True-Motion Estimation with 3-D Recursive Search Block Matching

Gerard de Haan, Paul W. A. C. Biezen, Henk Huijgen, and Olukayode A. Ojo

Abstract—**A new recursive block-matching motion estimation algorithm with only eight candidate vectors per block is presented. A fast convergence and a high accuracy, also in the vicinity of discontinuities in the velocity plane, was realized with such new techniques as bidirectional convergence and convergence accelerators. A new search strategy, asynchronous cyclic search, which allows a highly efficient implementation, is presented. A new block erosion postprocessing proposal further effectively eliminates block structures from the generated vector field. Measured with criteria relevant for the field rate conversion application, the new motion estimator is shown to have a superior performance over alternative algorithms, while its complexity is significantly less.**

I. Introduction

VARIOUS algorithms for consumer display scan rate conversion have been proposed [1]–[8], but their common drawback is decreased dynamic resolution. The alternative provided by motion compensation techniques [9]–[12] seems, due to the complexity of the motion estimator, to be by far too expensive for consumer television applications, probably for a long time to come. The existing simpler, albeit still expensive [13], motion estimation algorithms [14]–[16], on the other hand, cause artifacts in the up-converted images that are considered to be worse than the blur due to nonmotion compensated field rate doubling. To illustrate the above, Fig. 1 shows the blur effect of a field repetition algorithm on a sequence, showing a girl (Renata) moving in front of a zooming and panning camera. With motion compensation techniques this blur, which increases with increasing speed, can be eliminated in many picture parts, as shown in Fig. 2. However, the perceptual effect of the errors, which is clearly visible in Fig. 2, started the thinking about a new motion estimation method specifically suitable for field rate conversion. It was concluded from the pictures in Figs. 1 and 2 that a strong local distortion seems worse than a global degeneration, even when the latter option yields a significantly higher mean squared error. The starting hypothesis for the present research, therefore, was that the generated velocity field should be smooth in the first place, and that accuracy, at least initially, could be considered to be of secondary importance. Consequently, the algorithm must contain elements that impose a smoothness constraint on the resulting motion vectors.

Manuscript received December 12, 1992; revised June 2, 1993.

The authors are with the Visual Communication Systems Group, Philips Research Laboratories, 5600 JA Eindhoven, the Netherlands.

IEEE Log Number 9211372.

Fig. 1. Motion blur with a field repetition algorithm is clearly noticeable, but not very annoying.

Fig. 2. With motion compensation the artifacts can be very unnatural. In this example the OTS algorithm of [16] was used.

On the other hand, the additional constraints should not lead to a computationally complex algorithm, such as that known from computer vision research [17]. The hardware consequences, if possible, should be taken into consideration from the very beginning of the algorithm design.

II. 1-D Recursive Search

The block-matching algorithms that are most attractive for VLSI implementation limit the number of candidate vectors to be evaluated. This can be realized through recursively optimizing a previously found vector (prediction). This vector can be a spatially [16] or a temporally neighboring result [15]. The performance of these algorithms in the application of field rate conversion, however, proves to be rather poor. Fig. 2, e.g., was obtained apply-

ing the one-at-a-time search of [16]. The resulting smoothness of these algorithms is insufficient; this is assumed to be caused by the large number of evaluated candidate vectors, located around the spatial or temporal prediction value. This can cause strong deviation from this prediction, i.e., inconsistencies in the velocity field, as the vector selection criterion applied in block matching (minimum match error) is no guarantee for obtaining true motion vectors. A more successful algorithm for obtaining true motion vectors, phase plane correlation [18]–[20], allows the match-error criterion to select only between a very limited number of very likely candidate vectors. Similarly, if spatially or temporally neighboring displacement vectors are believed to reliably predict the displacement, a recursive algorithm should enable true motion estimation, if the amount of updates around the prediction vector is limited to a minimum. The updates were originally [21] arranged around the prediction value, similar to the candidate set $CS^N(\underline{X},t)$ in the last (Nth) step of 2-D logarithmic search, according to [22]. The spatial prediction however, was excluded from the candidate set:

$$CS^i(\underline{X},t) = \left\{\underline{C} \in CS^{\max} | \underline{C} = \underline{D}^{i-1}(\underline{X},t) + \underline{U}, \underline{U} = \begin{pmatrix} \pm L \\ 0 \end{pmatrix} \vee \begin{pmatrix} 0 \\ \pm L \end{pmatrix}\right\} \quad (1)$$

where L is the update length, which is measured on the frame (pixel) grid, $\underline{X} = (X,Y)^T$ is the position on the block grid, t is time, and the prediction vector $\underline{D}^{i-1}(\underline{X},t)$ is selected according to:

$$\underline{D}^{i-1}(\underline{X},t) \in \begin{cases} \{\underline{0}\}, & (i=1) \\ \{\underline{C} \in CS^{i-1}(\underline{X},t) | e(\underline{C},\underline{X},t) \le e(\underline{F},\underline{X},t), \forall \underline{F} \in CS^{i-1}(\underline{X},t)\}, & (i>1) \end{cases} \quad (2)$$

and the candidate set is limited to a set $CS^{\max}$:

$$CS^{\max} = \{\underline{C} | -N \le C_x \le +N, -M \le C_y \le +M\} \quad (3)$$

The resulting estimated displacement vector $\underline{D}(\underline{x},t)$, which is assigned to all pixel positions, $\underline{x} = (x,y)^T$, in the block $B(\underline{X})$ of size $X * Y$ with center $\underline{X}$:

$$B(\underline{X}) = \{\underline{x} | X_x - X/2 \le x \le X_x + X/2 \wedge X_y - Y/2 \le y \le X_y + Y/2\} \quad (4)$$

equals the candidate vector $\underline{C}(\underline{X},t)$ with the smallest error $e(\underline{C},\underline{X},t)$:

$$\forall \underline{x} \in B(\underline{X}): \underline{D}(\underline{x},t) \in \{\underline{C} \in CS^i(\underline{X},t) | e(\underline{C},\underline{X},t) \le e(\underline{F},\underline{X},t), \forall \underline{F} \in CS^i(\underline{X},t)\} \quad (5)$$

Errors are calculated as summed absolute differences (SAD):

$$e(\underline{C},\underline{X},t) = \sum_{\underline{x} \in B(\underline{X})} |F(\underline{x},t) - F(\underline{x} - C, t - n.T)| \quad (6)$$

where $F(\underline{x},t)$ is the luminance function and T the field period. The block size is fixed to $X = Y = 8$, although experiments indicate little sensitivity of the algorithm to this parameter. The prediction vector $\underline{D}^{i-1}(\underline{X},t)$ is a previously calculated displacement vector taken from a position spatially adjacent to the current block. Many options can be thought of that are completely analogous to options mentioned in, e.g., [23], and [24] for the pel-recursive algorithms: The result from the block at the left-hand side, the block above, the best of these two, or a weighted average of resulting vectors in a spatial neighbourhood can be considered. We will return to the subject later and conclude here with the general relation between the (one) spatial prediction vector (from now on indicated with $\underline{S}(\underline{X},t)$ rather than $\underline{D}^{i-1}(\underline{X},t)$), and the vectors previously calculated:

$$\underline{S}(\underline{X},t) = \underline{D}(\underline{X} - \underline{SD},t) \quad (7)$$

where $\underline{SD}$ ($SD_x \gtrsim 0$ and $SD_y \gtrsim 0$) points from the center of the block from which the prediction vector is taken to the center of the current block.

