Comparisons between Sub-pixel Estimation techniques in H.264/AVC and VC-1 video coding standards

WISSAL HASSEN [1] , MBAINAIBEYE JÉRÔME [2] AND HAMID AMIRI [1]

Signal, Image and Information Technologies laboratory
[1] The Electrical Engineering Department of National Engineering School of Tunis, TUNISIA
[2] Polytechnic High Institute of Mongo, CHAD
WISSAL.HASSEN@enit.rnu.tn, jerome.mbai@enit.rnu.tn , hamid.amiri@enit.rnu.tn

Abstract: The VC-1 is an advanced video standard developed by Microsoft, while the H.264/AVC is developed by the ITU-T Video Coding Experts Group together with the ISO/IEC JTC1 Moving Picture Experts Group. Both standards use advanced techniques of compression to reduce redundancies in a video sequence. Although the Motion Estimation technique plays a fundamental role to reduce the temporal redundancy, it is still not enough in the case of small Motion Estimation. To ensure a good temporal prediction, some video standards propose a technique of Fractional or Sub-pixel Motion Estimation. This paper shows the efficiency of this technique, it presents the used algorithms in these standards and discusses the effectiveness of each one. Comparisons are made by using two video quality assessment metrics as well as a visual evaluation. The computation time, which is fundamental for real time transmission, in experimental results is also an important evaluation criterion in this work.

Key-Words: Block-matching, H.264/AVC standard, image interpolation, image quality assessment, Fractional Motion Estimation, VC-1 standard, video coding.

1 Introduction

The Video compression consists to reduce or even remove the redundant video data so as to reduce the size of storage and the bitrate transmission of digital video file. Indeed, a video sequence presents two kinds of redundancies; a spatial redundancy due to the repetition of blocks of pixels in each image which is processed with an intra-frame encoding and a temporal redundancy related to the repetitions of the same data in many successive frames which is treated with an inter-frame encoding. The Motion estimation [1] is the key of temporal compression; it treats the sequence as a group of successive images, the first is considered as the reference image and the rest of the group constitute the predictive images. The motion estimation can be summarized in two steps: the first step is the motion vector estimation which represents the displacements of predictive image blocks relative to the reference image blocks. Therefore, compressed video is composed by the motion vector and frames differences, between predictive images and reference image, both encoded by an entropy coding.

In this work, we treat the sub-pixel motion estimation based on image interpolation technique which is proposed by several video standards [2, 3]. Indeed, the motion vector estimation is established through a Block Matching algorithm which decomposes the treated frames into rectangular sections or 'blocks' called Macro-Blocks (MBs). Then, for each Mb in the predictive frame, the Block Matching algorithm searchs its similar in the reference frame. The similar Mb or the matching Mb is that which minimizes the value of the mean square error (MSE). The main idea of the fractional motion estimation is to interpolate the reference frame to increase the estimation accuracy of the motion vector.

Our contribution consists to the study of fractional motion estimation techniques used in H.264/AVC [1] and VC-1 video standards [5]. In fact, both standards are based on the fractional motion estimation with interpolation. Although each standard uses its own interpolation technique; the H.264/AVC uses bilinear and 6-tap filter interpolation while VC-1 standard operates the bicubic interpolation. The evaluation of these different methods is done according to the accuracy and the parameters of motion vector estimation (the Mb and search area p) using objective and subjective image quality assessment metrics. The Full search algorithm which estimates the similarity between Mbs at each possible location in the search area and gives the highest PSNR among any other block matching algorithm [1] is used in our simulations. Experimental tests are

conducted on two video sequences, Football and Foreman, with significant motion dynamics and different resolutions (CIF and QCIF).
This paper is decomposed into three sections: in the first section, we present the principle of motion estimation by interpolation. The second part deals with the two interpolation algorithms used by H.264/AVC and VC-1 standards. Finally, we present the evaluation of these algorithms and the obtained results in the last and third section.

2 The principal of motion estimation

Motion estimation and compensation is a technique for reducing the temporal redundancy in a video sequence. A video sequence is divided into groups of pictures (GOPs). Each GOP is composed of 12 pictures: I or P pictures. The first pictures of a GOP must be an I-picture and is coded only with an intra-frame encoder. Predictive-coded pictures (P-pictures) are coded using reference frames which can be a previous I-frame or P-frame. B-picture or bidirectional picture is another kind of image that forms a GOP, it is predicted from two frames into two inverse temporal directions; these two frames are a just previous frame and the just next frame.
In the motion estimation, current frame (the frame to predict) and reference frames are decomposed into rectangular Mbs. Then, the displacement of each Mb of the current frame is estimated based on the reference frame as shown in Fig.1. After that, this motion vector is used in the motion compensation stage in order to provide the predicted frame from the reference frame [1]. The error of prediction, named the difference frame or residual, is encoded rather than the current frame itself. Also the estimated motion information has to be transmitted.

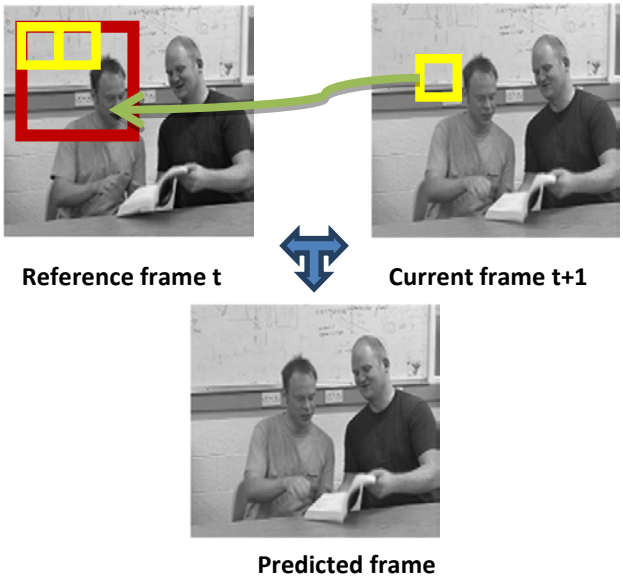

Fig. 1 Prediction of an image by Block Matching

The decoder estimates the current frame from data already decoded: the reference frame, the motion vector and the residual. Each Mb in the current frame passes through a stage of search to determine the 'best' matching Mb or the similar Mb in the reference frame. This search can be carried out by making a comparison between the Mb in the current frame and the possible Mbs in a fixed search area (P) in the reference frame. The search is performed by one of block matching algorithms [1]. A popular matching criteria is the Mean Squared Error (MSE) calculated between the current Mb and the reference Mb and provides a measure of the remaining energy in the difference block. This process of finding the best match is known as motion estimation. The MSE for N x M sample block can be calculated as follows:

$$\mathrm{MSE} = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} \left(C_{ij} - R_{ij}\right)^2 \quad (1)$$

Where C_{ij}, is a sample of the current Mb, R_{ij} is a sample of the reference Mb. The offset between the current Mb and the position of the candidate Mb called motion vector is also transmitted after having encoded by an entropy coding [6].

3 Subpixel motion vector estimation

Although the motion estimation is based on the motion vector search in a search window of the reference image, the fractional motion estimation increases the size of this window for more precision at the searching stage. Therefore, the search in a subsampled image requires more computation than that in the original image. In spite of this complexity, subpixel motion estimation can significantly outperform integer motion estimation

which is due to the fact that object will not necessarily move by an integral number of pixels between successive video frames. Searching subpixel locations as well as integer locations is likely to find a good match in a larger number of cases. Given the importance of this topic, several studies have focused on this area, this part deals specifically with the predicted image quality improvement by increasing the prediction accuracy of the motion vector estimation.

