Scene-Effect Detection and Insertion MPEG Encoding Scheme for Video Browsing and Error Concealment

Soo-Chang Pei, *Fellow, IEEE,* and Yu-Zuong Chou

***Abstract*—For an MPEG coding scheme, the encoder can do more than video compression. In this paper, a novel MPEG codec embedded with scene-effect information detection and insertion is proposed to provide more functionality at the decoder end. Based on the macroblock (MB) type of information that is generated simultaneously in the encoding process, a single-pass automatic scene-effect insertion MPEG coding scheme can be achieved. Using the USER_DATA of picture header, the video output bitstreams by our method still conform to the conventional MPEG decoding system. The proposed method provides a solution toward upgrading the existing MPEG codec with low complexity to accomplish at least two major advantages. Precise and effective video browsing resulting from the scene-effect extraction can significantly reduce the user's time to look up what they are interested in. For video transmission, the bitstreams containing scene-effect information can obtain better error concealment performance when scene changes are involved. Compared with the gain it achieves, the payout of our algorithm is very worthy with comparatively small efforts.**

***Index Terms*—Error concealment, MPEG, scene change, scene cut, video analysis, video transmission.**

I. Introduction

AS THE Internet becomes more and more popular and multimedia can be widely spread and applied, the video data booms to a tremendous amount and keeps on increasing rapidly. In addition to the problem of large memory storage, to browse and retrieve among this huge amount of video databases is also very difficult. MPEG codec nowadays is widely used to save the storage space and proven to be very effective. VCD and DVD players, which are based on the MPEG standard, have also become the common appliance in families. Since most video sequences are stored in MPEG compressed form, many browsing and retrieving algorithms based on MPEG compressed video have been proposed recently. Hence, to achieve this purpose, the scene change detection method, acting as the initial step to extract the video editing points also arouses a lot of interests. A number of researches have focused on scene-change detection methods [1]–[28] and many effective methods are developed. Although the scene-change detection mechanism at the decoder end works effectively, it still puts great efforts on the decoder structures. It violates the rule of MPEG coding scheme to keep the decoder as simple as possible and to concentrate the complex part at the encoder end. As a result, it is a preferred way to detect the interesting frame, such as scene changes, at the encoder side and insert the indications of the findings in the video bitstreams. Thus, video browsing can work by means of extracting the specific indications instead of complex analysis and collection of huge amount of overhead data. This idea is simple and intuitive but very attractive. However, it is not as easy as it seems. The difficulty lies in how to provide a novel MPEG encoding scheme with effective scene-effect detection, and how to design the format of overhead scene-effect indication to make the video bitstreams still conform to the conventional existing MPEG decoder. It is crucial to provide a novel solution with low complexity on the conventional operative MPEG encoder.

In addition to video storage, MPEG also makes video transmission possible because it significantly reduces large amount of transmission data. However, due to the property of predictive coding, the penalty of propagation error to video quality is quite huge. Even a single bit error can produce the corruption of a series of macroblocks (MBs) and these erroneous blocks will propagate to all GOP. To solve this problem, many error-resilience and error-concealment methods have been proposed [29]–[54]. Error concealment method uses the spatial or temporal redundancy to conceal the corrupted MBs. Although the temporal error concealment is proven effective, scene changes often occur and can do much harm for sudden loss of temporal redundancy. As a result, if the scene change is found, the error concealment should be adjusted correspondingly. For example, the bidirectional (B) frames should choose the reference P frame of the same video clip to conceal the erroneous block because the temporal redundancy is still valid. In other words, the scene-change information is also very helpful for error concealment algorithm. Therefore, besides easy video browsing and description, a more precise error concealment mechanism can be also obtained if the scene-effect indication is added in the headers of MPEG bitstreams.

To conclude from the above discussions, the benefit of inserting a scene-effect indication is clear. Nevertheless, how to minimize the efforts to MPEG encoding when the scene-effect detection and indication insertion is involved also becomes an important issue. In this paper, a novel MPEG encoding scheme with low-complexity scene-effect detection method and simultaneous insertion is proposed. The scene-effect de-

Manuscript received July 3, 2003; revised February 17, 2004. This work was supported by the National Science Council of Taiwan, R.O.C., under contract NSC 93-2219-E-002-004 and by the Ministry of Education under Contract 89-E-FA06-2-4. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Wen-Tsung Chang.

The authors are with the Department of Electrical Engineering National Taiwan University Taipei, Taiwan, R.O.C. (e-mail: pei@cc.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/TMM.2005.850961

1520-9210/$20.00 © 2005 IEEE

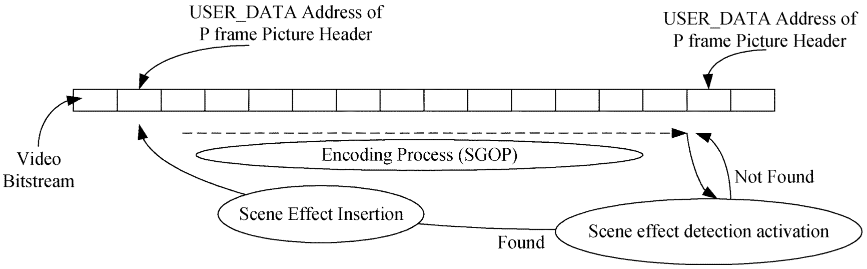
USER_DATA Address of P frame Picture Header
USER_DATA Address of P frame Picture Header
Video Bitstream
Encoding Process (SGOP)
Not Found
Scene Effect Insertion
Found
Scene effect detection activation

