimated. The joint temporal-vertical interpolation means hereby a good compromise: Low-motion data is well approximated, and high-motion data shows moderate to good quality. This feature fits further nicely into the perception properties of the Human Visual System which demands high quality for still picture content but not for quickly moving objects.

3. CONCLUSIONS AND OUTLOOK

In the two-dimensional case, we can conclude that the technique of combined 2-D vertical and horizontal filtering with included weighting/switching does not justify its increased complexity as compared to vertical sixth-order filtering. Vertical interpolation further outperforms non-linear filtering

with almost all test images. The use of median-type filters can therefore not be recommended because of their high complexity. Many-tap vertical interpolators yield usually better results than short-tap filter, but this depends strongly on the source statistics. Visually, second-order filters provide satisfactory results.

Considering the three-dimensional case, a combination of vertical and temporal filtering is best for the interpolation of video. The use of spatial filters of higher orders is advantageous. Especially tenth-order filters yield the best results when weighted with temporal estimates. The use of temporal information for interpolation increases the performance typically by 3–6 dB as compared to spatial filtering.

The application of adaptive non-linear filters is of course not limited to error concealment; the filters proposed here may also be used for e.g. data compression, sampling rate alterations, image zooming, and video scan conversion.

4. REFERENCES

- [1] Khalid Sayood, *Introduction to Data Compression*, Morgan Kaufmann Publishers, San Francisco (CA, USA), 2000.
- [2] Peter Borgwardt, “Handling interlaced video in H.26L,” Tech. Rep. N57r2, ITU-T Q.6/SG 16 (VCEG), Sept. 2001.
- [3] Tor A. Ramstad, “Digital methods for conversion between arbitrary sampling frequencies,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, no. 3, pp. 577–591, June 1984.
- [4] Jaakko Astola, Pekka Heinonen, and Yrjö Neuvo, “Linear median hybrid filters,” *IEEE Trans. Circuits, Syst.*, vol. 36, no. 11, pp. 1430–1438, Nov. 1989.
- [5] Bing Zheng and Anastasios N. Venetsanopoulos, “Image interpolation based on median-type filters,” *Opt. Eng.*, vol. 39, no. 9, pp. 2472–2482, Sept. 1998.