Project Number :	RACE 2110
Project Title :	HAMLET

CEC Deliverable No. :	R2110/WP2/DS/S/006/b1
Contractual Date of Delivery :	Y2/M9
Actual Date of Delivery :	Y2/M9
Title of Deliverable :	**Specification of a Generic Format Converter**
Workpackage involved :	WP2
Nature :	Specification
Authors :	S. Pigeon (UCL) L. Vandendorpe (UCL) L. Cuvelier (UCL) B. Maison (UCL)

Abstract :

This specification investigates the definition of a generic format converter that is accepting at its I/O both interlaced and progressive formats (25Hz or 50Hz based). These conversions will make use of the general sampling theory which was proposed recently to handle interlaced images. This specification especially defines a 4:2:2 deinterlacer.

Keywords :

Scanning Format Conversions, Deinterlacer, Reinterlacer, Motion Estimation.

Contents

Specification of a Generic Format Converter

1 Introduction

Image format transcoding has become necessary in order to get rid of all the incompatibilities due to the ever increasing number of image processing applications and their associated image formats. The purpose of the RACE 2055/TRANSIT project, recently came to conclusion, was to develop both low-cost and high-cost format conversion algorithms able to work with different scan modes (interlaced and progressive), aspect ratios (16:9 and 4:3) and field/frame rates (50Hz and 60Hz) [1, 2]. Among others, the work of TRANSIT was meant to bridge the gap between the television (interlaced) and the computer world (multimedia progressive applications). It also wanted to demonstrate how transcoding functions could work in real time between different television formats (CCIR 601, EDP, HDI and HDP).

On the other hand, the Scanning Format Extension of HAMLET has to study the influence of scanning formats on the coding efficiency, performed in an MPEG2 framework. The HAMLET Extension intends to investigate the advantages of using a progressive format as an intermediate format for the coding of interlaced images [3, 4].

Progressive scanning is the most direct approach to represent two-dimensional images. However, in the early years of television, an interlaced format was chosen in order to efficiently save bandwidth. Even if this latter format introduces some well known artefacts such as interline twitter, line crawling and field aliasing, these effects were not so annoying at the time of early television, mainly due to the limited spatial definition and the limited brightness range of the cameras and displays at that time. Today, with the progress in technology, these artefacts become more obvious. In such a context, the advent of the future digital television may be seen as a good opportunity to bring a change in the scanning format [5].

When working with digital video, digital image compression has to be performed in order to transmit the data with a reasonable bit rate. Since compression is performed, the picture quality is no longer directly linked to the resolution of the picture but depends on how compression is achieved. However the picture resolution gives an upper limit to the obtainable picture quality. Considering a compression algorithm like MPEG2, the picture quality may vary according to the bit rate at the coder output, the quality of the motion estimation/compensation and the scanning format used to encode the sequence.

Concerning this last point, it has been found that coding progressive sequences improves the subjective quality of the decoded sequence [6] [8], even if the latter is displayed in an interlaced format. Although a progressive format contains twice as many pixels as interlaced, the amount of information is not twice as large but may be seen twice as redundant. A good coding method eliminates this redundancy. So, even if an interlaced scheme is chosen for the future digital television, an intermediate progressive format – generated inside the codec – may be useful in order to provide a better quality of the displayed sequence (interlaced or progressive) thanks to an improved motion estimation/compensation step and decorrelation procedure. A progressive format would also simplify further signal processing [5].

Although the HAMLET definition of a generic format converter and the TRANSIT work are related, specificities of the coding application in HAMLET must be taken into account to lead to some further developments. Unlike TRANSIT, the HAMLET Generic Format Converter is basically meant to be placed in front of the coder, converting interlaced sources into progressive in order to get the best coding efficiency. To achieve this result, a particular attention must be paid to the quality of the construction of the progressive sequence from the interlaced input. At first, the perceptual quality of the reconstruction must be considered. It already has been done within TRANSIT. Different deinterlacing solutions were compared and discussed from that point of view. But more important for our application is the fact that the progressive sequence has to represent the "analog" scene hidden behind the interlaced input as well as possible in order to improve the coding efficiency. In other words, the fields added to the interlaced sequence in order to convert it into a progressive format must be spatially and temporally coherent with the already existing fields. This can only be achieved by the use of motion estimation and compensation. In particular, the calculation of the motion vectors must be finely tuned in order to recover the initial temporal and vertical correlation existing inside the progressive sequence as if it was directly shot from the "analog" scene by the appropriate camera. The analysis inside HAMLET will be based on the *general sampling theory* which was proposed recently [7, 8] to handle interlaced images and proved to be successful.

After presenting the coding scheme on which the generic format converter has to be integrated (section 2), this deliverable will introduce the generalized sampling theorem (section 3) and its use for the motion estimation that is required to perform reliable format conversions (section 4). The next section will be devoted to format conversions involving interlaced and progressive scanning formats (section

5) and will also make use of the generalized sampling theorem. Finally, we will analyze other conversions like those between different spatial (SDTV/HDTV - section 6) and temporal (25Hz/50Hz frame frequency - section 7) formats.

2 Coding Scheme

As within HAMLET's framework, the generic format converter (GFC) will work in close relationship with a coder, a study case as represented in figure 1 may be proposed for this work.

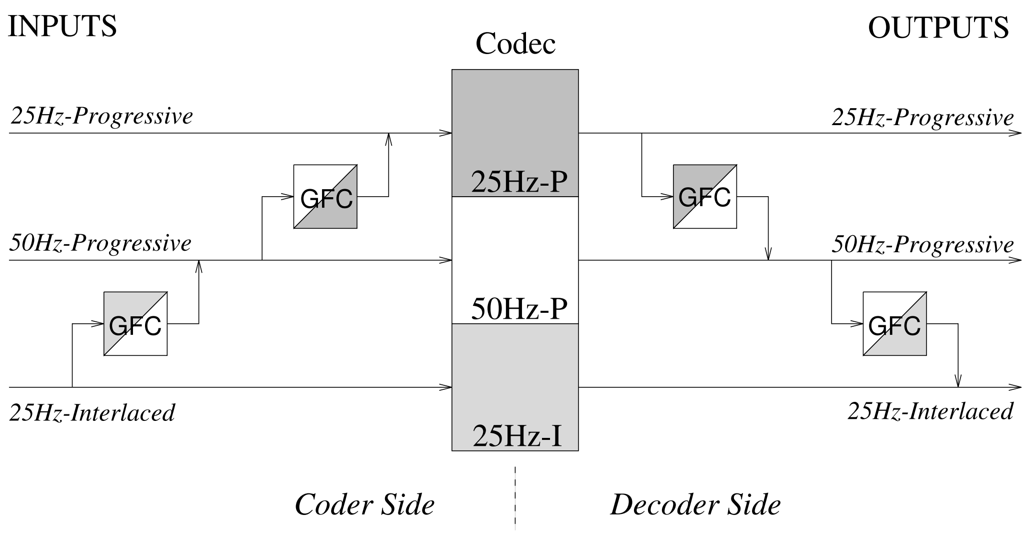

Figure 1: Study case for the generic format converter

Although a generic format converter is, at first sight, meant to work in a more general context – as represented in figure 2 – the representation of the former figure (figure 1) has the advantage to distinguish the coder and decoder sides. Indeed, as the decoder is intended to be replicated in each receiver set, economical considerations will force the market to offer, at this side, low-cost format conversions only. On the opposite, the coder side is able to deal with high-cost algorithms in order to get the best coding efficiency. Such distinction between the coder and the decoder sides will be used further in this work.

