논문96-1-1-07

# 3D Deinterlacing Algorithm Based on Wide Sparse Vector Correlations

Yeong-Taeg Kim

**Abstract**

In this paper, we propose a new 3-D deinterlacing algorithm based on wide sparse vector correlations and a vertical edge based motion detection algorithm, which is an extension of the deinterlacing algorithm proposed in [10, 11] by the authors. The prooised algorithm is developed mainly for the format conversion problem encountered in current HDTV system, but can also be aplicable to the double scan conversion problesm frequently encountered in ths NTSC systems. By exploiting the edge oriented spatial interpolation based on the wide vector correlations, visually annoying artifiacts caused by interlacing such as a serrate line, line crawling, a line flicker, and a large area flicker can be remarkably reduced since the use of the wide vectors increases the range of the edge orientations that can be detected, and by exploiting sparse vectors correlations the H/W complexity for realizing the algorithm in applications cam be significantly simplified. Simulations are provided indicating thet the proposed algorithm results in a high performance comparable to the performance of the deinterlacing algorithm, based on the wide vector correlations.

## I. Introduction

In the development of current HDTV systems, it is indispensable to employ an interlaced to progressive rate conversion (IPC) system due to the variety of the standard source formats adopted in HDTV Initially, IPC algorithms were developed for NTSC systems, where interlacing scheme is employed to use a channel bandwidth efficiently, to reduce the intrinsic aliasing caued by interlacing such as a serrate line, line crawling, a line flicker, raster line visibility, and a field flicker. Such artifacts become increasingly objectionable with larger displays. To lessen such artifacts, many algorithms have been proposed in literature [1]-[8], and the consensus is to recover the lines unavailable in the interlaced images through signal processing. Examples of the algorithms include a simple line doubling scheme, vertical filtering, edge direction dependent deinterlacing [3], nonlinear interpolation schemes based on a weighted median filter [4], based on a FIR median hybrid interploation [5], based on direction dependent median filtering [6]. and motion adaptive schems [7, 8].

Signal Processing Lab, Samsung Electronics Co., Suwon, Korea

Recently in [10], a deinterlacing algorithm based on weighted wide vector correlations was proposed by the author for the format conversion problem encountered in the development of current HDTV system. This algorithm introduces the use of wide vectors in estimating the spatial correlations at a certain point, which is found to be effective in detecting various edge orientations to determine the direction of spatial interpolation. One drawback of this aproach is found in its H/W complexity for computing the correlations between wide vectors for various directions. To lessen the H/W complexity while maintaining the performance of the algorithm based on wide vector correlations, we explicitly introduce the use of sparse vectors in this paper. Whereas the use of wide vector correlations increases the capability of the algorithm in detectomg varopis edge-orientations, the use of sparse vectors makes it possible to implement the algorith, with less H/W complexity. The detailed descriptions of the proposed algorithm along with simulation results are followed in the subse-

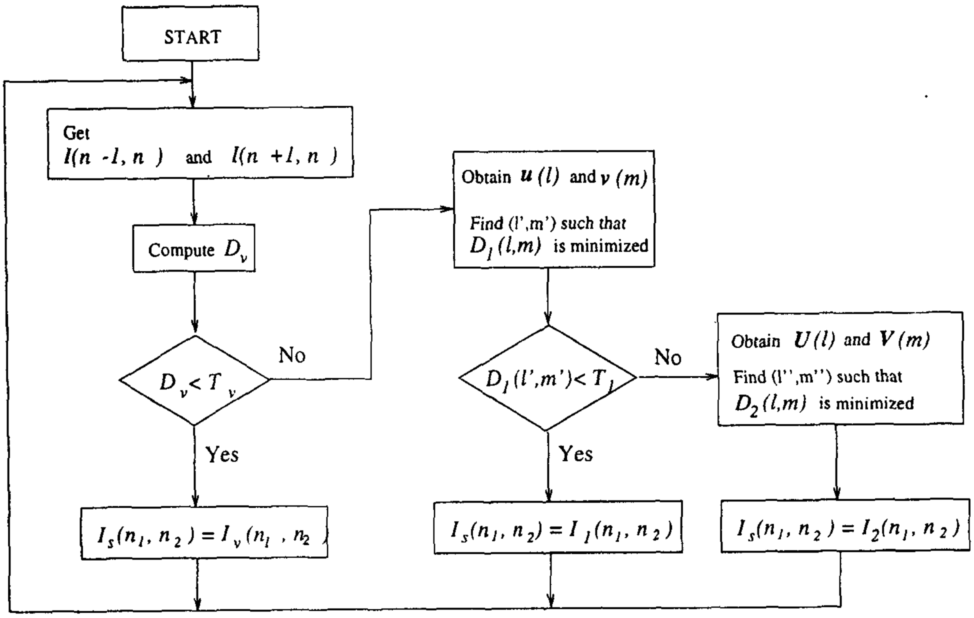


Figure 1 : The flowchart of the proposed 2-D deinterlacing algorithm based on wide sparse vector corrlations.

quent sections.