Nowadays, there are two main standards, H.264/AVC and VC-1, that introduce an advanced technique of motion estimation based on the subpixel motion estimation [7] and on the fact that the best estimation cannot be found using integer pixels grids rather than by fractional pixel accuracy.

In Fig.2 we present the subpixel motion estimation effect on the residual image which is the difference between the current image to be encoded and the reference image previously coded. After the stage of motion estimation, the encoder determines a motion vector which represents the displacement of blocks relative to a reference image and coded by an entropic encoder. Then in the stage of motion compensation, the encoder determines the predicted image called also compensated image from the motion vector and current image. The difference between the predicted image and the current image gives the prediction error or residual Image. This residual will be encoded by an intra-frame encoder. By comparing the three images in Fig 2: (a) residual image using integer pixel accuracy, (b) residual image using half-pixel accuracy and (c) residual image using quarter-pixel accuracy, we see that the energy of the residual image decreases when the motion vector is estimated with more precision. Indeed, increasing the precision of motion vector estimation allows reconstructing a predicted image which is very similar to the original image. Therefore, the residual image shows a minimum energy. In this part we deal with two sub-pixel motion estimation algorithms used in the H.264/AVC and the VC-1 standards.

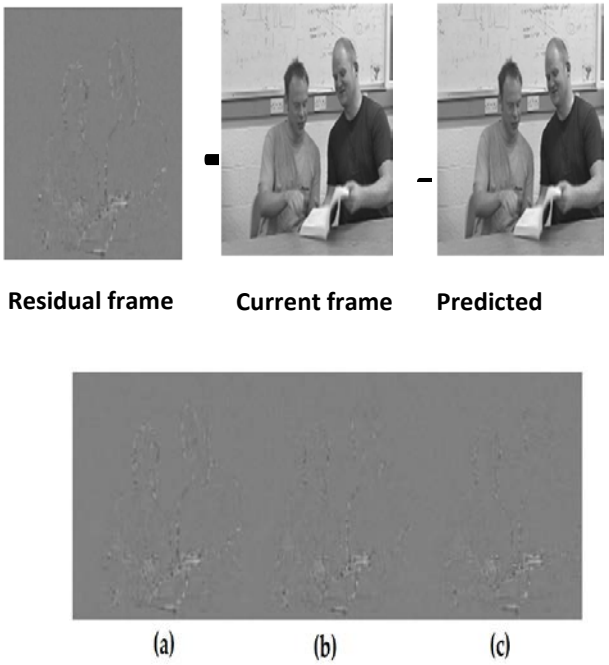

Fig.2 Motion estimation with different accuracy: the impact on the residual image (a) integer pixel accuracy, (b) half-pixel accuracy and (c) quarter-pixel accuracy

The first version of H.264/AVC standard is developed by the Joint Video Team (JVT) in May 2003 and its current version updated in April 2012. However VC-1 standard, which was initially developed as a proprietary video format by Microsoft, was released as a SMPTE video codec standard in April 2006. The standard VC-1 is today a supported standard found in Blu-ray Discs, Windows Media and Microsoft Silverlight framework. In this section we present the used sub-pixel estimation techniques in each standard.

To perform the fractional motion estimation, the advanced video standards proceed by a step of reference image interpolation followed by the search of matching block using the Block Matching algorithm. The Mean Square Error (MSE) calculated between the current Mb and the reference Mb provides a measure of the remaining energy in the difference block. When this energy is minimal, the candidate block is the matching block and the displacement between block gives the motion vector.

```
function [MSE, MV]= Integer-pixel-accuracy-ME(current
Mb,x1, x2 ,r)
% r = search area
% x1= current frame
% x2= reference frame
% Blocksize =bs

current Mb = x1(y:y+bs-1, x:x+bs-1)

       for Dy = -r:1:r % vertical search range
          for Dx = -r:1:r % horizontal search range
             Reference Mb= x2(y+Dy, x+Dx)
             diffblk = current Mb - Reference Mb
             mse = sum (sum (diffblk ^ 2)) / (bs^ 2)
          end
       end

 MSE= mse
 MV =[Dy Dx]
```

Fig. 3 Matlab function1 : Motion Estimation nested loops with Integer-pixel accuracy using Full Search Algorithm

```
function [MSE, MV1]= Sub-pixel-accuracy-ME(current
Mb,x1,x2,r,MV,s)
% r = search area
% x1= current frame
% x2= reference frame
% x2int= interpolated x2 at level s
% Blocksize =bs
% MV =[My Mx]
% s= interpolation level
current Mb = x1(y:y+bs-1, x:x+bs-1)
dy = s*(y+ My)-1;
dx = s*(x+ Mx)-1;

for Dy = -r*s:1:r*s  % vertical search range
   for Dx = -r*s:1:r*s  % horizontal search range
         cy= dy+Dy:s:dy+Dy+s*bs-1;
         cx= dx+Dx:s:dx+Dx+s*bs-1;
    Reference Mb = x2int(cy, cx)
    diffblk = current Mb - Reference Mb
    mse = sum (sum (diffblk ^ 2)) / (bs^ 2)

 end
end

MSE= mse
MV1 =[Dy Dx]
```

Fig. 4 Matlab function 2: Motion Estimation nested loops with Sub-pixel accuracy using Full Search Algorithm, *s* is the interpolation level

We present in fig.3 and fig.4 a Matlab simulation of two functions that show respectively the integer-pixel accuracy motion estimation and the sub-pixel accuracy motion estimation with interpolation. The level of interpolation (2, 4, 8 or more) is noted by "s" in the Matlab code. The used Block Matching Algorithm is using the Full Search Algorithm. The Interpolation technique and the level of interpolation are two parameters that characterize each standard. We explain the interpolation techniques of each standard in the following section.

4 Subpixel motion estimation in H.264/AVC standard

Motion estimation starts from two images: the reference image and the current image, the algorithm scans the current image Mb per Mb. For each current Mb, the algorithm searches its similar in the reference image. The Mb that minimizes the MSE, located at (y + Dy, x + Dx), is selected as the best matching Mb. The motion vector of the current Mb is MV = (Dy Dx). The subpixel motion estimation begins at this stage. It starts with the location found in the previous step (y + Dy, x + Dx). The algorithm restarts the search in the interpolated reference image. Depending on the level of interpolation noted "*s*", the second step is repeated to determine the motion vector as shown in the section of code that we simulated by Matlab in Fig.5.

```
x1 = reference frame
x2 = current frame
imx = image width
imy = image height
r = search area
bs = Macro block size

for y=1:bs:imy
  for x=1:bs:imx

    current Mb = x2(y:y+bs-1, x:x+bs-1);
    best_mse = 256^2; % best MSE is initialized at 256^2
    best_MV = [0 0]; % best MV is initialized at [0 0]

% Integer-pixel motion vector estimation
    [MSE, MV]= Integer-pixel-accuracy-ME(current Mb, x1 ,r)
    {Comparison of the MSE vs best_mse
     Decision}

% Refine the best motion vector to 1/2 pixel accuracy
   [MSE1, MV1]= Sub-pixel-accuracy-ME(current Mb,x1,r,MV,2)
   {Comparison of the MSE vs best_mse
    Decision}

% Refine the best motion vector to 1/4 pixel accuracy
    [MSE2, MV2]= Sub-pixel-accuracy-ME(current Mb,x1,r,MV,4)
    {Comparison of the MSE vs best_mse
     Decision}

% Refine the best motion vector to 1/8 pixel accuracy
    [MSE3, MV3]= Sub-pixel-accuracy-ME(current Mb,x1,r,MV,8)
    {Comparison of the MSE vs best_mse
     Decision}
 end
end
```

Fig. 5 The main Algorithm of Sub-pixel Motion Estimation

Before moving from one level of interpolation to a new level, the H.264/AVC standard introduces a new step called discussion; indeed it compares the new value of MSE (N_MSE) with the last found value of MSE (L_MSE). If the N_MSE value is greater than the L_MSE value, the algorithm stops at this stage, otherwise the algorithm proceeds to an advanced level of interpolation.

