

IC FOR MOTION-COMPENSATED DE-INTERLACING, NOISE REDUCTION, AND PICTURE-RATE CONVERSION

Gerard de Haan

Philips Research Laboratories, Eindhoven, The Netherlands

ABSTRACT

An IC¹ for consumer television applies motion estimation and compensation for high quality video format conversion. The chip achieves perfect motion portrayal for all sources including 24, 25, and 30 Hz film material, and many display formats. The true-motion vectors are estimated with a sub-pixel resolution and are used to optimally de-interlace video broadcast signals, perform a motion compensated picture rate conversion and improve temporal noise reduction.

1 INTRODUCTION

Picture sequences come in various picture rates: film material in 24, 25 and 30 Hz, and video usually in 50 and 60 Hz. Television displays, on the other hand, are commercially available with picture rates of 50, 60 and 100 Hz, and either progressive or interlaced scanning. Simple picture rate converters repeat pictures until the next arrives [1,2], which results in blur and/or judder when motion occurs. Similarly, de-interlacing sometimes results from repetition, or averaging of neighbouring lines. The more advanced de-interlacing concepts apply vertical-temporal processing [3,4,5,6], but even these degrade those parts of images where motion occurs.

Some years ago, consumer television ICs appeared using motion estimation (ME) and compensation (MC) to achieve high performance conversion for even moving sequences [7,8]. Those circuits made breakthroughs in motion estimation to give high quality motion compensation for a consumer price. Indeed, they received various international awards², and are still unsurpassed in quality. Products based on similar concepts have been announced [9,10], but are not yet available.

This paper shows new progress, introducing a new IC that handles more input and output formats, and applies significantly improved algorithms for ME, MC de-interlacing, MC picture interpolation, and MC noise reduction.

1. The IC is available commercially as SAA4992.
2. The 1995 EISA European Video Innovation Award, the 1995 ICCE first place Outstanding Paper Award, and the 1998 ICCE second place Outstanding Paper Award. The first TV equipped with the IC received the 1996 EISA Television of the Year Award.

2 THE ALGORITHMS

This section describes the new elements of the algorithms for motion estimation, motion compensation, de-interlacing and noise reduction, and compares them with the previous generation of this scan conversion IC.

2.1 Motion estimation

As before, we used a 3-D recursive block matcher, based on [11], for motion estimation. This estimator, like any block matcher, estimates displacement vectors to minimize a match error calculated for blocks of (8 by 8) pixels. This match error function³ does not guarantee that the vector closest to the true object motion is found. The crux in designing a good block matcher is therefore not to test unlikely motion vectors. The 3-D RS block matcher does this by introducing constraints based upon two simple and very effective assumptions:

1. Relevant objects are larger than blocks, and
2. Objects have inertia.

The consequence of assumption 1 is that the vector describing the velocity of the object in the current block can be found in at least one of the neighbouring blocks. The implication is that it makes no sense to evaluate all possible vectors within the search range $CS_{\vec{X}}^{\max}$; it should be sufficient to evaluate candidate vectors, \vec{C} , taken from the spatial neighbours. This gives a candidate vector set CS :

$$CS(\vec{X}, n) = \left\{ \vec{C} \in CS^{\max} \mid \vec{C} = \vec{D}(\vec{X} + \begin{pmatrix} iX \\ jY \end{pmatrix}, n) \right\} \quad (1)$$

$i, j = -1, 0, -1$

where X and Y are the block width and height respectively, n is the picture number, \vec{X} is the position on the block grid, and \vec{D} is the output displacement vector. There are two problems with assumption 1:

- Not all neighbours are immediately available (causality problem)
- At initialization, all vectors are zero...

The first problem is solved by assumption 2; those vectors

3. We used the sum of absolute differences between pixels in the current block and pixels in the block shifted over a candidate vector in the previous image.

