NONLINEAR POLYPHASE IMAGE RATE CONVERSION

O. Franzen, H. Schröder

University of Dortmund, Circuits and Systems Lab, Dortmund, Germany

ABSTRACT

A new algorithm for the motion vector based interpolation of temporal intermediate images is proposed in this paper. This algorithm uses dedicated weighted median (WM) filters for each interpolation phase (polyphase WM filters) and therefore allows a correction of faulty estimated motion vectors up to a certain degree. Since motion estimation on natural image scenes always suffers from errors in the estimated motion vector field, this robustness of the proposed algorithm against motion vector errors is essential to achieve a high interpolation quality. A new design method for error tolerant polyphase WM filters is presented in the paper. Combining the polyphase WM filters with a spatial nonlinear band separation allows a combination of the error tolerance of the WM filters with the detail preservation advantages of other interpolation techniques. The results of the proposed algorithm are compared to existing interpolation algorithms.

1. INTRODUCTION

As there exist lots of different video standards with specific spatial as well as temporal sampling rates, format conversion between different standards is an important task in the field of video signal processing. Since format conversion requires the interpolation of (spatial and/or temporal) intermediate samples, the quality of the interpolation result strongly depends on the utilized interpolation algorithm. In the following we will focus on temporal interpolation techniques.

In order to achieve a high subjective interpolation quality in still as well as in moving areas of the image, motion vector based interpolation algorithms have to be used as this allows a correct positioning of moving objects in the intermediate images to be interpolated. However, due to noise, coverage/uncoverage, violations of the motion parameter model and other degradations motion estimation for natural images scenes always suffers from errors in the estimated motion vector field. Therefore it is of great significance that the interpolation algorithm possesses a certain robustness against erroneous motion vectors.

2. VECTORBASED IMAGE INTERPOLATION

In a temporal image format conversion, different interpolation phases depending on the input and output image repetition rate can be distinguished. For the conversion task from 50 Hz to 100 Hz which is very popular in Europe for flicker reduction the output image is either at the same temporal position as the input image or it is temporally located exactly in the middle of two input images. However, other conversions imply much more different interpolation phases. For instance regarding the conversion from 50 Hz to 60 Hz six different interpolation phases can be distinguished which substantially differ in the temporal position of the output image according to the temporally neighboring input images. Regarding an output image (denoted by Φ_k) temporally located between the previous input image F_{n-1} and the following input image F_n the so called projection factor p_{left} can be defined by the temporal distance of the output image to the previous input image normalized with the temporal distance between two successive input images. A second projection factor p_{right} being complementary to p_{left} can be defined by $p_{right}=1-p_{left}$. As discussed in [3], a temporal image rate conversion can be divided into k_{max} different cyclically repeating interpolation phases each characterized by the corresponding projection factor, with k_{max} being dependent on the input and output image repetition rates.

2.1. Intermediate image interpolation algorithms

The basic principle of motion vector based intermediate image interpolation algorithms is illustrated in Fig. 1. Starting from the pixel to be interpolated a positioning of the filter masks in the previous and following input picture is done using the projected motion vector. In addition, recursive elements resp. pixels can be included in the filter. The different interpolation algorithms basically differ in the interpolation filter or the interpolation filter algorithm being applied. Apart from simple interpolation techniques, like e.g. pixel shifting or weighted pixel averaging, more sophisticated filter were proposed in the literature. Several algorithm proposal are presented in [4], e.g. based on a 3-tap median filter. Advanced interpolation concepts which make use of a median based choice among different interpolation results are also presented in [4]. In case of faulty motion vectors these advanced algorithms can lead to a significantly better suppression of the visibility of interpolation artifacts, when compared to simple linear interpolation techniques, because they revert to a suitable (e.g. static) fallback mode. Other interpolation techniques based on median filters are presented in [5].