## Ⅱ. 2-D Deinterlacing Algorithm Based on Wide Sparse Vector Correlations

Let $\{I(n_1, n_2)\}$ denote a given image and $\{I(n_1, n_2)\}$ denote a low-pass filtered image of $\{I(n_1, n_2)\}$. We assume that the given image $\{I(n_1, n_2)\}$ is interlaced, that is, only odd lines or even lines of $\{I(n_1, n_2)\}$ are observed at a certain field interval. For the notational simplicity in the sequel, it will be assumed that $(n_1, n_2)$ represents the sampling point whose value is not available in interlace mode at a certain field. Thus, $I(n_1, n_2)$ will designate the signal value that we wish to estimate from a given interlaced image. The proposed algorithm is composed of the following steps : *(I) vertical filtering mode, ( Ⅱ ) narrow vector correlation mode, and ( Ⅲ ) wide sparse-vector correlation mode.* The detailed description of the proposed 2-D algorithm based on the use of wide sparse vector correlations is in order referring to the flow chart shown in Fig. 1.

As a first step, we consider a vertical interpolation. Due to its simple operation and implementation, a vertical interpolation is widely used in practice for the problem at hand. In many cases, it provides pleasing results since an interlaced image can be regarded as a decimated image by 2 in vertical direction and a vertical filter has a low-pass filter characteristic. The vertical interpolation is accomplished simply by taking the weighted combination of the nearest neighbor samples in vertical direction. Letting $Iv(n_1, n_2)$ as the estimation of $I(n_1, n_2)$ by the vertical interpolation using $2N$ neighbor lines, it can be expressed as

$$Iv(n_1, n_2) = \sum_{i=-N+1}^{N} w_i \, I(n_1+2i-1, n_2), \qquad (1)$$

where $w_i$ is the weight of each sample. This method intrinsically introduces blurring, however, since the vertical sampling rate in the interlaced image typically is fall behind the Nyquist sampling rate in real image, Hence, if there exists a vertical edge between the samples $I(n_1-1, n_2)$ and $I(n_1+1, n_2)$ the vertical filtering smears out the details and, as a consequence, it introduces some annoying visual artifacts such as a serrate line and line flickering. Hence, it can be easily understood that detecting the existence of an edge in vertical direction in applying vertical interpolation is of importance, and the method should be adjusted accordingly.

For determining the correlation between the neightbor lines in vertical direction, we consider the following quantity :

$$Dv = | \, I(n_1-1, n_2) - I(n_1+1, n_2) \, | , \qquad (2)$$

where we used the low-pass filtered signal in order not to decrease the reliability on this quantity that might be affected by some unwanted high frequency components in a given image such as an impulsive noise. Letting $I_S(n_1, n_2)$ as the output of the proposed 2−D algorithm, the estimate of $I(n_1, n_2)$ is given as

$$I_s(n_1, n_2)=I_v(n_1, n_2), \text{ if } D_v<T_v \tag{3}$$

*where* $T_v$ is a constant. For the case in which $D_v \geq T_v$, however, we note that the estimation in (1) is not reliable since $D_v \geq T_v$, implies the existence of an edge in vertical direction. Hence, for $D_v \geq T_v$, an imterpolation method appropriately chosen based on edge−orientation should be taken into account to prevent unwanted artifacts. Some algorithms in this category are discussed in [3]. In our algorithm, for $D_v \geq T_v$, we move to the following step utilizing a correlation between narrow vectors within a small region.

Let us define the 3−long vectors u($l$) and v($m$), respectively, as

$$u(l)=\begin{bmatrix} I(n_1-1,\ n_2+l-1) \\ I(n_1-1,\ n_2+l) \\ I(n_1-1,\ n_2+l+1) \end{bmatrix} \overset{\text{let}}{=} \begin{bmatrix} u_{-1}(l) \\ u_0(l) \\ u_1(l) \end{bmatrix} \tag{4}$$

and

$$v(m)=\begin{bmatrix} I(n_1+1,\ n_2+m-1) \\ I(n_1+1,\ n_2+m) \\ I(n_1+1,\ n_2+m+1) \end{bmatrix} \overset{\text{let}}{=} \begin{bmatrix} v_{-1}(m) \\ v_0(m) \\ v_1(m) \end{bmatrix} \tag{5}$$

where $l$ and $m$ represent the respective positions of those vectors in the $n_2$ axis with respect to the interpolation point $(n_1, n_2)$. Based on these vectors, we define the weighted absolute difference between them as

$$D_1(l, m)=\sum_{i=-1}^{1} | u_i(l)-S_i(m) | C_i \tag{6}$$

*where* $C_i$ is a normalized weight. Note that this quantity can be used to monitor how much the vectors, u($l$) and v($m$), are alike. For instance, the vectors u($l$)and v($m$) will represent the same pattern if and only if $D_1(l, m)=0$. Clearly, estimating $D_1(l, m)$ for different values of $l$ and $m$ can be a guideline to detect the direction to which the interpolation would be followed. Hence, we find the respective locations of u($l$) and v($m$), denoted as $l'$ and $m'$, such that $D_1(l', m')$ is to be the minimum over a searching region $l, m \in [-R, R]$. In other words, we find $(l', m')$ such that

$$D_1(l', m')=Min\{D_1(l, m) \mid l=-R, \cdots, R, \text{ and } m=-R, \cdots, R\}. \tag{7}$$

Fig. 2. shows the respective locations of the vectors u($l$) and v($m$) for all possible ($l'$ $m'$) satisfying (7) when $R=1$, where the samples in u($l$) are connected by arrows with the associated samples in v($m$) in computing $D_1(l, m)$ given in (6). This figure clearly indicates how the sample $I(n_1, n_2)$ should be interploated from the neighbor sample when ($l'$ $m'$) is found.