Fig. 1. Illustration of Picture-Header insertion method.

tection method is based on MB-type information generated in the encoding processes. In addition, only simple analysis on MB-type information can achieve effective scene-effect detection [23]. It is worthwhile to note that only one-pass processing is needed, that is to say, our method creates very little efforts on conventional MPEG encoding scheme. To switch the existing encoder to our proposed scheme, the cost is acceptably low.

In order to evaluate our method, the overheads for computations, memory and encoding data are taken into considerations and experimented. It is shown that the cost is so insignificant that this method is very practical to be embedded into conventional MPEG codec. Moreover, the corresponding decoding scheme is also proposed to provide easy browsing, effective video description and smarter error concealment mechanism. Experimental results show that our method works very well and effectively.

II. Proposed Method

It is obvious that to insert the scene-effect information into the bitstream can do a lot of good to the applications of the decoder side. Very fast and precise video browsing can be obtained and more effective error concealment to deal with scene effects can provide better quality under the noisy transmission environments. The benefits is evident, and the only question lies in how many efforts is needed to detect and insert the scene-effect information into the compressed video sequences. The most straightfoward and intuitive way is a two-phase method, in which the first pass is for scene-change detection for raw data and the second for the video compression. In the first pass, a scene-change detection method based on image difference or histogram comparison is used to decide which frame header needs to be inserted scene-change information and then the video sequence and the detection information are put into the conventional MPEG encoding scheme. This method works but the computation overhead is enormous, and moreover, the above encoding scheme brings out great efforts on the conventional encoding mechanism. In other words, the cost is tremendous, either from software design or hardware structure, to integrate the first phase detection algorithm into widely-used MPEG coding scheme.

Accordingly, in order to reduce the efforts to be integrated into conventional MPEG coding scheme, the embedded scene-change detection algorithm, which achieves based on the information collected during the conventional encoding process, is adopted. During this single phase process, video compression and scene-change detection perform simultaneously, and as a result, there is very low impact and integration cost for present MPEG coding systems. In this paper, an integration information insertion coding mechanism with a scene-change detection method based on the MB-type information is proposed. Our proposed method can be easily integrated into the present MPEG system with only MB-type information collection and little computation overhead.

There exists one another problem: how to insert the scene-effect information conforming to the existing MPEG format. It is mandatory that the conventional existing decoding tool should be capable of handling the video sequence with the picture header (PH) containing scene effects. The MPEG standard reserves the USER_DATA field to insert useful data for some specific purpose and this field can be used to indicate the scene effects. In MPEG, USER_DATA is defined by the users for their specific applications, which follows the USER_DATA_START_CODE that is the bit string 000 001B2 in hexadecimal. It identifies the beginning of user data and the user data continues until receipt of another start code. In this paper, the format of USER_DATA for scene effects will be discussed in the later paragraph.

In order to lower the storage and compuation load, we leverage our past research which addressed a scene-effect detection method based on MB-type information. This method exploit the computation of motion estimation at the encoder side and only simple analysis can detect the scene effect such as abrupt and gradual scene change, flashlight or caption, etc. This method benefits from low computation load and small storage space, and also provides a good solution toward a single phase scene-effect PH insertion coding scheme.

Single phase algorithm, although it reduces the integration complexity, faces one another difficulty. The encoding process has been finished befor the scene-effect detection completes. As a result, the USER_DATA in the PH has to be inserted in advance no matter whether this frame is a scene-change frame or not. In order to lower the overhead data for scene-effect identification, not all frames' PHs are inserted USER_DATA. Because the P frame is decoded in advance of the following B frames, it is practicable that the scene-effect indication is only appended to P and I frames. This concept is illustrated in Fig. 1 and the pseudocode of the proposed algorithm is shown in Fig. 2.

As to the format of USER_DATA to represent the scene effects, two kinds of information should be carried. One is for showing which frame in this SGOP (PBB…) is a scene-change frame, and the other is for indicating what kind of the scene effect is. For the above purpose, the USER_DATA for scene effects are shown in Tables I–III. The 2 bytes following the user_data_code is defined as the scene-effect message (SEM).

```
Pseudo Code of encoding algorithm
Frame=0;
While (Frame!=EndOfPictureStream)
{
   If (FrameType(FrameIndex)==Iframe)
      {
        If (SceneEffectIf==Yes)
           {
             InsertSceneEffectIntoGOPHeader();
             SceneEffectIf=No;
           }
        Coding(FrameIndex);
        FrameIndex++;
      }
  else if (FrameType(FrameIndex)==Pframe)
      {
        HeaderPos=RecordPictureHeader();
        Coding(FrameIndex);
        PframeMBTypeInformation=CollectMB(FrameIndex);
        FrameIndex++;
      }
else // means (FrameType(FrameIndex)==Bframe)
      {
        if   (FrameIndex==LastBFrameOfSGOP)
           {
              Coding(FrameIndex);
              BframeMBTypeInformation=CollectMB(FrameIndex);
              FrameIndex++;
SceneEffectDetection(BframeMBTypeInformation,PframeMBTypeInformation);
              SceneEffectInsertion(HeaderPos);
              SceneChangeIF=Yes;
           }
          else
          {
              Coding(FrameIndex);
              BframeMBTypeInformation=CollectMB(FrameIndex);
              FrameIndex++;
          }
      }
}
```

Fig. 2. Pseudocodes of scene-effect detection algorithm.