III. 2-D Recursive Search

A recursive search block-matching algorithm as discussed thus far has the drawback of one-dimensional convergence. Hence, at boundaries of moving objects a run-in occurs, which visibly deteriorates the contours when the vectors are applied for temporal interpolation of pictures. It is already well known from pel-recursive techniques (e.g., [23]) that, with predictions calculated from a two-dimensional area or even a (motion compensated) three-dimensional space, the convergence can be improved considerably. In this section, a new two-dimensional prediction strategy is introduced that, in contrast to known techniques, does not dramatically increase the complexity of the hardware. The fundamental difficulty with a one-dimensionally recursive algorithm is that it cannot cope with discontinuities in the velocity plane, as those occurring particularly at the boundaries of moving objects. The first impression may be that smoothness constraints exclude good step response in a motion estimator design. The dilemma of combining smooth vector fields with a good step response can be circumvented, however, as will be shown in this section. The method was earlier published in [21] and [25].

Let us assume that the discontinuities in the velocity plane are spaced at a distance that enables convergence of the recursive block matcher in between two discontinuities. When this assumption is satisfied, the recursive block matcher yields the correct vector value at the first side of the object boundary and starts converging at the opposite (second) side. The convergence direction here points from side one to side two. Either side of the contour can be estimated correctly, depending on the convergence direction chosen, though not both simultane-

ously. Therefore, it seems attractive to concurrently apply two estimator processes, as indicated in Fig. 3, with opposite convergence directions. It can then be decided which of these two estimators yields the correct displacement vector at the output. As a selection criterion, the already available SAD of both vectors can be used. Bidirectional convergence, hereinafter referred to as 2-D C, can then be formalized as a process that yields a displacement vector:

$$\forall \underline{x} \in B(\underline{X}):\ \underline{D}(\underline{x},t) = \begin{cases} \underline{D}_a(\underline{X},t), & (e(\underline{D}_a,\underline{X},t) \le e(\underline{D}_b,\underline{X},t)) \\ \underline{D}_b(\underline{X},t), & (e(\underline{D}_a,\underline{X},t) > e(\underline{D}_b,\underline{X},t)) \end{cases} \quad (8)$$

where

$$e(\underline{D}_a,\underline{X},t) = \sum_{\underline{x}\in B(\underline{X})} |F(\underline{x},t) - F(\underline{x}-\underline{D}_a,t-T)| \quad (9)$$

and

$$e(\underline{D}_b,\underline{X},t) = \sum_{\underline{x}\in B(\underline{X})} |F(\underline{x},t) - F(\underline{x}-\underline{D}_b,t-T)| \quad (10)$$

while $\underline{D}_a$ and $\underline{D}_b$ are found in a spatial recursive process as described in Section II, updating, respectively, prediction vectors $\underline{S}_a(\underline{X},t)$:

$$\underline{S}_a(\underline{X},t) = D_a(\underline{X} - \underline{SD}_a,t) \quad (11)$$

and $\underline{S}_b(\underline{X},t)$:

$$\underline{S}_b(\underline{X},t) = D_b(\underline{X} - \underline{SD}_b,t) \quad (12)$$

where

$$\underline{SD}_a \neq \underline{SD}_b. \quad (13)$$

As indicated in (13), the two estimators have unequal spatial recursion vectors $\underline{SD}$. If the two convergence directions are opposite (or at least different, as will be discussed later), then 2-D C solves the run-in problem at the boundaries of moving objects. This is because one of the estimators will have converged already at the position where the other is yet to do so. Hence, the concept combines the consistent velocity field of a recursive process with the fast step response as required at the contours of moving objects. The attractiveness of a convergence direction varies significantly for hardware. Referring to Fig. 4, the predictions taken from blocks 1, 2, or 3, are convenient for hardware. Block 4 is less attractive, as it complicates pipelining of the algorithm (the previous result has to be ready before the next can be calculated). Block 5 is not attractive, as the causality problem must be solved by reversing the line scan. This costs several line memories in the hardware. Finally, blocks 6, 7, and 8 are totally unattractive, as they require field memories for reversing the vertical scan direction. When applying only the preferred blocks, the best implementation of 2-D C results with predictions from blocks 1 and 3 for estimators a and b, respectively. The angle between the convergence directions can be enlarged by taking predictions from blocks P and Q in Fig. 4 (equally attractive for hardware),

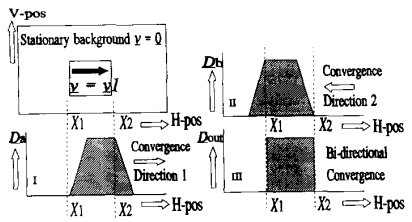


Fig. 3. The bidirectional convergence principle.

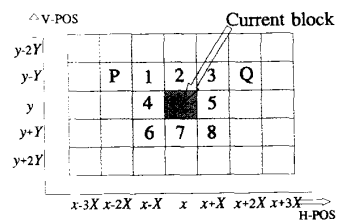


Fig. 4. Locations around the current block, from which the estimation result could be used as a spatial prediction vector.

though the spatial prediction distance then increases. This was, however, found to perform worse experimentally than 2-D C with the predictions taken from blocks 1 and 3. A solution for the suboptimal situation of the convergence directions turns out to be necessary and will be discussed in the next section. Fig. 5 shows the convergence directions for the preferred design, applying the most convenient spatial predictors.

IV. 3-D Recursive Search

Both estimators (a and b) in the algorithm thus far produce four candidate vectors each ($\underline{C}_n$) by updating their spatial predictions $\underline{S}_a(X,t)$ and $\underline{S}_b(X,t)$. The best candidate, selected with the SAD criterion, is assigned as the resulting displacement vector $\underline{D}(x,t)$ to all pixels in the block $B(\underline{X})$ of size $X * Y$ and centered at $\underline{X}$ in the present field. The spatial predictions were selected to yield two perpendicular diagonal convergence axes:

$$\underline{S}_a(\underline{X},t) = \underline{D}_a\left(\underline{X} - \begin{pmatrix} X \\ Y \end{pmatrix}, t\right),$$
$$\underline{S}_b(\underline{X},t) = \underline{D}_b\left(\underline{X} - \begin{pmatrix} -X \\ Y \end{pmatrix}, t\right). \quad (14)$$

The suboptimal situation arises from the assumed difficulty in obtaining convergence, which is (partly) opposite to the scanning of the pictures. There is a method that realizes convergence from the bottom to the top of the screen (see [26]). Predictions can be taken (e.g., from spatial positions 6–8 in Fig. 4) without causality problems if the results from a previously calculated vector field are used. The underlying assumption is that the displacements between two consecutive velocity planes, due to movements in the picture, are small compared to the block size. This assumption enables the definition of a third and a fourth estimator, c and d, in addition to the estimators a

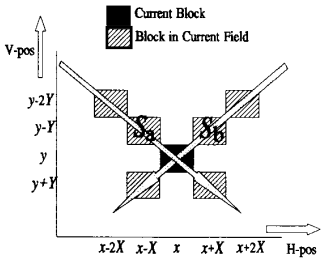


Fig. 5. Location of the spatial predictions of estimators a and b with respect to the current block. The arrows indicate the convergence directions.