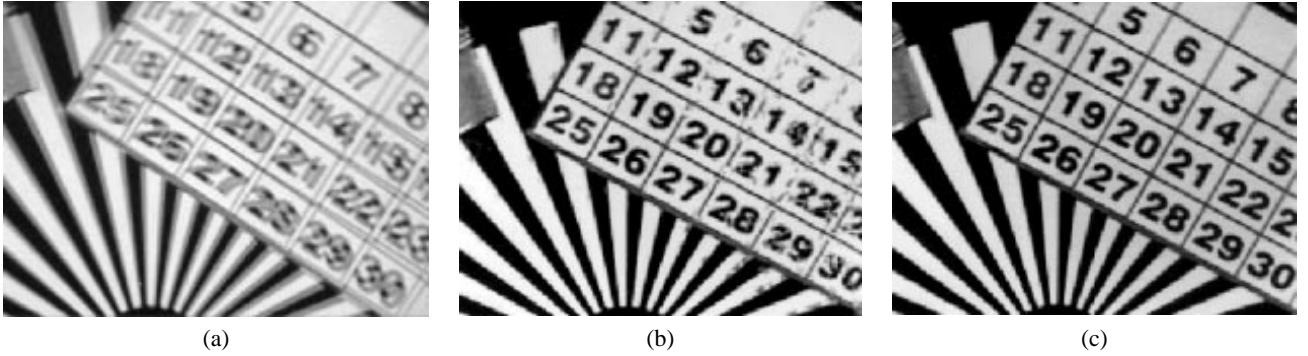


Fig.1 Photographs of a screen detail comparing picture rate conversion using non-motion compensated temporal averaging (a), motion compensated interpolation using vectors from a full search block matcher (b), and motion compensated interpolation from the new IC using vectors from the 3-D Recursive Search block matcher (c).

that have not yet been calculated in the current image are taken from the corresponding location in the previous vector field, profiting from object inertia. If the blocks are scanned from top left to bottom right, the candidate set is defined as:

$$CS(\vec{X}, n) = \left\{ \vec{C} \in CS^{\max} \mid \vec{C} = \vec{D}(\vec{X} + \begin{pmatrix} kX \\ -1Y \end{pmatrix}, n) \vee \right. \quad (2)$$

$$\left. \vec{C} = \vec{D}(\vec{X} + \begin{pmatrix} iX \\ jY \end{pmatrix}, n-1) \right\}$$

$$k = -1, 0, 1 \quad i = -1, 0, 1 \quad j = 0, 1$$

This candidate set implicitly assumes spatial and/or temporal consistency¹.

The second problem can easily be solved by adding an update vector, given by the sum of either of the spatial candidates and a “noise vector”. Rather than actually using a noise vector, we found that the update vector could be taken cyclically from an update set, e.g.:

$$US_i(\vec{X}, n) = \left\{ \begin{matrix} \vec{u}_y, & -\vec{u}_y, & \vec{u}_x, & -\vec{u}_x, \\ 2\vec{u}_y, & -2\vec{u}_y, & 3\vec{u}_x, & -3\vec{u}_x \end{matrix} \right\} \quad (3)$$

where we introduce $\vec{u}_x = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, and $\vec{u}_y = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

It turned out possible to reduce the number of candidate vectors further by simply omitting some of the spatio-temporal predictions from the candidate set.

Figure 1 illustrates the advantages of motion compensated

interpolation over non-motion compensated temporal interpolation. This figure also shows the benefit of using true-motion vectors. The output picture from the current IC, using a 3-D Recursive Search block matcher, improves on that from the common full search block matcher. The full search works well in coding applications, but clearly is inadequate for the more critical video format conversion.

The accuracy of the motion vectors was increased over the previous generation from integer to a quarter of a pixel, according to research described in [12]. Essentially, this only implies that the “random” update vector, mentioned above, is allowed to have fractional values, while the vector prediction memory can store the motion vectors with this increased resolution. The update set defined in eq.3 was extended by adding vectors from the following set:

$$US_j(\vec{X}, n) = \left\{ \frac{1}{4}\vec{u}_y, \quad -\frac{1}{4}\vec{u}_y, \quad \frac{1}{4}\vec{u}_x, \quad -\frac{1}{4}\vec{u}_x \right\} \quad (4)$$

Evaluating sub-pixel accurate candidate vectors requires interpolation on the pixel grid, for which we used a straightforward bi-linear interpolation. Furthermore, the range of the vectors was almost doubled in both dimensions, which implies a further increase in vector prediction memory capacity.

Another new aspect of the motion estimator affects the performance on (fast) camera manipulations like pan, tilt and zoom. As well as taking the spatial and temporal prediction vectors as candidates from a spatio-temporal neighbourhood, we calculated a prediction vector, \vec{C}_p , from a parametric motion model. This can describe pans, tilts and zooms of the camera²:

1. If the assumption is false, this consistency in the vector field results anyway, because no other candidate vectors are available.

