Chapter 4

Field Coding on H.264 + RFGS

At the end of Chapter 3, we claimed that field coding is inherently suitable for error resilience and error concealment. A further discussion of field coding on H.264+RFGS is given in this chapter.

4.1 On Base Layer

The base layer of H.264 + RFGS is coded by the H.264/AVC standard. In the H.264/AVC standard, field coding is already included. H.264 provides a macroblocks-adaptive frame/field coding (MBAFF) scheme which makes it possible to arbitrarily choose frame coding or field coding for every macroblock [2]. Since it is typically more efficient to code static regions with frame coding mode while to code dynamic regions with field coding, MBAFF coding usually achieves higher coding efficiency than frame coding.

Here we put emphasis on the topic of error resilience. We take field coding as an error resilience tool and apply it to the coding of the whole sequence. In the following subsections, we'll discuss the comparison between field coding and frame coding in terms of their error resilience capability.

4.1.1 Proposed Types of Field Coding

In field coding, we partition each frame into two fields and code these two fields individually. Compared with frame coding, field coding offers more prediction modes in the operation of inter-frame motion compensation. Fig 4.1 shows a general prediction mode of frame coding. When the reference frame number is set to 1, the previous frame is the only choice for P frames. As to field coding, there are four prediction modes as shown in Fig. 4.2. The details of these modes are to be described below.

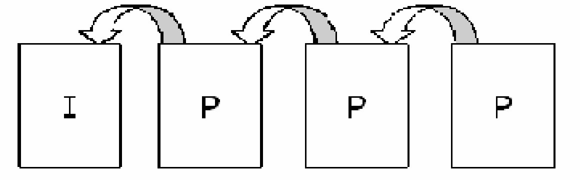


Fig 4.1 A general prediction mode of frame coding

Fig. 4.2(a) shows the default type of field coding in H.264, called as top-oriented prediction field coding (TOPFC) in this thesis. In this scheme, except intra fields, “top field” refers to the top field of the previous frame, while “bottom field” refers to the bottom field of the same frame. This scheme is unfavorable to error resilience because there would be too much dependency between top fields and bottom fields. Once there is a lost in a top field, the corresponding bottom field will also be influenced. Moreover, considering coding efficiency, the coding rate of TOPFC is 1.91 times of the original one, even much higher than that of FMO (see Table 4.1). The above descriptions indicate that TOPFC would be a bad choice as an error resilience tool.

Based on TOPFC, a modified scheme with the least dependency between the top field and bottom field of the same frame is proposed. This coding structure is named as parallel prediction field coding (PPFC) and is shown in Fig. 4.2(b). There is only one reference field for each field of a P frame and this reference field is restricted to the same kind of field in the previous frame. For example, a top field in a P frame can only refer to the top field of the previous frame. Besides, the coding rate ratio between PPFC and the original frame coding is 1.19, less than TOPFC and FMO. Since the two aforementioned disadvantages of TOPFC are solved in PPFC, PPFC becomes a more suitable structure for error resilience.

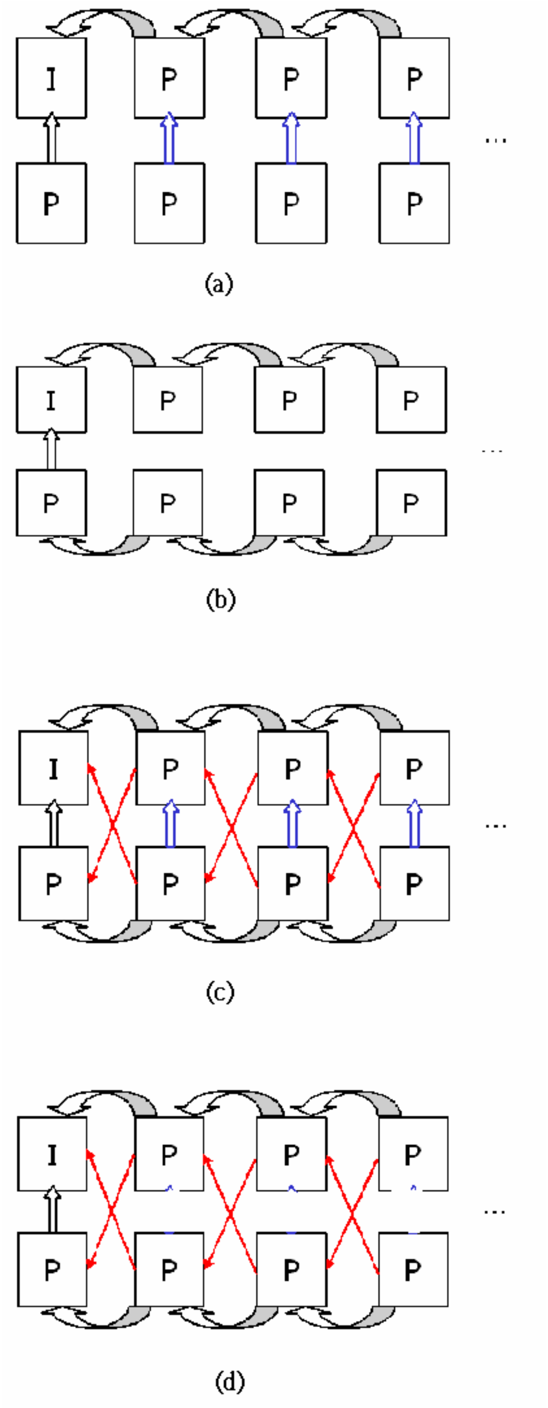


Fig. 4.2 Four prediction schemes of field coding

(a) Top-field Oriented Prediction Field Coding (TOPFC)

(b) Parallel Prediction Field Coding (PPFC)

(c) Crossing Prediction Field Coding (CPFC)

(d) Full Search Prediction Field Coding (FSPFC)

However, in the aspect of coding efficiency, the PPFC structure is still not a satisfactory choice. Fig. 4.2(c) shows the third type of field coding scheme: crossing prediction field coding (CPFC). In this coding scheme, each field of a P frame may make reference to either the top field or the bottom field of the previous frame. Another structure with an even higher coding efficiency is shown in Fig. 4.2(d). This coding scheme is named full search prediction field coding (FSPFC). In this scheme, the bottom field of a P frame may refer to not only the two fields of the previous frame but also the top field of the current frame. The dependency between the top and bottom fields in FSPFC is higher than that in PPFC. In Table 4.1, it shows that the coding rate of the FSPFC structure is only 1.03 times of the original frame coding. Due to the consideration of coding efficiency, FSPFC, instead of PPFC, is adopted in this paper. However, the improvement of coding efficiency causes the reduction of error resilience capability. The effect of error propagation becomes slightly more serious now. In the next subsection, an effective and simple enough error concealment algorithm is to be proposed for the FSPFC coding structure.

Table 4.1 Coding rate records of Field Coding (QP=42, PL=200 bytes)

	Frame coding	TOPFC	PPFC	CPFC	FSPFC
Coding rate ratio	1	1.91	1.19	1.14	1.03

Test sequences: Foreman, Stefan, Table, Container

Before getting into the next subsection, an experiment is given here to prove that FSPFC is really more error resilient than frame coding. Fig. 4.4 shows the comparison of error resilience performance between frame coding and FSPFC, under the setting of QP=42 and PL=200 bytes. We can see that the coding efficiency of frame coding is slightly higher than that of field coding. As the packet loss rate (PLR) increases, the PSNR performance of field coding gradually exceeds that of frame coding. However, the difference is not significant.

If the operation of post-processing is taken into consideration, the difference between frame coding and FSPFC becomes more apparent. In frame coding, it would be difficult to conceal lost macroblocks since all the neighboring pixels are lost together. These lost macroblocks are usually concealed by temporal replacement. On the contrary, in field coding, besides temporal concealment, spatial interpolation is also feasible. The lost information in one field can be efficiently concealed by simply interpolating the information in the other field. As shown in Fig. 4.4, with the employment of post-processing, the superiority of FSPFC becomes more significant with the increase of packet loss rate.

In Fig. 4.4, we only use the simple linear interpolation method as the post-processing method. However, in general a bitstream is coded in order, with top field first and then bottom field. As the top field is being decoded, the information of bottom field is not available yet. Hence, for the concealment of frame-coded videos or the top fields of field-coded videos, we use direct temporal replacement, as shown in Fig. 4.3(a). On the other hand, for the concealment of the bottom fields of field-coded videos, we may either use temporal replacement or direct spatial interpolation as illustrated in Fig. 4.3(b).