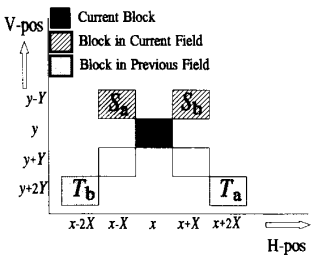


Fig. 6. The relative positions of the spatial predictors $\underline{S}_a$ and $\underline{S}_b$ and the convergence accelerators $\underline{T}_a$ and $\underline{T}_b$.

and b and also generating four candidate vectors by updating the prediction value. Selecting the predictions for estimators c and d from positions 6 and 8 in Fig. 4, respectively, yields additional convergence directions exactly opposite to those of estimators a and b. The resulting design, however, is not symmetrical in its convergence behaviour, as the convergence speed is much lower due to the temporal component in the prediction delays of estimators c and d. Recognizing this limitation, there are simpler options to benefit from candidates taken from a previous vector field. Rather than choosing the additional estimators c and d, it is suggested here to apply vectors from positions opposite to the spatial prediction position as additional candidates in the already defined estimators. This saves hardware, as fewer errors have to be calculated. Furthermore, working with fewer candidates reduces the risk of inconsistency [27]. The concept now is that a fifth candidate in each original estimator, a temporal prediction value from the previous field ($\underline{T}_a$ and $\underline{T}_b$ for estimators a and b, respectively), accelerates the convergence of the individual estimators by introducing a look ahead into the convergence direction. These convergence accelerators CA's are not taken from the corresponding block in the previous field ($\underline{D}(\underline{X}, t-T)$), but from a block shifted diagonally over r blocks and opposite to the blocks from which the spatial predictions $\underline{S}_a$ and $\underline{S}_b$ result:

$$\underline{T}_a(\underline{X},t) = \underline{D}\left(\underline{X} + r \cdot \begin{pmatrix} X \\ Y \end{pmatrix}, t-T\right),$$

$$\underline{T}_b(\underline{X},t) = \underline{D}\left(\underline{X} + r \cdot \begin{pmatrix} -X \\ Y \end{pmatrix}, t-T\right). \quad (15)$$

Increasing r implies a larger look ahead, but the reliability of the prediction decreases with increasing spatial distance (r), as the correlation between the vectors in a velocity plane can be expected to drop with increasing distance. $r = 2$ has been experimentally found to be best for a block size of 8*8 pixels. The resulting relative positions are drawn in Fig. 6. For the resulting 3-D RS block-matching algorithm, the displacement vector $\underline{D}(\underline{x}, t)$ is calculated according to equations (8), (9), and (10), where $\underline{D}_a(\underline{X}, t)$ and $\underline{D}_b(\underline{X}, t)$ result from estimators a and b, respectively, calculated in parallel with the candidate set CS_a:

$$CS_a(\underline{X},t) = \left\{\underline{C} \in CS^{\max} | \underline{C} = \underline{D}_a\left(\underline{X} - \begin{pmatrix} X \\ Y \end{pmatrix}, t\right) + \underline{U},\right.$$

$$\left.\underline{U} = \begin{pmatrix} \pm L \\ 0 \end{pmatrix} \vee \begin{pmatrix} 0 \\ \pm L \end{pmatrix}\right\} \cup \left\{\underline{D}\left(\underline{X} + 2 \cdot \begin{pmatrix} X \\ Y \end{pmatrix}, t-T\right)\right\} \quad (16)$$

and CS_b:

$$CS_b(\underline{X},t) = \left\{\underline{C} \in CS^{\max} | \underline{C} = \underline{D}_b\left(\underline{X} - \begin{pmatrix} -X \\ Y \end{pmatrix}, t\right) + \underline{U},\right.$$

$$\left.\underline{U} = \begin{pmatrix} \pm L \\ 0 \end{pmatrix} \vee \begin{pmatrix} 0 \\ \pm L \end{pmatrix}\right\} \cup \left\{\underline{D}\left(\underline{X} + 2 \cdot \begin{pmatrix} -X \\ Y \end{pmatrix}, t-T\right)\right\} \quad (17)$$

while errors are assigned to candidate vectors using the SAD criterion of equation (6). Due to the faster convergence, the CA's are particularly advantageous at the top of the screen, where the spatial process starts converging. Furthermore, they improve the temporal consistency.

V. Updating Strategy

The optimization of the constant L (the update length) in the definitions (16) and (17) of the candidate sets of each estimator $CS_a(\underline{X}, t)$ and $CS_b(\underline{X}, t)$ reveals some conflicting demands. A small integer value of the update allows convergence to accurate output vectors, whereas a relatively high value is expected to enable quicker convergence. It is possible to circumvent the choice by adapting L individually for estimators a and b, e.g., to the SAD error value of the spatial prediction. This option was indicated in [21]. Alternatives, however, were investigated (see also [26] and [28]).

It is conceivable to bring down the number of candidate vectors deviating from $\underline{0}$ to the very limit, one only. Any other number is possible as well without losing either of the aforementioned options on the average or adding excessive complexity to determine the candidates. Apart from the zero update, only one update vector $\underline{U}(\underline{X}, t)$ is added to the spatial predictions $\underline{S}_a(\underline{X}, t)$ and $\underline{S}_b(\underline{X}, t)$ in both estimators a and b. The components of the second update originally were generated with two independent pseudorandom generators. However, experiments indicate

that there is no need for these random generators, as a simple cyclic update generator realized with a counter and a look-up table (LUT) already yields good results, particularly when the counter runs asynchronously with the scanning of the blocks in a picture. For this final version of the 3-D RS block matcher that will be used in the evaluation section, the displacement range again is limited to $CS^{\max}$:

$$CS^{\max} = \{\underline{C} \mid -N \le C_x \le +N, -M \le C_y \le +M\} \quad (18)$$

and the proposed candidate set $CS_a(\underline{X}, t)$ for estimator a applying this updating strategy, hereinafter referred to as asynchronous cyclic search (ACS), is found as:

$$CS_a(\underline{X}, t) = \left\{\underline{C} \in CS^{\max} | \underline{C} = \underline{D}_a\left(\underline{X} - \begin{pmatrix} X \\ Y \end{pmatrix}, t\right) + \underline{U}_a(\underline{X}, t)\right\} \cup \left\{\underline{D}\left(\underline{X} + 2 \cdot \begin{pmatrix} X \\ Y \end{pmatrix}, t - T\right), \underline{0}\right\} \quad (19)$$

where the update vectors $\underline{U}_a(\underline{X}, t)$ are found as

$$\underline{U}_a(\underline{X}, t) \in \{\underline{0}, \underline{\text{lut}}(N_{bl}(\underline{X}, t) \bmod p)\} \quad (20)$$

where N_{bl} is the output of a block counter, $\underline{\text{lut}}$ is a look-up table function, and p is not a factor of the number of blocks in a picture. For estimator b, similarly, the candidate set $CS_b(\underline{X}, t)$ is found as

$$CS_b(\underline{X}, t) = \left\{\underline{C} \in CS^{\max} | \underline{C} = \underline{D}_b\left(\underline{X} - \begin{pmatrix} -X \\ +Y \end{pmatrix}, t\right) + \underline{U}_b(\underline{X}, t)\right\} \cup \left\{\underline{D}\left(\underline{X} + 2 \cdot \begin{pmatrix} -X \\ Y \end{pmatrix}, t - T\right), \underline{0}\right\} \quad (21)$$

while, possible from the same LUT, updates $\underline{U}_b(\underline{X}, t)$ are generated as

$$\underline{U}_b(\underline{X}, t) \in \{\underline{0}, \underline{\text{lut}}((N_{bl}(\underline{X}, t) + \text{offset}) \bmod p)\} \quad (22)$$

which differ from $\underline{U}_a(\underline{X}, t)$ due to the integer offset added to the value of the block rate counter. The estimator n ($n = a, b$) yields a vector $\underline{D}_n(\underline{X}, t)$ chosen from the candidate set $CS_n(\underline{X}, t)$ [see (19) and (21)] such that it minimizes the matching error:

$$e(\underline{C}, \underline{X}, t) = \sum_{\underline{x} \in B(\underline{X})} |F(\underline{x}, t) - F(\underline{x} - \underline{C}, t - T)| \quad (23)$$

where the match error is summed over a block $B(\underline{X})$, defined as

$$B(\underline{X}) = \{\underline{x} | X_x - X/2 \le x \le X_x + X/2, X_y - Y/2 \le y \le X_y + Y/2\}. \quad (24)$$

The best of the two vectors resulting from estimators a and b is selected in the output multiplexer and assigned to all pixels in $B(\underline{X})$ [see (8)].