In order to discriminate SEM from other user defined data, the first four bits of the first byte is set 1111 to stand for SEM. The least significant four bits of the first bit is used for representing the scene-effect location in this SGOP. The last byte of SEM is designed to indicate the scene-effect types which are classified and listed in Table I.

TABLE I
FORMAT OF SCENE EFFECT USER_DATA IN THE PICTURE HEADER

USER_DATA_CODE	MSB 6 Bits	LSB 2 Bits
SEMUDI	LOSE	SETYPE

SEMUDI: Scene Effect Message User Data Identifier (1111)
LOSE: Location Of the Scene Effect, SETYPE: Scene effect type

TABLE II
FORMAT OF LOSE FIELD OF SEMUDI

LOSE (MSB 3bit)	Description
000	Starting point (not in this SGOP)
001	Scene effect starting point occurs at P frame
010	Scene effect starting point occurs at 1st B frame
011	Scene effect starting point occurs at 2nd B frame
100	Scene effect starting point occurs at 3rd B frame
101	Scene effect starting point occurs at 4th B frame
110	Scene effect starting point occurs at 5th B frame
111	Scene effect starting point occurs at 6th B frame
LOSE (LSB 3bit)	**Description**
000	Ending point (not in this SGOP)
001	Scene effect ending point occurs at P frame
010	Scene effect ending point occurs at 1st B frame
011	Scene effect ending point occurs at 2nd B frame
100	Scene effect ending point occurs at 3rd B frame
101	Scene effect ending point occurs at 4th B frame
110	Scene effect ending point occurs at 5th B frame
111	Scene effect ending point occurs at 6th B frame

TABLE III
FORMAT OF SETYPE FIELD OF SEMUDI

SETYPE	Description
0x00	No Effects
0x01	Abrupt Scene Change
0x10	Gradual Scene Change
0x11	Fast Panning

In our proposed scene-effect detection method, three types of scene effects, including abrupt scene change, gradual scene change, and fast panning sequence, can be detected. Based on our past research [23], the abrupt scene change can be found from the prediction pattern of MB-type information of B frames. As to fast panning and gradual scene change, significant number of Intra-coded MBs of P frame is used for indication and MB-type information of B frames can used to distinguish these two apart. With these three scene-effect indications, it is essential for fundamental video browsing. However, it is worthwhile to note that the scene-effect detection method can be extended to other effects certainly, such as caption or flashlight detection. It depends on what functionality and what key frame the system needs to provide. Compared with conventional MPEG coded, our proposed method with scene-effect indication requires three kinds of overheads: computational overhead, memory overhead and data overhead. Computational overhead means that the operations for scene-effect detection, PH position recording and insertion, and all other extra process compared with the conventional coding scheme. Memory overhead represents the extra memory space requirement to store MB-type information and the information relative to PH insertion. As to data overhead, it comes from the USER_DATA of PHs of P and I frames. Because these three parameters will determine the efficiency of our proposed method, we design a series of experiments to verify our coding scheme.

A. Computational Overhead

A 30-min video (i.e. 30 × 60 × 30 frames) is encoded by the conventional MPEG TM5 codec and our proposed method, respectively; and the computational time is recorded. The test

TABLE IV
COMPUTATION COMPARISONS OF CONVENTIONAL AND PROPOSED ENCODING METHOD

Video Sequence	Conventional	Proposed
Seq1 (30-min, 44 scene change)	6hr:30min:27sec	6hr:33min:49sec
Seq2 (12-min, 26 scene change)	2hr:11min:13sec	2hr:12min:45sec

TABLE V
OVERHEAD TIME OF CONVENTIONAL AND PROPOSED METHODS (CIF, secs)

Sequence without scene effects	A(900)	B(450)	C(1100)
Conventional method	302.53	144.34	390.45
Proposed method	303.11	144.79	391.10
Sequence with scene effects	**A(856)**	**B(600)**	**C(900)**
Conventional method	288.39	206.72	306.46
Proposed method	290.14	207.31	308.02

TABLE VI
MEMORY OVERHEAD EVALUATION

Type	Ordinary (bytes)	Extra memory (bytes)
Frame	352x240x1.5 (4:2:0)	22x15x1

platform is based on a PIII 866 PC with 256 M RAM. The experimental results are listed in Table IV and Table V. From these tables, only 0.1% computational overhead is required to handle the extra process of our scene-effect detection and PH insertion. It also implies that our proposed method is very practical in real-world video applications.