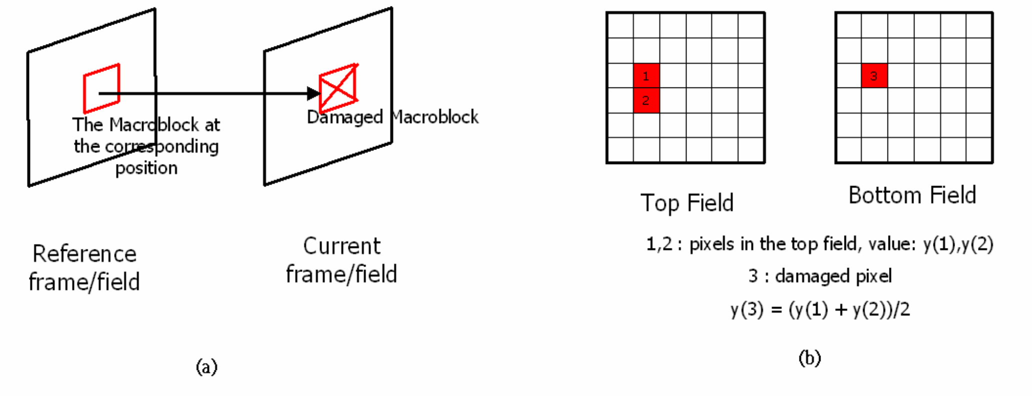


Fig. 4.3 Two basic error concealment methods.
(a) Direct temporal replacement.
(b) Direct spatial interpolation.

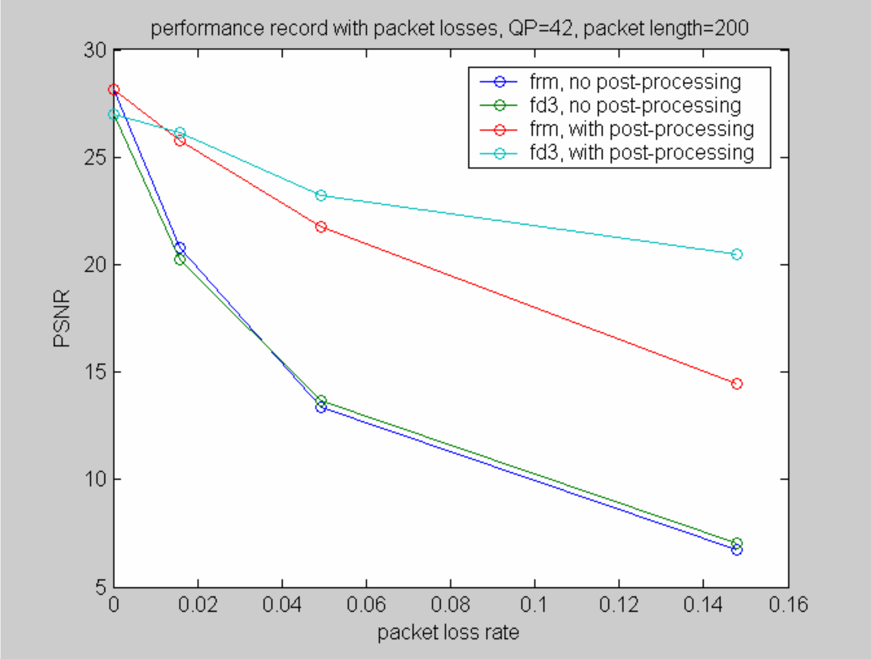


Fig. 4.4 Comparison of error resilience performance between FSPFC and frame coding

Blue line: Frame coding, without post-processing
Green line: FSPFC, without post-processing
Red line: Frame coding, with post-processing
Light blue line: FSPFC, with post-processing

4.1.2 Error Concealment

We have shown that field coding could be much more error-resilient than frame coding, when error concealment is applied. In this subsection, the techniques of error concealment are further discussed. Since the decoding process is expected to be real-time, low complexity and short delay time are the two major concerns in the development of these error concealment methods.

In field coding, there are two fields for each frame. In PPFC, expect for I frames, top fields and bottom fields are coded independently. Due to the close relationship between top fields and bottom fields, this PPFC scheme is very suitable for error concealment. However, due to the consideration of coding efficiency, FSPFC, instead of PPFC, is adopted in this thesis. In the FSPFC scheme, each bottom field may refer to its corresponding top field. Hence, these top fields should be better protected and the strategies of error concealment would be different for different types of fields.

4.1.2.1 Error Concealment for Bottom Fields

When a macroblock is lost in a bottom field, the probability of having a macroblock loss at the same place of the corresponding top field is quite small. Hence, in most cases, for each lost macroblock in the bottom field, we can assume that the related information in the top field, including motion vectors and residuals, would be available. This top-field information could be used to conceal the errors in the bottom field.

Basically, the error concealment methods for bottom fields could be classified into 2 major approaches:

1. Spatial interpolation: A detailed discussion can be found in [15]. Since the order of spatial filter only slightly influences the PSNR performance, the simplest form is used in this paper. That is, the upper pixel and the lower pixel in the top field are used to interpolate the lost pixels in the bottom field. The interpolation equation is expressed as below:

   $$P_b(m, n) = (P_t(m, n) + P_t(m, n+1))/2$$

   where b denotes bottom field, t denotes top field, and P(m, n) denotes the intensity value at pixel (m, n).
2. Temporal concealment: In [14], when doing temporal concealment, the information of the next frame must exist and this makes both buffer size and delay time increase at the decoder side. Here, for real time applications, a different approach of temporal concealment is adopted. In field coding, the two corresponding MB’s in the top and bottom fields are interleaved with each other. Since these two MB’s locate almost at the same place of the image frame, it is reasonable to assume the motion vectors of these two MB’s are very similar to each other. Fig. 4.5 shows the relative positions between an MB and its neighbor

MV's. Table 4.2 shows the expectation of motion vector difference (MD) between an MB and its neighboring MB's. It can be seen that the motion vector difference between an MB and its corresponding MB in the top field is the smallest. Since in Fig 4.7 we find by experiments that the power of difference of DCT coefficients is proportional to the value of motion vector errors statistically, we can say that the motion vector at the corresponding MB in the top field does offer the best estimation for the motion vector at the bottom field.

In H.264, a 16x16 MB is partitioned into sixteen 4x4 blocks; i.e., there are 16 motion vectors in each MB. However, the minimum transmission unit is still an MB and thus the minimum unit of data loss would be an MB. In practice, the motion vector of every lost 4x4 block in a bottom field is replaced by that of the corresponding 4x4 block in the top field.

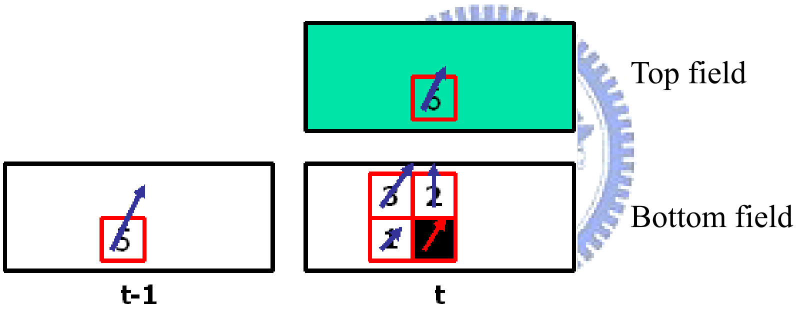


Fig. 4.5 Diagram of the relative positions between an MB and its neighbor MV's

Black MB: current MB

MB with number 1 – 6: neighbor MB's of the current MB

Table 4.2 Expectation of MD between current MB and its neighbor MB's (QP=36, PL=200 bytes, 'Foreman')

Number of neighbor MBs (according Fig.4.5)	1	2	3	1+2	5	6
Exp. of MD x	3.06	3.82	4.56	3.31	4.60	2.54
Exp. of MD y	1.24	1.60	1.75	1.29	1.91	1.05