3 Generalized Sampling Theorem

In this section, the reader will be reminded of results concerning the generalized sampling theorem applied to the vertical direction of pictures [7].

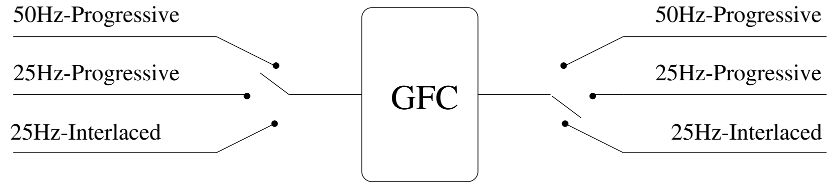

Figure 2: General study case

From the Nyquist theorem [9], it is well-known that any sequence whose spectrum is limited to $1/T$ can theoretically be recovered exactly as long as it is sampled at a rate greater than $2/T$. Actually, this condition is sufficient but not necessary. In 1956, Yen [10] published a generalized form of the Nyquist theorem and the associated interpolation formulas. He showed that any signal whose spectrum was limited to $1/T$ could theoretically be recovered exactly from N sequences at the same rate $2/NT$ but yet with different phases.

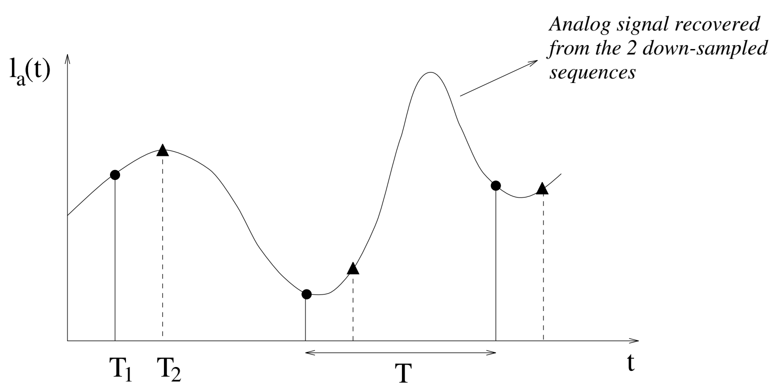

Figure 3: Non-uniform sampling

Let us focus on the case of two sequences. We denote by $l_a(t)$ the analog signal which is sampled. It is assumed to have a bandwidth limited to $1/T$. A sequence of samples y_1 taken at locations $nT + T_1$ (see figure 3) is defined by :

$$y_1(t) = \sum_{n=-\infty}^{\infty} l_a(nT + T_1)\delta(t - nT - T_1) \quad (1)$$

and has a spectrum related to that of the analog signal by :

$$Y_1(\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \exp^{-2\pi jnT_1/T} \; L_a(\omega - 2\pi n/T) \quad (2)$$

A typical situation of spectrum repetition is shown in figure 4. It appears that when the spectrum is bandlimited to $1/T$ and the sampling rate is $1/T$, only two repeated versions of the initial spectrum interfere at the same time for any value of the frequency. It thus means that such a sequence of samples provides, for instance from 0 to $1/T$, a linear combination of the initial spectrum with two complex weights which are known when the sampling phase is known.

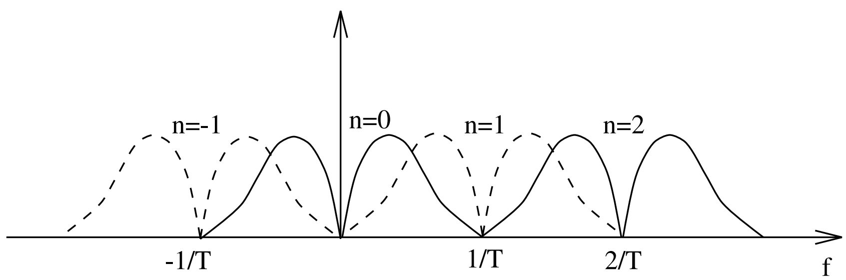

Figure 4: Repetition of the spectra around n/T

A second sequence y_2 with another sampling phase T_2 would provide another linear combination :

$$y_2(t) = \sum_{n=-\infty}^{\infty} l_a(nT + T_2)\delta(t - nT - T_2) \tag{3}$$

$$Y_2(\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \exp^{-2\pi j n T_2/T} \; L_a(\omega - 2\pi n/T) \tag{4}$$

Considering frequences between 0 and 1/T, we can rewrite equations 2 and 4 as follows :

$$Y_1^+(\omega) = \frac{1}{T} \left[L_a^+(\omega) + e^{\frac{-2\pi j T_1}{T}} L_a^-(\omega - \frac{2\pi}{T}) \right] \tag{5}$$

$$Y_2^+(\omega) = \frac{1}{T} \left[L_a^+(\omega) + e^{\frac{-2\pi j T_2}{T}} L_a^-(\omega - \frac{2\pi}{T}) \right] \tag{6}$$

were $L_a^-(\omega)$ and $L_a^+(\omega)$ represent respectively the negative and positive parts of the baseband spectrum (n=0 – see figure 4). This makes a linear system with L_a^- and L_a^+ as unknowns and means that by solving these two equations, the spectrum of the analog input signal can be recovered from 0 to $1/T$. The same result holds from $-1/T$ to 0. Consequently, we are able to completely recover the input signal.

Let us denote by $w^+(t)$ and $w^-(t)$ the impulse responses of ideal bandpass filters, passing only frequencies between 0 and $1/T$, and between $-1/T$ and 0 respectively. We denote by $Y_i^+(\omega)$ and $Y_i^-(\omega)$ the versions of y_i filtered by means of w^+ and w^-. The result provided by the solution of the systems is as follows. The spectra for positive and negative frequencies are respectively computed as :

$$L_a(\omega > 0) = \alpha Y_1^+(\omega) - \beta Y_2^+(\omega) \tag{7}$$
$$L_a(\omega < 0) = \gamma Y_1^-(\omega) - \delta Y_2^-(\omega) \tag{8}$$

where

1. the constants are given by :

$$\alpha = \gamma^* = \frac{T \exp^{-2\pi j T_2/T}}{\exp^{-2\pi j T_2/T} - \exp^{-2\pi j T_1/T}} = \frac{T \exp^{\pi j (T_1 - T_2)/T}}{2j \sin\left[\pi(T_1 - T_2)/T\right]} \tag{9}$$

$$\beta = \delta^* = \frac{T \exp^{-2\pi j T_1/T}}{\exp^{-2\pi j T_2/T} - \exp^{-2\pi j T_1/T}} = \frac{T \exp^{-\pi j (T_1 - T_2)/T}}{2j \sin\left[\pi(T_1 - T_2)/T\right]} \tag{10}$$

   where * denotes complex conjugation.