B. *Memory Overhead*

The extra requirement of memory space is shown in Table VI. For each MB, only one type information is required in our detection method. Thus, for a SGOP containing 1P2B frames (352 × 240), only 330 × 3 byte of memory space is required to be recorded. Compared with the memory space used to store the decoded reference image $(352 \times 240 \times 1.5 \times 3)$, the overhead memory is quite small and acceptable.

C. *Data Overhead*

Owing to the insertion of PH header, the video data stream will be elonged. Equation (1) shows the ratio of extra bits to the original data stream

$$R = \frac{\left(\frac{N}{M}\right) \times 16 + 16}{\frac{BitRate}{n}}$$

where N is the number of pictures in a GOP and M is the distance from a P frame to its previous P frame in a GOP. $BitRate$ is in the unit of bit per second and n represents the number of GOP in a second. For the case of a 4-Mbps bitstream and 30 frames per second ($N = 15$, $M = 3$, $n = 30/N = 2$)

$$R = 0.0048\%.$$

It is clear that the ratio is very insignificant.

D. *Easy Embedding*

Besides the advantage of lower overhead, the proposed method also benefits the architectural advantage. The scene-effect detection based on MB-type information is very easy to be integrated into conventional MPEG encoding scheme. Fig. 3 illustrates the simpleness of the integration. The scene-effect detection engine can operate totally separately without interfering the other functional blocks of the conventional MPEG encoder. The result of scene-effect detection will be used to modify the pre-pending user data field in the PH, as Fig. 1 shows. The change of MPEG encoding architecture is in very small scale and the video sequence will carry the scene-effect information for the decoder's further processing. The advantage of this kind of sequences will be discussed in the next section.

III. ADVANTAGES OF PROPOSED METHOD

As what we have mentioned, if the PH contains the scene-effect information, a corresponding MPEG decoder can provide more precise and efficient video browsing, an effective video description, and more accurate error concealment. We will describe these three advantages, respectively.

A. *Video Browsing*

A traditional way to browse video sequences is forwarding-and-searching. For MPEG sequences, the I-frame will be decoded as the key frame for browsing. Unfortunately, the number of I frames is large and makes the video browsing imprecise and time-consuming. Taking a 90-min movie video, for example, there exist $90 \times 60 \times 2 = 10\,800$ I-frames to be browsed. It is a tremendous large number. Actually, a 90-min movie may contain only 100 to 300 video clips, that is, 100 to 300 scene changes. If the scene changes can be found as the key frames for browsing, the user can save a lot of time and more precise video browsing is obtained. The browing decoder is quite simple and only the user_data of GOP header are required to be checked. A simulation to compare the efficiency of proposed algorithm and conventional browsing method have been tested and the results are shown in Table VII.

It is obvious from Table VII that our method provides a more effective and efficient video browsing, which significantly reduce the computation load and relieve the browser from the tedious searching task. However, it is noted that the actual video browsing is an interactive action and Table VII simply shows the improvement of browsing by the reduction of key frames.

B. *Accurate Error Concealment*

Recently, video transmission gets more and more important because the internet become common and widely spread. MPEG makes video transmission practical owing to the significant reduction of data amount but it also brings great impact. The characteristic of temporal prediction and variable length coding makes MPEG very sensitive to transmission error. Error propagation can do great harm to decoded video sequences. As a