The vector $\underline{0}$ in the candidate sets $CS_a(\underline{X}, t)$ and $CS_b(\underline{X}, t)$ [(19) and (21)] improves the performance for small stationary image parts but introduces a serious risk of disturbing the convergence. In Section VI we will return to this issue. The optimal search area will most likely reveal a symmetrical distribution of candidate vectors around the spatial prediction vector, as there is no preferred velocity direction. This can be translated into the *a priori* expectation that the output of the generators (the modulo p counter plus LUT) on the average should be zero. Further, the variance of the vertical update is smaller than that of the horizontal update, as large horizontal movements seem to occur more frequently than large vertical movements. It may be advantageous to adapt the distributions at the output of the update generator to, e.g., the match errors obtained, in the sense that a narrower distribution is chosen in case of small errors. These topics can be subject for further research. Good results (see the evaluation in Section IX) were obtained from an estimator with the ACS strategy where the LUT contained the following updates:

$$US_n = \left\{\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \begin{pmatrix} 0 \\ -2 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 3 \\ 0 \end{pmatrix}, \begin{pmatrix} -3 \\ 0 \end{pmatrix}\right\}. \quad (25)$$

To realize a symmetrical distribution around $\underline{0}$ with p updates, a number of $\underline{0}$ updates and symmetrical pairs of the other updates can be added at will to arrive at a value of p that is not a factor of the number of blocks in a field [30].

VI. Further Emphasis on Smoothness

It happens that blocks shifted over very different vectors with respect to the current block contain the same information, particularly on periodic structures. A block matcher therefore will randomly select one of these vectors due to small differences in the matching error caused by noise in the picture. If the estimate is used for temporal interpolation, very disturbing artifacts will be generated in the periodic picture part. For the 3-D RS block matcher, the spatial consistency could guarantee that, after reaching a converged situation at the boundary of a moving object, no other vectors will be selected. This, however, functions only if none of the other candidate vectors that yield an equally good matching error are ever generated. A number of risks jeopardize this constraint:

1) An element of the update sets $US_a(\underline{X}, t)$ and $US_b(\underline{X}, t)$ may equal a multiple of the basic period of the structure.
2) "The other" estimator may not be converged, or may be converged to a value that does not correspond to the actual displacement.
3) Directly after a scene change, it is possible that one of the convergence accelerators $\underline{T}_a(\underline{X}, t)$ or $\underline{T}_b(\underline{X}, t)$ yields the threatening candidate.

It is possible to improve the result, as far as risks mentioned under 1) and 3) are concerned, by adding penalties to the error function related to the length of the differ-

ence vector between the candidate to be evaluated and some neighbouring vectors. This was mentioned in [31]. For the 3-D RS block matcher a very simple implementation is realized with a penalty depending on the length of the update:

$$e(\underline{C}, \underline{X}, t) = \sum_{\underline{x} \in B(\underline{X})} |F(\underline{x}, t) - F(\underline{x} - \underline{C}, t - T)| + \alpha \cdot \|\underline{U}(\underline{X}, t)\| \quad (26)$$

For the CA's, this is not applicable, but a fixed penalty equal to α, independent of the difference with $\underline{S}_a(\underline{X}, t)$ or $\underline{S}_b(\underline{X}, t)$, proved to give satisfactory results. Experimental optimization of α led to a value equal to 0.24% of the maximum error value. Other experiments, however, indicate that fixed penalties for all updates can be applied. Optimization led to values for these fixed penalties of, respectively, 0.4%, 0.8%, and 1.6% of the maximum error value, for the cyclic update, the convergence accelerator, and the fixed $\underline{0}$ candidate vector. This last candidate especially requires a large penalty, as it introduces the risk of convergence interruption in flat areas.

The risk mentioned under 2), however, is not reduced with these update penalties. The situation described typically occurs if a periodic part of an object enters the picture from the blanking or appears from behind another object in the image. In that situation, one of the two estimators can converge to the wrong vector value, since there is no boundary moving along with the periodic picture part to prevent this. Therefore, an experiment was set up, with $\underline{S}_a(\underline{X}, t)$ set to the value of $\underline{S}_b(\underline{X}, t)$ if

$$e(\underline{D}_a, \underline{X} - \underline{SD}_a, t) > e(\underline{D}_b, \underline{X} - \underline{SD}_b, t) + Th \quad (27)$$

where Th is a fixed threshold, and, reversely $\underline{S}_b(\underline{X}, t)$ is set to the value of $\underline{S}_a(\underline{X}, t)$ in case

$$e(\underline{D}_b, \underline{X} - \underline{SD}_b, t) > e(\underline{D}_a, \underline{X} - \underline{SD}_a, t) + Th \quad (28)$$

This attempt to cope with the problem through linking the two estimators was based upon the following hypothesis: In the first blocks next to the horizontal blanking interval (or the "other object"), one estimator will be converged, whereas the other must start from its initialization value, or from the velocity of the other object. Usually this will result in a remarkable difference in the match errors $e(\underline{D}_a, \underline{X}, t)$ and $e(\underline{D}_b, \underline{X}, t)$ of the two estimators a and b. Substitution of the spatial prediction vector of the worst matching estimator by the spatial prediction vector of the best matching estimator prevents the algorithm from converging to the wrong value. The threshold Th in (27) and (28), above which the predictions are linked, turned out to be useful, as the advantage of two independent estimators would otherwise be lost. In [29], an adapted version of the penalties of updating is mentioned, applying coarser quantization of the match errors in an attempt to arrive at the same goal.

VII. Block Erosion to Eliminate Blocking Effects

If the motion information is limited to one vector per block of pixels, motion compensation sometimes introduces visible block structures in the interpolated picture. The block sizes commonly used in block matching are in a range that gives rise to very visible artifacts [32]. A post-filter on the vector field can overcome this problem. The option was mentioned in [15] but has the drawback that discontinuities in the vector field are blurred as well. Therefore a post-operation is introduced in this section; it eliminates fixed block boundaries from the vector field without blurring contours. Furthermore, an option is found that prevents vectors that did not result from the estimator from being generated. This is attractive mainly for algorithms that yield vectors that have a poor relation to the actual object velocities. In case of a full search method, for example, it is not unlikely that the average of two vectors yielding a low match error will result in a vector that gives a bad match. The method was published in [26] and [33]. In case of a velocity field for which one vector per block is available, it is proposed to divide each block $B(\underline{X})$

$$B(\underline{X}) = \{\underline{x} | X_x - X/2 \le x \le X_x + X/2 \wedge X_y - Y/2 \le y \le X_y + Y/2\} \quad (29)$$

to which a vector $\underline{D}(\underline{x}, t)$ is assigned, into four sub-blocks $B_{i,j}(\underline{X})$,

$$B_{i,j}(\underline{X}) = \left\{\underline{x} | X_x - (1-i) \cdot \frac{X}{4} \le x \le X_x + (1+i) \cdot \frac{X}{4} \wedge X_y - (1-j) \cdot \frac{Y}{4} \le y \le X_y + (1+j) \cdot \frac{Y}{4}\right\} \quad (30)$$

where the variables i and j take the values $+1$ and -1. To the pixels in each of the four sub-blocks $B_{i,j}(\underline{X})$ a vector $\underline{D}_{i,j}(\underline{x}, t)$ is assigned:

$$\forall \underline{x} \in B_{ij}(\underline{X}): \underline{D}_{ij}(\underline{x}, t) = \underline{D}_{ij}(\underline{X}, t), \qquad i, j = 1, -1 \quad (31)$$

where

$$\underline{D}_{ij}(\underline{X}, t) = \underline{\text{med}}\left[\underline{D}\left(\underline{X} + i \cdot \begin{pmatrix} X \\ 0 \end{pmatrix}, t\right), \underline{D}(\underline{X}, t), \underline{D}\left(\underline{X} + j \cdot \begin{pmatrix} 0 \\ Y \end{pmatrix}, t\right)\right]. \quad (32)$$