2. The impulse responses denoted by $w^+(t)$ and $w^-(t)$ are given by :

$$\begin{aligned} w^+(t) &= \frac{\exp^{j\pi t/T}}{T} \frac{\sin(\pi t/T)}{(\pi t/T)} = \frac{\exp^{j\pi t/T}}{T} \operatorname{sinc}\left[\pi t/T\right] \\ w^-(t) &= \frac{\exp^{-j\pi t/T}}{T} \frac{\sin(\pi t/T)}{(\pi t/T)} = \frac{\exp^{-j\pi t/T}}{T} \operatorname{sinc}\left[\pi t/T\right] \end{aligned} \tag{11}$$

As a consequence, $l_a(t)$ is recovered by taking the inverse Fourier transform of $L_a(\omega)$:

$$\begin{aligned} l_a(t) &= \frac{1}{2\pi} \int_{-\pi/T}^{\pi/T} \left[\alpha Y_1^+(\omega) + \gamma Y_1^-(\omega)\right] \exp^{j\omega t} \, d\omega \\ &- \frac{1}{2\pi} \int_{-\pi/T}^{\pi/T} \left[\beta Y_2^+(\omega) + \delta Y_2^-(\omega)\right] \exp^{j\omega t} \, d\omega \end{aligned} \tag{12}$$

This means that l_a is recovered by adding the versions of y_1 and y_2 filtered by means of $[\alpha w^+(t) + \gamma w^-(t)]$ and $[-\beta w^+(t) - \delta w^-(t)]$ respectively. We then find that the original signal can be recovered from the two sequences of samples by means of :

$$
\begin{aligned}
l_a(t) &= \sum_{k_1=-\infty}^{\infty} y_1(k_1T+T_1)\,\frac{\sin\left[\pi(t-k_1T-T_2)/T\right]}{\sin\left[\pi(T_1-T_2)/T\right]} \\
&\times \ \operatorname{sinc}\left[\pi(t-k_1T-T_1)/T\right] \\
&- \sum_{k_2=-\infty}^{\infty} y_2(k_2T+T_2)\,\frac{\sin\left[\pi(t-k_2T-T_1)/T\right]}{\sin\left[\pi(T_1-T_2)/T\right]} \\
&\times \ \operatorname{sinc}\left[\pi(t-k_2T-T_2)/T\right] \qquad (13)
\end{aligned}
$$

This formalism is relevant for the vertical description of interlaced images. As a matter of fact, if the vertical sampling distance is denoted h within a frame, the theoretical range of the vertical resolution is upper limited by $1/2h$. In each field, there is a 2-fold vertical downsampling compared to the vertical frame sampling rate. As a consequence, each field contains aliasing. Nevertheless, considering the assumptions made previously about the motion, two successive fields provide two vertical sets of points which, in general, correspond to different vertical sampling phases. Therefore, by adequately combining the vertical information of two successive fields, the analog information and the associated vertical resolution "hidden behind these two fields" can theoretically be recovered exactly. In particular, the complete vertical resolution can be recovered.

4 Motion Estimation Using the General Sampling Theorem

This section is devoted to the way the results presented in the previous section apply to the problem of motion estimation in interlaced sequences.

In the present work, the motion of any field will be estimated between this field and the two previous ones. We make the assumption that the motion from field 1 to field 3 is uniform. Actually, it is only required that the images can be partitioned into areas with a translational motion. We nevertheless make the calculations for the whole picture, but for the vertical direction only. The analog luminance signal at the time of field 1 is denoted $l_a(y)$ and the luminance sequences corresponding to fields 1 to 3 are denoted by l_1 to l_3. Let us denote by y_0 the vertical motion between two successive fields. With such a motion, the analog luminance signals corresponding to fields 2 and 3 are given by $l_a(y-y_0)$ and $l_a(y-2y_0)$.

The problem of estimating the motion has to be analyzed for two situations depending on the parity of the field for which the motion has to be estimated. We first handle the problem of estimating the motion of field 3 and next that of field 4.

4.1 Estimation for an odd field

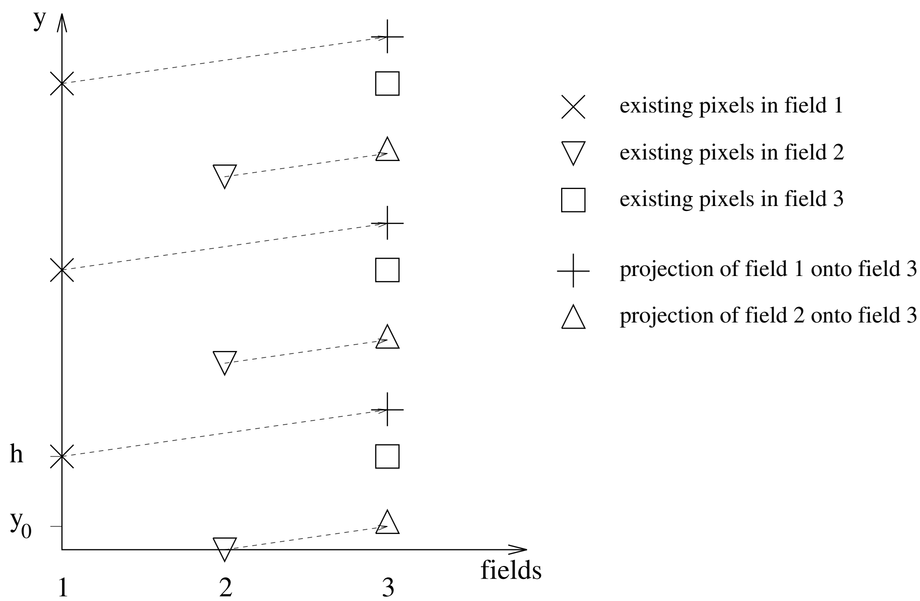

Figure 5: Motion estimation of field 3

The problem is illustrated by figure 5. We assume that fields 1 and 3 provide samples of their associated luminance signals in vertical positions $2k_1h - h$ and $2k_3h - h$ respectively, and that field 2 provides samples located in $2k_2h$, where the k_i are all integers. The parameter h is defined as half the distance between two lines within a field, or the distance between two lines within a frame. If y_0 is assumed to be the correct motion vector, it means that we are able to compute a prediction of the lines of field 3 from the information contained in fields 1 and 2. If the assumption of a uniform motion holds, the lines $2k_1h - h$ of field 1 provide samples located in $2k_1h - h + 2y_0$ in field 3. Similarly, the lines $2k_2h$ of field 2 provide samples located in $2k_2h + y_0$ in field 3. If we assume that the analog luminance is vertically bandlimited to $1/2h$, we know 2 sequences of samples for field 3 and we are able to recover the exact analog signal associated with this field. This is done by applying formula 13 with $T = 2h$, $T_1 = 2y_0 - h$ and $T_2 = y_0$. This analog signal taken at locations $2k_3h - h$ provides an estimate of the exact lines of field 3 from the information contained in fields 1 and 2.