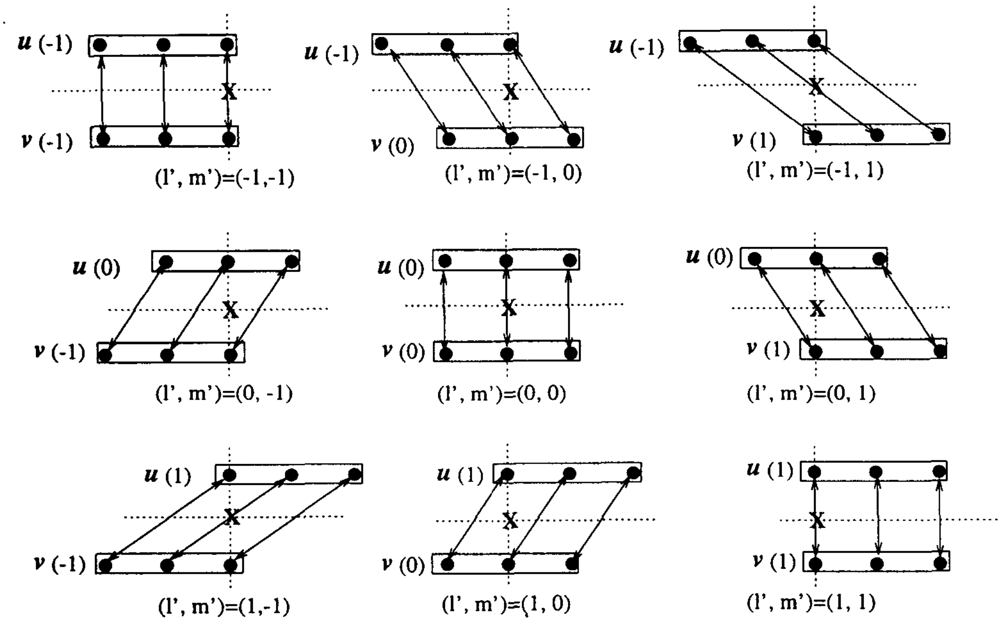


Figure 2 : The respective locations of the Vectors u($l$) and v($m$) for all possible ($l'$, $m'$) when $R$=1

Hence, the directions of the strongest spatial correlations are clearly shown in Fig. 2 for all $(l'\ m')$. In our algorithm, $R=1$ is used not only to save the associated H/W complexity, but also to prevent an error in detecting the direction of the strongest spatial correlation. Based on the values of $(l'\ m')$ found, the proposed algorith, estimates $I(n_1, n_2)$ as

$$I_1(n_1, n_2)=(I(n_1-1, n_2+p)+I(n_1-1, n_2+q)+I(n_1+1, n_2+r)+I(n_1+1, n_2+s))/4 \quad (8)$$

where

$$(p, q, r, s)= \begin{cases} (0,0,0,0), & \text{if}(l', m')=(-1,-1), (0,0), (1,1) \\ (-1,0,0,1), & \text{if } (l', m')=(-1,0), (0,1) \\ (-1,-1,1,1), & \text{if } (l', m')=(-1,1) \\ (0,1,-1,0), & \text{if } (l', m')=(0,-1), (1,0) \\ (1,1,-1,-1), & \text{if } (l', m')=(1,-1) \end{cases} \quad (9)$$

whose functionals can be easily understood from Fig. 2

The method described previously is improper, however, when there is a high correlation in the direction of low slope around the point $(n_1, n_2)$. Note from Fig. 2 that the method based on narrow vector correlations can detect the direction of the spatial correlation ranging from $-45°$ to $45°$. It is obvious that the value $D_1(l', m')$ will be large if there exists a high correlation in the direction of low slope. Thus, we compare the quantity $D_1(l', m')$ with a given constant $T_1$, and interpolate $I(n_1, n_2)$ as If the value $D_1(l', m')$ exceeds $T_1$, it is highly probable that the edge-orientation around the interpolation point is associated with a low slope. Note that simply enlarging the searching bound, $R$, does not guarantee that the narrow vector correlation based method can handle such case correctly as long as we use 3-long vectors $u(l)$, $v(m)$. To account for the existence of the high correlation in low direction when $D_1(l', m')\geq T_1$, we propose to proceed the next step based on wide sparse vector correlations.

$$I_s(n_1, n_2)=I_1(n_1, n_2), \text{ if } D_1(l', m') < T_1 \quad (10)$$

Define the wide sparse vectors $U_S(l)$ and $V_S(m)$, defined as

$$U_S(l)=\begin{bmatrix} I(n_1-1, n_2-L-l+m) \\ I(n_1-1, n_2+l) \\ I(n_1-1, n_2+L+l) \end{bmatrix} \overset{\text{let}}{=} \begin{bmatrix} U_{-1}(l) \\ U_0(l) \\ U_1(l) \end{bmatrix} \quad (11)$$

and

$$V_S(m)=\begin{bmatrix} I(n_1+1, n_2-L-l+m) \\ I(n_1+1, n_2-l+m) \\ I(n_1+1, n_2+L-l+m) \end{bmatrix} \overset{\text{let}}{=} \begin{bmatrix} V_{-1}(m) \\ V_0(m) \\ V_1(m) \end{bmatrix} \quad (12)$$