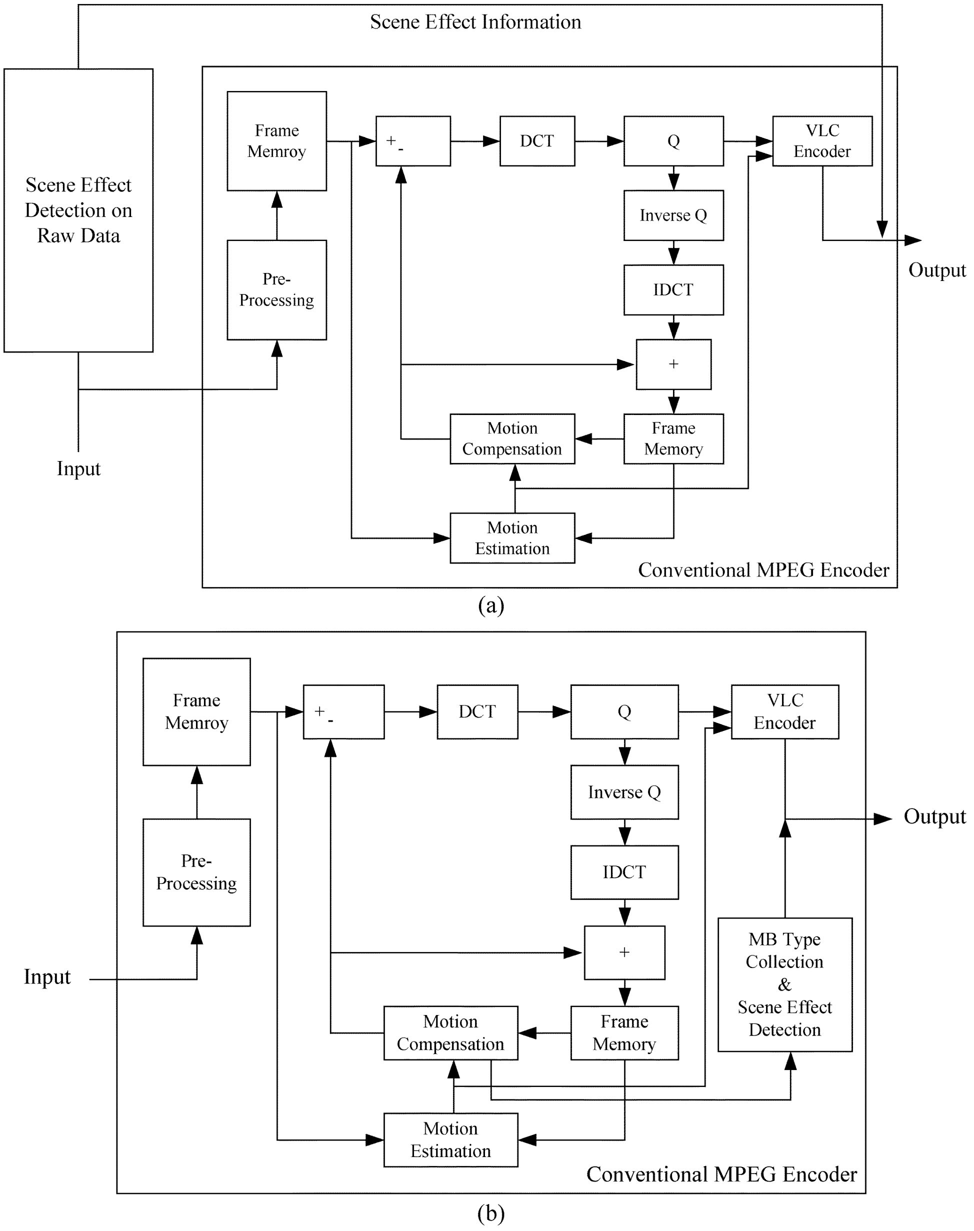

Fig. 3. How to embed our scene-effect detection engine into conventional MPEG encoder: (a) two-pass solution and (b) proposed single-pass solution.

TABLE VII
COMPARISON OF DECODING TIME FOR BROWSING

Video Sequence	Conventional	Proposed
Video 1, 90 min, 136 scene changes	270.34 secs	11.44 secs
Video 2, 76 min, 211 scene changes	245.22 secs	21.56 secs
Video 3, 66 min, 146 scene changes	221.78 secs	11.93 secs

result, error-resilient coding to resist error and error concealment to recover the damaged video stream nowadays arouse more and more attention. Regarding error concealment, temporal error concealment is widely used and proven superior to spatial error concealment in most of cases. However, the scene change may do great harm to temporal error concealment because it cause the temporal corelation entirely ineffective. As a result, a special treatment should be adopted to handle the error concealment under the circumstance of scene changes. Because the scene effect is known after the PH decoding, the choice of error concealment strategy can be made more precisely. If an abrupt scene change is indicated in the PH of this P frame, it implies that the temporal correlation is invalid for error concealment. As a result, once the error is encountered in the decoding process of the P frame, the spatial error concealment is adopted. For the B frames in this SGOP, the error concealment direction is toward the reference frame in the same shot. Figs. 4 and 7 illustrates this idea.

As to the dissolve sequences, the temporal error concealment is also ineffective because the temporal correlation is almost insignificant. A more precise error concealment based on the linear property can be applied to obtain better quality of decoded frames. When the SGOP is decoded as a dissolve SGOP, which is indicated by PH of the P frame, the corrupted MBs of B frames can be concealed by the interpolation of two reference frames.

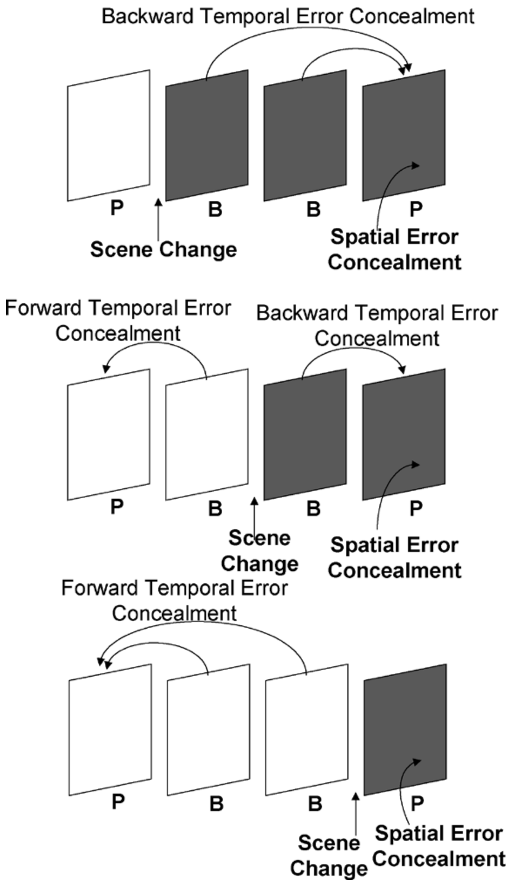
Backward Temporal Error Concealment
P
B
B
P
Scene Change
Spatial Error Concealment
Forward Temporal Error Concealment
Backward Temporal Error Concealment
P
B
B
P
Scene Change
Spatial Error Concealment
Forward Temporal Error Concealment
P
B
B
P
Scene Change
Spatial Error Concealment

Fig. 4. Solution to scene-change error concealment.