Let us define the integer (q) and fractional (r) parts of the motion vector by :

$$y_0 = 2h(q + r) \tag{14}$$

where $0 \leq r < 1$. In addition, we define new indices :

$$k_{1,3} = k_3 - 2q + j \quad (15)$$
$$k_{2,3} = k_3 - q + j \quad (16)$$

Therefore, the estimates are computed by :

$$\hat{l}_3(2k_3h - h) = \sum_{j=-\infty}^{\infty} w_{13}(j) l_1(2k_{1,3}h - h) + w_{23}(j) l_2(2k_{2,3}h) \quad (17)$$

The weights are defined by the following equations :

$$w_{13}(j) = (-1)^j \operatorname{sinc}[\pi(j + 2r)] \quad (18)$$
$$w_{23}(j) = (-1)^{j+1}\, 2 \sin(\pi r) \operatorname{sinc}[\pi(j + r + 0.5)] \quad (19)$$

The motion estimation procedure will provide the value of y_0 which minimizes a distance between the known luminance samples of field 3 and their associated estimates computed from fields 1 and 2 by means of formula 17. Any classical method can be used. For instance, block matching with exhaustive search or gradient-based approaches can be considered. It should nevertheless be mentioned that different interpolation formulas have to be used for each new vector estimate. However, the computation load can be reduced by limiting the possible values for y_0 and quantizing the motion vector. It is worth mentioning that the weights depend on the fractional part r of the motion vector only, and the samples which have to be used depend on the integer part q only.

Remarks should be made about 2 particular situations :

1. When $r = 0$, the motion is exactly a multiple of $2h$ and no interpolation is required. Lines of field 3 (resp. 4) are predicted by the lines of field 1 (resp. 2).

2. When $r = 0.5$, the motion is exactly the distance between 2 lines in a frame. It corresponds to the singular situation where the sampling phases of the 2 sequences are the same. Therefore, from the point of view of interpolation, it is not possible to recover the analog signal exactly. As regards the motion estimation problem the situation is however very easy, because the lines of fields 3 and 4 are also lines of fields 1 and 2. Again, no interpolation is required for the motion estimation. However, we feel it better to build an estimate from the average value obtained from fields 1 and 2 rather than use the information of one of the two fields only.

4.2 Estimation for an even field

We assume that fields 2 and 4 provide samples of their associated luminance signals in vertical positions $2k_2h$ and $2k_4h$ respectively, and that field 3 provides samples located in $2k_3h - h$. If y_0 is assumed to be the correct motion vector, it means that we are able to compute a prediction of the lines of fields 4 from the information contained in fields 2 and 3. If the assumption of a uniform motion holds, the lines $2k_2h$ of field 2 provide samples located in $2k_2h + 2y_0$ in field 4. Similarly, the lines $2k_3h - h$ of field 3 provide samples located in $2k_3h - h + y_0$ in field 4. If we assume that the analog luminance is vertically bandlimited to $1/2h$, we know two sequences of samples for field 4 and we are able to recover the exact analog signal associated with this field. For field 4, we recover the analog luminance signal by applying the formulas of section 3 with $T = 2h$, $T_1 = 2y_0$ and $T_2 = y_0 - h$. This analog signal taken at locations $2k_4h$ provides an estimate of the exact lines of field 4. Considering the integer and fractional parts of the motion vector, we define new indices :

$$k_{2,4} = k_4 - 2q + j \tag{20}$$
$$k_{3,4} = k_4 - q + j \tag{21}$$

Therefore, the estimates are computed by :

$$\hat{l}_4(2k_4h) = \sum_{j=-\infty}^{\infty} w_{24}(j) l_2(2k_{2,4}h) + w_{3,4}(j) l_3(2k_{3,4}h - h) \tag{22}$$

The weights are defined by the following equations :

$$w_{24}(j) = (-1)^j \operatorname{sinc}[\pi(j + 2r)] \tag{23}$$
$$w_{34}(j) = (-1)^j \, 2 \sin(\pi r) \operatorname{sinc}[\pi(j + r - 0.5)] \tag{24}$$

The motion estimation procedure will provide the value of y_0 which minimizes a distance between the known luminance samples of field 4 and their associated estimates computed from fields 2 and 3 by means of formula 22.

5 Conversions between Interlaced and Progressive Scanning Formats

This section deals with conversions between progressive and interlaced formats. As it refers to the coder side (see figure 1), an interlaced to progressive conversion may use some high-cost techniques. This is fortunate since low-cost linear deinterlacing (e.g. spatial and temporal linear interpolation) does not offer a sufficient picture quality at the input of the coder in order to improve the coding efficiency. Concerning the decoder side, the opposite conversion - progressive to interlaced - has to be performed. As it basically reverts to throw lines away, this conversion can easily be achieved and meets the "low computational cost" requirement needed on this side.

5.1 25Hz-Interlaced to 50Hz-Progressive Conversion

Motion compensated deinterlacing (MCD) is really similar to motion estimation (see section 4). Instead of estimating existing lines, one has to compute missing points (see figure 6). In field 3, the lines located in $2k_3h$, and, in field 4, the lines located in $2k_4h - h$ have to be estimated.

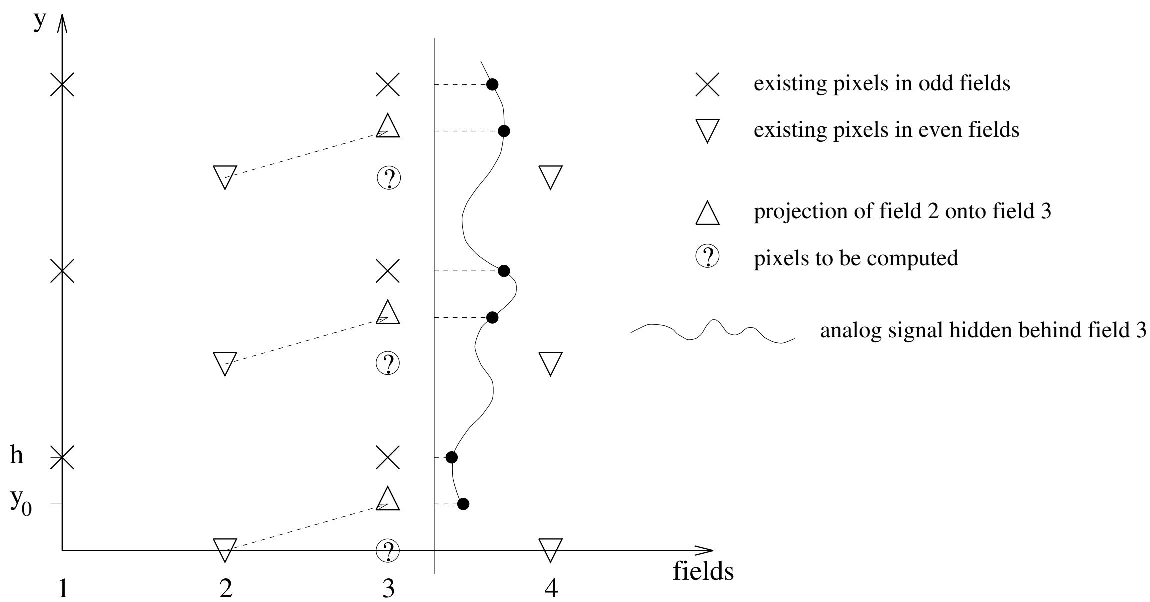

Figure 6: Deinterlacing the third field

Nevertheless, some comments can be made. The motion estimation step always requires that the current field be compared with a prediction made from two different ones. Concerning MCD, there are several possibilities. A first one is to follow the motion estimation step, namely to compute the missing lines of the current field from the two previous fields, as suggested in section 4. Another possibility is to build the estimation of the missing lines from the previous field and the current one as represented in figure 6 : the generalized sampling theorem can be applied to the existing pixels of field 3 and the projections of field 2 in order to reconstruct the analog signal hidden behind field 3 and then resample this signal at the points of the missing pixels. From a practical point of view, this second solution is more attractive. As a matter of fact, the storage of 3 fields would be required in forward/backward MCD. In the first solution, 5 fields would have to be stored. Besides, it has been shown in [8] the that the second solution is also more efficient. Therefore, the 3-field solution is preferred and only the filters corresponding to this solution will be derived. Again two situations associated with the parity of the current field have to be considered.