H.264/AVC standard uses the 6-tap filter for the half-pixel interpolation and a bilinear interpolation to achieve quarter-pixel precision or more. This allows encoder to calculate frames at the accuracy of half-pixel before starting the encoding process. Subsequently, we briefly introduce the principle of each interpolation method.

4.1 The 6-tap filter interpolation

Obviously, the search on a sub-sampled image requires more computation than the integer searches described earlier. In spite of the increased complexity, sub-pixel motion estimation can significantly outperform integer motion estimation. Indeed, a moving object does not necessarily move to an integer number of pixels between successive video frames. Searching sub-pixel locations as well as integer locations is likely to find a good match in a larger number of cases. Pixels at half-pixel position are obtained by applying the 6-tap filter described in this section.

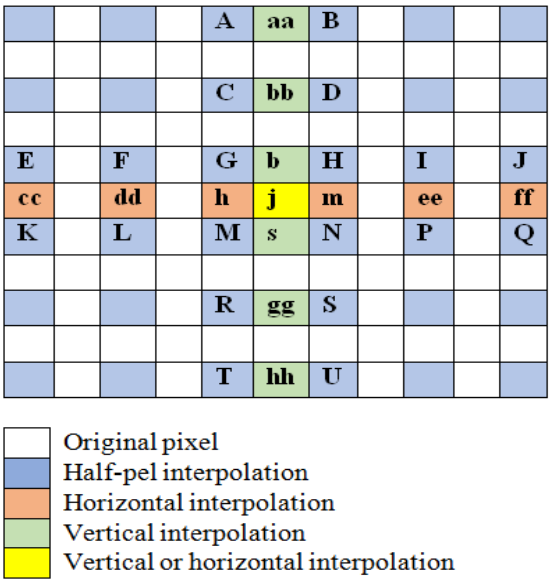

Fig. 6 The 6-tap filter interpolation

As presented in fig.6, each half-pixel sample that is adjacent to two integer samples, is interpolated from integer-position samples using a 6-tap filter with weights (1/32,−5/32, 5/8, 5/8,−5/32,1/32). For example, half-pixel sample *b* is calculated from the six horizontal integer samples E, F, G, H, I and J as given in equation (2).

$$b = \text{round}((E - 5F + 20G + 20H - 5I + J)/32) \quad (2)$$

Similarly, *h* is interpolated by filtering A, C, G, M, R and T. Once all of the horizontal and vertical samples adjacent to integer samples have been calculated, the remaining half-pel positions are calculated by interpolating between six horizontal or vertical half-pel samples from the first set of operations. For example, *j* is generated by filtering cc, dd, h, m, ee and ff. The six-tap interpolation filter is relatively complex but produces an accurate fit to the integer-sample data and hence good motion compensation performance.

4.2 The Bilinear image interpolation

Pixels at quarter pixel position are obtained by Bilinear interpolation which will be described in this section. The bilinear interpolation is an interpolation method that can calculate the value of a function at any point from its two nearest neighbors. It is a widely used method to resize an image. It consists in carrying out a linear variation of intensities in each direction. Consider the case of an image of 2×2 pixels shown in the Fig.7, whose levels are A1, A2, B1 and B2. The calculations are carried out according to this approach; the first step is to interpolate the horizontal lines and determine the values of P1 and P3, then performed the second interpolation on vertical lines and calculate values of P2 and P4. Finally, a central interpolation is effected with the four nearest neighbors and the value of P5 is calculated.

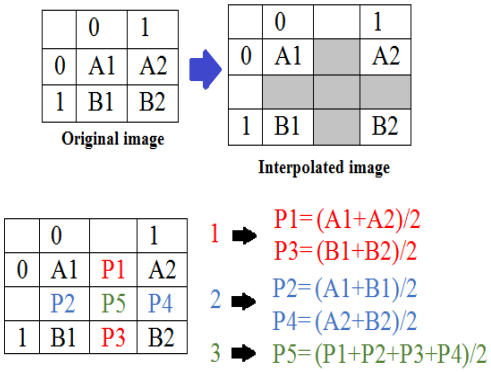

Fig. 7 The Bilinear interpolation

5 Subpixel Estimation in VC-1 standard

To perform motion estimation of one half-pixel and also of quarter-pixel or more, an interpolation between pixels is performed previously on the reference image. The VC-1 standard uses the Bicubic interpolation; the algorithm starts with the Bicubic interpolation between the samples of the search area in the reference frame to form a higher-resolution interpolated region. The first interpolation is performed to have half-pixel precision, to achieve quarter-pixel precision, interpolation is performed again.

Then, the algorithm searches the best match Mb in the interpolated region. Finally, the algorithm subtracts the samples of the matching region from

the samples of the current Mb to form the difference block or residual which is coded with an entropic encoder. As we have explained the interpolation techniques used by H.264/AVC, we present in this section the principle of Bicubic interpolation. Similar to the bilinear interpolation, the Bicubic interpolation [9] uses information from an original pixel, and 16 surrounding pixels to determine the color of new pixels which are created from the original pixel.

The Bicubic interpolation is an extension of cubic interpolation. Indeed, the Bicubic is a cubic interpolation applied in two dimensions (horizontal and vertical direction). The interpolated surface in Bicubic interpolation is smoother than corresponding surfaces obtained by bilinear interpolation used in H.264/AVC.

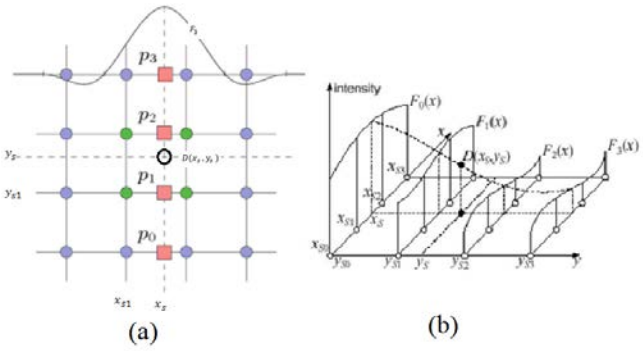

Fig. 8 The bicubic interpolation

In contrast to bilinear interpolation, which only takes 4 pixels into account, Bicubic interpolation considers 16 pixels as shown in Fig.8 (a). Images obtained with Bicubic interpolation are smoother and have fewer interpolation artifacts.