The weighting of the interpolation can be evaluated by the error-free decoded MB subgroup, MBSGOP [23]. For P frames, the linear property also can be applied to recover the error MBs. If there exists three consecutive reference frames in the dissolve sequence found by the PHs, the successful error-free co-located decoded MBs in the two of these three frames can be used to predict the corrupted MB of the other one. Interpolation method is applied for the error of the middle one and the extrapolation is for the latest reference frame. This method is shown in Figs. 5, 6, and 8and it is noted again the parameters of interpolation and extrapolation is calculated by the successful error-free decoded MBs.

A series of experiments are tested to verify our proposed scene-change dependent error concealment algorithm. Fig. 9 shows the comparison between conventional temporal error concealment and proposed method. From Fig. 9, the effectiveness of error concealment of our method is obviously better than the conventional one because the more accurate prediction reference frame is chosen. Tables VIII– X show the statistics of the PSNR performance comparison. It is found that about 3–5 dB is gained by adopting our method. Fig. 10 shows similar performance for the proposed algorithm, and about 3–5 dB in PSNR is gained, corresponding to different bit error rates (BERs). In all the experiments, it is worthwhile to emphasize that only the GOP with a scene change is taken into consideration and PSNR calculation. It is because the error resulting from the poorly concealed block due to the scene change will not propagate to the next GOP.

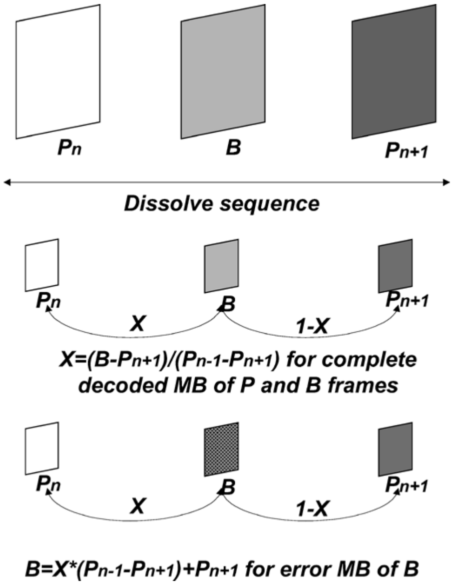
Pn
B
Pn+1
Dissolve sequence
Pn
X
B
1-X
Pn+1
X=(B-Pn+1)/(Pn-1-Pn+1) for complete decoded MB of P and B frames
Pn
X
B
1-X
Pn+1
B=X*(Pn-1-Pn+1)+Pn+1 for error MB of B

Fig. 5. Interpolated temporal error concealment method to handle B frames in the dissolve sequence.

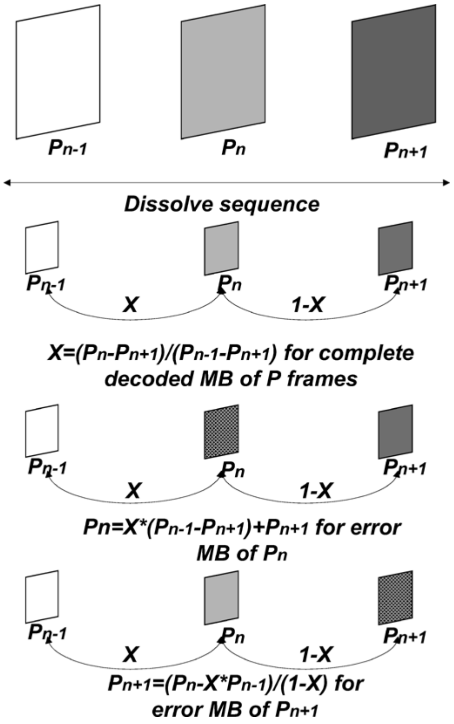
Pn-1
Pn
Pn+1
Dissolve sequence
Pn-1
X
Pn
1-X
Pn+1
X=(Pn-Pn+1)/(Pn-1-Pn+1) for complete decoded MB of P frames
Pn-1
X
Pn
1-X
Pn+1
Pn=X*(Pn-1-Pn+1)+Pn+1 for error MB of Pn
Pn-1
X
Pn
1-X
Pn+1
Pn+1=(Pn-X*Pn-1)/(1-X) for error MB of Pn+1

Fig. 6. Illustration of interpolated and extrapolated temporal error concealment to handle P frames in the dissolve sequence.