5.1.1 Filter derivation

We assume that the correct motion vector y_0 has been found and that fields 3 and 4 have to be deinterlaced. The generalized interpolation formulas have to be adapted. The samples of field 2 located in $2k_2h$ provide samples of l_3 located in $2k_2h + y_0$. The lines of field 3 provide samples of l_3 located in $2k_3h - h$. This is illustrated by figure 6. Therefore, the generalized interpolation can be applied with $T_1 = y_0$ and $T_2 = -h$. Defining $k_{2,3} = k_3 - q + j, k_{3,3} = k_3 + j$, the estimates are computed by :

$$\hat{l}_3(2k_3h) = \sum_{j=-\infty}^{\infty} v_{23}(j)l_2(2k_{2,3}h) + v_{33}(j)l_3(2k_{3,3}h - h) \tag{25}$$

The weights are defined by the following equations :

$$v_{23}(j) = (-1)^j \operatorname{sinc}[\pi(j+r)] / \cos(\pi r) \tag{26}$$
$$v_{33}(j) = (-1)^j \operatorname{sinc}[\pi(j-0.5)] \sin(\pi r)/\cos(\pi r) \tag{27}$$

Similarly, field 4 can be deinterlaced. Defining

$$k_{3,4} = k_4 - q + j \tag{28}$$
$$k_{4,4} = k_4 + j \tag{29}$$

the estimates are computed from fields 3 and 4 by :

$$\hat{l}_4(2k_4h - h) = \sum_{j=-\infty}^{\infty} v_{34}(j)l_3(2k_{3,4}h - h) + v_{44}(j)l_4(2k_{4,4}h) \tag{30}$$

The weights are defined by the following equations :

$$v_{34}(j) = (-1)^j \operatorname{sinc}[\pi(j+r)] / \cos(\pi r) \tag{31}$$
$$v_{44}(j) = (-1)^{j+1} \operatorname{sinc}[\pi(j+0.5)] \sin(\pi r)/\cos(\pi r) \tag{32}$$

As mentioned previously, when $r = 0.5$, it is not possible to recover the analog signal exactly. This situation corresponds to a motion which is an odd multiple of the distance between two lines within a frame, namely h. Consequently, this is equivalent to having two sequences produced with the same sampling phase. In this situation, a fallback mode has to be defined. We decided to compute the missing point by averaging between the two surrounding lines in the current field.

All formulas were derived for the case of forward prediction, namely, prediction of the current information from the past. Equivalent formulas can be easily derived for the backward prediction.

5.1.2 Nyquist-shaped interpolators

In this subsection we adapt the idea already proposed in [7]. When the values of the weights defined above are computed, it appears that the coefficient decay is low and that a large number of coefficients are required to obtain good accuracy. This is due to the sinc function associated with the ideal rectangular bandpass filter. A faster decay of the coefficients may be expected from other interpolators. As a matter of fact, motion vectors can be very local. In order not to alter the local aspect of the motion information, it is necessary to avoid using pixels which are too far away from the center of the reference matching area. In other words, long filters should be used in large areas where many blocks have the same motion. Nevertheless, one should be careful along the borders of such areas. Conversely, for small objects whose motion is different from that of the surrounding area, small filters should be preferred.

In this section, alternative interpolators are studied. They are obtained from the modulation of a lowpass prototype filter. We again derive the interpolation formulas associated with the generalized sampling theorem [7]. The 2 filters $w^+(t)$ and $w^-(t)$ are obtained by shifting the same lowpass prototype filter on the frequency axis . We then have :

$$\begin{aligned} W^-(\omega) &= W(\omega+\omega_0) \\ W^+(\omega) &= W(\omega-\omega_0) \end{aligned} \tag{33}$$

where $\omega_0 = \pi/T$. The impulse responses then fulfill the following relationships :

$$\begin{aligned} w^-(t) &= \exp^{-j\omega_0 t} w(t) \\ w^+(t) &= \exp^{j\omega_0 t} w(t) \end{aligned} \tag{34}$$

Let us denote by y_1 and y_2 the 2 sequences of samples taken at the same rate $1/T$ and with phases T_1 and T_2. If we denote by $Y_i^+(\omega)$ and $Y_i^-(\omega)$ the versions of Y_i filtered by means of w^+ and w^- respectively, the spectrum of the reconstructed input signal can be written as :

$$X(\omega) = \alpha Y_1^+(\omega) + \gamma Y_1^-(\omega) - \beta Y_2^+(\omega) - \delta Y_2^-(\omega) \quad (35)$$

We have :

$$\alpha w^+(t) + \gamma w^-(t) = T \frac{\sin[\pi(T_1 - T_2 + t)/T]}{\sin[\pi(T_1 - T_2)/T]} w(t) \quad (36)$$

$$\beta w^+(t) + \delta w^-(t) = T \frac{\sin[\pi(T_2 - T_1 + t)/T]}{\sin[\pi(T_1 - T_2)/T]} w(t) \quad (37)$$

As regards the choice of the prototype filter, one has to take into consideration what follows. The ideal interpolation filter preserves the full bandwidth up to $1/2h$. However, because of the Kell factor [5], the bandwidth is actually limited to $K/2h$, with $K \simeq 0.7$. This means that the repeated spectra will not interfere in the interval $|f| < (1-K)/2h$ and the bandpass filters may have a transition in this interval. In order to recover the exact spectrum repeated around the 0-frequency, it is necessary for a non-ideal filter to have a Nyquist shape around 0. On the other hand, a Nyquist behavior of $w(t)$ guarantees that the zeroes of the impulse response are at the same locations as those of the ideal impulse response, which means that the existing samples are not modified by the interpolation. These considerations support the choice of a Nyquist filter for $w(t)$.

If we take a prototype filter $w(t)$ which is of the Nyquist type, the frequency response is defined by :

$$W(f) = \begin{cases} 1 & |f| < (1-p)/(2T) \\ 1/2 + 1/2 \sin[\pi T(1/(2T) - |f|)/p] & (1-p)/(2T) \leq |f| \leq (1+p)/(2T) \\ 0 & |f| > (1+p)/(2T) \end{cases} \quad (38)$$

with p being the roll-off factor. The corresponding impulse response is :

$$w(t) = \frac{1}{T} \frac{\cos[\pi p t/T]}{1 - 4p^2t^2/T^2} \operatorname{sinc}[(\pi t)/T] \quad (39)$$

We see that compared with the ideal lowpass filter there is an additional windowing factor in the impulse response. On the other hand, in order to have a transition in an interval of width $2(1-K)/T$, we should chose the roll-off factor $p = 2(1-K)$.