2.2. Intermediate image interpolation with WM filters

The different interpolation techniques, e.g. presented in [4], differ among other things in their fallback mode attempting to suppress strongly visible interpolation artifacts. A *correction* of faulty motion vectors (in terms of a correct positioning of moving objects according to the motion phase in face of a faulty motion vector estimation result) allows none of them. In contrast, the interpolation algorithm presented in [1] which is based on weighted median (WM) filters allows a correction of faulty motion vectors up to a certain degree for the conversion from 50 Hz to 100 Hz. This algorithm used for the temporal interpolation of the intermediate image Φ_k can be explained using Fig. 1. The output of the interpolation filter is given by a weighted median including input pixels covered by the vector addressed weight masks W_{n-1} in the previous as well as W_n in the following input image. In addition, already calculated pixels of the actual output image covered by the weight mask W_Φ can be included in the weighted median processing as recursive filter elements. The calculation of the output pixel $\Phi_{WM}(\vec{x})$ can be expressed by

$$\Phi_{wm}(\vec{x}) = \text{med}\begin{Bmatrix} W_\Phi \lozenge \Phi_{wm}(\vec{x}); W_{n-1} \lozenge F_{n-1}(\vec{x} - p_{left} \cdot \vec{v}_{est}); \\ W_n \lozenge F_n(\vec{x} + (1 - p_{left}) \cdot \vec{v}_{est}) \end{Bmatrix}$$

with all pixels covered by one of the masks being fed into the filter with duplication according to the corresponding filter weight.

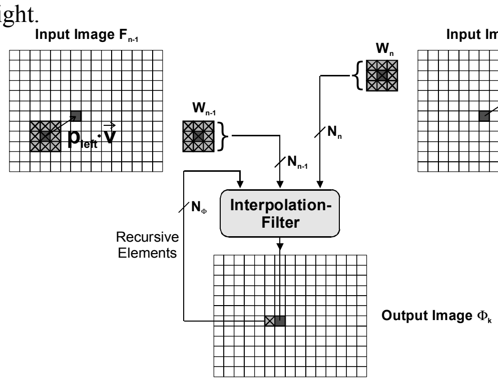

Figure 1: Vector based intermediate image interpolation

2.2.1. Correction of vector errors by WM filters

For the sake of clearness the correction of faulty motion vectors by WM filters should be explained using a pure 1D interpolation problem for a horizontally moving edge. At first, an intermediate "image" interpolation for the interpolation phase p_{left}=1/2 being characteristic for a conversion from 50 Hz to 100 Hz is considered. It is assumed, that the edge moves with a velocity v_{real}=6 pixel/image whereas the motion estimator estimated a faulty motion v_{est}=0 pixel/image. According to the estimated vector being zero, the interpolation filter masks are positioned centered to the position to be interpolated both in the previous and following input image, as depicted in Fig. 3. If linear interpolation filters are used, a blurring of the edge in the intermediate image is inevitable. But using a WM (here W_{n-1}=W_n={1,1,1,7,1,1,1}) filter, the moving edge is correctly interpolated at the position $0.5{\cdot}v_{real}$ of half the horizontal motion although an incorrect velocity v_{est}=0 was used to position the interpolation masks. This means, that in the example presented here an estimation error of Δx=6 with Δx=v_{real}–v_{est} can be corrected by the WM filter. This error correction is based on the shift property of WM filters which is explained more detailed in [1].