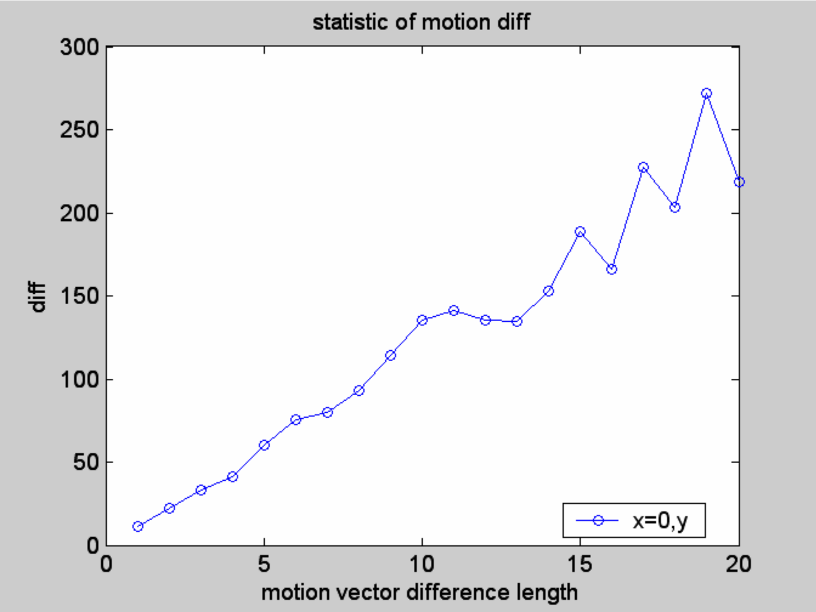


Fig. 4.6 Power of differences of DCT coefficients versus motion vector errors

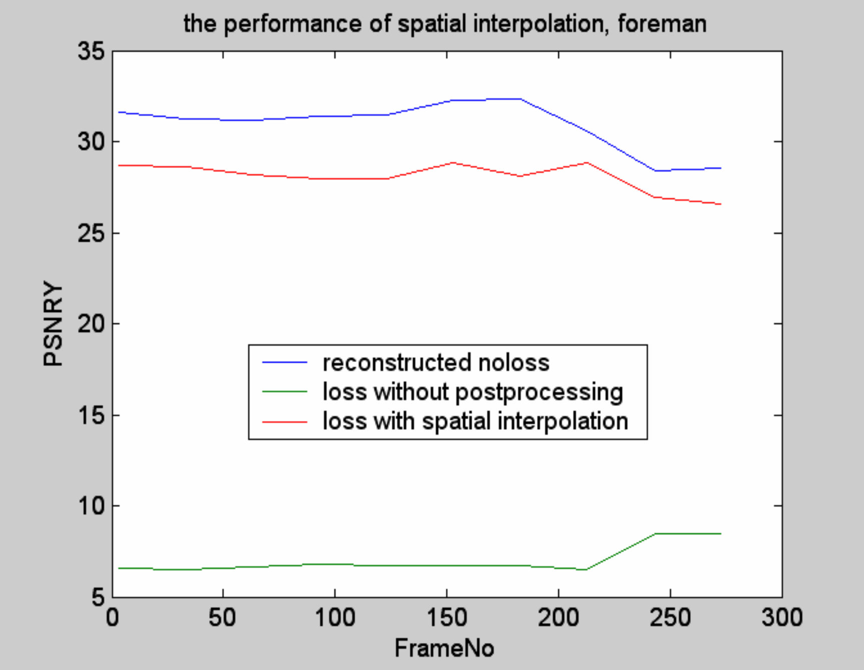


Fig. 4.7 Performance of the concealment method ‘Spatial interpolation’

Sequence: Forman

Blue line: reconstructed frame with no loss

Green line: reconstructed frame with the whole bottom field being lost

Red line: reconstructed frame with loss + spatial interpolation

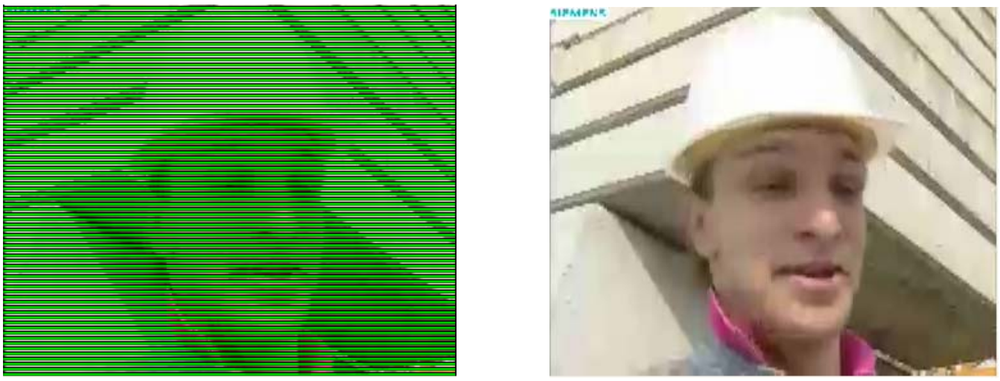


Fig. 4.8 Experimental result of spatial interpolation

(a) Frame with the whole bottom field being lost.

(b) The concealed frame (by spatial interpolation).

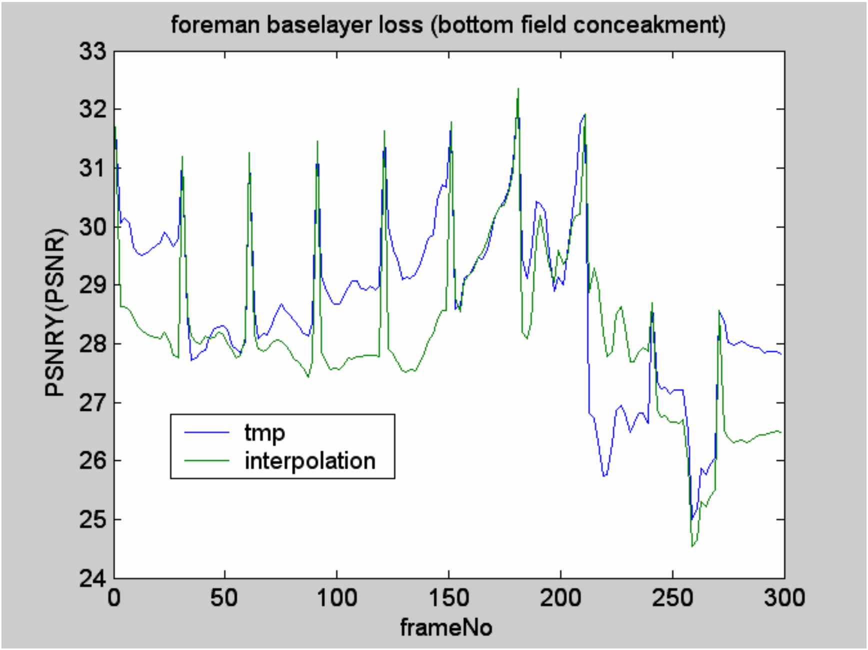


Fig. 4.9 Simulation results of the ‘Foreman’ sequence. In this simulation, the bottom field of the (30N+3)th frame is lost, with N = 0, 1, …, 9.

Blue line: Using temporal concealment

Green line: Using spatial interpolation

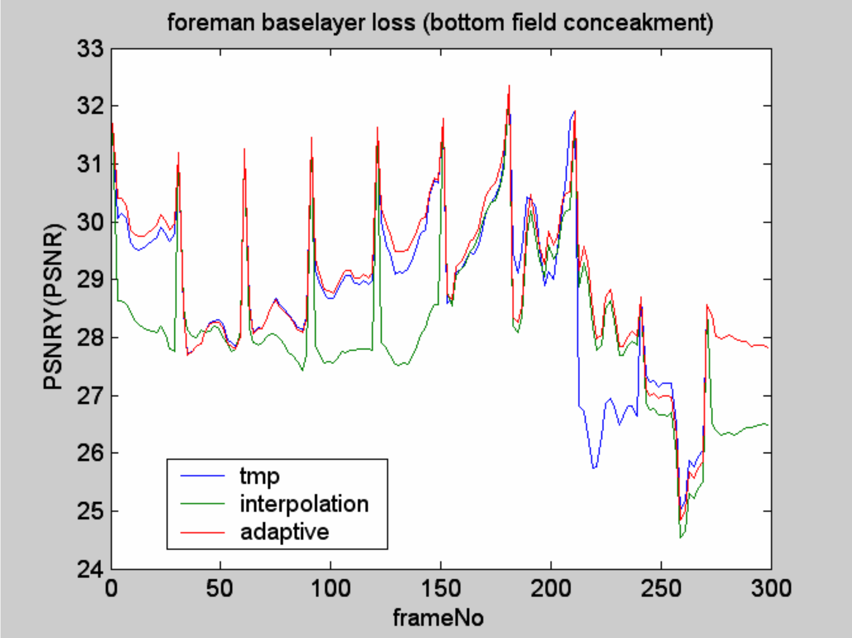


Fig. 4.10 Simulation results of the 'Foreman' sequence. In this simulation, the whole bottom field of the (30N+3)th frame is lost, with N = 0, 1, …, 9. When the magnitude of motion vector is larger than 2 pixels, spatial interpolation is used; otherwise, temporal concealment is used.