We then find that the original signal can be estimated by means of :

$$
\begin{aligned}
l_a(t) &= \sum_{k_1=-\infty}^{\infty} l_1(k_1T+T_1)\,\frac{\sin\left[\pi(t-k_1T-T_2)/T\right]}{\sin\left[\pi(T_1-T_2)/T\right]} \\
&\times \frac{\cos\left[\pi p(t-k_1T-T_1)/T\right]}{1-4p^2(t-k_1T-T_1)^2/T^2}\,\mathrm{sinc}\left[\pi(t-k_1T-T_1)/T\right] \\
&- \sum_{k_2=-\infty}^{\infty} l_2(k_2T+T_2)\,\frac{\sin\left[\pi(t-k_2T-T_1)/T\right]}{\sin\left[\pi(T_1-T_2)/T\right]} \\
&\times \frac{\cos\left[\pi p(t-k_2T-T_2)/T\right]}{1-4p^2(t-k_2T-T_2)^2/T^2}\,\mathrm{sinc}\left[\pi(t-k_2T-T_2)/T\right]
\end{aligned} \tag{40}
$$

All formulas derived in subsection 5.1.1 for non-Nyquist shaped filters can be modified consequently. This modification is performed by replacing the terms of the form $\mathrm{sinc}(\pi x)$ by $\mathrm{sinc}(\pi x)\,\cos\left[\pi p x\right]/(1-4p^2x^2)$.

5.2 25Hz-Interlaced to 25Hz-Progressive Conversion

As represented in figure 1, the 25Hz-interlaced to 25Hz-progressive conversion may be seen as the cascade of two modules : a 25Hz-interlaced to 50Hz-progressive converter (see the previous section) followed by a 50Hz-progressive to 25Hz-progressive converter (see section 7).

5.3 50Hz-Progressive to 25Hz-Interlaced Conversion

50Hz-progressive to 25Hz-interlaced conversion can easily be achieved by filtering the progressive source through a vertical low-pass filter and discarding the useless lines in order to generate the interlaced fields. The low-pass filtering is used to reduce the vertical definition according to the Kell factor. This factor is meant to reduce the line flicker that appears on bright and sharp horizontal edges when displayed in an interlaced format [5]. The impulse response of such filter (known as "HHI filter") is listed below and its frequency response is represented in figure 7.

-4	8	25	-123	230	728	230	-123	25	8	-4	/1000

Table 1 : Pre-interlacing filter

If the progressive sequence to be converted comes from a deinterlaced source, a low-pass filtering is not mandatory since the Kell factor is already present in the

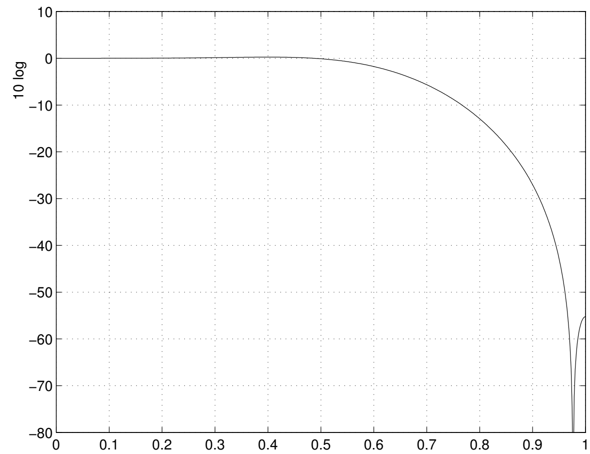

Figure 7: Pre-interlacing HHI filter

progressive sequence. However, if this progressive sequence has been encoded then decoded, some high-frequencies could still be present due to the coding process. Then, a low-pass filtering would be useful again.

5.4 25Hz-Progressive to 25Hz-Interlaced Conversion

Again, as represented in figure 1, a 25Hz-progressive to 25Hz-progressive conversion may be seen as the cascade of two modules : a 25Hz-progressive to 50Hz-progressive converter (see section 7) followed by a 50Hz-progressive to 25Hz-interlaced converter (see the previous section).

6 Conversions Between Different Spatial Formats

In this section, we will focus on conversions between different spatial resolutions, as needed for conversions between High-Definition and Standard-Definition television formats. Again, a distinction has to be made between the two different scanning formats.

6.1 Conversions Between Progressive Formats

When working with a progressive format, spatial conversions are easily processed by using a chain of down-/up-sampling and low-pass filtering.

6.1.1 Down-conversion

As the scaling factor may be fractional, a digital down-conversion requires the use of both up- and down-samplers, as illustrated in figure 8 in the particular case of a 3/2 down-conversion.

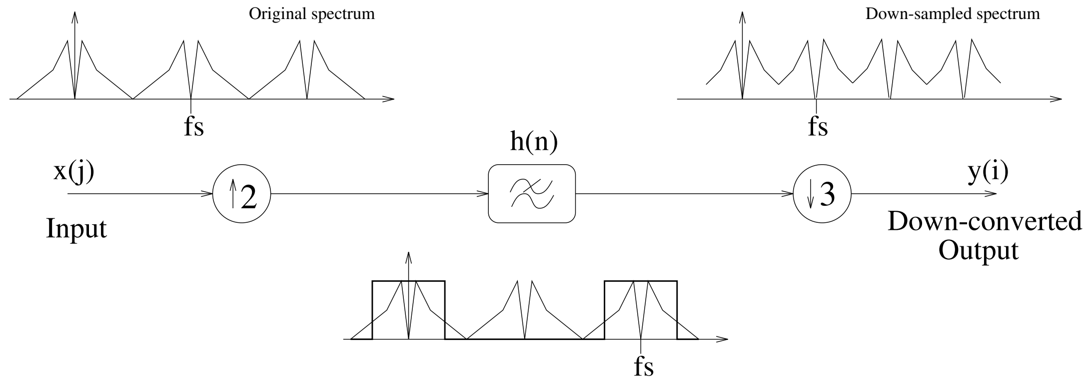

Figure 8: Down-sampling by a factor of 3/2

The low-pass filter is used to avoid the aliasing effect and suppresses the frequencies that could fold back to the low spectrum during the down-sampling operation.

As these operations are not time invariant – due to the up-/down-sampling operators – they cannot be written in terms of a simple filtering expression but as a filter whose impulse response periodically depends on the output samples to be calculated.

Let us define the down-sampling factor as N/M, where M and N are integers and $N > M$. Let us also denote $h(n)$ the approximation of the ideal low-pass filter that cuts off all frequencies below $f_s/2N$, were f_s represents the sampling frequency. If $x(n)$ are the input samples, the down-converted sequence y(i) is given by :

$$y(i) = \sum_j h(iN - jM)x(j) \tag{41}$$

Assuming a low-pass filter of the $2K + 1$ th order (K integer), $h(k) = 0$ for $-K < k < K$ and the index number j has only to cover integers situated between $\frac{Ni-K}{M} \leq j \leq \frac{Ni+K}{M}$.

Further considerations can be found in [15].

6.1.2 Up-conversion

Once again, as the scaling factor may be fractional, a digital up-conversion requires the use of both up- and down-samplers as illustrated in figure 9.