Suppose the function values are F_0, F_1, F_2 and F_3. The interpolated surface presented by Fig.8 (b) can then be calculated as in the equation (3). The interpolation problem consists of determining the 16 coefficients a_{ij}.

$$P(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij} x^i y^j, \quad 0 \le x \le 1, \quad 0 \le y \le 1 \qquad (3)$$

The above equation has 16 unknowns a_{ij}, and requires 16 boundary conditions to provide a unique solution for the coefficients. These boundary conditions are formed using the four control points; the four tangent vectors along the x-direction at the points, the four tangent vectors along the y-direction, and the four twist vectors. The bi-cubic surface patch obtained by solving the above linear system of equations is given by equation (4)

$$P(x,y) = [F_0(x)\ F_1(x)\ F_2(x)\ F_3(x)] \times \begin{bmatrix} P(0,0) & P(0,1) & P_y(0,0) & P_y(0,1) \\ P(1,0) & P(1,1) & P_y(1,0) & P_y(1,1) \\ P_x(0,0) & P_x(0,1) & P_{xy}(0,0) & P_{xy}(0,1) \\ P_x(1,0) & P_x(1,1) & P_{xy}(1,0) & P_{xy}(1,1) \end{bmatrix} \begin{bmatrix} F_0(y) \\ F_0(y) \\ F_0(y) \\ F_0(y) \end{bmatrix} \qquad (4)$$

6 Results of implementation and evaluation

The implementation is developed with MATLAB using the Full search Block Matching algorithm to ensure finding the optimal matching block. The simulated sub-pixel motion estimation based on h.264/AVC standard and VC-1 standard is operated to evaluate the effectiveness of used interpolation methods for each standard based on objective and subjective assessment metrics. The evaluations are also provided in terms of temporal complexity. In this study we have used two test video sequences: the Football sequence composed by 121 frames with QCIF resolution (176 x144), shown in Fig 9 (a) and the Foreman sequence composed by 121 frames with CIF resolution (352x288), shown in Fig 9 (b). The video frequency is set to 25 frames per second. Tested video sequences are treated by a successive Group Of Pictures (GOP). Each GOP is ordered as "IPPP" with a size of 12 frames and the I-frame is used as a reference frame to predict the rest of the GOP.

Fig. 9 Football (a) and foreman (b) video sequences

The main goal is to predict the current frame using a reference frame and to estimate the motion vector. Thereby each frame is divided into Mbs (the Mb size varied from 4x4 to 16x16). Then for each Mb in the current frame, the Mean Square Error (MSE) is calculated between this block and other blocks in the reference frame. The search of the best matching block in the reference frame is performed on varying search area from [x= -7: 7, y= -7 : 7] to [x= -16 : 16, y= -16 : 16]. The block giving the minimum value of the MSE is the matching block and the motion vector value is the vertical

displacement (y) and horizontal displacement (x) for this Mb.

In our simulation, we have tested many level of interpolation to perform the subpixel motion vector estimation. The assessment of tested results using Bilinear, 6-tap filters and Bicubic interpolation techniques are evaluated using two metrics: the Peak Signal-to-Noise Ratio (PSNR) and the Structural SIMilarity (SSIM).

The Peak Signal-to-Noise Ratio (PSNR) is a classic metric used to compare two frames; it can determine the level of distortion of a compressed image from its source. The PSNR is usually expressed in terms of the logarithmic decibel scale, high PSNR indicates that the reconstruction is of high quality. The PSNR is calculated using the mean squared error (MSE) by the equation (5), *i* indicate the Red, Green and blue colour image component. The PSNR of an RGB frame is the average value of the three colour components.

$$\mathrm{PSNR_i} = 10.\log_{10}\left(\frac{255^2}{\mathrm{MSE_i}}\right) \quad (5)$$

The second metric used to evaluate the image quality is the SSIM or The Structural SIMilarity. The SSIM is used to measure the similarity between two images. The SSIM index is a full reference metric and the measuring of image quality based on an initial uncompressed image as reference. SSIM is designed to improve the traditional methods like peak signal-to-noise ratio (PSNR) and mean squared error (MSE), which have proven to be inconsistent with human vision system [8].

Table 1 and Table 2 show the mean PSNR value and the elapsed time of the decoded first GOP for variable Mb size (w) and search area (p) with different sub-pixel accuracy using H.264/AVC estimation and VC-1 estimation respectively for Football and Foreman sequences. We note that when the interpolation level increases and the subpixel estimation accuracy increases therefore, the quality of the reconstructed image is better; the reason is that the interpolation generates more precision in the motion vector estimation. These results are confirmed by the results of Table 3 and Table 4 presenting the mean SSIM value of the decoded first GOP for variable Mb size (w) and search area (p) with different sub-pixel accuracy using H.264/AVC estimation and VC-1 estimation respectively for Football and Foreman sequences.

Nevertheless, the execution time of each algorithm progresses as soon as we increase the accuracy of estimation and it is increased almost three times when we go to a higher level of interpolation. This deduction is very clear in Fig.14 that presenting, in histogram graph , the average time of execution for each method of estimation with different accuracies. In addition, we note that the bicubic interpolation used by the VC-1 standard requires more computation time than the techniques used by H.264/AVC standard.

Table 1 The mean value of the PSNR and the elapsed time of the decoded first GOP of Football sequence for variable block size (w) and search area (p) with different accuracy using H.264 and VC-1 estimation

		Without interpolation		**1/2 interpolation pixel**				**1/4 interpolation pixel**				**1/8 interpolation pixel**			
				H.264/AVC		**VC-1**		**H.264/AVC**		**VC-1**		**H.264/AVC**		**VC-1**	
w	p	PSNR	TIME	PSNR	TIME	PSNR	TIME	PSNR	TIME	PSNR	TIME	PSNR	TIME	PSNR	TIME
4	7	30,659	2,543	31,253	6,765	31,318	7,363	31,574	25,030	31,665	26,712	31,668	94,046	31,746	95,191
4	10	30,814	3,988	31,466	12,619	31,522	13,669	31,801	48,228	31,887	49,181	31,888	190,329	31,957	188,341
4	16	30,959	8,524	31,641	73,894	31,668	30,692	31,987	117,064	32,061	119,262	32,070	461,244	32,136	503,317
8	7	30,598	1,189	31,007	2,279	31,125	2,031	31,249	6,589	31,357	6,728	31,296	25,321	31,421	26,105
8	10	30,665	1,568	31,085	3,826	31,191	3,488	31,323	12,493	31,436	12,766	31,366	48,486	31,498	51,673
8	16	30,702	2,687	31,128	8,638	31,223	7,809	31,351	30,221	31,469	29,982	31,401	122,908	31,524	130,185
16	7	29,998	0,861	30,238	1,011	30,453	0,867	30,425	2,161	30,627	2,096	30,450	6,922	30,652	6,981
16	10	30,044	0,961	30,301	1,416	30,495	1,259	30,469	3,834	30,676	3,732	30,494	13,322	30,691	13,466
16	16	30,053	1,284	30,299	2,780	30,497	2,443	30,470	8,721	30,681	8,598	30,494	32,596	30,697	41,943

Table 2 The mean value of the PSNR and the elapsed time of the decoded first GOP of Foreman sequence for variable block size (mb) and search area (p) with different accuracy using H.264 and VC-1 estimation