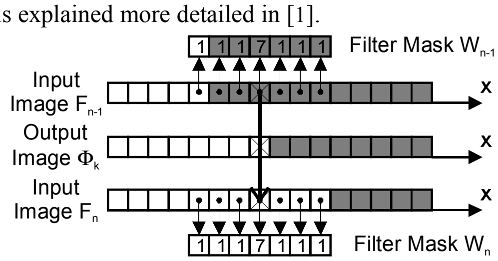

Figure 2: Interpolation with WM filters

3. POLYPHASE WM INTERPOLATION FILTERS

The interpolation algorithm proposed in [1] allows a positioning of moving objects on half of the real motion vector displacement even in presence of motion estimation errors and is therefore able to surpass other algorithms for the conversion from 50 Hz to 100 Hz. However, other interpolation phases require a positioning not on half but on another fraction (given by p_{left}) of the full displacement, as demonstrated by the following example, which again regards the horizontally moving edge with v_{real}=6 and a faulty vector estimation of v_{est}=0. In contrast to the previous example with p_{left}=1/2, an interpolation phase p_{left}=1/6 (e.g. for a conversion from 50 Hz to 60 Hz) is now regarded. An interpolation using the filter mask depicted in Fig. 2 leads to a positioning of the moving edge at half of the real motion vector displacement in the interpolated image, what would mean a correct object positioning for p_{left}=1/2, but a mispositioning according to the actual interpolation phase p_{left}=1/6 which requires a positioning of moving objects on one sixth of the real motion. In contrast, the WM filter masks depicted in Fig. 3 being adapted to the actual interpolation phase allow this correct positioning, although an estimation error of Δx=6 again was assumed.

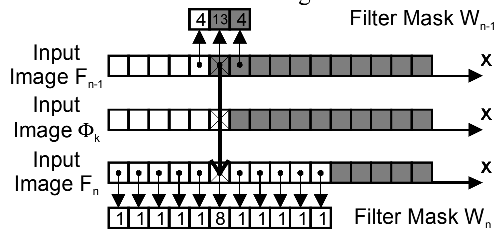

Figure 3: Interpolation with polyphase WM filters

The presented example clarifies that the use of dedicated WM filters for each interpolation phase (polyphase WM filters) is mandatory in order to achieve a correct positioning of moving edges (which serve as model for the boundaries of large moving objects) in spite of faulty motion vector estimations.

3.1. A new design technique for polyphase WM filters

As there did not exist suitable design method for median based interpolation filters, heuristically designed filters have been used in interpolation algorithms presented in the literature[1,2,4,5]. Therefore a new design technique for vector error tolerant WM filters is presented in this paper, consisting of two design steps. In the first step, the conditions for the filter weights are derived, which have to be fulfilled for a filter being be able to correct faulty motion vectors. In the second step, an optimum interpolation filter is designed using integer programming.

3.3.1. Deriving the conditions for the filter weights

In order to design a vector error tolerant weighted median interpolation filter, the constraints on the filter coefficients to achieve the demanded error tolerance have to be formulated. The notation used in this paper is explained using Fig. 4. Again a moving edge with v_{real}=6 and the interpolation phase with p_{left}=1/2 is considered, whereas in contrast to recent examples an estimated velocity of v_{est}=3 is assumed. Let x_c denote the correct position of the moving edge in the output image Φ_k and x_{n-1} resp. x_n denote the position of the moving edge (according to the real velocity v_{real}) in the input images F_{n-1} resp. F_n. The positions f_{n-1} and f_n are the central positions of the filter masks in the previous and following input image in order to interpolate the output pixel position x_c. Then a "Difference of the Central Position of the filter mask" ΔCP for the previous and following image can be defined as $\Delta CP_n = x_n - f_n$ resp. $\Delta CP_{n-1} = f_{n-1} - x_{n-1} = \Delta x - \Delta CP_n$.