Blue line: using temporal concealment

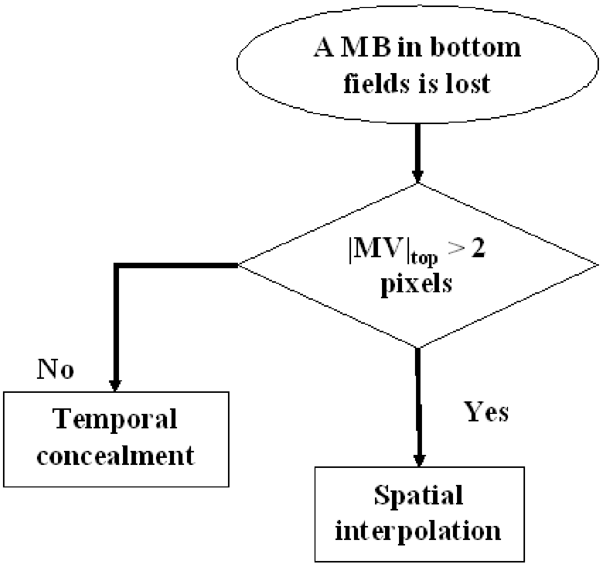
Green line: using spatial interpolation

Red line: using adaptive concealment

In the above two methods, we can find that the performance of spatial interpolation is reasonably good. Fig. 4.7 shows the experimental result of spatial interpolation by evaluation of dB and Fig. 4.8 gives an example of damaged frame and concealed frame. However, for these regions with less temporal change, the performance of temporal concealment outperforms that of spatial interpolation. Fig. 4.9 shows the simulation results of spatial interpolation and temporal concealment individually. Hence, we may use the magnitude of motion vectors to adaptively select spatial interpolation or temporal concealment. When motion vectors are larger, spatial interpolation is preferred. Otherwise, temporal concealment would be used for stationary parts. In Fig. 4.10, we show the simulation results based on spatial interpolation, temporal concealment, and the proposed adaptive method. The proposed method achieves the best performance most of time.

As to how the threshold of motion vector should be selected in this adaptively method, so far we use a statistically experimental value instead because a theoretical value has not been derived yet.

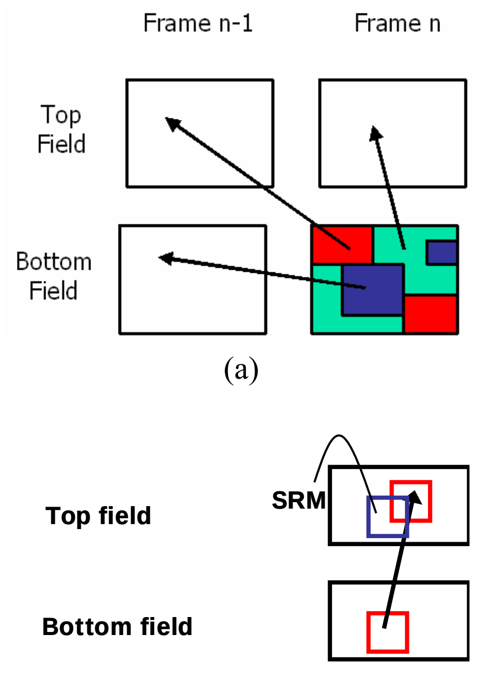
In summary, the diagram of the proposed method is plotted as below:



4.1.2.2 Error Concealment for Top Fields

Next, the error concealment for the top field is to be considered. The main difference between top field concealment and bottom field concealment is that the bottom field is allowed to refer to its corresponding top field while the top field is not allowed to refer to its corresponding bottom field, as illustrated in Fig. 4.11(a).

With the FSPFC structure, since some regions in the bottom field may refer to the corresponding top field, the loss in the top field may cause some regions in the bottom field to be affected. Hence, it would be impractical to use the information of these referring regions in the bottom field to conceal the regions in the corresponding top field. We call the corresponding regions in the top field as ‘spatially referred regions’ (SRR) (Fig. 4.11(b)).



(a)

(b)

Fig. 4.11 An illustration of the correlations between a bottom field and its reference fields.

(a)The red, green and blue regions indicate these regions that refer to previous top field, previous bottom field, and current top field, respectively.

(b) Spatially referred regions (SRR)

Since a 16x16 MB is partitioned into sixteen 4x4 blocks, the concealment of each block can be done individually. Here we propose two strategies for the concealment of an SRM block. If there is any neighbor block whose corresponding block in the bottom field does not refer to the top field, like the green regions in Fig. 4.13, the motion vector of the lost block could be replaced directly by the motion vector of the neighbor block. If more than one neighbor block meet the conditions, the order of priority is shown in Fig. 4.12. In this thesis, we name this method 'Temporal Concealment based on Neighboring Motion Vectors' (TCNMV) method.

However, sometimes a connected SRM region could occupy a large area, like the black regions in Fig. 4.13, and the TCNMV method is no longer applicable. In this case, the motion vectors of these SRM blocks are directly replaced by the motion vectors of the corresponding blocks in the previous top field. We name this method as

‘Temporal Concealment based on Anterior Motion Vectors’ (TCAMV) method. The principle of TCAMV is to be described below.

There are two major conditions that may result in a large SRM region. One condition is the occurrence of a large change in the current frame. The other condition is the presence of a smooth region. The first condition means that the change between the current frame and the previous frames is dramatic and thus the temporal correlation between these two frames is small. The second condition means the correlation between the top field and the bottom field is strong. In both cases, temporal compensation may be inefficient and the MB’s in the bottom field tend to refer to MB’s in the top field. In the case of smooth regions, the large size of an SRM region implies that the smooth region could be a large one. Hence, using the motion information of the corresponding blocks in the previous top field may get a satisfactory result.

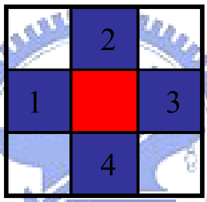
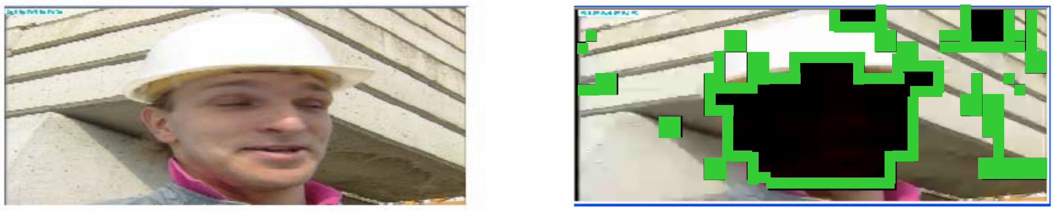


Fig. 4.12 A block (red) and its neighboring blocks (blue).
The number indicates the order of priority.



(a) (b)

Fig. 4.13 Regions in the top field that are referred by MB’s in the bottom field
Test sequence: the Foreman sequence.

(a) Original top field.

(b) Reconstructed field. The green regions indicate the SRR regions which can be concealed by TCNMV, while the black regions indicate the SRR regions which can be concealed by TCAMV.

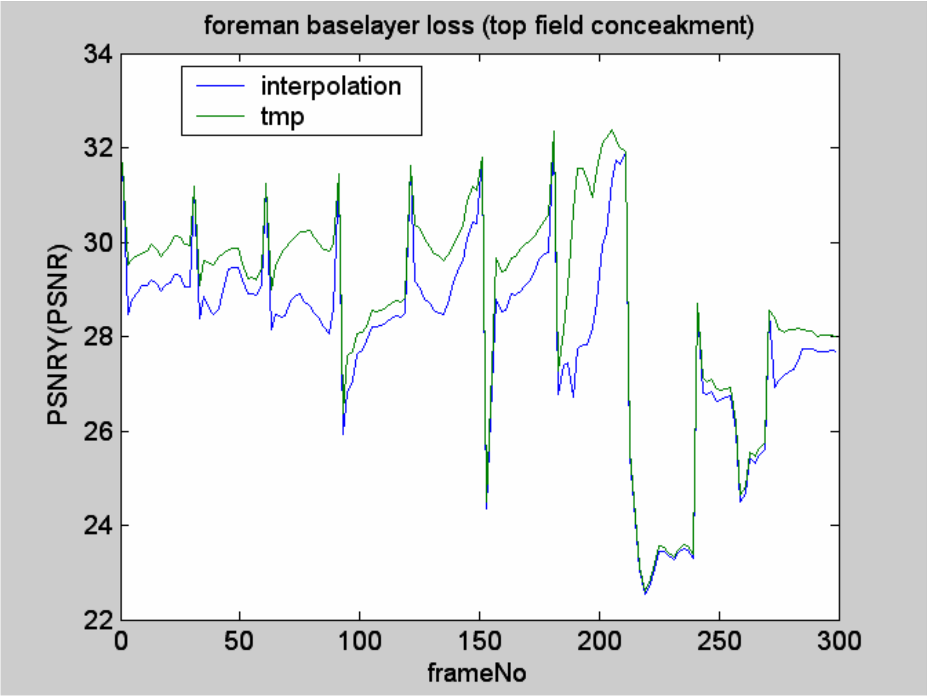


Fig. 4.14 Simulation results of the ‘Foreman’ sequence. In this simulation, the whole top field of the (30N+3)th frame is lost, with N = 0, 1, …, 9.

Blue line: using temporal concealment.

Green line: using spatial interpolation

For these non-SRM blocks, the error concealment operation techniques similar to those methods mentioned in Secion 4.1.2.1 are applicable. Fig. 4.14 shows the simulation results. For SRM blocks, TCNMV method and (TCAMV) method are used and temporal concealment and spatial interpolation are applied individually for the rest. Compared the two curves in Fig. 4.14, we find the temporal concealment always performs better. This is because spatial interpolation is suitable for smooth regions but smooth regions usually belong to non-SDE regions. Hence, the chance of using spatial interpolation becomes quite small. For this reason, we simplify the concealment method of top field by taking off the spatial interpolation.