The $\underline{\text{med}}$ function is a median on the x and y vector components separately:

$$\underline{\text{med}}(\underline{X}, \underline{Y}, \underline{Z}) = \begin{pmatrix} \text{median}(X_x, Y_x, Z_x) \\ \text{median}(X_y, Y_y, Z_y) \end{pmatrix} \quad (33)$$

Because of this separate operation, a new vector that was neither in the block itself nor in the neighbouring blocks, can be created. To prevent this, it could be checked whether the new vector is equal to one of the three input

vectors. And in case

$$\underline{\text{med}}\left(\underline{D}\left(\underline{X}+i\cdot\begin{pmatrix}X\\0\end{pmatrix},t\right),\underline{D}(\underline{X},t),\underline{D}(\underline{X}+j)\cdot\begin{pmatrix}0\\Y\end{pmatrix},t\right)\right)$$
$$\notin\left\{\underline{D}\left(\left(\underline{X}+i\cdot\begin{pmatrix}X\\0\end{pmatrix},t\right),\underline{D}(\underline{X},t),\right.\right.$$
$$\left.\left.\underline{D}\left(\left(\underline{X}+j\cdot\begin{pmatrix}0\\Y\end{pmatrix},t\right)\right)\right\},\qquad(34)$$

to the pixels in the quadrant the original vector is assigned:

$$\forall\underline{x}\in B(X):\quad \underline{D}_{ij}(\underline{x},t)=\underline{D}(\underline{X},t).\qquad(35)$$

Fig. 7 illustrates the process, showing with shading the areas from which the vectors are taken to calculate the result for sub-block H_{-1-1} the neighbouring blocks E and G, and block H itself. The block erosion can be repeated for the sub-blocks in case of large initial blocks. Each sub-block is then again subdivided into four parts.

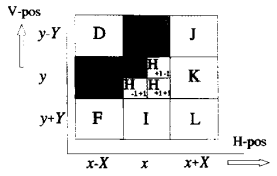


Fig. 7. The center block H divided into four sub-blocks, $H_{i,j}$, $i=+/1$ and $j=+/1$.

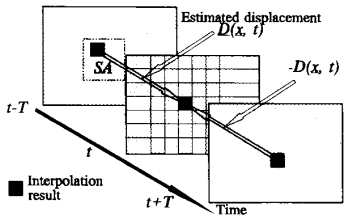


Fig. 8. A motion compensated average is calculated of fields at instants $t-T$ and $t+T$, applying vectors estimated between field t and $t-T$.

VIII. Evaluation Tools

A. The M2SE Quality Indicator for Estimated Vectors

To indicate the performance of a motion estimator, often a mean square prediction error (MSE) is used, or the entropy of the prediction error. Both are objective and relevant for coding applications of motion estimation. For field rate conversion applications, however, these measures, as already mentioned in the introduction, are not very relevant. Therefore, in this section a first attempt to arrive at objective motion estimator evaluation tools for field rate conversion is presented. As the proposed criteria are new, their assumed validity will be illustrated with pictures from vector fields, which allow a subjective comparison of the performance of some of the best algorithms.

The first proposed performance indicator is a modified mean square prediction error (M2SE). The quintessence of the modification is that the validity of the vectors is extrapolated outside the temporal interval on which they are calculated. The extrapolation, because of object inertia, is expected to be more legitimate if the vectors represent true motion than if they indicate only a good match between blocks of pixels.

As illustrated in Fig. 8, displacement vectors $\underline{D}(\underline{x},t)$ are calculated between the previous field at time $t-T$ and the present field at t, according to the definition given in (5) and (6). With vectors so defined, output sequences are created by interpolating each output field as the motion compensated average from two successive input fields, using displacement vectors from various motion estimation algorithms under evaluation. Interpolated output fields are thus found as

$$F_{mc}(\underline{x},t)=\tfrac{1}{2}\cdot[F(\underline{x}-\underline{D}(\underline{x},t),t-T)+F(\underline{x}+\underline{D}(\underline{x},t),t+T)].\qquad(36)$$

To calculate the proposed performance indicator, the squared pixel differences between the interpolated output and the original input field are summed over a field excluding a boundary area and normalized with respect to the number of pixels in this measurement window. Further, the resulting figures, obtained from five different input test sequences, are averaged. Hence the resulting M2SE performance criterion can be written as

$$\text{M2SE}(t)=\frac{1}{5}\cdot\sum_{s=1}^{5}\frac{1}{P\cdot L}\cdot\sum_{\underline{x}\in\text{MW}}[F_s(\underline{x},t)-F_{mc}(\underline{x},t)]^2\qquad(37)$$

where the index s identifies the test sequence (1, 2, 3, 4, or 5) to which the luminance function $F_s(\underline{x},t)$ belongs and on which also $F_{mc}(\underline{x},t)$ is calculated. $P\cdot L$ is the number of pixels in the measurement window MW that equals the entire image, excluding a margin of N pixels wide horizontally and M vertically, where M and N define the vector range of the motion estimator according to (3). To enable convergence of algorithms that include the use of temporal predictors, it is proposed to calculate the M2SE, defined in (37), in the fourth field of each sequence. This seems reasonable, as the human observer also needs some time to interpret pictures after a scene cut. The five test sequences were selected to provide critical test material and include several categories of difficulties. This broad range of difficulties was assumed to make the M2SE more meaningful, though the inhomogeneous data set obviously implies a rather high variance on the figures.

The low velocities and the fine details of the Renata test sequence (see Fig. 1) make it most likely that algorithms will be well converged in the fourth field, so that the M2SE measure, for this sequence, will indicate the accuracy of a method. This sequence was also applied in a version accelerated three times (highest velocity around 12 pel/T) to provide the second sequence in the test.

Now the fine details and the large difference in foreground/background velocity provide a critical test for covering and uncovering situations and, in case of a recursive algorithm, the convergence speed is tested. The Car & Gate sequence (for a picture see Fig. 19) was applied as the third test sequence. The old car shown drives towards the camera, which is zooming out. The gate is closing behind the car. The velocities are fairly low. A picture from the fourth sequence, Roller Coaster, is shown in Fig. 9. Fairly large displacements, up to 16 pel/T, are caused by the fast-moving train while, due to the looping, velocities in many directions occur. A stationary image part exists close to the fast-moving roller coaster. A picture of the fifth test sequence is shown in Fig. 10, the BBC-DRUM-TEXT sequence. A stationary text, VALVO AL, was superimposed on a picture attached to a rotating drum from which a fast (11–12 pel/T), but almost uniform, motion resulted. The main difficulty here is the strong discontinuity in the velocity field, which is caused by the superimposed text. It is challenging for algorithms that presume smoothness in the displacement vector field.