		Without interpolation		**1/2 interpolation pixel**				**1/4 interpolation pixel**				**1/8 interpolation pixel**			
				H.264/AVC		**VC-1**		**H.264/AVC**		**VC-1**		**H.264/AVC**		**VC-1**	
mb	p	PSNR	TIME	PSNR	TIME	PSNR	TIME	PSNR	TIME	PSNR	TIME	PSNR	TIME	PSNR	TIME
4	7	38,645	7,683	39,902	24,723	39,404	24,302	39,839	89,438	40,546	89,891	39,850	347,691	40,546	341,026
4	10	38,853	13,269	40,146	47,059	39,566	46,349	40,014	177,278	40,787	173,442	40,079	690,156	40,757	697,320

4	16	39,038	30,081	40,312	112,275	39,675	111,621	40,143	434,574	40,954	439,982	40,242	1726,600	40,901	1723,700
8	7	37,699	2,960	38,645	7,577	38,515	7,265	38,773	24,204	39,213	23,820	38,753	90,090	39,220	89,089
8	10	37,780	4,394	38,741	13,217	38,585	12,843	38,866	46,086	39,315	46,035	38,837	178,853	39,318	178,792
8	16	37,823	8,631	38,786	30,094	38,614	29,911	38,907	112,692	39,357	113,386	38,869	445,934	39,362	444,088
16	7	37,035	1,785	37,798	3,946	37,827	2,986	37,844	7,826	38,325	7,666	37,773	26,402	38,286	26,962
16	10	37,073	2,189	37,843	4,808	37,863	4,517	37,892	13,924	38,375	14,025	37,822	51,563	38,337	52,152
16	16	37,079	3,373	37,854	9,496	37,870	9,180	37,899	32,467	38,382	33,465	37,831	128,965	38,344	132,655

Table 3 The mean value of the luminance SSIM of the decoded first GOP of Football sequence for variable block size (w) and search area (p) with different accuracy using H.264 and VC-1 estimation

		Without interpolation	1/2 interpolation pixel		1/4 interpolation pixel		1/8 interpolation pixel	
w	p		H264/AVC	VC-1	H264/AVC	VC-1	H264/AVC	VC-1
4	7	0.9284	0.9377	0.9389	0.9432	0.9440	0.9444	0.9452
4	10	0.9311	0.9409	0.9421	0.9466	0.9471	0.9474	0.9483
4	16	0.9336	0.9438	0.9445	0.9495	0.9497	0.9503	0.9509
8	7	0.9210	0.9274	0.9302	0.9319	0.9342	0.9318	0.9350
8	10	0.9226	0.9289	0.9314	0.9331	0.9354	0.9329	0.9361
8	16	0.9234	0.9296	0.9321	0.9337	0.9361	0.9336	0.9367
16	7	0.9039	0.9076	0.9122	0.9078	0.9154	0.9053	0.9142
16	10	0.9049	0.9087	0.9130	0.9086	0.9161	0.9060	0.9149
16	16	0.9049	0.9086	0.9130	0.9085	0.9162	0.9060	0.9149

Table 4 The mean value of the luminance SSIM of the decoded first GOP of Foreman sequence for variable block size (w) and search area (p) with different accuracy using H.264 and VC-1 estimation

		Without interpolation	1/2 interpolation pixel		1/4 interpolation pixel		1/8 interpolation pixel	
w	p		H264/AVC	VC-1	H264/AVC	VC-1	H264/AVC	VC-1
4	7	0.9284	0.9377	0.9389	0.9432	0.9440	0.9776	0.9782
4	10	0.9311	0.9409	0.9421	0.9466	0.9471	0.9789	0.9795
4	16	0.9336	0.9438	0.9445	0.9495	0.9497	0.9797	0.9802
8	7	0.9210	0.9274	0.9302	0.9319	0.9342	0.9713	0.9721
8	10	0.9226	0.9289	0.9314	0.9331	0.9354	0.9719	0.9727
8	16	0.9234	0.9296	0.9321	0.9337	0.9361	0.9720	0.9727
16	7	0.9039	0.9076	0.9122	0.9078	0.9154	0.9637	0.966
16	10	0.9049	0.9087	0.9130	0.9086	0.9161	0.9643	0.9666
16	16	0.9049	0.9086	0.9130	0.9085	0.9162	0.9643	0.9666

In Fig.10 and Fig.11 we present respectively the mean PSNR value and the mean SSIM value of the decoded Foreman and Football sequences with different accuracy using H.264/AVC estimation and VC-1 estimation. The search area is fixed at 10 with variable Mb size. However in Fig.12 and Fig.13 we present respectively the mean PSNR value and the mean SSIM value of the decoded Foreman and Football sequence with different accuracy using H.264/AVC estimation and VC-1 estimation. The Mb size is fixed at 4 with variable search area. In the first place, we observe that the average value of PSNR increases by increasing the interpolation accuracy and that the quality of the reconstructed image using VC-1 estimation is better than that given by H.264/AVC estimation. Despite that computation time of the linear interpolation adopted by the H.624 / AVC standard is lower than the computation time of Bicubic interpolation used in VC-1 standard, we see that the predicted image quality using the second standard is better; the explanation is the fact that the Bicubic interpolation keeps well the image dynamics. This observation is confirmed by the SSIM values.

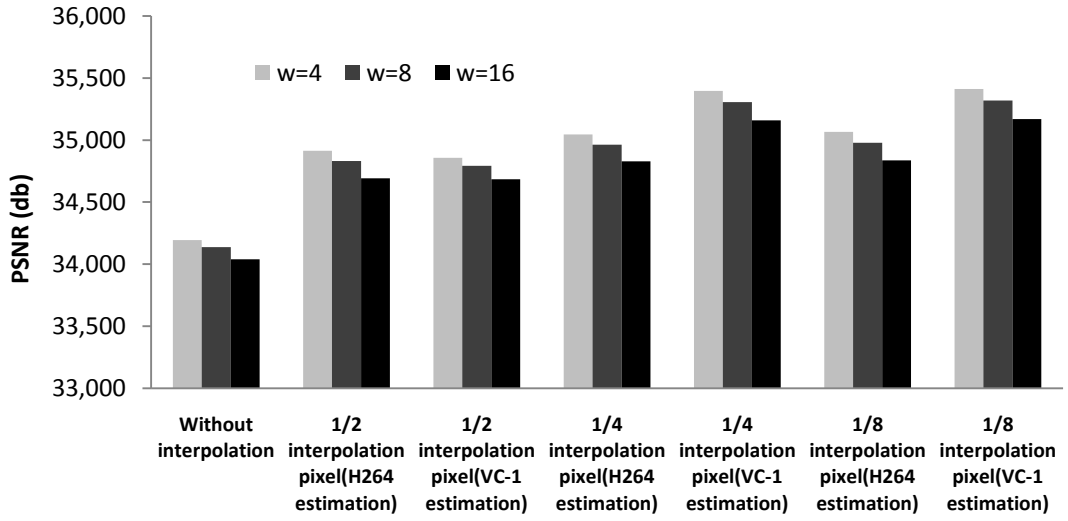

Fig. 10 The mean PSNR value of the decoded Foreman and Football sequences with different accuracy using H.264 estimation and VC-1 estimation. The search area is fixed at 10 with varied macro block size.