The diagram of the proposed method is plotted as below:

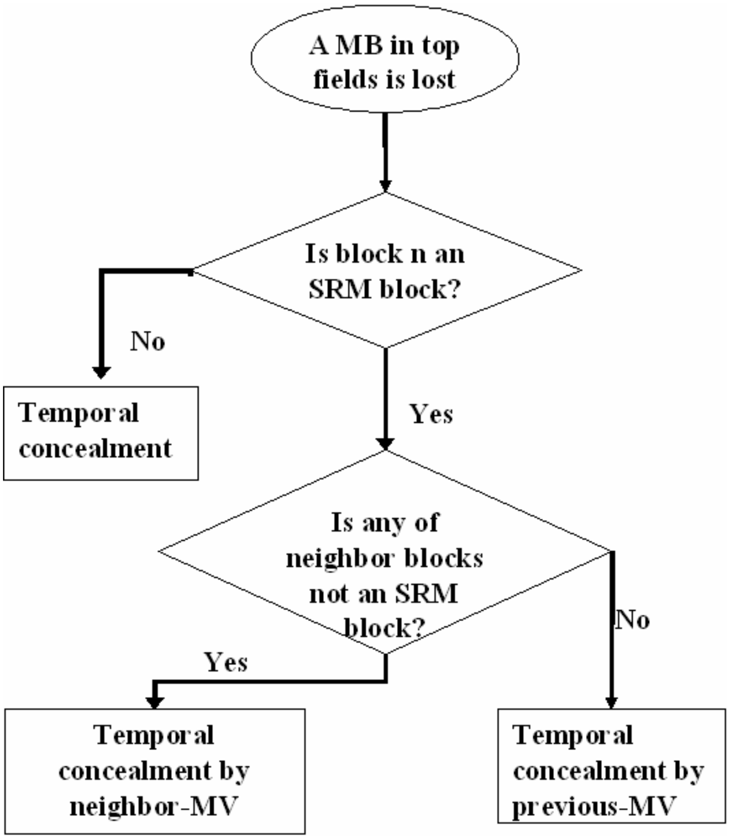


Fig. 4.15 shows the comparison of error resilience between PPFC and FSPFC. The concealment method used in both fields of PPFC is just the same as the concealment method used in the bottom field for the FSPFC structure. This is because there is no MB's in the bottom field referring to MB's in the top field with the PPFC structure. Since the motion vectors can be better recovered, Fig. 4.15 shows a more stable result with the PPFC structure However, considering the influence on coding efficiency, we still choose FSPFC as the structure for field coding.

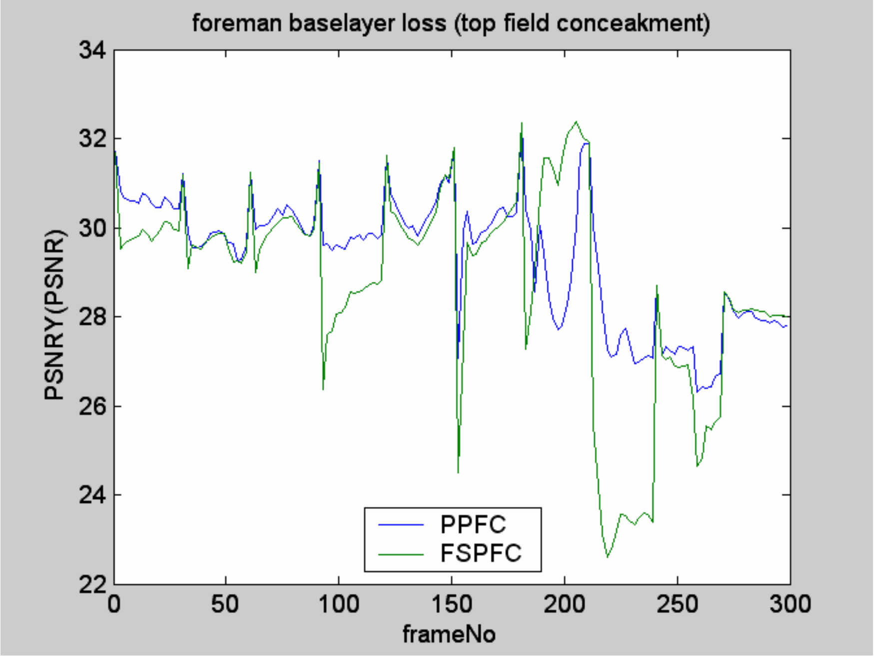


Fig. 4.15 Comparison of the error resilience between PPFC and FSPFC

Blue line: using PPFC

Green line: using FSPFC

4.2 On Enhancement Layer

We have realized how field coding on the base layer of H.264+RFGS enhances the error resilience of H.264. Next we apply the technique of field coding on the enhancement layer of H.264 + RFGS and want to find how this structure improves the error resilience capability of H.264 + RFGS coding.

4.2.1 Proposed Structure

On a base layer with field coding, we can select whether to apply field coding on the enhancement layer. The structure of field coding over the base layer + frame coding over the enhancement layer is abbreviated as fld-frm coding and is shown in Fig. 4.16. The coding process is described as below:

1. Finish the base layer coding and build a reconstructed frame of base layer.

2. Get the residual of the original frame by subtracting the reconstructed frame. The residual constructs the enhancement layer.
3. Do motion-compensation on the enhancement layer according to RFGS. Set α, and β. Since the motion vectors of the bottom fields is not always correct in FSPFC (See 4.1.2.2). The motion vectors in top fields are adopted and a pair of macroblocks in the enhance layer use the same motion vector in the top field. Notice that the minimum unit of H.264 is a 4x4 block. If there are different motion vectors in a macroblock of the base layer, the left-top one would be selected.
4. Code the enhancement layer in bit-plane coding.

Another choice is to apply field coding on the enhancement layer. We called this coding structure fld-fld coding (in Fig. 4.17). The coding process is described as below:

1. Build a reconstructed field for each base-layer field.
2. Get the residual of the original field by subtracting the reconstructed field. The residual constructs the enhancement layer.
3. Do motion compensation on the enhancement layer according to RFGS. Set α, and β. Here we have two types of enhancement-layer fields, top and bottom fields. For the same reason that the motion vectors of the bottom fields is not always correct in FSPFC, the motion vectors in top fields are adopted in both enhancement-layer top and bottom fields.
4. Code the enhancement-layer fields individually in bit-plane coding.

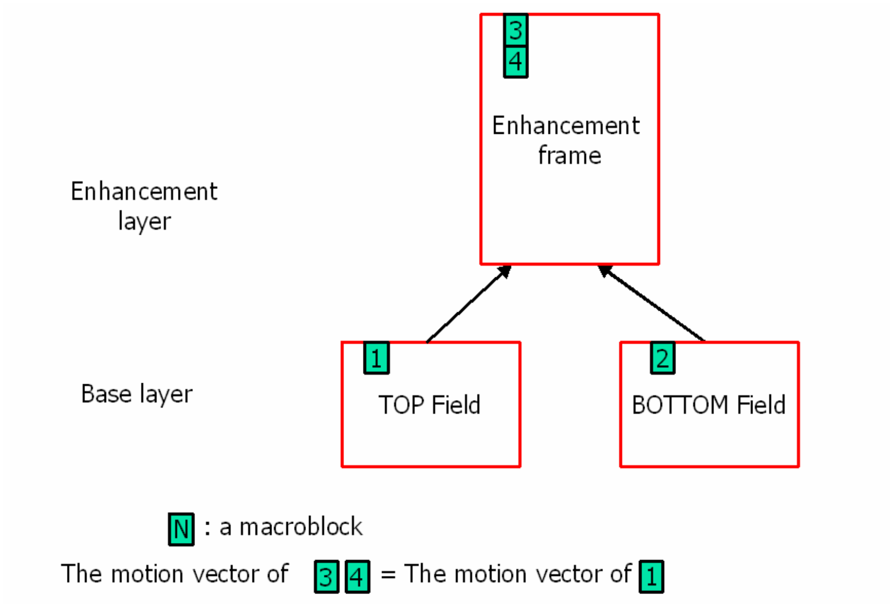


Fig. 4.16 The structure of field coding over base layer + frame coding over enhancement layer