2. A four parameter affine transformation suffices to describe this camera-caused motion.

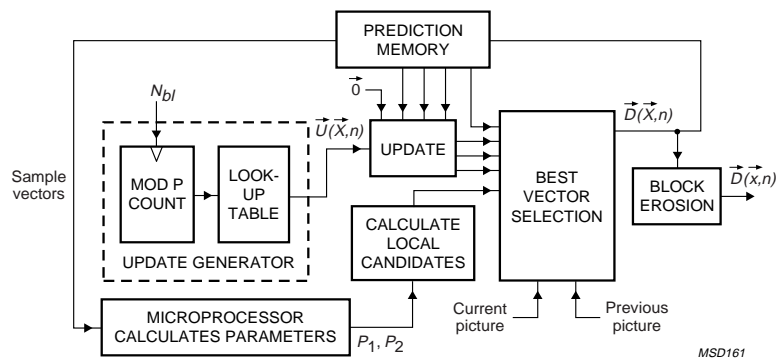


Fig.2 Block diagram of the 3-D Recursive Search block matcher. The new IC has an increased prediction vector memory that allows a larger range of motion vectors, and an increased (sub-pixel) resolution for the motion vectors. The update vector generator also generates sub-pixel updates. Finally, additional candidates are calculated from a parametric motion model. The parameters are determined in a microprocessor from sample vectors taken from fixed positions in the vector prediction memory.

$$\vec{C}_p(\vec{X},n) = \begin{pmatrix} p_1(n) + p_3(n)X \\ p_2(n) + p_4(n)Y \end{pmatrix} \quad (5)$$

The parameters of the parametric model, $p_1 \dots p_4$, are calculated by the microprocessor that controls the new IC. They are calculated from a set of 9 sample vectors taken from the most recent vector field. The parameters are fed back to the IC, which generates the local candidate from this model. A more formal description of this procedure can be found in an earlier publication on motion estimation [13]. Figure 2 shows the block diagram of the new motion estimator. The block erosion mentioned in this figure was also used in the previous generation video format conversion IC. It calculates vectors on the pixel grid from the output of the estimator, which yields a vector per block of 8 by 8 pixels [12].

2.2 The de-interlacing algorithm

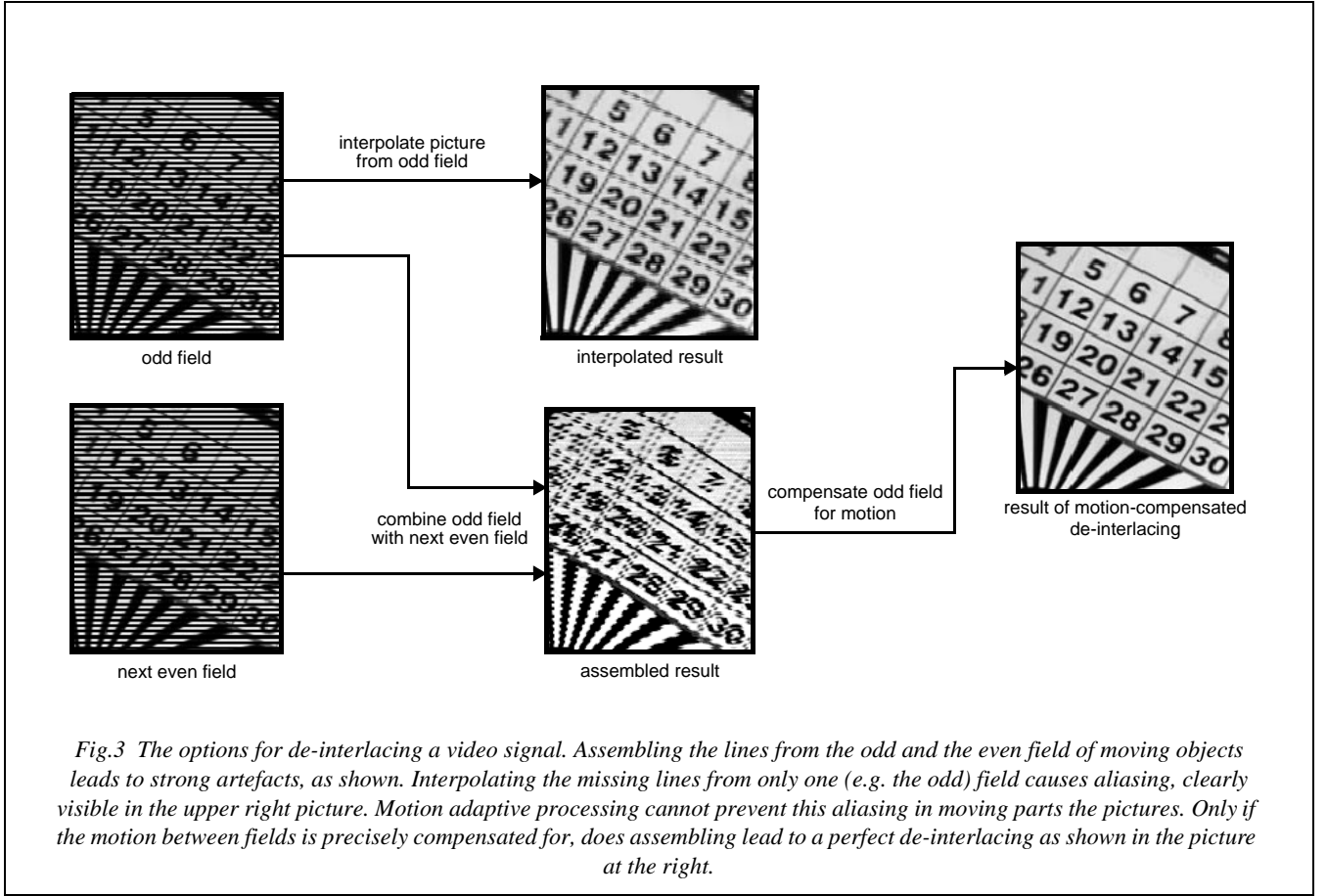
Interlacing is the common video broadcast procedure for transmitting the odd or even numbered picture lines alternately. De-interlacing attempts to restore the full vertical resolution, i.e. make odd and even lines available simultaneously for each picture. For stationary images this is a trivial task, as the information from two successive fields can be assembled into one full resolution frame. For moving images, however, this is not possible because odd and even fields no longer describe samples from the same image¹. Assembling the two fields anyway leads to totally unacceptable results, as illustrated in Figure 3. A common solution is to assemble only stationary (parts of the) images, and perform a so-called intra-field interpolation on a moving image (part). The flaw in this *motion adaptive* de-interlacing is that vertical details are lost in moving