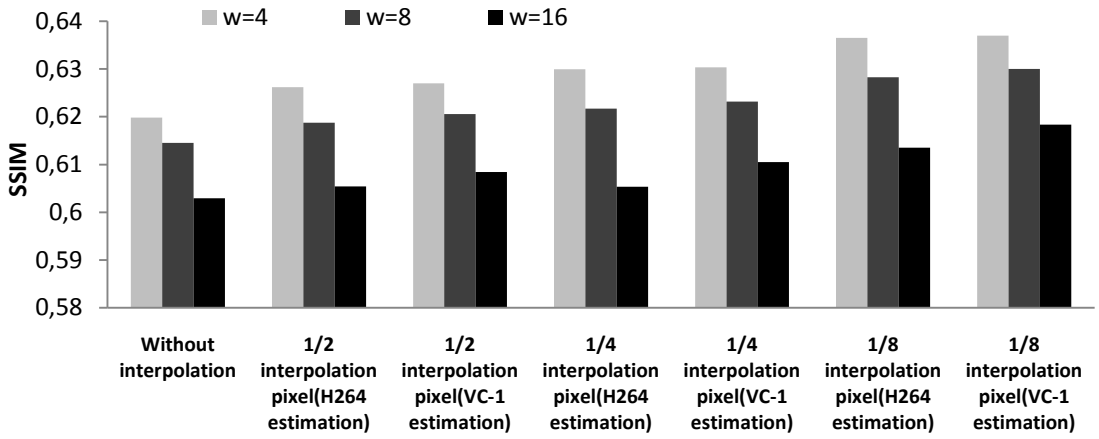

Fig. 11 The mean SSIM value of the decoded Foreman and Football sequence with different accuracy using H.264 estimation and VC-1 estimation. The search area is fixed at 10 with varied macro block size.

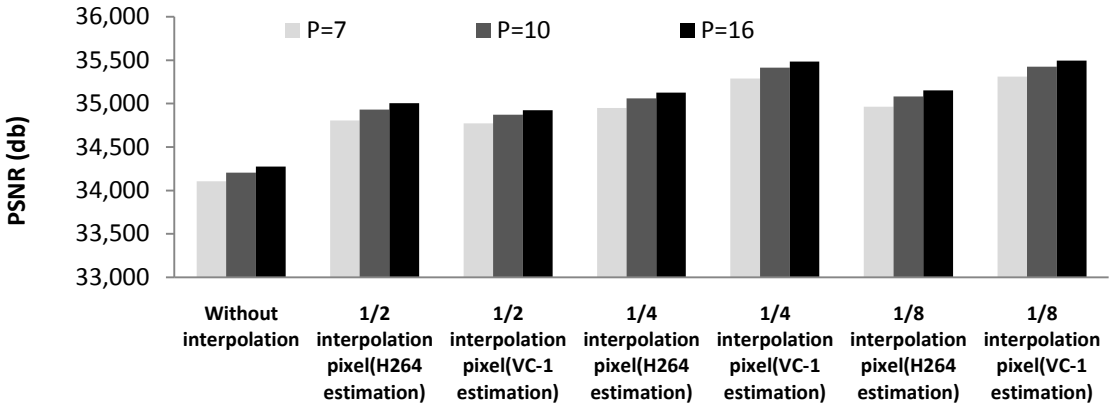

Fig. 12 The mean PSNR value of the decoded Foreman and Football sequence with different accuracy using H.264 estimation and VC-1 estimation. The macro block size is fixed at 4 with varied search area.

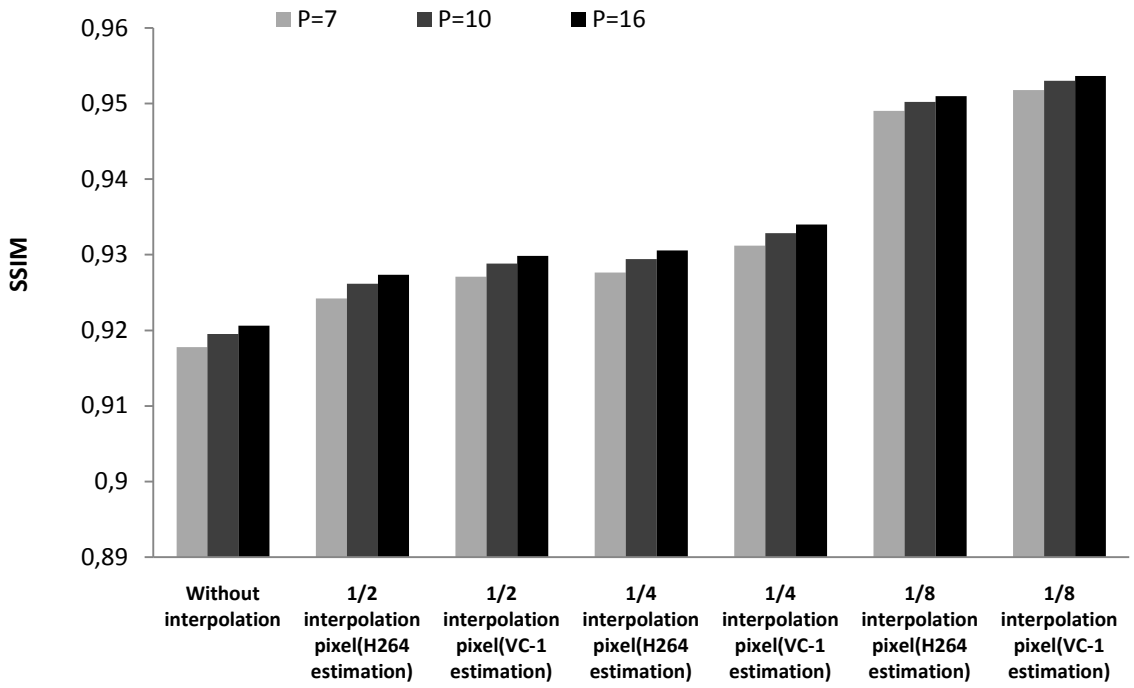

Fig. 13 The mean SSIM value of the decoded Foreman and Football sequence with different accuracy using H.264 estimation and VC-1 estimation. The macro block size is fixed at 4 with varied search area.

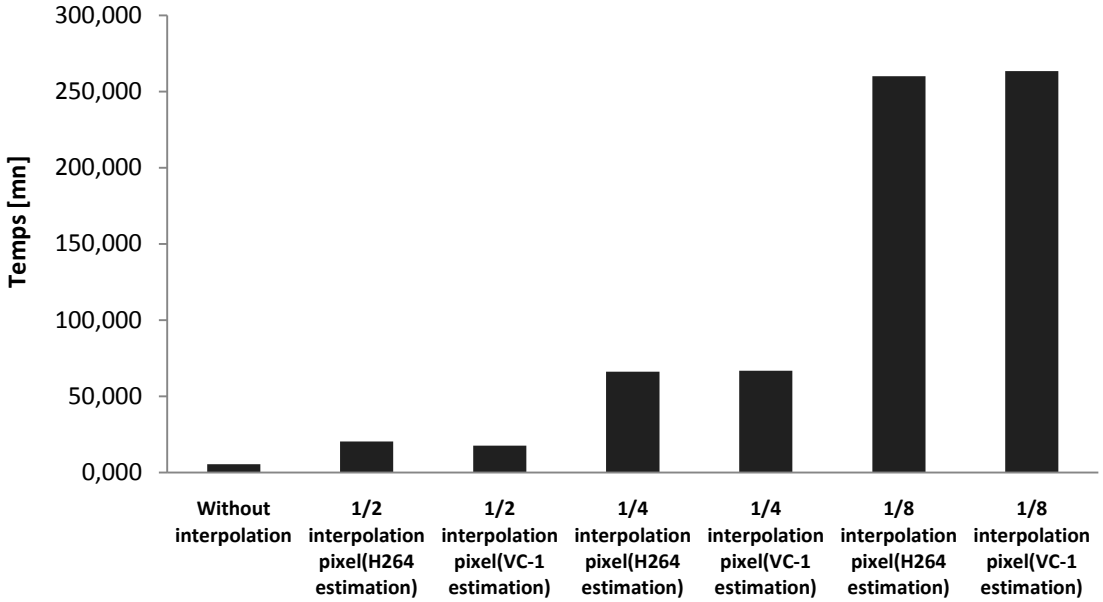

Fig. 14 The average time of execution for each method of estimation with different accuracy