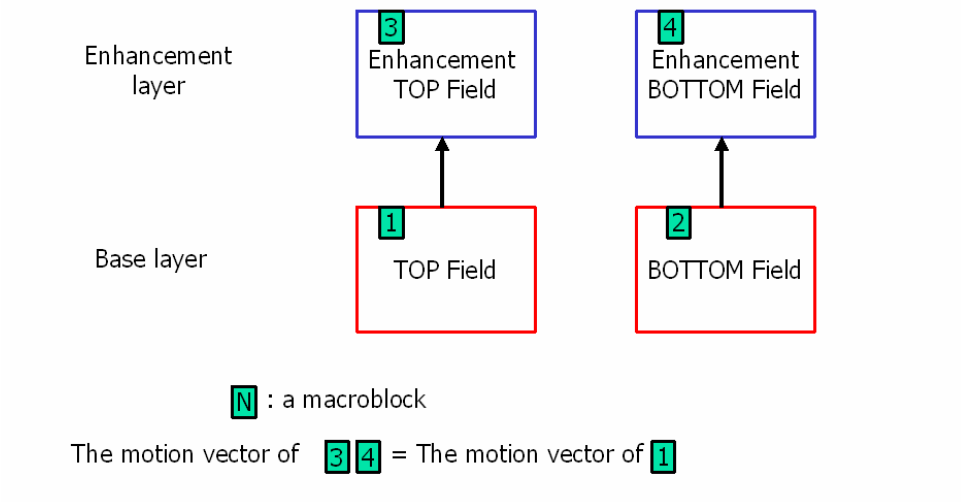
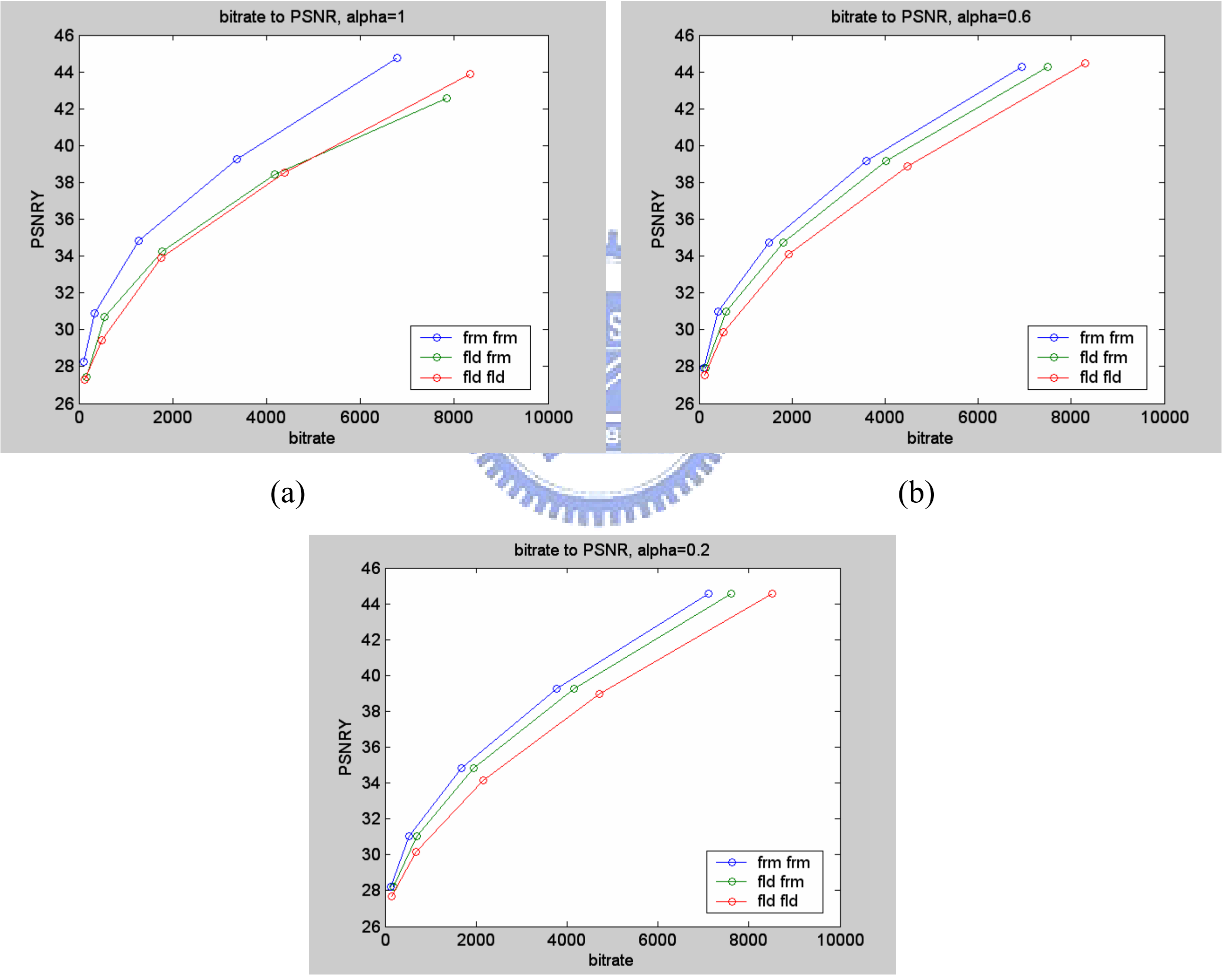


Fig. 4.17 The structure of field coding over base layer + field coding over enhancement layer

The coding efficiency of each coding structure is shown in Fig. 4.18. The entire frame coding (frm-frm coding) has the highest coding efficiency. When α is not close to 1, the performance curves of all three structures are quite stable and the order of coding efficiency is frm-frm coding, fld-frm coding, and then fld-fld coding. When α is close to 1, as shown in Fig 4.18 (a), there is a large performance decreasing in fld-frm coding and fld-fld coding if compared to frm-frm coding. When comparing the performance of different α's, as shown in Fig. 4.18 (b) (c), an unusual reduction of performance happens when α = 1. This problem occurs because of the motion compensation process. Since we only use motion vectors in the top field of the base layer, the motion vectors are probably not the optimal choices. This circumstance makes the highest coding efficiency not to happen at α =1 but at some other α value which is close to 1 (See Fig. 4.19).



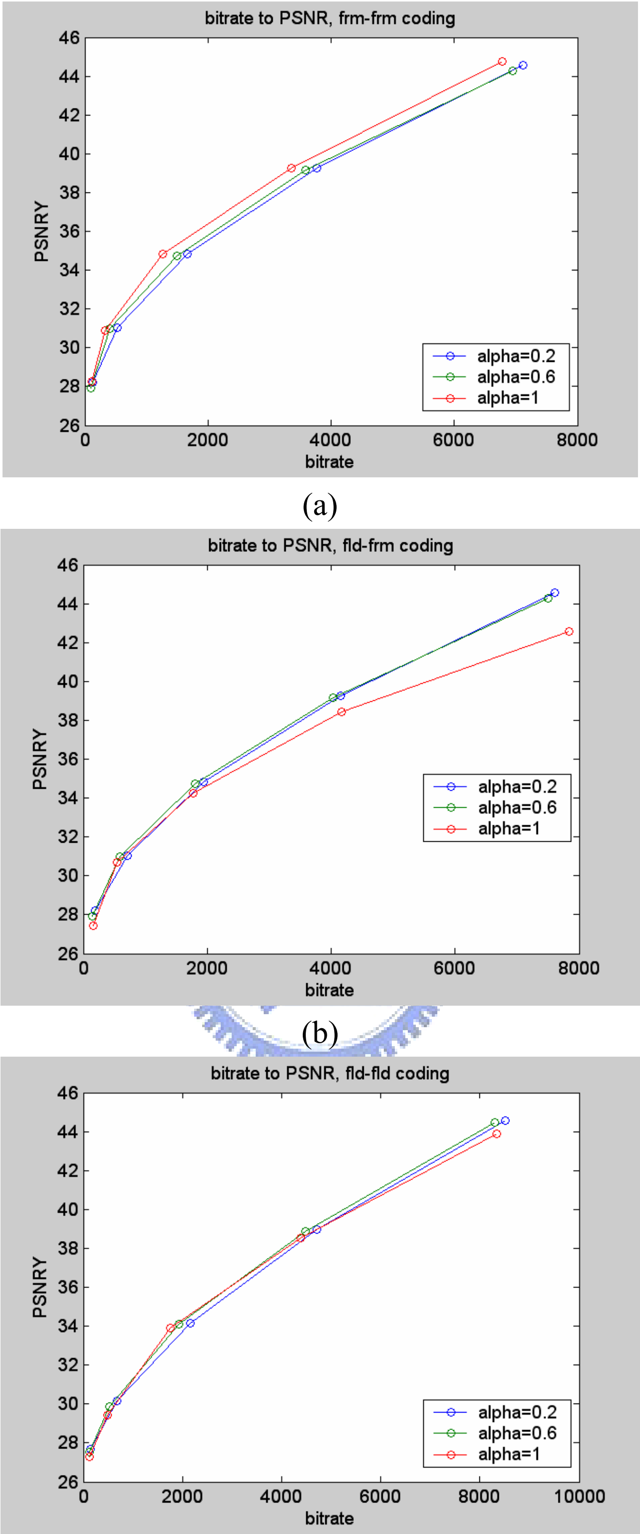
(a)

(b)

(c)

Fig. 4.18 The coding efficiency of H.264 + RFGS with frm-frm coding (blue), fld-frm coding (green), and fld-fld coding (red), respectively.