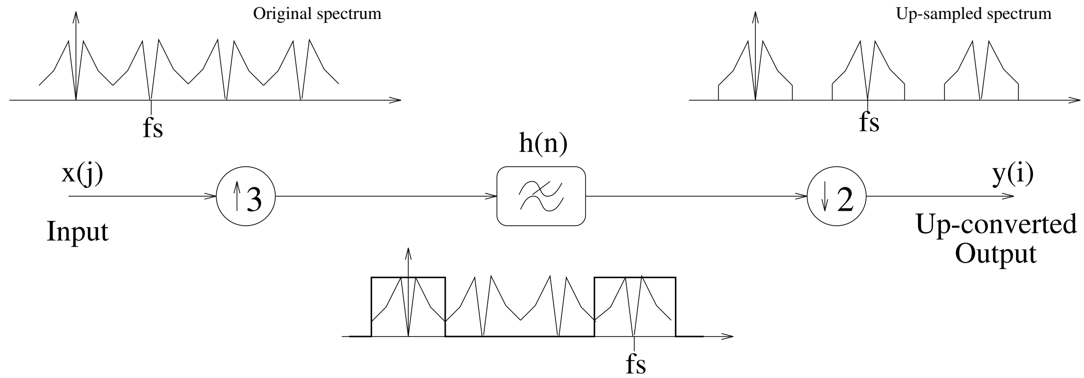

Figure 9: Up-sampling by a factor of 3/2

Again, the expression of the up-converted sequence may be expressed with a filter whose impulse response depends on the output samples to be calculated and equation 41 remains valid with an up-sampling factor of M/N $(M > N)$.

6.2 Conversions Between Interlaced Formats

A first low-cost solution would be to merge the interlaced fields into progressive frames and apply the same processing as in the previous subsection. Unfortunately, this technique produces ghost-effects and motion judder in moving parts of the picture. Another way would be to perform intra-field conversions. However, it does not take into account the aliasing present in each field.

In order to get rid of this intra-field aliasing, an improved method is based on the general sampling theory : the interlaced image is first deinterlaced (as explained in section 3) and its progressive version can be further up- or down-converted. As last step, reinterlacing has to be performed. Mixing all these operations together leads to an overall digital and motion-based method for interlaced image. Further equations were developed in [13] and [16].

7 Temporal Conversions

Temporal down- and up-conversions revert to the same problem as for spatial down- and up-sampling, but for the temporal dimension. Again, the same basic outline as represented in figures 8 and 9 may be used. This problem is however more complex since it involves many different aspects as explained hereafter.

As the video signal shots the scene at fixed moments – and thus performs a sampling process in the time dimension – temporal aliasing could appear in the original video sequence if no temporal low-pass pre-filtering has been performed during the capture of the analog scene. As a temporal filtering is hard to implement (it reverts to suppress from the scene all objects that are moving too fast compared to the frame rate), only a poor filtering is performed inside the camera thanks to the remanence effect in the pickup tube. Fortunately, the human eye seems not to be bothered by this temporal aliasing issued from the lack of pre-filtering.

On the other side, the sampling theory also defines a low-pass post-filtering in order to reconstruct the original signal by means of its samples. In the case of displaying television sequences, this post-filtering only counts upon the properties of the human vision [5], its remanence in particular and the remanence of the display. As this filtering is not perfectly suited for such application, some defects may be still visible like the *large area flickering* or some *judder* in quick moving parts of the scene. In order to reduce these defects, further attention must be paid during the capture of the sequence.

In order to avoid large area flickering, the display refresh frequency must generally be larger than 50Hz, depending on the brightness range of the screen. Concerning the judder effect that might be visible for quick motion present in sequences taken at low frame rate (below the 50Hz), it can be suppressed with a camera exposure time set to the elapsed time between two successive images. With such exposure time, quick motions are turned into a continuous blur and makes the transitions between successive images smoother for the eye.

Moreover, the judder effect occurs when the difference between two adjoining frames is too high for the eye to reconstruct the missing movement existing between these images. This effect depends on several parameters :

- *Frame rate.* Higher the frame rate, lower the differences between successive images.
- *Camera integration time.* Longer the integration time, smoother the transitions between successive images.
- *Viewing distance.* Closer the viewing distance, larger the visual angle and more visible the differences between successive images.

- *Screen/picture resolution.* As the viewing distance is related to the screen resolution (higher the resolution, closer the viewing distance), it also influences the judder visibility.
- *Screen brightness.* Higher the brightness, longer the retinal persistence. The eye then becomes more sensitive to the judder effect.
- *Picture contents.* Faster the movements of image objects, higher the difference between successive images.

Of course, these parameters are interacting: for example, we can reduce the exposure time (in order to improve the sharpness of the sequence) and still limit the judder effect by simultaneously increasing the shooting (and display) frequency.

7.1 Low-Cost Conversions (Decoder Side)

Taking into account the above general considerations, low-cost temporal conversions (decoder side) can be performed while offering a satisfactory sequence quality.

7.1.1 Down-Conversion

By taking care of performing a low-pass temporal pre-filtering (e.g. bilinear filtering), it is possible to down-convert a sequence by simply throwing images away. However, in order to achieve good results, the following conditions are required :

- the exposure time of the original sequence must be large enough in order to avoid multiple contours as result of the low-pass (digital) pre-filtering. Ideally, this exposure time should cover the elapsed time between two successive images.
- the low-pass pre-filtering must have enough taps in order to render a motion blur that covers the elapsed time between two successive images of the down-converted sequence. However, a longer filter implies an larger frame memory in order to perform the temporal filtering. It may run counter to the low-cost requirement considered here. This can be solved by using an appropriate exposure time at the camera.

With respect to these, the judder appearing in quick moving parts of the scene can be avoided but to the detriment of some increased blur in these same areas.

7.1.2 Up-conversion

If the exposure time of the input sequence is high enough, up-conversion to an higher frame rate can simply be performed by repeating existing frames. This solution is used for displaying films (24Hz sped up to 25Hz) on television (50Hz).

7.2 High-Cost Conversions (Coder Side)

When the shooting conditions are such as it is not possible to use some low-cost techniques or when there is a need to offer an improved quality and avoid additional blur in quick motion areas (broadcasting side), motion compensated conversions have to be used.

For conversions implying *progressive formats*, a usual interpolation techniques may be used. For example, the pixels of the missing frames can be obtained by a temporal interpolation along the motion direction. The motion estimation is first carried on a block-by-block basis and uses some iterative procedure in order to recursively compute the different displacement estimates with an increasing accuracy, starting from an initial guess. This initial estimate may be obtained from the last estimates of the surrounding blocks [14]. Let D_n (a 2x1 matrix) denote the displacement vector of the n^{th} block, the algorithm works as follows :

$$D_{n,i+1} = D_{n,i} - \begin{pmatrix} \sum(\nabla_H.\nabla_H) & \sum(\nabla_H.\nabla_V) \\ \sum(\nabla_H.\nabla_V) & \sum(\nabla_V.\nabla_V) \end{pmatrix}^{-1} . \begin{pmatrix} \sum(DFD.\nabla_H) \\ \sum(DFD.\nabla_V) \end{pmatrix} \qquad (42)$$

where ∇_H and ∇_V represent respectively the horizontal and vertical intensity gradients around the (x,y) pixel. DFD stands for *displaced frame difference* and is defined as follows :

$$DFD = DFD(x, y) = I(x, y, t) - I(x - D_x, y - D_y, t - T) \qquad (43)$$

where T represent the elapsed time between two frames. The summations referred in 42 extends over the pixels belonging to the considered block.

As the motion estimation is carried out on a block basis, the vector of each block represents an averaged value of the displacement within this block. In case of having two different moving objects within the same block (object borders), this value may be wrong and the interpolation incorrect. In case of motion compensation, this is not awkward since the compensation error is also transmitted along the estimated motion vectors. But this is not the case in our frame interpolation application. Therefore, a complementary pixel based motion estimation has to performed.