In the second place, we observe that when the Mb size increases, the quality of the reconstructed image is better. We deduce that with a smaller size of Mb we can control with more precision the small motions in an image sequence. Therefore the small size of Mb (w) gives a great PSNR value of reconstructed frame. This observation is confirmed by the SSIM value. In the third place, we note that when the search area size (noted p) increases, the quality of the reconstructed image is better. In effect, with a larger size of search area we can control with more accuracy the small motions in an

image sequence; consequently the large size of search area gives a great PSNR value of reconstructed frame. Those observations are confirmed by the SSIM evaluation.

From these results, we deduce that the quality of predicted frame varies according these three parameters: the size of search area (p), the Mb size (w) and the level of interpolation. Moreover the analyzed results show that the PSNR is higher and the SSIM is closer to 1 for small Mb size, large search area and great level of interpolation but the disadvantage is that the calculation time becomes more important.

Fig.15 and Fig.16 present respectively the PSNR and the SSIM curves of decoded Football sequence for different motion estimation accuracy using H.264/AVC estimation technique and VC-1 estimation technique at fixed Mb size (w = 4) and search area (p = 7). However, Fig.17 and Fig.18 present respectively the PSNR and the SSIM curves of decoded Foreman sequence for different motion estimation accuracy using H.264/AVC estimation technique and VC-1 estimation technique at fixed Mb size (w = 4) and search area (p = 7). From these results, we note that the best results in terms of PSNR and SSIM are given using the VC-1 standard with Bicubic interpolation and eight-pixel motion vector estimation. However, worst results are given by the motion estimation without interpolation. We note also that the PSNR values of Football sequence vary between 28 dB and 32 dB, however, for the Foreman sequence they vary between 37 dB and 41 dB. This is explained primarily by the fact that the Football video sequence has a very fast motion compared to Foreman video sequence and especially in images 43, 44 and 45 which are characterized by the sudden appearance of a player and the ball in the scene. The second explication is due to the video sequence resolution which is QCIF for the Football while it is CIF for Foremen sequence.

In Fig.19 we present the original foremen frame number 54, the predicted frame without interpolation, the predicted frame with 1/2 accuracy using 6-tap filter and the predicted frame using 1/8 accuracy of Bicubic interpolation. By Observing figure 19, we confirm our objective results which show that the best results are achieved when the level of interpolation is more important and that the Bicubic interpolation gives the better result of sub-pixel estimation.

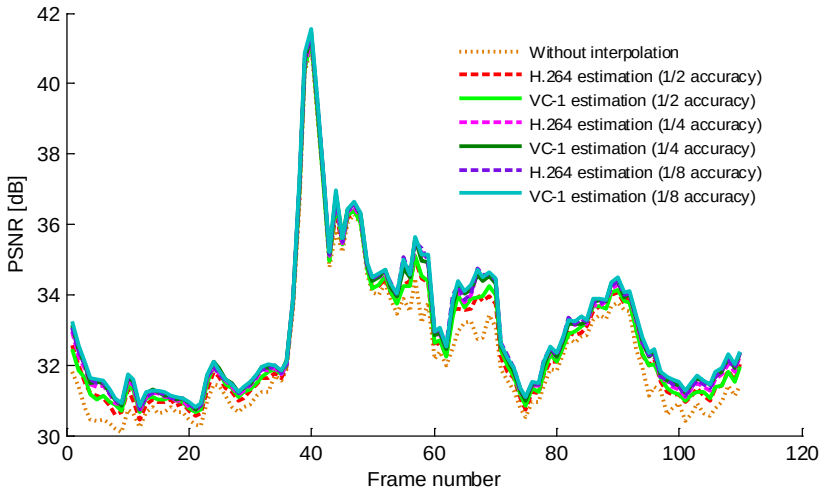

Fig. 15 PSNR curves of decoded Football sequence for different motion estimation accuracy using H.264/AVC estimation technique and VC-1 estimation technique at fixed Mb size (w=4) and search area (p=7)

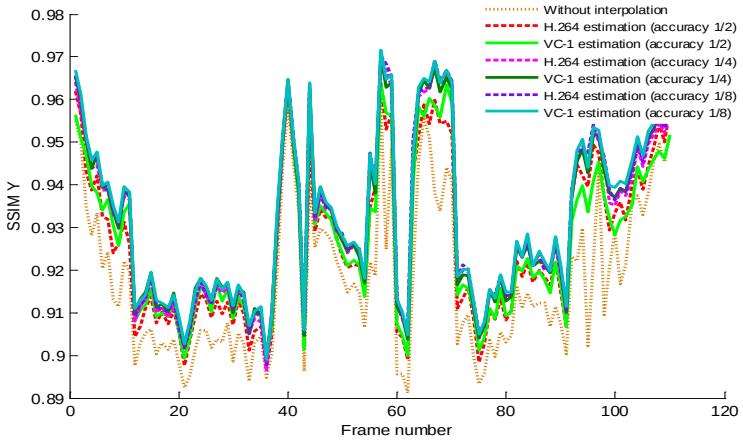

Fig. 16 SSIM curves of decoded Football sequence for different motion estimation accuracy using H.264/AVC estimation technique and VC-1 estimation technique at fixed Mb size (w=4) and search area (p=7)

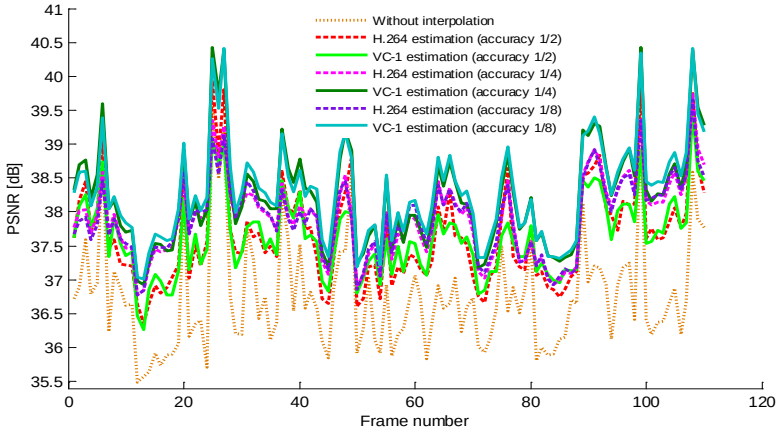

Fig. 17 PSNR curves of decoded Foreman sequence for different motion estimation accuracy using H.264/AVC estimation technique and VC-1 estimation technique at fixed Mb size (w=4) and search area (p=7)