The QP of base layer = 42 and β = 14400. (a) α = 1, (b) α = 0.6, (c) α = 0.2



(a)

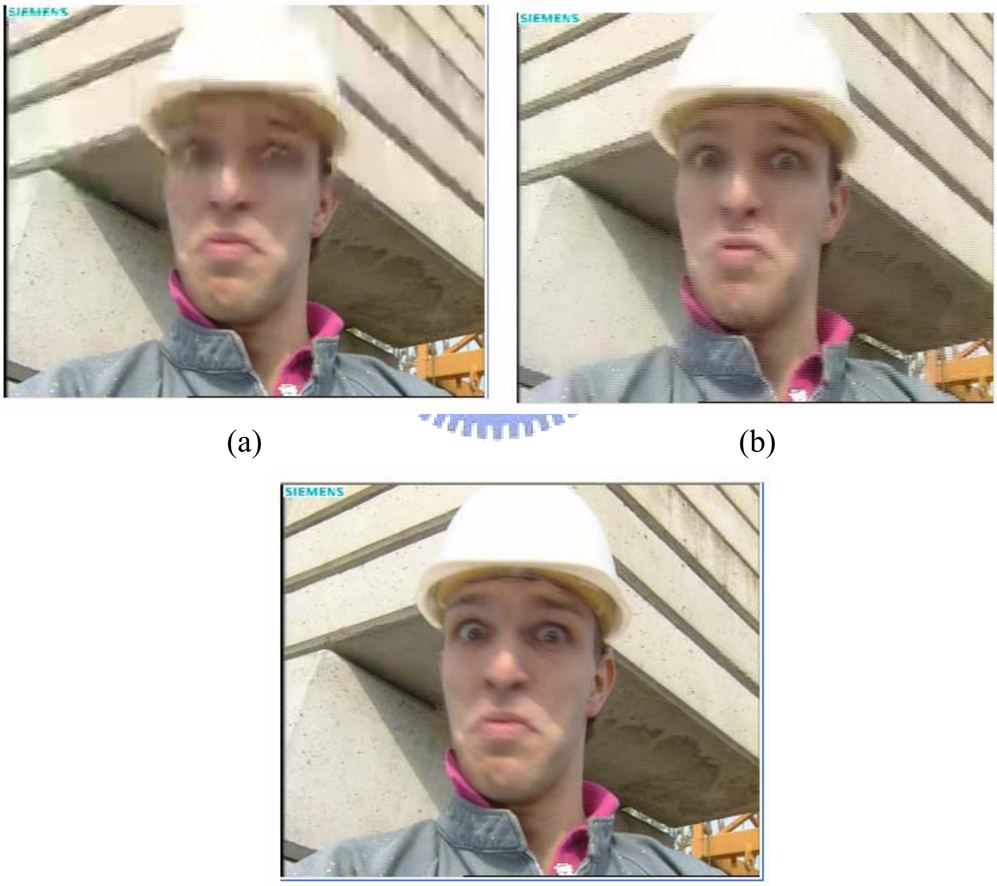
(b)

(c)

Fig. 4.19 The coding efficiency of H.264 + RFGS with α = 1 (red), α = 0.6 (green), and α = 0.2 (blue). The QP of base layer = 42 and β = 14400. (a) frm-frm coding. (b) fld-frm coding. (c) fld-fld coding.

4.2.2 Error Concealment

In this subsection, a discussion of error concealment on enhancement layer is given. Fig. 4.20 shows the simulation results when the enhancement layer information of half a frame is lost without error concealment. We can find that although the PSNR result of field coding is almost the same as that of frame coding (31.005:31.039), the visual quality of field coding is actually better than that of frame coding. This once again verifies that field coding is more error resilient than frame coding from the viewpoint of human vision.

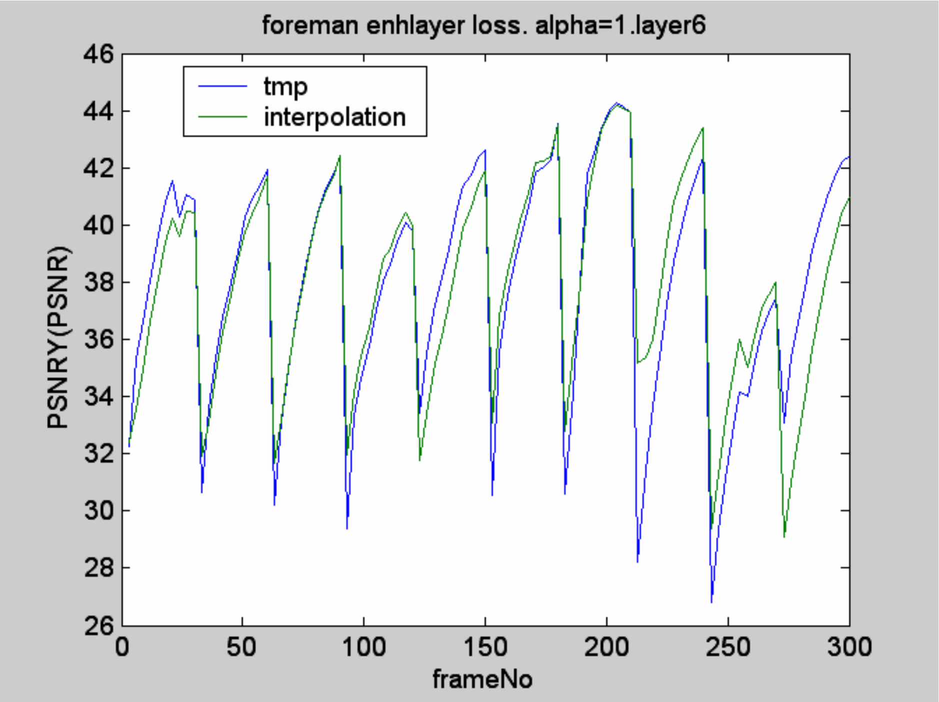


(a) (b)

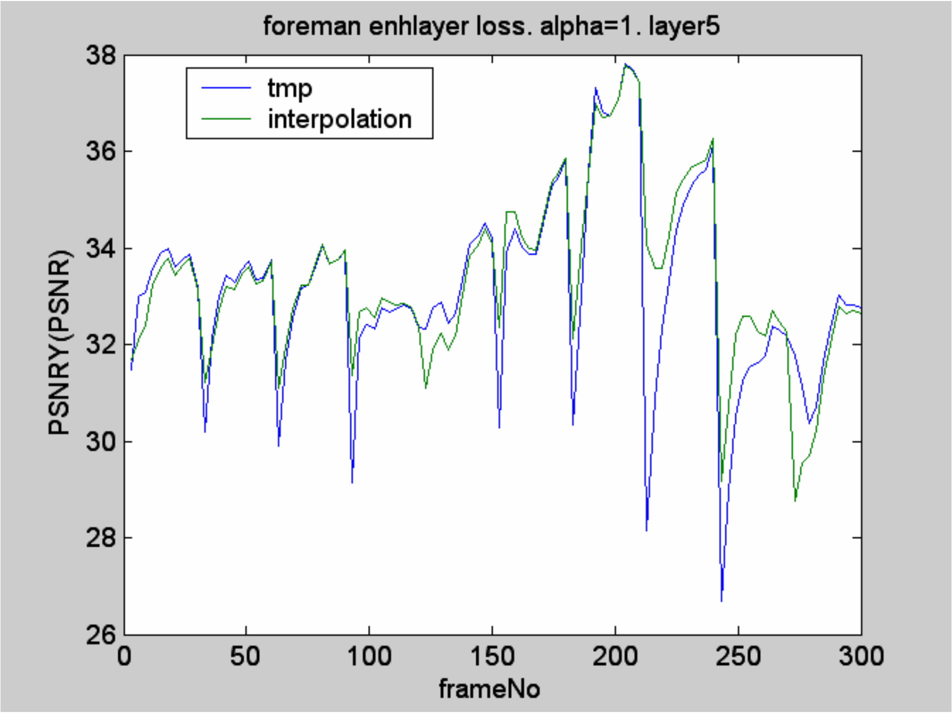
(c)

Fig. 4.20 Comparison the vision effect of frame and field coding,
Sequence: Foreman frameNo.92
(a) fld-frm coding: top-half frame lost, PSNRY=31.039dB
(b) fld-fld coding: bottom field lost, PSNRY=31.005dB
(c)Original frame