parts of the image. Aliasing may then show, as illustrated in Figure 3.

Consequently, perfect de-interlacing will only result if motion between successive fields is either absent, *or can be compensated for*. In the first consumer IC for motion compensated video format conversion [7], we applied motion-compensated field assembling. However, in order to protect the de-interlacer against erroneous motion vectors that may always occur in a practical implementation, we eliminated outliers from the resulting progressively scanned picture using a three tap vertical median filter. This error protection, and the fact that motion vector resolution was confined to integer values, limited the advantages of the motion compensated de-interlacing.

Our new design profits from the increased resolution of the motion vectors, which is now a quarter of a pixel. This already significantly improves the quality of the motion compensated de-interlacing, even if protected with a median filter [14]. On top of this improvement, however, we have designed and implemented a much improved de-interlacing algorithm [15]².

1. An exception occurs when film material is broadcast. Film is progressively scanned, and although the odd and even lines of a film picture are transmitted in separate fields, they originate from the same film image. So, they can be assembled to give the original progressive picture.
2. This design was first presented at the International Conference on Consumer Electronics in 1997, where it received an Outstanding Paper Award. It was further favourably evaluated against the relevant alternative de-interlacers in a recent overview article [14].



The main ingredient for success of this algorithm is an intelligent protection mechanism that, in contrast with the previous median filter, does not introduce harmonics (alias) in fine textures. This protection mechanism resembles the mixer of a temporal recursive noise filter. In this case, it mixes a motion-compensated prediction from a previously de-interlaced frame, $F_o(\vec{x}, n-1)$, with a simple fall-back intra-field interpolation, $F_i(\vec{x}, n)$, to calculate the output at interpolated pixels, $F_o(\vec{x}, n)$:

$$F_o(\vec{x}, n) = pF_i(\vec{x}, n) + (1-p)F_o(\vec{x} - \vec{D}, n-1) \quad (6)$$

The intelligence is in the mixer control that determines p . This is designed such that the resulting flicker for original pixels and interpolated pixels along the estimated motion trajectory becomes equivalent [14,15]:

$$p(\vec{x}, n) = \frac{|A+B| + \delta}{2|F_i(\vec{x}, n) - F_o(\vec{x} - \vec{D}, n-1)| + \delta} \quad (7)$$

with

$$\begin{aligned} A &= F_o(\vec{x} - \vec{u}_y, n) - F_o(\vec{x} - \vec{D} - \vec{u}_y, n-1) \\ B &= F_o(\vec{x} + \vec{u}_y, n) - F_o(\vec{x} - \vec{D} + \vec{u}_y, n-1) \end{aligned} \quad (8)$$

where δ , a small constant, prevents division by zero.