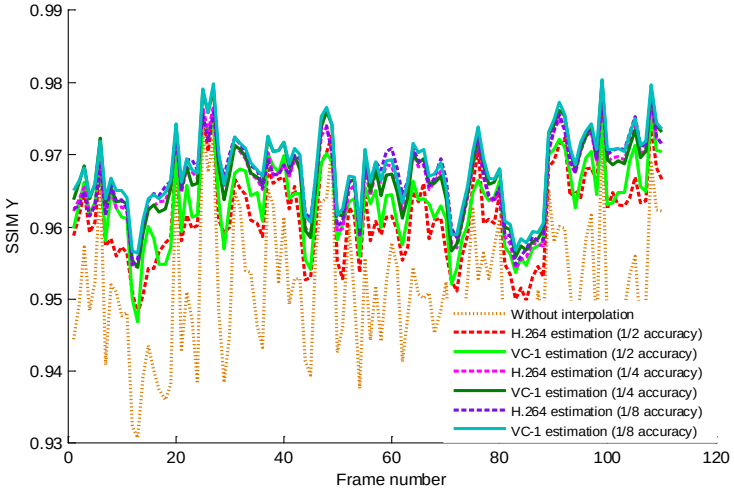

Fig. 18 SSIM curves of decoded Foreman sequence for different motion estimation accuracy using H.264/AVC estimation technique and VC-1 estimation technique at fixed Mb size (w=4) and search area (p=7)

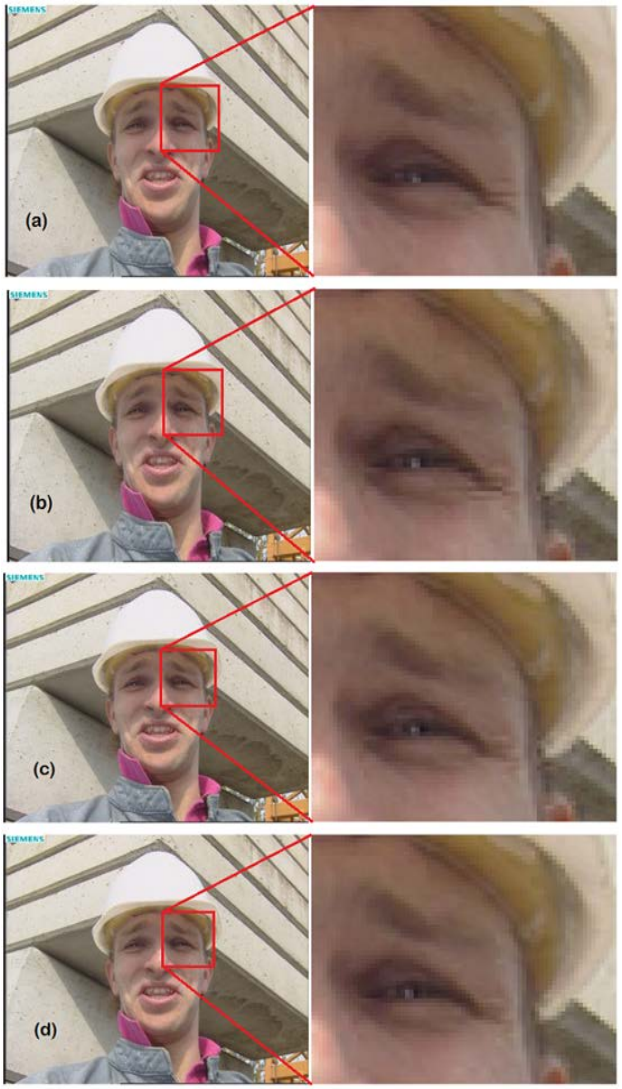

Fig. 19 The Different views for a part of foreman fame number 54: (a) the original frame, (b) predicted frame without interpolation

7 Conclusion and perspectives

The objective of this study was to compare the two subpixel motion estimation techniques used by H.264/AVC and VC-1 standards. The implementation of these motion estimation techniques is using the Full search Block Matching algorithm and Matlab implementation. The evaluations are operated using Football and Foreman video sequences. The simulation results are presented in terms of PSNR and SSIM. Firstly, we note that the motion vector estimation using sub-pixel accuracy gives the better results, than those given by Integer-pixel accuracy despite an important computational time of subpixel interpolation.

Secondly, we can conclude that the quality of predicted frames varies according to: the size of search area, the Mb size and the level and the function of interpolation.

Indeed, the analyzed results show that the PSNR is higher and the SSIM is closer to 1 for small Mb size, large search area and great level of interpolation.

The comparison of subpixel estimation of two compression standards led us to discover that the Bicubic interpolation used by the VC-1 standard gives the better quality of predicted frames but requires more computation time than the Bilinear and 6-tap filter interpolation techniques used by H.264/AVC standard.

In the perspectives of this work, we will integrate the subpixel interpolation implemented in the Separate Sign Coding with Motion Compensation (SSC-MC) based on Discrete Wavelet Transform which is our previous contributions [10][11].

References

[1] W. Hassen and H. Amiri, “Block Matching Algorithms for Motion Estimation”, the 39th IEEE IECON Annual Conference of the IEEE Industrial Electronics Society, pp. 136-139, November 2013.

[2] B. Cirod, “Motion-compensating prediction with fractional-pel accuracy”, IEEE Transactions on Communications, pp. 604-612, 1993.

[3] K. Panusopone, D. M. Baylon, “An analysis and efficient implementation of half-pel motion estimation”, IEEE Transactions on Circuits and Systems for Video Technology, pp. 724-729, 2002.

[4] Draft ITU-T Recommendation and Final Draft International Standard of Join t Video Specification, JVT-G050, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, May 2003.

[5] J.B. Lee, H. Kalva, “The VC-1 and H.264 Video Compression Standards for Broadband Video Services”, Springer, 2008.

[6] H.-M.Hang, Y. –M. Chou, S. “Chih.Cheng.Motion estimation for video coding standards”, J.VLSISignalProcess. Systems for Signal, Image, Video Technol. pp.113-136, November 1997.

[7] F. Urban, ‘’Implantation optimisée d'estimateurs de mouvement pour la compression vidéo sur plates-formes hétérogènes multi composants’’, Ph.D. dissertaion, Institut National des Sciences Appliquées de Rennes, 26 Mars 2008.

[8] Z. Wang, R. Hamid Sheikh and C. Alan Bovik. "Objective Video Quality Assessment The Handbook of Video Databases: Design and Applications", B. Furht and O. Marqure, ed., CRC Press, Chap. 41, Pages: 1041-1078, 2003
[9] W. K. Pratt, "Digital Image Processing", Wiley, New York, 1991.
[10] W. Hassen, J. Mbainaibeye, H. Amiri and C. Olivier, "A wavelet separate sign approach for video coding and motion compensation", Journal Of Telecommunications, Vol.11, Issue 1, pp.19-29, October 2011
[11] Wissal Hassen, Jérôme Mbainaibeye, Hamid Amiri & Christian Olivier, "A New Approach To Video Coding Based On Discrete Wavelet Coding And Motion Compensation", International Journal of Research and Reviews in Applied Sciences, Vol.11, Issue 2, pp.176-189, May 2012.