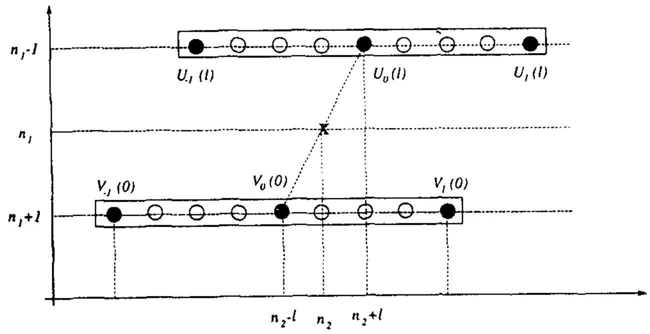


Figure 3 : The visual formulation of the wide sparse vectors $U_S(l)$ and $V_S(0)$

where we assume $2L+1\gg3$. Observe that the vectors $U_S(l)$ and $V_S(m)$ are 3-long in length. However, two objacent samples in $U_S(l)$ and $V_S(m)$ are $L$-samples away spatially. Hence, it can be said that the virtual length of the sparse vector is $2L+1$. Note also that the center locations of $U_S(l)$ and $V_S(0)$ are symmetric with respect to the point $(n_1, n_2)$, which is shown in Fig. 3 where the visual formulation of the wide sparse vectors $U_S(l)$ and $V_S(0)$ is illustrated. The vector $V_S(m)$ is then obtained by shifting $V_S(0)$ by $m$ in the $n_2$ axis.

With a similar motivation behind the definition of $D_1(l, m)$ give in (6), we define

$$D_2(l, m)=\sum_{i=-1}^{1} |U_i(l)-V_i(m)|\, W_i \quad (13)$$

where $W_i$'s are normalized weights. By estimating $D_2(l, m)$ over a searching region $l\in[-S, S]$, we also find $(l'', m'')$ such that

$$D_2(l'', m'')=Min\{D_2(l, m)\mid l=-S, \cdots, 0, \cdots, S, \text{ and } m=-1, 0, 1\}, \quad (14)$$

where we will limit the searching bound as $S\leq L$. Once the $(l'', m'')$ is found, then the interpolation of $I(n_1, n_2)$ by this method, $I_2(n_1, n_2)$, is expressed as

$$I_2(n_1, n_2)=\frac{I(n_1-1, n_2+l'')+I(n_1+1, n_2-l''+m'')}{2} \quad (15)$$

and thus, $I_s(n_1, n_2)=I_t(n_1, n_2)$.

## Ⅲ. 3-D Deinterlacing Algorith, Based on Edge-Dependent Motion Switching

In the previous section, we proposed a 2-D deinterlacing algorithm based on weighted wide sparse vector correlations. In this section, we propose a 3-D deinterlacing algorith, composed of the 2-D algorithm described in the previous section and an edge-dependent motion switching method described below.

Suppose $\{I^-(n_1, n_2)\}$ and $\{I^+(n_1, n_2)\}$ are the previous and next fields of the given interlaced image $\{I(n_1, n_2)\}$ . Based on these fields, the temporal interpolation for $I(n_1, n_2)$, which will be denoted as $I_t(n_1, n_2)$, is given as

$$I_t(n_1, n_2)=(I^-(n_1, n_2)+I^+(n_1, n_2))/2. \tag{16}$$

When there is no motion between $\{I^-(n_1, n_2)\}$ and $\{I^+(n_1, n_2)\}$ around $(n_1, n_2)$, it is clear that the quantity given in (16) will provide a good estimate of $I(n_1, n_2)$. However, if there occurs a motion, $I_t(n_1, n_2)$ will introduce an artifacts. Thus, detecting a motion in a proper fashion is an important issue in temporal interpolation in order not to cause serious visual artifacts. The basic philosophy behind our algorithm for detecting a motion is to reduce artifacts resulting from detection errors as much as possible.

Let us define the windows $W^-$ and $W^+$, respectively, as

$$W^-=\begin{bmatrix} 0 & I^-(n_1-2, n_2) & 0 \\ I^-(n_1, n_2-1) & I^-(n_1, n_2) & I^-(n_1, n_2+1) \\ 0 & I^-(n_1+2, n_2) & 0 \end{bmatrix} = \begin{bmatrix} 0 & W_1^- & 0 \\ W_2^- & W_3^- & W_4^- \\ 0 & W_5^- & 0 \end{bmatrix} \tag{17}$$

and

$$W^+=\begin{bmatrix} 0 & I^+(n_1-2, n_2) & 0 \\ I^+(n_1, n_2-1) & I^+(n_1, n_2) & I^+(n_1, n_2+1) \\ 0 & I^+(n_1+2, n_2) & 0 \end{bmatrix} = \begin{bmatrix} 0 & W_1^+ & 0 \\ W_2^+ & W_3^+ & W_4^+ \\ 0 & W_5^+ & 0 \end{bmatrix}. \tag{18}$$

Based on $W^-$ and $W^+$, we consider

$$D_t=\sum_{i=1}^{5} | W_i^- - W_i^+ | \alpha_i, \tag{19}$$

where $\alpha_i$'s are coefficients. The quantity $D_t$ is basically to measure the correlation between the two fields around $(n_1, n_2)$ in temporal direction. That is, if we observe large value of $D_t$, we can say that there occurs a motion around $(n_1, n_2)$. Hence, for $D_t>T_m$, where $T_m$ is a constant, it is clear that the output of the proposed 2-D algorithm, $\hat{I}(n_1, n_2)$, should be given by