This implies that as soon as the motion vector becomes somewhat unreliable (i.e. flicker results along the motion trajectory for original pixels), the mixer automatically passes more of the interpolated intra-field luminance to the output. This prevents artefacts due to erroneous motion vectors. Figure 4 shows the improved output from this de-interlacer against one of the best non-motion compensated methods (vertical temporal filtering).

2.3 Motion-compensated up-conversion

Motion compensated picture interpolation would be straightforward if the motion vectors always gave an accurate and reliable description of temporal changes in a scene. The output image temporally located between two original input images could then be calculated by simply shifting either of the two images, $F(\vec{x}, n)$ or $F(\vec{x}, n-1)$, over the (inverse) motion vector, e.g.:

$$F_{mc}(\vec{x}, n) = (F(\vec{x} - \alpha\vec{D}(\vec{x}, n), n-1)) \quad (9)$$

where α would determine the temporal position of the interpolated image.

There are several reasons why this assumption is not valid



Fig.4 Photographs of a screen detail comparing one of the best non-motion compensated methods, vertical temporal filtering (a) with the de-interlacer in the current IC (b). The pictures are part of a sequence in which the calendar moves upwards with a velocity somewhere between 1 and 2 pixels per field period.

in practice. They include changing lighting conditions, covering and uncovering of objects, and vector errors due to noise or periodic structures. Motion-compensated picture interpolation should therefore be designed so that it is robust against the errors resulting from violating the assumption. A first improvement results by motion compensating *both* neighbouring images, rather than just one of them, and using the average as the interpolated image:

$$F_{mca}(\vec{x}, n) = \frac{1}{2}(F(\vec{x} - \alpha \vec{D}(\vec{x}, n), n-1) + F(\vec{x} + (1 - \alpha) \vec{D}(\vec{x}, n), n)) \quad (10)$$

Now, if vectors are incorrect, the blurring is less objectionable than mispositioned sharp objects. A further improvement was published in [16] where we proposed a median filter instead of the averaging operation. This median had a third input formed by the non-motion compensated average, Av , of the neighbouring images:

$$F_{mcm}(\vec{x}, n) = \text{med}(F(\vec{x} - \alpha \vec{D}(\vec{x}, n), n-1), Av, F(\vec{x} + (1 - \alpha) \vec{D}(\vec{x}, n), n)) \quad (11)$$

with:

$$Av = \frac{1}{2}(F(\vec{x}, n) + F(\vec{x}, n-1)) \quad (12)$$

and:

$$\text{med}(a, b, c) = \begin{cases} a & , (b \leq a \leq c \vee c \leq a \leq b) \\ b & , (a \leq b \leq c \vee c \leq b \leq a) \\ c & , (\text{otherwise}) \end{cases} \quad (13)$$

This non-linear up-conversion filter yields perfect motion-compensated interpolation for perfect vectors, since two of the three pixels are identical and therefore determine the output signal. If the vector is locally unreliable, the two motion compensated pixels will have a different luminance value, and the larger this difference, the larger the chance that the non-motion compensated picture average is switched to the output. This greatly improves the robustness, and was used in the previous generation format conversion IC [7].

The new design perfects this concept, adapting the interpolation strategy to the local characteristics of the vector field. The background to this improvement is the observation that interpolation errors mainly occur in parts of the picture where there are discontinuities in the vector field. A more robust interpolator (like the one in eq.11) can be used in areas with spatially inconsistent vectors, with a less robust interpolator that yields a sharper output image elsewhere (e.g. using eq.10). The “flaw” of robustness can then be concentrated in areas that profit from it. A concise and more formal description of this new adaptive interpolator has been published in [17].

2.4 Motion-compensated noise reduction

Noise reduction was already part of the video processing in the previous generation IC [7]. The algorithm was that of a motion adaptive temporal first order recursive filter. The temporal delay was one field period, implemented using the same field memory required for the motion estimator and the up-converter. Interlacing was taken into account by alternating the delay in the recursive filter between $1/2(l-1)$ and $1/2(l+1)$, where l is the total number of lines in a frame [18]. The output of the noise filter, $F_F(\vec{x}, n)$, was:

$$F_F(\vec{x}, n) = kF(\vec{x}, n) + (1 - k)F_F(\vec{x} + (-1)^{N_F} \vec{u}_y, n - 1) \quad (14)$$

where N_F is the field number, and k controls the recursion and is determined by the output of the motion detector.