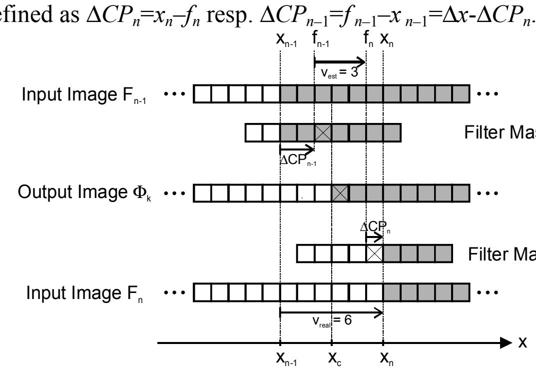

Figure 4: Notation for the filter design

For a spatially continuous image representation $\Delta CP_n = p_{right} \cdot \Delta x$ and $\Delta CP_{n-1} = p_{left} \cdot \Delta x$ holds and therefore ΔCP_{n-1} and ΔCP_n can be interpreted as the estimation error projected on the previous and following images. But as the new filter design technique is based on a spatially discrete pixel scheme, a mapping of fractional values of f_{n-1}, f_n, x_{n-1} and x_n to integer positions is presupposed in the following considerations. The definition $\Delta CP_{n-1} = \Delta x - \Delta CP_n$ guarantees that the sum of ΔCP_{n-1} and ΔCP_n equals the total estimation error. The notation for the weights of the filter masks in the following image is depicted in Fig. 5 with L_n denoting the filter expansion measured from the central position (resulting in a total filter mask length of $2L_n+1$). The notation for the filter masks for the previous image is simply given by replacing the index n by $n-1$. Recursive filter elements are not regarded in the filter design process.

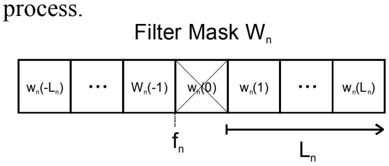

Figure 5: Notation for the filter masks

Let for the filter design the moving edge be modeled as a binary transition from a high luminance ($F(x)=1$, denoted by "H") to a low luminance ($F(x)=0$, denoted by "L"). Due to the stacking property of weighted median filters this model represents all other ideal transitions between arbitrary luminances. With x_c being the correct edge position (refer to Fig. 4) in the intermediate image according to the actual interpolation phase, for horizontal positions in the output image being smaller than x_c a high luminance (H) has to be interpolated whereas for positions greater or equal than x_c the luminance to be interpolated is low (L) as depicted in Fig. 6.

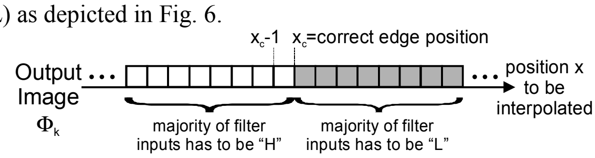

Figure 6: Constraints for correct edge interpolation

These constraints for the filter coefficients for achieving a correct edge positioning in the output image depicted in Fig. 6 can be formulated as

$$x = x_c: \quad \sum_{i<-\Delta CP_{n-1}} w_{n-1}(i) + \sum_{i<\Delta CP_n} w_n(i) < \sum_{i\ge -\Delta CP_{n-1}} w_{n-1}(i) + \sum_{i\ge \Delta CP_n} w_n(i)$$

$$x = x_c - 1: \quad \sum_{i<-\Delta CP_{n-1}+1} w_{n-1}(i) + \sum_{i<\Delta CP_n+1} w_n(i) > \sum_{i\ge -\Delta CP_{n-1}+1} w_{n-1}(i) + \sum_{i\ge \Delta CP_n+1} w_n(i)$$

using the notation introduced in Fig. 4 and Fig. 5.

3.3.2. Filter design by integer programming

The conditions for a correct interpolation of the moving edge can be formulated by two inequalities for the filter weights only depending on the projected estimation errors ΔCP_{n-1} resp. ΔCP_n. The filter design task addressed in this paper can be formulated as: "Design an error tolerant filter which (for the estimated velocity v_{est}) is able to fully correct a motion estimation error of Δx_{max} for the interpolation phase with a given p_{right}". Fully correction of the motion estimation error Δx_{max} for an estimated velocity v_{est} means, that for all real velocities v_{real} in the range of $[v_{est} - \Delta x_{max}; v_{est} + \Delta x_{max}]$ the edge is interpolated at the correct position, which leads to a set of $2\Delta x_{max}+1$ pairs of inequalities. All weighted median filters whose weights fulfill the whole set of $4\Delta x_{max}+2$ inequalities are solutions of the filter design task formulated above. In order to achieve a hardware expense being as low as possible, a minimization of the sum of the filter weights seems appropriate. This leads to the formulation of the filter design problem as an integer program: "Minimize the sum of the filter coefficients under the constraints for the motion vector error correction".