$$\hat{I}(n_1, n_2)=I_s(n_1, n_2), \tag{20}$$

since $D_t> T_m$ implies that a motion is detected. On the contrary, however, $D_t\leq T_m$ does not simply imply no-motion, but implies either "no motion" or "fast motion". It is well-known that a failure in discriminationg fast-motion from no-motion causes serious artifacts such as tearing artifact. However, theoretically, it is not possible to distinguish them by simply observing $D_t$ since the temporal sampling rate always behind the Nyquist rate in practice. Thus, most of the methods in motion adaptive interpolation algorithms are focused on how to reduce the artifacts resulting from the failure in detecting fast-motion. For this purpose, for $D_t\leq T_m$, we further consider

$$D= | I(n_1-1, n_2)-I(n_1+1, n_2) | , \tag{21}$$

which is the correlation in vertical direction, or which corresponds to the existence of an edge in vertical direction. It should be noted that the visual effect of artifacts resulting from a failure in detecting fast-motion is reduced as the value of $D$ increases. In other words, for large $D$, $I_t(n_1, n_2)$ would be good enough visually although when there occurs a fast motion. Thus, we have

$$\hat{I}(n_1, n_2)=I_t(n_1, n_2), \text{ if } D> T, \tag{22}$$

where $T$ is a constant. In case when $D\leq T$, we consider the quantity

$$\beta=\left[\frac{I(n_1-1, n_2)+I(n_1+1, n_2)}{2} - \frac{I^-(n_1, n_2)+I^+(n_1, n_2)}{2}\right], \tag{23}$$

which designates the difference between the line average and the temporal average. The quantity $\beta$ basically is then used to minimize an uncertainty in deciding a motion. That is, if $\beta$ is too large combined with $D\leq T$, it might be considered that there occurs a fast motion although $D_t\leq T_m$. This quantity was also addressed in [9]. Hence, based on the value of $\beta$, we have

$$\hat{I}(n_1, n_2) = I_S(n_1, n_2), \text{ if } \beta > T_s \quad (24)$$

$$\hat{I}(n_1, n_2) = I_I(n_1, n_2), \text{ if } \beta \le T_s \quad (25)$$

## Ⅳ. Simulation Results

In order to illustrate the performance of the proposed algorithm in interlaced to progressive conversion problem, we present some simulation results in this section. Fig. 4 and 5 show the originally given image and its interlaced image composed of four different imgages. The resolution image we used is 720×480. In the interlaced image, we put zero-valued lines for the lines unavailable. For comparison, the results of applying line doubing method and vertical interpolation method are depicted in Fig. 6 and Fig. 7, respectively. In these results, blurring and severe artifacts are observed. Especially, serrate