In the current design, the complete de-interlaced previous picture is stored in the picture delay memory. This implies that for the temporal noise filter, the delay can be adjusted to the exact field time without interlace problems. This means the delay no longer has to alternate, so

$$F_F(\vec{x}, n) = kF(\vec{x}, n) + (1 - k)F_F(\vec{x}, n - 1) \quad (15)$$

A further improvement was made in the motion performance by using motion compensation in the recursion loop of the noise filter. Compensation for (fast) horizontal motion is most cost effective, and this occurs more frequently than fast vertical motion. We therefore limited the motion compensation to the horizontal vector component:

$$F_F(\vec{x}, n) = kF(\vec{x}, n) + (1 - k)F_F(\vec{x} - \begin{pmatrix} D_x \\ 0 \end{pmatrix}, n - 1) \quad (16)$$

where D_x is the x -component of the motion vector, found by the motion estimator.

The motion detector that controls k remains in the design. As before, it prevents excessive blurring in the event of vertical, or incorrectly estimated, motion. On average, however, less motion is detected because of the compensation. More effective, stronger filtering (smaller average value of k) therefore results.

3 VLSI DESIGN AND APPLICATION

With a total die size of 72 mm^2 , the new video format conversion IC is smaller than the first generation [7]. This is due to progress in technology: we used a 0.35 micron process compared with 0.8 micron in the earlier design. The complexity of the design has however grown considerably. The transistor count increased from roughly 1.10^6 to 4.10^6 , mainly caused by the increased on-chip memory necessary for the larger vector range and the sub-pixel resolution of the vectors.

The IC processes 8-bit luminance and chrominance, and supports (Y:U:V) 4:2:2 and 4:1:1 formats (the previous generation could handle 4:1:1 only). The capacity of the external memory may vary between one and three field memories. Picture rate conversion, e.g. when converting from 50 Hz broadcast material to 100 Hz displays, requires a capacity of at least two field memories. De-interlacing and film judder elimination, e.g. for showing 60 Hz interlaced broadcasts on a progressively scanned display, may use one or two field memories. The freedom in external memory size comes from the IC's embedded compression, which can be used to double the capacity of

the memory at the expense of a slight decrease in signal-to-noise ratio.

Figure 5 shows a block diagram of a typical application with the new IC, while Figure 6 shows a chip photomicrograph. Table 1 shows an overall comparison of the characteristics of the new video format conversion IC with the earlier design.

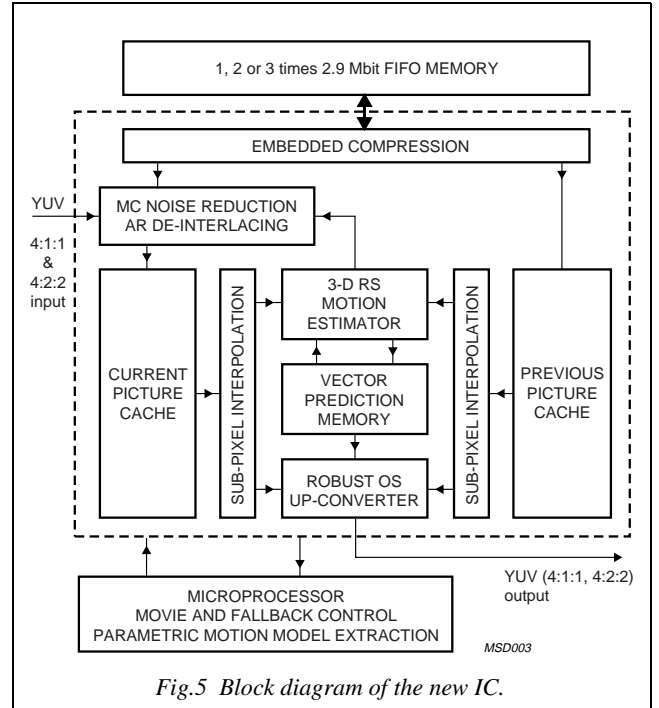


Fig.5 Block diagram of the new IC.