B. The Vector Field Smoothness Indicator

As discussed in the introduction, it was observed for motion compensated pictures that inconsistencies in the estimated displacement vector field are a major threat to quality. Inconsistencies could spoil the result to a level where the viewer prefers simple non-motion compensated field rate conversion techniques, such as field averaging or field repetition. It was concluded that the spatial smoothness of the velocity field is of major importance. The second performance indicator proposed for the evaluation of motion estimation algorithms, therefore, is a smoothness figure $S(t)$, defined as

$$S(t) = \sum_{\underline{X}} \sum_{k=-1}^{k=+1} \sum_{l=-1}^{l=+1} \left(\frac{8 \cdot N_b}{|\Delta_x(\underline{X}, k, l, t)| + |\Delta_y(\underline{X}, k, l, t)|} \right) \tag{38}$$

where $\underline{X}$ runs through all centers of the blocks within the fourth field of a test sequence, excluding the boundary for obvious reasons. N_b is the number of blocks in a field, and

$$\Delta_x(\underline{X}, k, l, t) = D_x(\underline{X}, t) - D_x\left(\underline{X} + \begin{pmatrix} k \cdot X \\ l \cdot Y \end{pmatrix}, t\right)$$

$$\Delta_y(\underline{X}, k, l, t) = D_y(\underline{X}, t) - D_y\left(\underline{X} + \begin{pmatrix} k \cdot X \\ l \cdot Y \end{pmatrix}, t\right) \tag{39}$$

This smoothness figure drops in case the motion estimator under test generates a vector field with many inconsistencies. A high score on this criterion, therefore, qualifies the algorithm as good for field rate conversion applications.

C. The Computational Overhead Indicator

As a final criterion, a standardized complexity measure is defined. It concerns the operations count of the algorithm, selecting a block size of 8*8 pixels (if applicable), and a motion vector range of ± 12 pel/T. The adder function is used as a unit for the operations count. Subtracters, comparison, and absolute value are assumed to yield the same complexity. Multiplications and divisions are supposed to cost 3 ops/pel, as their silicon area is approximately three times larger than that of an adder stage. This criterion can be seen as a first-order approximation of the hardware attractiveness, which makes it particularly relevant for *consumer* display conversion applications.

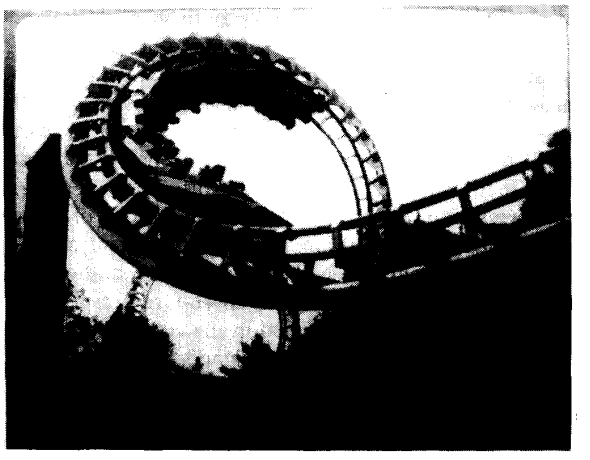


Fig. 9. A picture from Roller Coaster sequence used in the M2SE test.

Fig. 10. A picture from the BBC-DRUM-TEXT sequence.

IX. Evaluation Results

In addition to the new 3-D RS design, a fairly large number of algorithms are selected to provide a reference in the evaluation. The evaluated 3-D RS algorithm includes the asynchronous cyclic search of Section V, the penalties on updating of Section VI, the block erosion of Section VII, and sub-sampling on block and pixel level as mentioned in [26] in order to further reduce the operations count. From the block-matching methods, the full

search (FS) was included in the comparison, as this method probably yields the best possible result from all nonhierarchical block matchers. In order to include more hardware-attractive alternatives, the three-step and the four-step logarithmic search and the one-at-a-time search (OTS) methods were added to the reference list. Hierarchical methods are known to yield displacement vectors that correspond more closely to the true motion of objects in the image, and therefore they should reveal a better score on our quality scale. The two implementations (H2 and H3 for the two-level and the three-level, respectively) described in [34] were evaluated in the comparison. Phase plane correlation (PPC) was designed [18]–[20] at BBC Research for the demanding professional (television studio) field rate conversion applications. Therefore, it is regarded as representing the state of the art in motion estimation for field rate conversion as far as quality is concerned. The implementation of the PPC algorithm used is the one described in [9], though the block size was adapted from $16*16$ to $8*8$ pixels.

The M2SE scores of the algorithms are shown in Fig. 11. As can be seen from the figure, FS does not yield the best possible score on the M2SE scale, which reflects the influence of the modification in the more usual mean square prediction error criterion. Two quality groups can be distinguished: a high-quality group consisting of the PPC, the hierarchical and FS block matchers, and the 3-D RS method; and a low-quality group containing the efficient search algorithms.

The standard deviation on the M2SE scores is rather high, 14 in the high-quality category and more than 50 for the algorithms in the second group. This corresponds to our expectation, as the test sequences are very different in order to include many different categories of motion. On average, the 3-D RS block matcher's M2SE score is the best of all estimators apart from PPC. However, it should be noted that the comparison of the algorithms is not entirely justifiable for two reasons. A first complication is that the hierarchical block matcher H3 is the only estimator that calculates vectors for each block of size $2*2$ pixels. The other "good" algorithms have a block size of $8*8$. The second obstacle for a fair comparison is that the PPC does not estimate between fields t and $t-T$, to interpolate a result at t using $t-T$ and $t+T$, as proposed in subsection VIII-A. Instead, the implemented PPC algorithm estimates the displacements between $t-T$ and $t+T$ and uses these vectors to interpolate field t. This is believed to give a slight advantage to the PPC method.

The second performance indicator, the smoothness of the vector fields, has to be carefully interpreted, as no best smoothness figure can be given. It is, however, reasonable to classify an algorithm as "better" when its smoothness figure is higher, provided its M2SE figure is similar or even lower. In this sense, the graph shown in Fig. 12 clearly indicates a superior performance, in terms of vector consistency, of the 3-D RS block matcher over all alternative estimators. This is true not only on the average; it was found that the smoothness of the 3-D RS block matcher was better, for each of the (very different) test sequences, than that of any of the other tested algorithms. The smoothness score of H3 is omitted, as the smoothness criterion was designed for algorithms with one vector per block of $8*8$ or larger. The photographs at the back of this issue show that the smoothness resembles that of PPC. These photographs, Figs. 14 to 19, show for two typical pictures the vectors generated with the four best-performing algorithms. To this end, the vectors were coded as a color, which could be shown as an overlay on the pictures. As no completely objective quality measures for motion vector fields in the application of field rate conversion are known, the figures can help illustrate the criteria introduced in this paper. Table I shows how colors and vector values correspond.

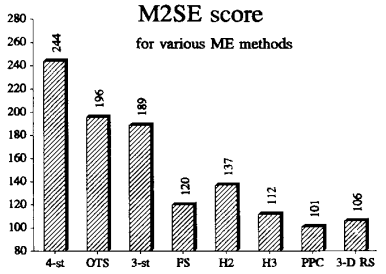


Fig. 11. Performance comparison of a set of motion estimation algorithms. To provide further reference, the non-motion compensated field average was calculated to yield an M2SE of 8855.

In the third comparison, the operations count of the 3-D RS block matcher, as it results from this paper, is compared with results from our reference algorithms. As Fig. 13 reveals, the 3-D RS design compares favorably, also in terms of operations count, with the algorithms listed. This is believed to be a consequence of taking the hardware considerations into account from the beginning of the algorithm design.