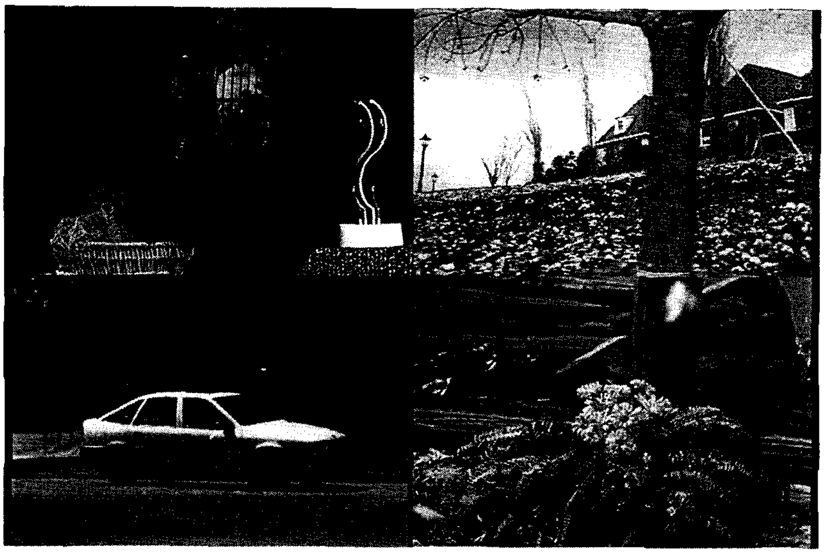

Figure 4. The originally given image composed 720×480 pixels,

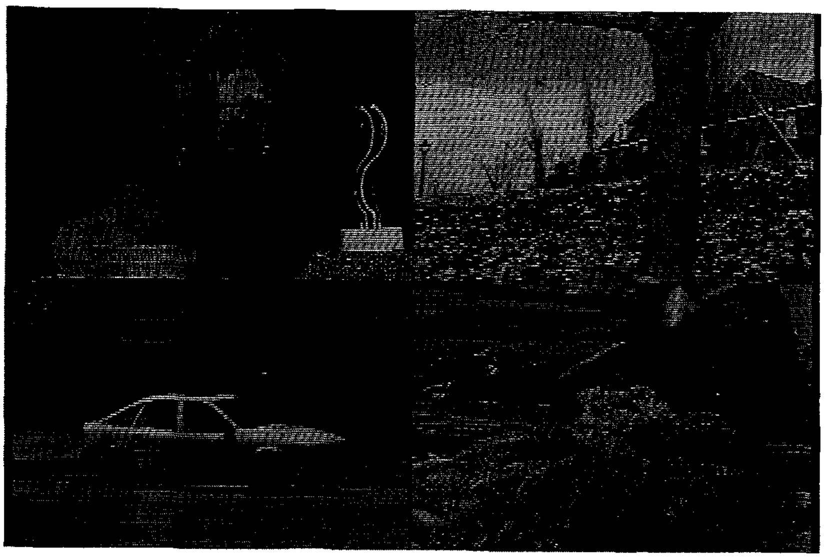

Figure 5 : The interlaced image obtained from Fig. 4

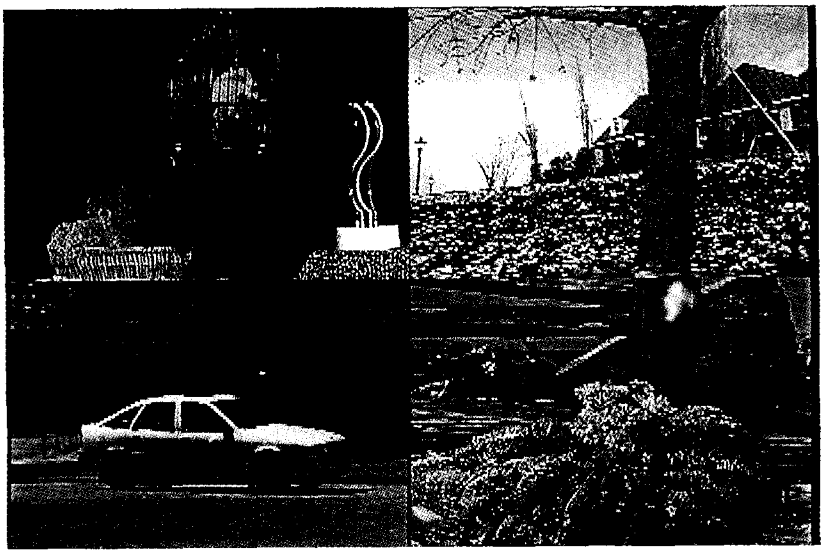

Figure 6 : The result of line doubling method.

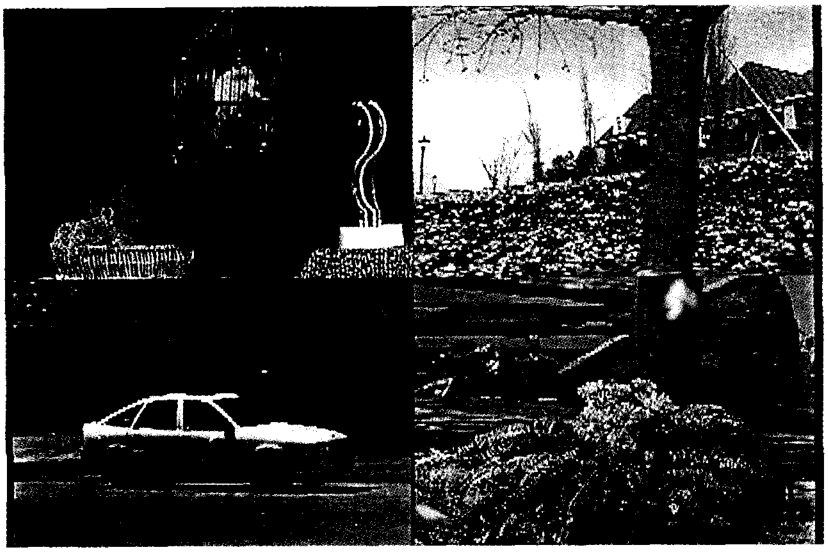

Figure 7 : The result of vertical filtering.

lines are introduced around edges in the direction of low slopes. The result of the algorithm based on weighted wide vector correlations proposed in [10] is shown in Fig. 8, where $L=4$ and $D_v=T_l=30$ are used, and we present the result of the proposed 2-D algorithm based on wide sparse vectors in Fig. 9 where we used the same parameters as the parameters used in obtaining the result shown in Fig. 8. The result of the proposed 3-D algorithm is shown in Fig. 12 and the result of a 3-D algorithm based on median approach is depicted in Fig. fig:med for comparison. For close comparisons of the algorithms, we magnified parts of the resulting images as shown in Fig. 10. From these results, first, it can be observed that the method based on wide vector correlatons shows a a high performance around various edges. Observe that clean lines are recovered around edges having low slopes with the method based on wide vector correlations. Netx, if we closely compare thae results in Figs. 8 and 9, the results of the proposed algorithm based on wide sparse vector corrlation is comparable to that of the wide vector correlation based algorithm. If we note that the proposed algorith, requires much less H/W

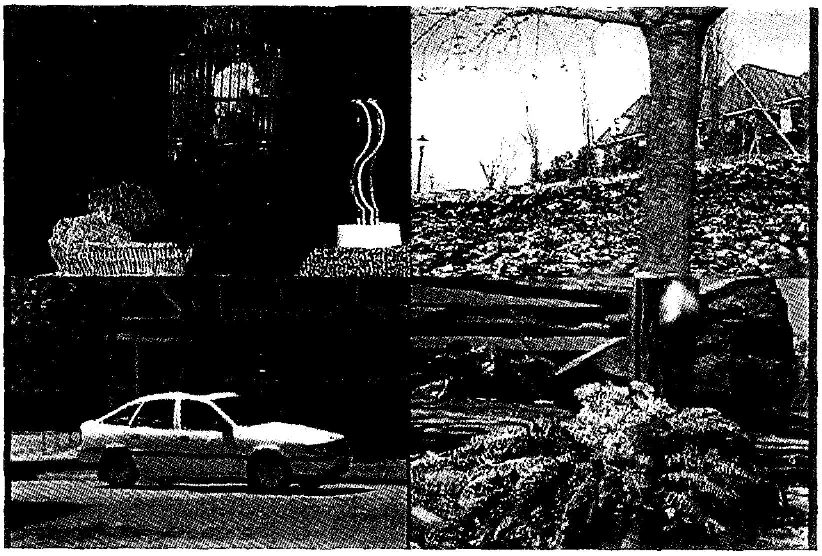

Figure 8 : The result of the algorithm based on weighted wide vector correlations.