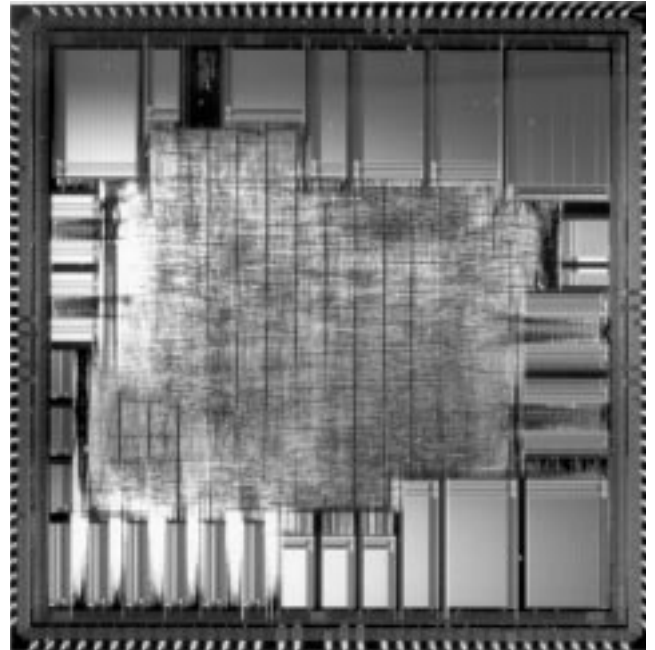


Fig.6 Photomicrograph of the new IC.

Table 1 New video format conversion IC compared to previous generation.

	Previous (1995) IC	New video format conversion IC
Process	CMOS 0.8 micron	CMOS 0.35 micron
Die size	97 mm ²	72 mm ²
Transistor count	1.10 ⁶	4.10 ⁶
Data clock	32/27 MHz	32/27 MHz
Package	PLCC84	QFP160
Dissipation	1.8 W	1.2 W
μP-interface	UART-bus	UART-bus
ME/MC range	32 × 18 (H × V)	64 × 24 (H × V)
Vector resolution	1 pixel	0.25 pixel
Data format	Y/U/V, 8-bit, 4:1:1	Y/U/V, 8-bit, 4:1:1 and 4:2:2
Video input	50 Hz/625/2:1, 60 Hz/525/2:1	50 Hz/625/2:1, 60 Hz/525/2:1
Video output	50/60/100/120 Hz, 2:1 and 1:1	50/60/100/120 Hz, 2:1 and 1:1
Film detector	2–2 pull-down	2–2 and 2–3 pull-down

4 CONCLUSION

This paper describes the second generation IC for motion compensated television format conversion. It includes all recent progress in motion estimation, motion compensation, de-interlacing and noise reduction. The IC converts 50 and 60 Hz broadcasts to display formats that may differ in picture frequency, with vertical scanning that can be either interlaced or progressive. Film material is automatically distinguished from video camera signals, and the conversion is adapted to give a judder-free motion portrayal of film scenes. Both 2–3 pull-down and 2–2 pull-down film judder can be eliminated. Efficient use of external memory comes from data compression on the IC, and by combining the video format conversion with on-chip motion compensated temporal noise reduction, continuous vertical zoom, and peaking functions.

To summarize, the IC ensures judder-free motion of film material, along with high quality video on interlaced and progressive displays regardless of their picture frequency.

ACKNOWLEDGMENT

The author wishes to thank the many colleagues from Philips Research, Semiconductors, and Consumer Electronics for their contribution to the success of this project.