3.3.2. 2D filters and additional filter constraints

The design method presented above can easily be extended for the design of 2D interpolation filters and arbitrary additional constraints (like e.g. detail preservation in case of a correct motion vector) can easily be added as long as they can be formulated as linear constraints for the filter coefficients.

4. NONLINEAR SPATIAL BAND SEPARATION

By including the additional constraints mentioned in the previous section in the filter design process, it is guaranteed, that in case of a correction motion estimation even very fine image details can be preserved during the interpolation process. However, if the motion vector is faulty, and the error correction ability of the filter is stipulated, small image details can be suppressed by the interpolation filter. As the filter design is based on the model of a moving edge and the error correction ability therefore only holds for objects being larger than the filter masks, a spatial band separation is an appropriate extension for the proposed algorithm in order to preserve images details being too small to match the moving edge model. Using median separation filters, it can be achieved, that large objects (and especially the object edges) are completely extracted in the "lows" channel and fine image details are extracted in the "highs" channel. In contrast, linear separation filter would lead to an unfavorable distribution of e.g. edges on both bands and therefore imply the risk of loosing image sharpness.

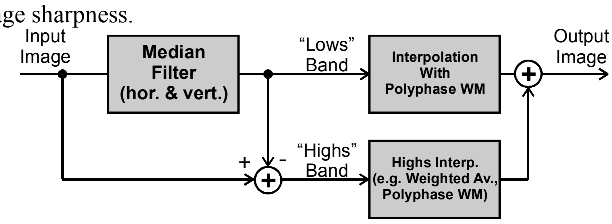

Figure 7: Nonlinear band separated interpolation architecture

Using median based band separation filters, the above mentioned polyphase WM filters for the "lows" channel interpolation and a technique with a spatially less extended filter (e.g. polyphase WM with lower correction interval or weighted average) for the "highs" channel interpolation, a further improvement of the interpolation quality can be achieved. The corresponding interpolation architecture is show in Fig. 7.

5. SIMULATION RESULTS

Some exemplary interpolation results for different intermediate image interpolation algorithms are depicted in Fig. 8. Apart from the adjacent input pictures F_{n-1} and F_n the interpolation result Φ_{dyn} of a so called "dynamic" 3-tap median (of two vector addressed input pixels and one non vector addressed average value as fallback pixel) according to [4], the result Φ_{WM} of a weighted median filter according to [1] and the result $\Phi_{\text{poly-WM}}$ of the proposed polyphase WM filters (with $\Delta x_{max}=\Delta y_{max}=4$) are shown in Fig. 8. It can be seen, that the dynamic median falls back to the non vector based averaged input pixel in moving areas with faulty estimated motion vectors and therefore leads to a blurring of moving objects in this areas. The WM filter does not blur the image, but it leads to a mispositioning of the moving objects and tends to a deletion of parts of the moving objects in areas with faulty motion vectors. In contrast, the polyphase WM allows a correct positioning and avoids blurring also in the critical areas and therefore visibly achieves the best interpolation results.
The influence of including a band separation is depicted in Fig. 9. The band separation was done by cascaded 5-tap horizontal and vertical median filters. For the $\Phi_{\text{poly-WM,sep}}$ result, a polyphase WM interpolation with $\Delta x_{max}=\Delta y_{max}=4$ was applied in the lows channel, in the highs channel a weighted average interpolation was used. This result using band separation is compared to the result Φ_{dyn} of a 3-tap median and $\Phi_{\text{poly-WM}}$ of a polyphase WM interpolation without band separation, whereby for the sake of clarity of the presentation a zero vector field was assumed for all the three compared interpolation techniques. It can be seen, that the band separation approach can avoid double contours (unlike the 3-tap median) at the football player due to the error correction ability, but on the other side allows a better detail preservation (e.g. in the grass area) in comparison with the non band separated polyphase WM approach.