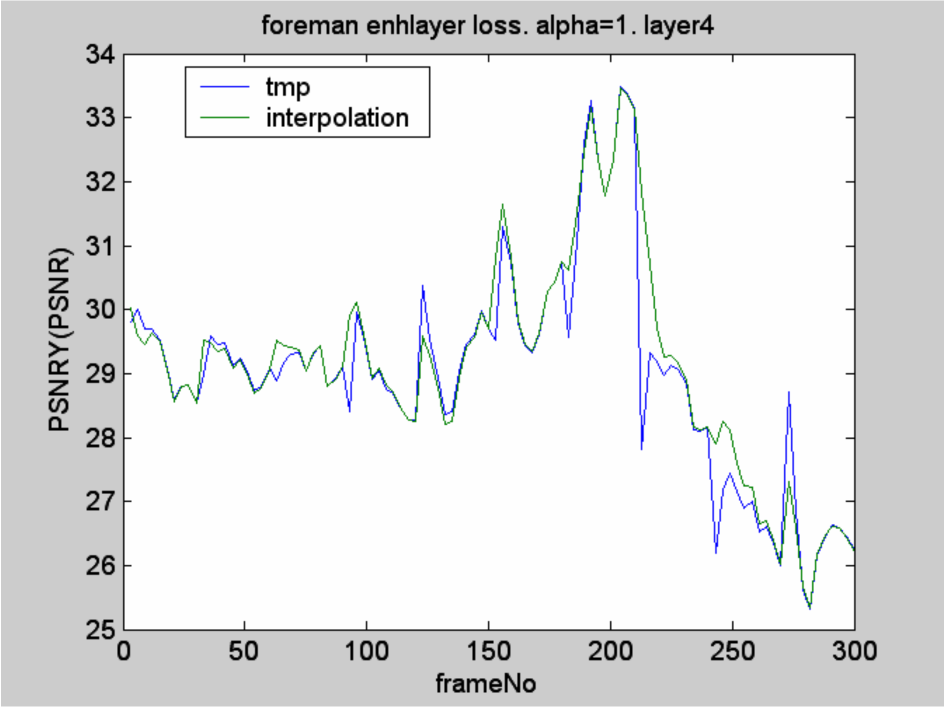
Since both the top and bottom fields in the enhancement layer are mutually independent, concealment methods for enhancement layer are similar to that for the bottom field in the base layer. The major difference is that no motion vector is lost when data in enhancement layer are lost. The simulation results of these two methods are shown in Fig. 4.21. Similar to base layer, we can find an adaptive concealment method on enhancement layer, as shown in Fig. 4.22. When there is no motion, temporal concealment is used. Otherwise, interpolation is used.



(a)



(b)

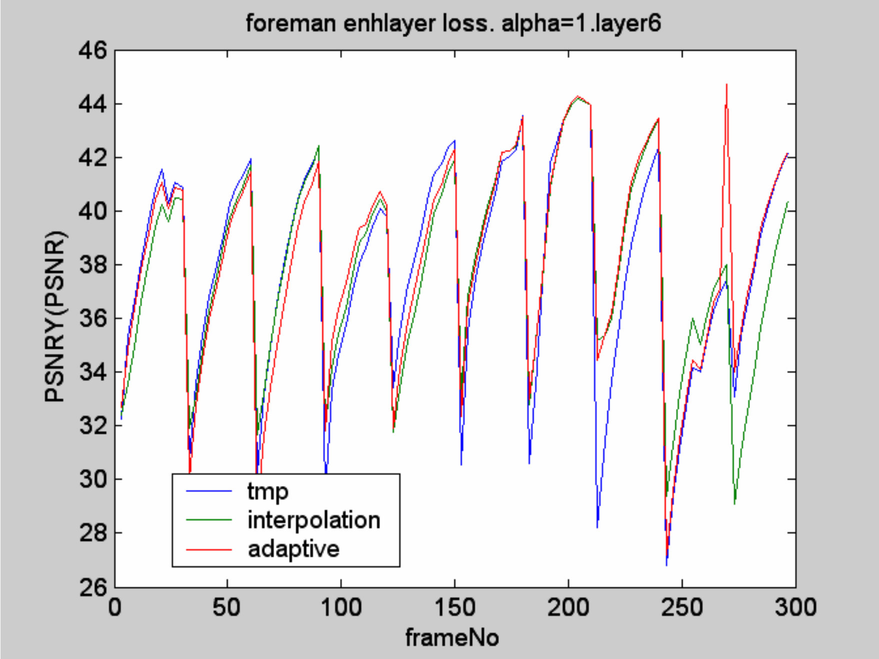


(c)

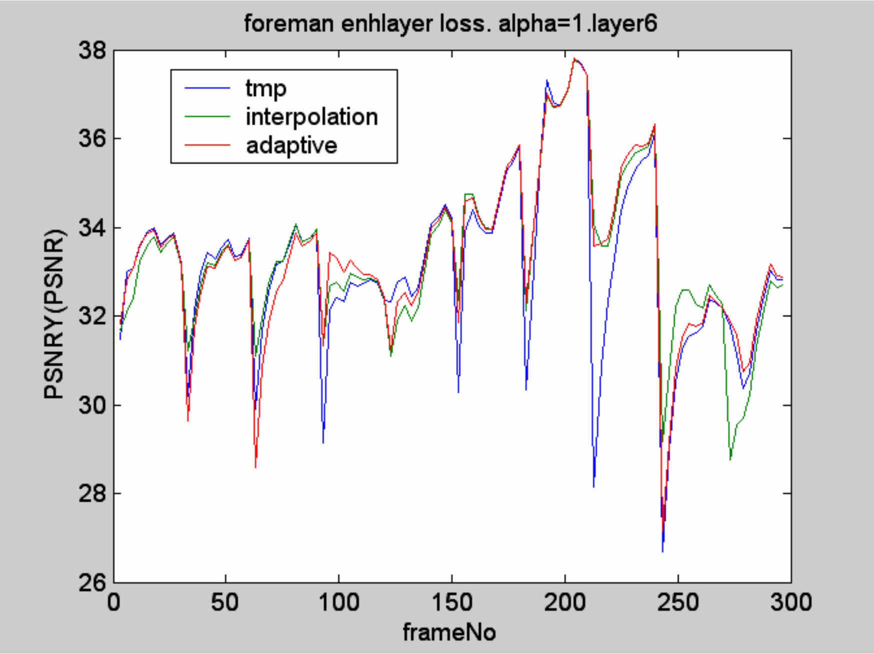
Fig. 4.21 Simulation results of concealment methods

(a) With 6 bit-planes in enhancement layer
(b) With 5 bit-planes in enhancement layer
(c) With 5 bit-planes in enhancement layer

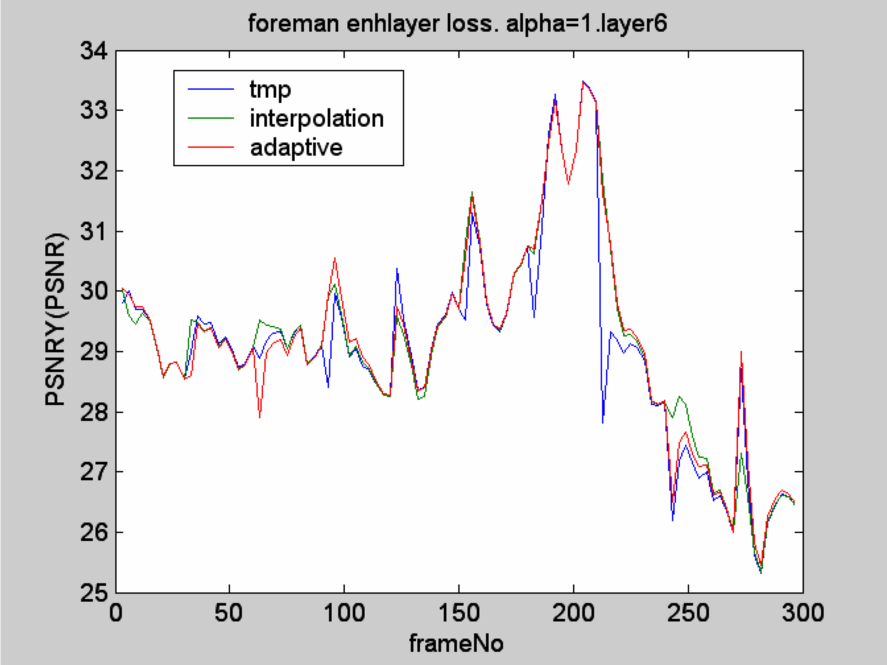
Blue line: using temporal concealment
Green line: using spatial interpolation


foreman enhlayer loss. alpha=1.layer6
PSNRY(PSNR)
46
44
42
40
38
36
34
32
30
28
26
0
50
100
150
200
250
300
frameNo
tmp
interpolation
adaptive

(a)


foreman enhlayer loss. alpha=1.layer6
PSNRY(PSNR)
38
36
34
32
30
28
26
0
50
100
150
200
250
300
frameNo
tmp
interpolation
adaptive

(b)



(c)

Fig. 4.22 Simulation results of concealment methods

(a) With 6 bit-planes in enhancement layer

(b) With 5 bit-planes in enhancement layer

(c) With 5 bit-planes in enhancement layer

Blue line: using temporal concealment

Green line: using spatial interpolation

Red line: adaptive method