IV. Conclusions

In this paper, a novel idea and the corresponding implementation to insert the scene-effect information into the video bitstream is proposed. In order to lower the impact on the existing MPEG coding scheme, we presented a single-pass MPEG codec with low-complexity scene-effect detection. The method based on MB-type information from our past studies benefits from the simultaneous data generation and simple analysis of low computational complexity. Moreover, USER_DATA of a PH is applied to represent the scene effect

```
AbruptSceneChangeIf=No;
AbruptSceneChangeInGOP=No;
Frame=0;
While (Frame!=EndOfPictureStream)
{
  If (FrameType(FrameIndex)==Pframe)
    {
      If (ExtractPictureHeader(FrameIndex)==AbruptSceneChange)
        {
          AbruptSceneChangeInGOP=Yes;
          AbruptSceneChangeIf=Yes;
          SceneChangePos=ExtractSceneChangePos();
          Decode(FrameIndex, SpatialErrorConcealment,ForwardPrediction);
          ErrorMBPos=RecordErrorMB(FrameIndex);
          //ErrorConcealMethod=SpatialErrorConcealment;
        }
      else
        {
          AbruptSceneChangeIf=No;
        Decode(FrameIndex, TemporalErrorConcealment,ForwardPrediction);
        }
      FrameIndex++;
    }

  else if (FrameType(FrameIndex)==Bframe)
    {
      if (AbruptSceneChangeIf==Yes)
      {
       If ( IndexOfSGOP(FrameIndex)<=SceneChangePos)
           {
            Decode(FrameIndex, TemporalErrorConcealment,ForwardPrediction);
           }
       else
         {
            Decode(FrameIndex, SpatialErrorConcealment,ForwardPrediction);
         }
      else
      {
            Decode(FrameIndex, SpatialErrorConcealment,NULL);
      }
       FrameIndex++;
     }
    }
else // means (FrameType(FrameIndex)==Iframe)
    {
      if   (AbruptSceneChangeInGOP==Yes)
        {
          ReverseCompensation(ReferenceFrameOfPreviousPframe,ErrorMBPos);
          AbruptSceneChangeInGOP =No;

        }
       Decode(FrameIndex);
       FrameIndex++;
    }
}
```

Fig. 7. Pseudocodes of abrupt scene-change error concealment.

```
GradualSceneChangeIf=No;
GradualSceneChangeInGOP=No;
Frame=0;
While (Frame!=EndOfPictureStream)
{
  If (FrameType(FrameIndex)==Pframe)
    {
      If (ExtractPictureHeader(FrameIndex)==GradualSceneChange)
        {
         If (GradualSceneChangeIf ==Yes)
         {
          Decode(FrameIndex, TemporalErrorConcealment,ExpolationPrediction);
          SceneChangePos=ExtractSceneChangePos();
          ErrorMBPos&=RecordErrorMB(FrameIndex);
         }
         else
        {
         GradualSceneChangeInGOP=Yes;
         GradualSceneChangeIf=Yes;
         SceneChangePos=ExtractSceneChangePos();
         Decode(FrameIndex, SpatialErrorConcealment,ForwardPrediction);
         ErrorMBPos=RecordErrorMB(FrameIndex);
        }
      else
        {
         GradualSceneChangeIf=No;
         Decode(FrameIndex, TemporalErrorConcealment,ForwardPrediction);
        }
      FrameIndex++;
    }

  else if (FrameType(FrameIndex)==Bframe)
    {
      if (AbruptSceneChangeIf==Yes)
      {
       if ( InsideDissolve(IndexOfSGOP(FrameIndex), SceneChangePos)==Yes)
         Decode(FrameIndex, TemporalErrorConcealment,InterpolationPrediction);
       else
         Decode(FrameIndex, SpatialErrorConcealment,ForwardPrediction);
      }
    else
      Decode(FrameIndex, SpatialErrorConcealment,NULL);

       FrameIndex++;
     }
    }
else // means (FrameType(FrameIndex)==Iframe)
    {
      if   (AbruptSceneChangeInGOP==Yes)
        {
          ReverseCompensation(ReferenceFrameOfPreviousPframe,ErrorMBPos);
          AbruptSceneChangeInGOP =No;

        }
       Decode(FrameIndex);
       FrameIndex++;
    }
}
```

Fig. 8. Pseudocodes of dissolve error concealment.

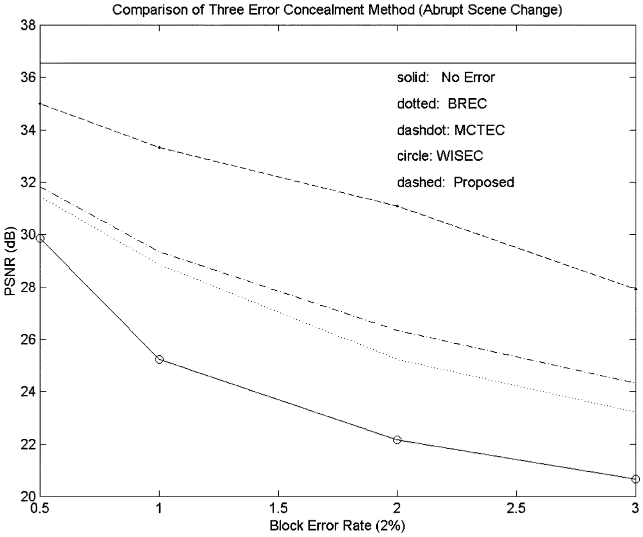

Fig. 9. Plot of comparisons of four methods (abrupt scene change).