6. SUMMARY AND CONCLUSION

A new algorithm for the motion vector based temporal intermediate image interpolation using dedicated weighted median filter for each interpolation phase (polyphase WM) is introduced in this paper. In addition, a new filter design technique is presented, allowing a fast and flexible design of such polyphase WM interpolation filters which are able to correct errors in the motion vector field up to a given degree. The polyphase WM is compared to other existing interpolation technique and can achieve a significantly better interpolation result due to its vector error correction ability. A combination of the proposed algorithm with nonlinear spatial band separation techniques allows a further improvement of the interpolation quality by combining the vector error correction ability of the polyphase WM with the superior detail preservation of algorithms with spatially less extended interpolation filters. Future research will be addressed to an adaptation of the filter characteristics to the image and vector field properties.

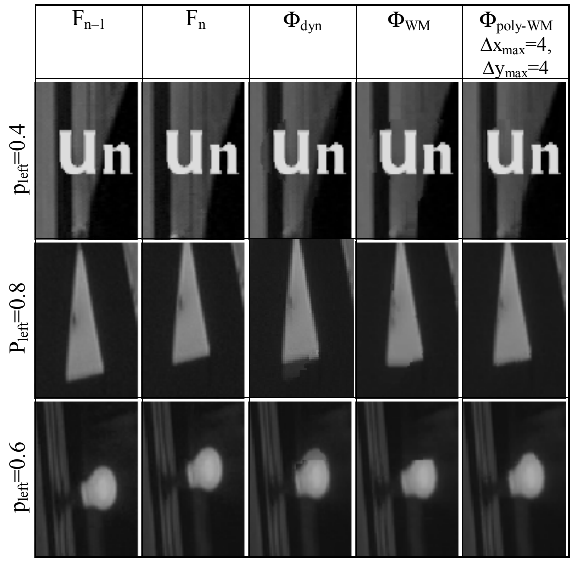

Figure 8: Simulation results (3-tap Median, WM, polyph. WM)

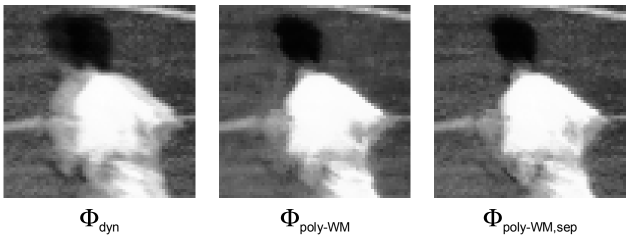

Figure 9: Simulation results (3-Tap Median, polyphase WM without/with nonlinear band separation)

7. ACKNOWLEDGEMENTS

The authors would like to thank the German Research Society (DFG) and the Micronas Munich GmbH for financially supporting the work on error tolerant interpolation filter design resp. the work on polyphase WM interpolation algorithms.

8. REFERENCES

[1] H. Blume, "Nonlinear Vector Error Tolerant Interpolation of Intermediate Video Images by Weighted Medians", *Signal Processing: Image Communication*, Vol. 14, pp. 851-868, 1999

[2] M. Braun, M. Hahn, J.-R. Ohm, M. Talmi, "Motion-Compensating Real-Time Format Converter for Video on Multimedia Displays", *Proc. of ICIP-97, Santa Barbara, CA, USA, October 26th-29th*, pp. 125-128, 1997

[3] O. Franzen, M. Schu, C. Tuschen, H. Schröder, "Intermediate Image Interpolation using Polyphase Weighted Median Filters", *Proc. of SPIE Electronic Imaging, San Jose, USA, January 20th-26th*, pp.306-317, 2001

[4] O.A. Ojo, G. de Haan, "Robust Motion-Compensated Video Upconversion", *IEEE Transactions on Consumer Electronics*, Vol. 43, pp. 1045-1056, 1997

[5] M. Schu, M, G. Scheffler, C. Tuschen, A. Stolze, "System on Silicon for Motion Compensated Scan Rate Conversion, Picture-In-Picture Processing, Split Screen Applications and Display Processing", *IEEE Trans. on Consumer Electronics*, Vol. 45, pp. 842-850, 1999