X. Conclusion

A block-recursive motion estimation algorithm was proposed in this paper. The bidirectional convergence principle enabled combination of the apparently conflicting demands for smoothness and yet steep edges in the velocity field. The method turns out to be very successful, even after adaptation for simple hardware. An additional new element is the use of what we have called convergence accelerators, a special type of temporal predictor, which are shifted with respect to the current position in order to create look-ahead function in the convergence direction. A new and very efficient updating strategy, asynchronous cyclic search, was introduced. Finally, with block erosion, block structures could effectively be removed from the resulting vector field.

Using three new test criteria, the suitability of motion estimators for use in consumer television with motion compensated field rate doubling was tested. Particularly,

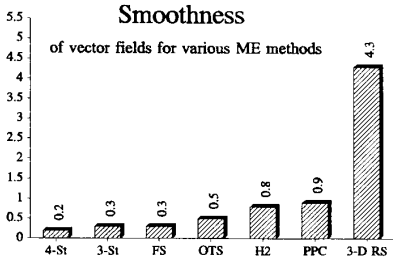


Fig. 12. Comparison of the vector field consistency score of various block-matching algorithms.

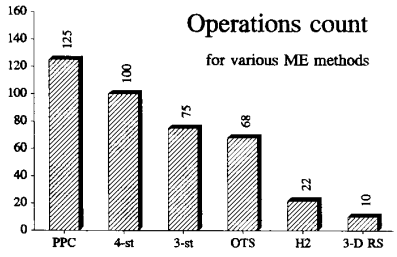


Fig. 13. Operations count of some estimators. FS and H3 are left out, but they have an operations count of 1895 and 1400 ops/pel, respectively.

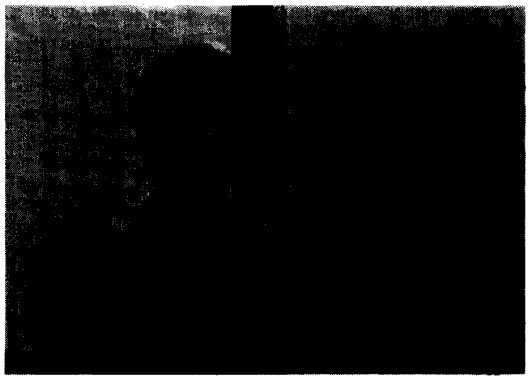


Fig. 14. Motion vectors of the 3-D RS block matcher, shown as a color overlay on the accelerated Renata sequence.

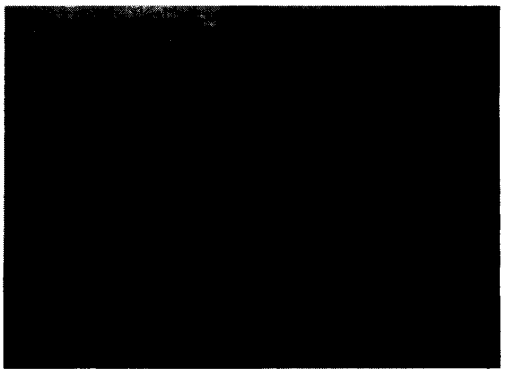


Fig. 15. Motion vectors of the phase plane correlation algorithm, shown as a color overlay on the accelerated Renata sequence.

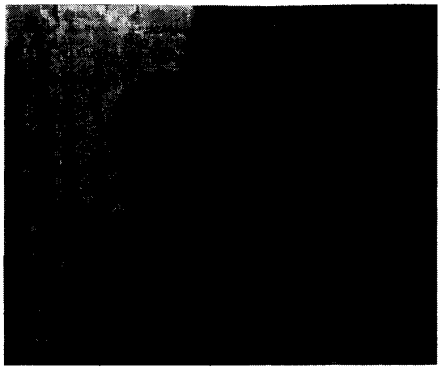


Fig. 16. Motion vectors of the three-level hierarchical block matcher, shown as a color overlay on the accelerated Renata sequence.

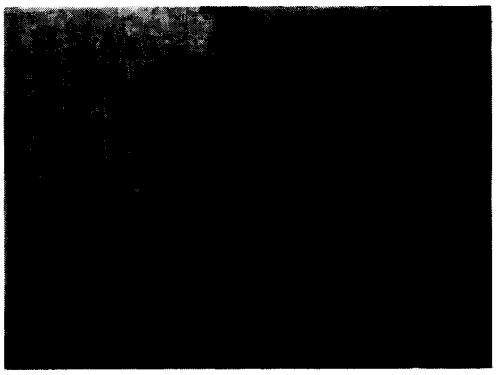


Fig. 17. Motion vectors of the full-search block matcher, shown as a color overlay on the accelerated Renata sequence.

the smoothness indicator for the estimated vector field together with the photographs of the visualized vector fields are believed to provide relevant information when judging the performance of a motion estimator intended for motion compensated field rate conversion. As no optimal smoothness figure is known, the M2SE criterion is believed to provide a useful instrument to verify that the smoothness constraint is not exaggerated. Finally, the operations count allowed an early indication of the hardware attractiveness.

It can be concluded that the newly designed motion estimation algorithm is emerging as the most attractive of the tested block-matching algorithms in the application of consumer field rate conversion. The presented vector fields leave little room for doubting that the 3-D RS estimator outperforms all tested algorithms in the comparison, while the operations count suggests that implementation is possible for a (much) lower price.

Fig. 18. Motion vectors of the 3-D RS block matcher, shown as a color overlay on the Car & Gate sequence.

Fig. 19. Motion vectors of the phase plane correlation algorithm, shown as a color overlay on the Car & Gate sequence.

TABLE I
COLORS USED TO VISUALIZE MOTION; EACH COLOR IS USED TO INDICATE POSITIVE AND NEGATIVE VALUES OF A VECTOR COMPONENT, BUT SUCH THAT A MINIMAL RISK OF CONFUSION RESULTS

Vector/Color-Overlay Coding Table							
Color	Grey	Yellow	Cyan	Green	Violet	Red	Blue
Positive	0	1	2	3	4	5	> 6
Negative		< −6	−5	−4	−3	−2	−1

ACKNOWLEDGMENT

The authors wish to thank the members of the Information Theory Group of the Delft University of Technology, and J. Biemond in particular, for their valuable help with the preparation of the text of this paper.

REFERENCES

[1] E. J. Berkhoff, U. E. Kraus, and J. G. Raven, "Application of picture memories in television receivers," *IEEE Trans. Consumer Electron.*, vol. CE-29, no. 3, pp. 251–258, Aug. 1983.

[2] C. Hentschel, "Linear and nonlinear procedures for flicker reduction," *IEEE Trans. Consumer Electron.*, vol. CE-33, no. 3, pp. 192–198, Aug. 1987.

[3] ——, "Comparison between median filtering and vertical edge controlled interpolation for flicker reduction," *IEEE Trans. Consumer Electron.*, vol. CE-35, no. 3, pp. 279–289, Aug. 1989.

[4] ——, "Television with increased image quality" (in German), Ph.D. dissertation, Tech. Univ. Braunschweig, Berlin, Feb. 1990.

[5] G. Huerkamp, "On flicker reduction in television receivers" (in German), Ph.D. dissertation, Dept. Elec. Eng., Univ. Dortmund, February, 1990.

[6] R. N. Jackson and M. J. J. C. Annegarn, "Compatible systems for high-quality television," *SMPTE J.*, pp. 719–723, July 1983.

[7] U. E. Kraus, "On the prevention of large-area flicker in consumer television receivers" (in German), *Rundfunktechnische Mitteilungen*, vol. 25, pp. 264–269, 1981.