As example, for a pixel to be interpolated, we can consider the motion vector of each surrounding block. Then, the vector which gives the minimum absolute difference between the displaced pixels belonging to the previous field and the next field is selected for the interpolation. The minimum absolute difference is computed on smaller blocs, typically 3x3 pixels.

Since the interpolated fields have to be inserted between those of the original sequence (e.g. 50Hz to 60Hz frame rate conversion, see [17]), an assignment problem may occur since the blocks, once translated along the attached movement, do not necessary create a complete partition in the interpolated field, as shown in figure 10.

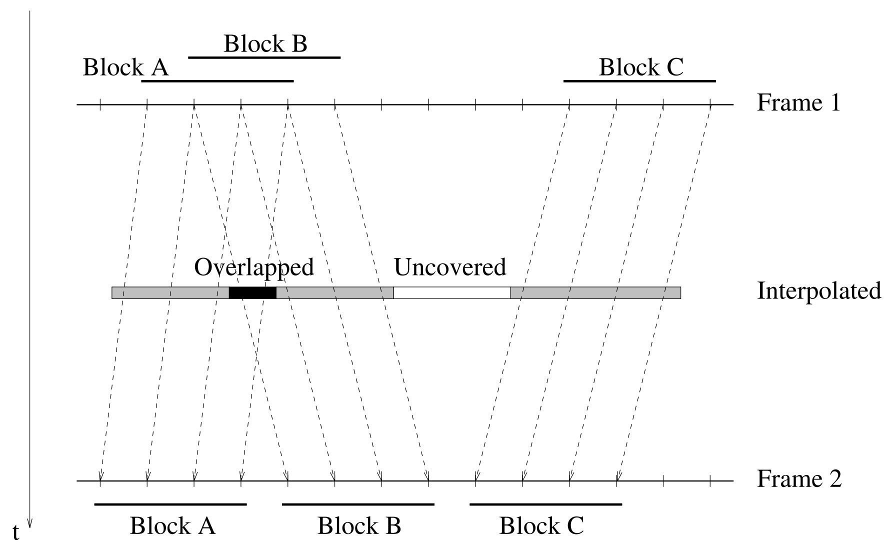

Figure 10: Partition problem inside the interpolated field

This problem can be solved by extending the matching area outside the elementary segmentation blocks in order to ensure at least the full covering of the inserted field. Uncovered areas will then disappear by giving rise to growing overlapped areas. For these areas, a choice has to be made between several block candidates. The best matching one can be selected by the use an absolute minimum absolute difference criteria.

Furthermore, a simple interpolation along the direction of the motion does not treat uncovered regions correctly. When the DFD of the best matching blocks rates high, then a block that is present in one frame but covered in the previous one has to be envisaged. In such a case, it is better to leave the interpolation technique and not consider information from both frames surrounding the frame to be interpolated anymore, but consider information from the future only.

For the *interlaced format*, a deinterlacer has to be used prior to the temporal interpolator and the up-converted sequence has then to be reinterlaced. Again, using an algorithm that makes use of the general sampling theory improves the deinterlacing step and the up-conversion by the same way.

8 Conclusion

This deliverable has specified a deinterlacer that makes use of the general sampling theory. This theory handles the aliasing present in each field of an interlaced sequence and allows a perfect reconstruction of the progressive frame hidden behind those fields. Such deinterlacer has been used within the HAMLET/WP2 in order to study the coding efficiency of deinterlaced (progressive) sequences compared to their interlaced sources.

Other conversions, spatial as well as temporal, were also considered here and further references about the generalized sampling theory have been made in order to improve these conversions when interlaced formats were involved.

References

[1] J. Pandel, "Introduction and Overview about the TRANSIT Project", *European Workshop and Exhibition on Image Format Conversion and Transcoding*, Berlin, Germany, March 1995.

[2] Final Report on Format Conversions Algorithms, *CEC RACE/TRANSIT Deliverable n° 2055/LEP/SPG/B/031/a1*, February 1994.

[3] L. Vandendorpe, L. Cuvelier, B. Maison and P. Delogne, "Coding of Deinterlaced Image Sequences", *IEEE International Conference on Image Processing (ICIP-94)*, Austin, Texas, November 1994, pp. 943-946

[4] B. Chupeau and P. Salmon, "Motion Compensated Deinterlacing for Studio Applications", *International Workshop On HDTV'93*, Ottawa, Canada, October 1993.

[5] S. Pigeon and P. Guillotel, "Advantages and Drawbacks of Interlaced and Progressive Scanning Formats", *CEC RACE/HAMLET Deliverable n° R2110/WP2/DS/R/004/b1*, June 1995.

[6] S. Pigeon, "Progressive versus Interlaced Coding", *CEC RACE/HAMLET Document n° R2110/WP2/UCL/I/5*, July 1995.

[7] P. Delogne, L. Cuvelier, B. Maison, B. Van Caillie and L. Vandendorpe, "Improved Interpolation, Motion Estimation and Compensation for Interlaced Pictures", *IEEE Transactions on Image Processing*, Vol. 3, No. 5, September 1994, pp. 482-491.

[8] L. Vandendorpe, L. Cuvelier, B. Maison, P. Queluz and P. Delogne, "Motion Compensated Conversion from Interlaced to Progressive Formats", *Signal Processing : Image Communication*, Vol. 6, No. 3, June 1994, pp. 193-211.

[9] A. J. Jerri, "The Shannon sampling theorem - its various extensions and applications : a tutorial review", *Proceedings of the IEEE*, Vol. 65, No. 11, November 1977, pp. 1565-1596.

[10] J. L. Yen, "On the nonuniform sampling of bandlimited signals", *IRE Trans. Circuit Theory*, Vol. CT-3, December 1956, pp. 251-257.

[11] F. Kretz et J. Sabatier, "Echantillonnage des images de télévision : analyse dans le domaine spatio-temporel et dans le domaine de Fourier", *Annales des Télécommunications*, Vol. 36, Nos. 3-4, Avril 1981, pp. 231-273.

[12] P. Delogne, O. Poncin, B. Van Caillie and L. Vandendorpe, "Compatible coding of digital interlaced HDTV", *IEEE Journal on Selected Areas in Communications*, Vol. 11, No. 1, January 1993, pp. 146-152.

[13] L. Vandendorpe, L. Cuvelier, B. Maison and P. Delogne, "Improved Motion-compensated Conversion between Interlaced TV and HDTV formats", *International Workshop On HDTV'93*, Ottawa, Canada, October 1993.

[14] M. Lamnhabhi and J.J. Lhuillier, "Motion Compensated Time Rate Conversion of Video Sequences", Signal Processing of HDTV, ElsevierScience Publishers B.V., 1988.

[15] R.E. Crochiere and L.R. Rabiner, *Multirate Digital Signal Processing.* Prentice-Hall, 1983.

[16] L. Vandendorpe, P. Delogne, B. Maison and L. Cuvelier, "MMSE Design of Interpolation and Down-sampling Filters in the context of Periodic Nonuniform Sampling", to appear in *IEEE Transactions on Signal Processing*, August 1994, 31 p.

[17] L. Vandendorpe, P. Delogne and D. Boucquey, "Generalized Interpolators For Advanced Movement-Compensated 50Hz-60Hz Conversion Of Interlaced Sequences", to appear in : *Proceedings of ICIP'95.*