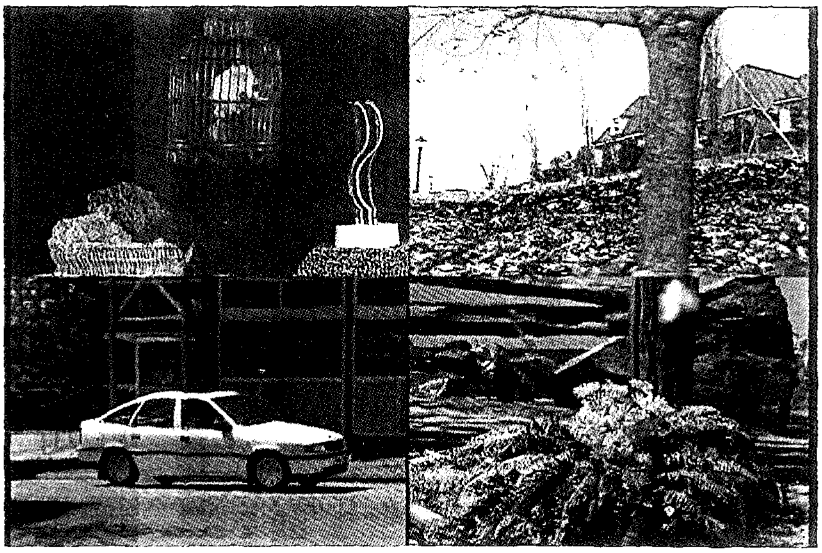

Figure 9 : The result of the proposed algorithm based on wide sparse vector correlation.

complexity than deinterlacing algorithm based on wide vector correlation, the advantage of the proposed algorithm in this paper is clearly seen.

## V. Conclusion

In this paper, we have presented a 3-D algorithm for IPC based on wide sparse vector correlations leading to a high performance in deinterlacing and a neat H/W structure. Throughout simulations, it has been shown that the proposed algorithm is comparable to the deinterlacing algorithm utilizing wide vector correlations in terms of performance. Besides, the proposed algorithm requires much less H/W complexity than the algorithm based on wide vector correlations in [10]. Thus, it can be seen that the proposed algorithm is not only suitable for interpolating a missed sample around various edges, but also suitable for H/W implementation in applications.

## References

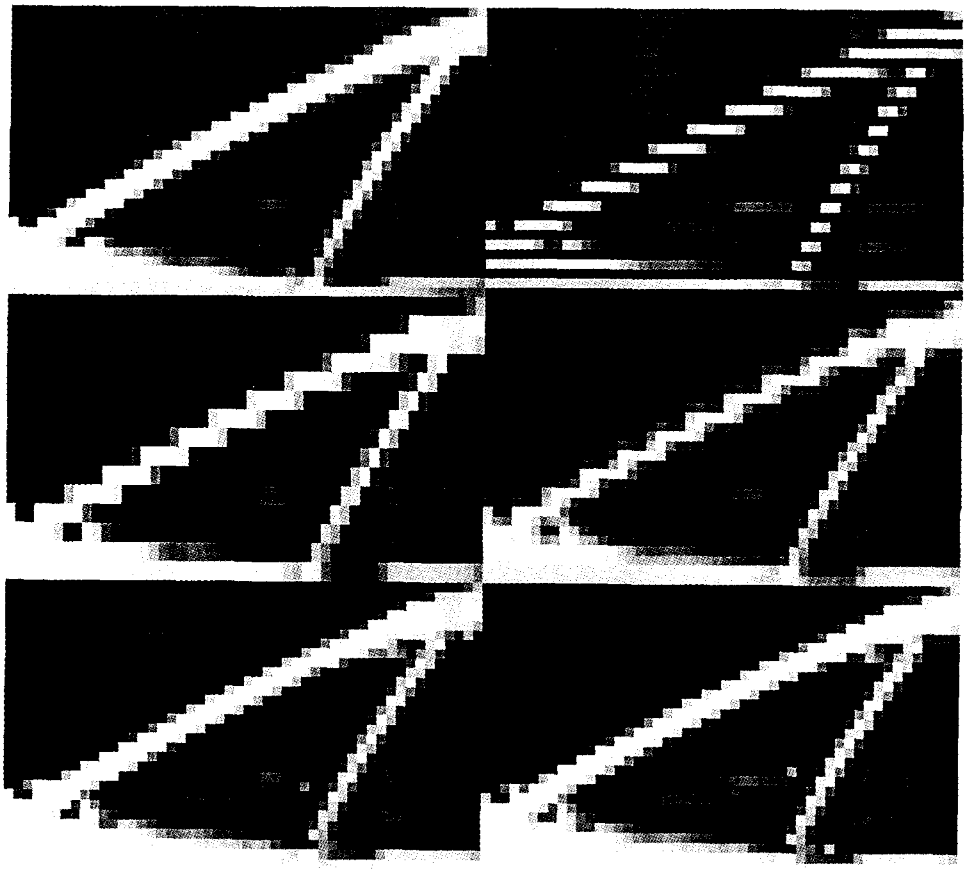

Figure 10. Enlarged view of the reconstruction for comparison. ***Left column(top to bottom)*** : Original image, reconstructed image using line doubling method, and the reconstructed image using wide vector corrleaiton based alogrithm. ***Right column (top to bottom)*** : Interlaced image, reconstructed image using vertical filtering and reconstructed image using the proposed algorithm based on wide sparse vector correlation.