REFERENCES

- [1] V. D'Alto, A. Cremonesi, C. Heintz, K. Oistamo, J. Urban, M. Karlsson, A. Vindigni and S. Dal Poz, "A highly integrated processor for improved quality television", *Digest of the ICCE*, Chicago, Jun. 1995, pp. 42-43.
- [2] C. v. Reventlow, M. Mencke, G. Scheffler, M. Schu, F. Petter, B. Schätzler, J. Krause, and R. Schwendt, "A low cost 100 Hz upconversion MCM", *Digest of the ICCE*, Jun. 1997, Chicago, pp. 410-411.
- [3] *Preliminary data sheet of Genesis gmVLD8, 8 bit Digital Video Line Doubler*, version 1.0, Jun. 1996.
- [4] Data sheet of Philips SAA4990H, "Progressive scan, Zoom, and Noise reduction IC (PROZONIC)", available from the Internet at: www.semiconductors.philips.com.
- [5] A.M. Bock, "Motion-Adaptive Standards Conversion Between Formats of Similar Field Rates", *Signal Processing: Image Communication*, Vol. 6, no. 3, Jun. 1994, pp. 275-280.
- [6] N. Seth-Smith and G. Walker, "Flexible Upconversion for High Quality TV and Multimedia Displays", *Digest of the ICCE*, Jun. 1996, Chicago, pp. 338-339.
- [7] G. de Haan, J. Kettenis, and B. De Loore, "IC for motion compensated 100 Hz TV, with a smooth motion movie-mode", *IEEE Tr. on CE*, vol. 42, May 1996, pp. 165-174.
- [8] R.J. Schutten and G. de Haan, "Real-time 2-3 pull-down elimination applying motion estimation/compensation on a programmable device", *IEEE Tr. on CE*, Vol. 44, No. 3, Aug. 1998, pp. 930-938.
- [9] H. Blume and H. Schröder, "Image Format Conversion – Algorithms, Architectures, Applications", in *Proceedings of the ProRISC/IEEE Workshop on Circuits, Systems and Signal Processing*, Mierlo, The Netherlands, November 1996, pp. 19-37.
- [10] M. Braun, M. Hahn, J.R. Ohm and M. Talmi, "Motion-Compensating Real-Time Format Converter for Video on Multimedia Displays", *Proc. IEEE Intern. Conf. on Image Processing*, Santa Barbara, Oct. 26-29, 1997, pp. 125-128.
- [11] G. de Haan, P. Biezen, H. Huijgen, and O. Ojo, "True motion estimation with 3-D recursive search block-matching", *IEEE Tr. on Circ. and Syst. for Video Techn.*, vol. 3, No. 5, Oct. 1993, pp. 368-388.

- [12] G. de Haan, and P. Biezen, "Sub-pixel motion estimation with 3-D recursive search block-matching", *Signal Processing: Image Communication* 6, 1994, pp. 229-239.
- [13] G. de Haan and P. Biezen, "An efficient true-motion estimator using candidate vectors from a parametric motion model", *IEEE Tr. on Circ. and Syst. for Video Techn.*, Vol. 8, no. 1, Mar. 1998, pp. 85-91.
- [14] G. de Haan and E.B. Bellers, "Deinterlacing – An overview", *Proceedings of the IEEE*, Vol. 86, No. 9, Sep. 1998, pp. 1839-1857.
- [15] G. de Haan and E.B. Bellers, "De-interlacing of video data", *IEEE, tr. on Consumer Electronics*, Vol. 43, no. 3, Aug. 1997, pp. 819-825.
- [16] G. de Haan, P. Biezen and O.A. Ojo, "An Evolutionary Architecture for Motion-Compensated 100 Hz Television", in *IEEE Tr. on Circ. and Syst. for Video Techn.*, Vol. 5, No. 3, June 1995, pp. 207-217.
- [17] O.A. Ojo and G. de Haan, "Robust motion-compensated video up-conversion", *IEEE Tr. on CE*, Vol. 43, No. 4, Nov. 1997, pp. 1045-1055.
- [18] J.G. Raven, "Noise suppression circuit for a video signal", UK Patent Application no. GB 2083317 A, August 1981.



BIOGRAPHY

Gerard de Haan was born in Leeuwarden, The Netherlands, on April 4, 1956. He received a B.Sc., M.Sc., and Ph.D. from Delft University of Technology in 1977, 1979 and 1992 respectively. In 1979 he joined the Philips Research Laboratories in Eindhoven. He has led research projects in the area of image processing, participated in European projects, and coached students from various universities. Since 1988, he has taught at the Philips Centre for Technical Training. In 1991/1992, he was a visiting researcher in the Information Theory Group of Delft University. In 1994, he was a Guest Editor for *Signal Processing: Image Communications* for a special issue on Video Format Conversion. At present, he is a Senior Scientist in the Television Systems group of Philips Research, and has a particular interest in algorithms for motion estimation, scan rate conversion, and image enhancement. His work in these areas has resulted in some 45 papers, more than 40 patents and patent applications, and several commercially available ICs. He was the first place winner in the 1995 ICCE Outstanding Paper Awards program, the second place winner in 1997 and in 1998, and the 1998 recipient of the Gilles Holst Award. The Philips "Natural Motion" television concept, based on his PhD-study, received the European Innovation Award of the Year 95/96 from the European Imaging and Sound Association. Dr. de Haan is a Senior Member of the IEEE.