[8] H. Schröder, B. Wendland, and G. Huerkamp, "Modes for flicker-free television display: A comparison" (in German), *Fernseh und Kinotechnik*, no. 4, pp. 134–139, 1986.

[9] G. M. X. Fernando, D. W. Parker, and P. T. Rogers, "Motion compensated display field rate conversion of bandwidth compressed HD-MAC pictures," in *Proc. 3rd Int. Workshop on HDTV and Beyond*, Torino, Italy, 1989.

[10] T. Reuter, "Improved TV standards conversion with 3-dimensional motion compensating interpolation filter," in *Proc. Club de Rennes Young TV Researchers Conf.*, Cambridge, MA, Oct. 1988.

[11] D. P. Siohan and B. Choquet, "Field-rate conversion by motion estimation/compensation," in *Signal Processing of HDTV*, L. Chiariglione, Ed. Amsterdam: Elsevier, 1988, pp. 319–328.

[12] M. Sugimoto, T. Fujio, and Y. Ninomiya, "Second generation HDTV standards converter," in *Proc. 14th Int. Symp. Extended TV and HDTV Systems*, Montreux, Switzerland, June 1985.

[13] L. De Vos, "VLSI-architectures for the hierarchical block-matching algorithms for HDTV applications," *SPIE*, vol. 1360, *Visual Communication and Image Processing*, pp. 398–409, 1990.

[14] T. Koga, K. Iinuma, A. Hirano, Y. Iilima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing," in *Proc. NTC 81*, New Orleans, LA, 1981, Paper G5.3.1.

[15] Y. Ninomiya and Y. Ohtsuka, "A motion-compensated interframe coding scheme for television pictures," *IEEE Trans. Commun.*, vol. COM-30, no. 1, 1982.

[16] R. Srinivasan and K. R. Rao, "Predictive coding based on efficient motion estimation," *IEEE Trans. Commun.*, vol. COM-33, no. 8, pp. 888–896, 1985.

[17] J. Konrad and E. Dubois, "A comparison of stochastic and deterministic solution methods in Bayesian estimation of 2-D motion," in *Proc. 1st European Conf. Computer Vision*, Antibes, France, Apr. 1990.

[18] S. C. Dabner, "Real time motion measurement hardware: Phase correlation unit," BBC Research Report BBC RD 1990/11.

[19] G. A. Thomas, "TV picture motion measurement," European Patent Application EP-A 0 261 137, *European Patent Bulletin*, no. 91/42.

[20] G. A. Thomas, "Television motion measurement for DATV and other applications," BBC Research Report BBC RD 1987/11.

[21] G. de Haan and H. Huijgen, "New algorithm for motion estimation," in *Proc. 3rd Int. Workshop on HDTV and Beyond*, Torino, Italy, 1989.

[22] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. COM-29, no. 12, 1981.

[23] J. N. Driessen, L. Böröczki, and J. Biemond, "Pel-recursive motion field estimation from image sequences," *J. Visual Commun. Image Represent.*, 1991.

[24] J. N. Driessen, "Motion estimation for digital video," Ph.D. dissertation, Delft Univ. of Tech., Sept. 1992.

[25] G. de Haan and H. Huijgen, "Motion estimation," European Patent Application EP-A 0 415 491, *European Patent Bulletin*, no. 91/10.

[26] G. de Haan and H. Huijgen, "Motion estimation for TV picture enhancement," in *Proc. 4th Int. Workshop on HDTV and Beyond*, Torino, Italy, 1991.

[27] G. de Haan, "Motion estimation and compensation," Ph.D. dissertation, Delft Univ. Tech., Sept. 1992.

[28] G. de Haan and H. Huijgen, "Video image motion vector estimation with asymmetric update region," European Patent Application EP-A 0 474 276, *European Patent Bulletin*, no. 92/11.

[29] P. H. N. de With, "A simple recursive motion estimation technique for compression of HDTV signals," in *Proc. Conf. Image Processing and its Applications*, Maastricht, the Netherlands, Apr. 1992.

[30] G. de Haan and H. Huijgen, "Motion estimation," European Patent Application 92.201.274.5, *European Patent Bulletin*, to be published.

[31] T. Reuter, "A modified blockmatching algorithm with vector reliability checking and adaptive smoothing," in *Proc. 3rd Int. Conf. Image Processing and its Applications*, Warwick, England, July 1989.

[32] M. Miyahara and K. Kotani, "Block distortion in orthogonal transform coding, analysis, minimization, and distortion measure," *IEEE Trans. Commun.*, vol. COM-33, no. 1, pp. 90–96, 1985.

[33] G. de Haan and H. Huijgen, "Motion vector processing device," European Patent Application EP-A 0 466 981, *European Patent Bulletin*, no. 92/04.

[34] R. Thoma and M. Bierling, "Motion compensating interpolation considering covered and uncovered background," in *Signal Processing: Image Communication 1*. Amsterdam: Elsevier, 1989, pp. 191–212.

Gerard de Haan was born in Leeuwarden, the Netherlands, on April 4, 1956. He received the M.Sc. degree (with honors) and the Ph.D. degree in electrical engineering from Delft University of Technology, Delft, The Netherlands in 1979 and 1992, respectively.

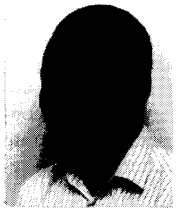
In 1979 he joined the Philips Research Laboratories in Eindhoven, the Netherlands, were he has been active as a Research Scientist in the Visual Communication Systems Group. He led various projects related to HDTV (signal origination, transmission, coding/decoding, and digital recording), which concerned algorithm as well as hardware design. He participated in the EUREKA HD-MAC project and in the EUREKA Working Party on Upconversion. From 1991 to 1992 he was a Visiting Researcher in the Information Theory Group of Delft University, and since 1989 he has taught courses on multidimensional signal processing for the Philips Centre for Technical Training. At present, he has a particular interest in algorithms for motion estimation/compensation, and consumer display scan rate conversion.

Paul W. A. C. Biezen was born in Goirle, the Netherlands, on March 22, 1965. He received the degree in electrical engineering at the Technische Hogeschool Rijswijk, Rijswijk, the Netherlands, in 1990.

In August 1990 he started working at the Philips Research Laboratories in Eindhoven, the Netherlands, as a Research Assistant in the Visual Communications Systems Group. He contributed to the video signal processor chip, worked on HDMAC for a short period, and designed hardware for real-time video signal processing. At present he contributes to the research in the area of motion estimation, motion compensation, and television scan rate conversion.

Henk Huijgen was born in 's Hertogenbosch, the Netherlands, on October 17, 1962. He received the degree in technical computer science at the Hogere Technische School Breda, Breda, the Netherlands, in 1985.

In 1986 he started working at the Philips Research Laboratories in Eindhoven, the Netherlands, as a Research Assistant in the Visual Communications Systems Group. He designed programmable hardware for real-time video processing and realized equipment for an experimental HDTV studio. He contributed to the research on motion estimation, and realized a prototype of the estimator. At present he is working on digital signal processing for television picture enhancement.



Olukayode A. Ojo was born in Ijaka-Oke, Nigeria, on August, 15, 1965. He received the bachelor's degree in electrical engineering at the University of Ibadan, Ibadan, Nigeria, in 1987, and in 1990 he received the master's degree in electronic engineering from the Philips International Institute, Eindhoven, the Netherlands.

He has been active as a Research Scientist in the Visual Communications Systems Group of the Philips Research Laboratories in Eindhoven. His current work focuses generally on digital signal processing for video applications, with particular emphasis on picture quality enhancement.