[1] C. Hentschel, "Linear and Non-linear Processing for flicker reduction," *IEEE Tr. on Consumer Electronics*, vol. 33, no. 3, pp. 192-198, 1987.

[2] C. Hentschel, "Comparison Between Median Filtering and Vertical Edge Controlled Interpolator For Flicker Reduction," *IEEE Trans. on Consumer Electronics*, vol. 35, no. 3, pp.279-289, August 1989.

[3] D. Bagni, R. Lancin, S. Landi, and S. Tubaro, "HD-TV Spatio-Temporal Upconversion," *Proc. Int. Workshop on HDTV*, 1994.

[4] J. Huhola, A. Nieminen, J. Salo, and Y. Neuvo, "Scan Rate Conversion Using Weighted Median Filtering," *Proc. IEEE ISCAS-89*, Portland, USA, pp. 433-436, May 1989.

[5] A. Lehtonen and M. Renfors, "Non-linear Quincunx Interpolation Filtering," *Proc. SPIE's Visual Communications and Image Processing*, Lausanne, Switzerland, pp. 135-132. Oct 1990.

[6] T. Doyle, "Interlaced to sequential Conversion for EDTV Applications," pp. 421-430, *Signal Processing of HDTV*, L. Chiariglione Ed., Elsevier Science Publishers, North Holland, 1988.

[7] N. Suzuki *et al.*, "Improved Synthetic Motion Signal For Perfect Motion-Adaptive Pro-Scan Conversion in HDTV Receivers," *IEEE Trans. on Consumer Electronics*, vol. 33, no. 3, pp.266-271, August 1989.

[8] C. P. Markhauser, "Motion Adaptive Pro-Scan Converter with Two Dimensional Contour Enhancement," *IEEE Trans. on Consumer Electronics*, vol. 36, no. 2, pp. 110-114, May 1990.

[9] R. Simonetti *et al.*, "A Deinterlacer for IQTV Receivers and Multimedia Applications," *IEEE Trans. on Consumer*

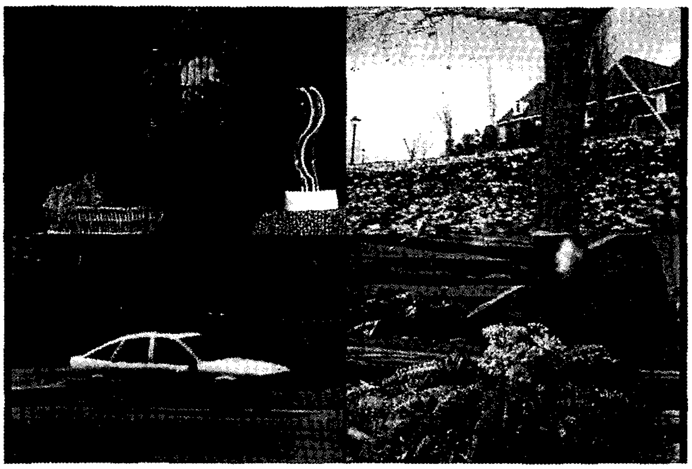

Figure 11 : The result of 3-D algorithm based on median approach.

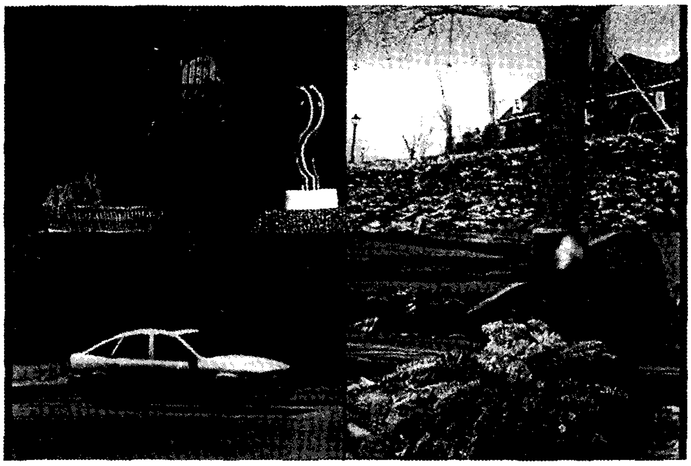

Figure 12 : The result of the 3-D algorithm

*Electronics*, vol. 39, pp.234240, Aug. 1993.

[10] Y.-T. Kim, "A Deinterlacing Algorithm Based on Weighted Wide Vector Correlations," *Proc. International Conference on Signal Processing Applications and Technology*, Boston, MA, USA, Oct. 24-26, 1995.

[11] Y.-T. Kim and Y. H. Cho, "Motion Adaptive Deinterlacing Algorithm Based on Wide Vector Correlations and Edge Dependent Motion Switching," *Proc. International Workshop on HDTV*, Taipei, Taiwan, ROC, Nov. 15-17, 1995.

## 저 자 소 개

金 永 鐸
1962년 11월 13일생.
1988년 2월 연세대학교 전자공학과 졸업.
1992년 1월 University of Delaware 전기공학 석사.
1993년 8월 University of Delaware 전기공학 박사.
현재 삼성전자 신호처리연구소 선임연구원