and keeps the consequent video stream conformable to conventional MPEG codec. Integrated with the above two ideas, a single-pass automatic scene-effect insertion MPEG codec is proposed and experimented. Based on our evaluation, less than 1% overhead on the computational complexity, memory space, and extra data is required to achieve our proposed codec. This is very attractive for conventional MPEG codecs to adopt. At the decoder end, the video bitstream with the scene-effect information can provide some useful functions. First, because the scene effects are known in advance of picture decoding, the video browsing can precisely focus on those containing specific scene effects, such as scene changes. A lot of browsing time and computation efforts can be saved by this design. Besides, the scene-effect information can be applied to help to describe video clips. For a higher level of video retrieval, scene-effect information can cooperate with some other video descriptors, such as color histogram and edge contour, to achieve the purpose of video description. For example, a TV commercial clip would show frequent scene changes and short video clip span and in the basketball sequences, frequent and steady appearance of fast panning effects could be found. In other words, to embed the scene-effect information into video headers may also provide a useful video descriptor in the decoder end. For video transmission, we also examine a scene-effect dependent

TABLE VIII
SIMULATION OF DIFFERENT LENGTH OF DISSOLVE SEQUENCE (4 Mbps, $\mathrm{BER} = 0.5\%$, $\text{Total Block error rate} = 5\%$).

Location	No Error (dB)	BREC (dB)	MCTEC (dB)	WISEC (dB)	Proposed (dB)
Seq A, 24 scene changes	35.55	31.77	31.43	29.75	33.45
Seq B, 32 scene changes	35.13	31.16	31.10	29.10	33.31
Seq C, 28 scene changes	35.83	31.84	31.53	29.58	33.79

Where M0: Error free decoding, M1: BREC, M2: MCTEC, M3: WISEC, M4: Proposed

TABLE IX
SIMULATION OF DIFFERENT LENGTH OF DISSOLVE SEQUENCE ($\mathrm{BER} = 1\%$, $\text{Total Block error rate} = 10\%$)

Location	No Error (dB)	BREC (dB)	MCTEC (dB)	WISEC (dB)	Proposed (dB)
Seq A, 24 scene changes	35.55	29.91	29.65	26.87	32.72
Seq B, 32 scene changes	35.13	29.33	28.97	26.05	32.19
Seq C, 28 scene changes	35.83	29.96	29.51	26.76	32.64

Where M0: Error free decoding, M1: BREC, M2: MCTEC, M3: WISEC, M4: Proposed

TABLE X
SIMULATION OF DIFFERENT LENGTH OF DISSOLVE SEQUENCE ($\mathrm{BER} = 2\%$, $\text{Total Block error rate} = 20\%$)

Location	No Error (dB)	BREC (dB)	MCTEC (dB)	WISEC (dB)	Proposed (dB)
Seq A, 24 scene changes	35.55	26.53	26.21	22.40	29.99
Seq B, 32 scene changes	35.13	25.86	25.39	22.25	29.64
Seq C, 28 scene changes	35.83	26.86	26.31	22.67	29.96

Where M0: Error free decoding, M1: BREC, M2: MCTEC, M3: WISEC, M4: Proposed

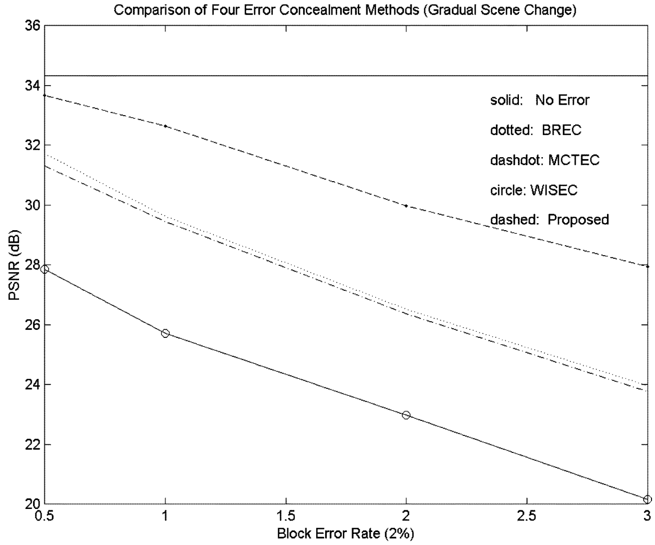

Fig. 10. Plot of comparisons of four methods (gradual scene change).

error concealment to handle the inaccurate frame prediction when scene change is involved. Based on the experiments, it gains more than 3 dB in PSNR by this algorithm.

REFERENCES

[1] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full motion search for object appearances," *Visual Database Systems, II*, pp. 119–133, 1991.

[2] F. Arman, R. Depommier, A. Hsu, and M.-Y. Chiu, "Content-based browsing of video sequences," in *Proc. ACM Multimedia'94*, Aug. 1994, pp. 97–103.

[3] F. Arman, A. Hsu, and M. Y. Chiu, "Feature management for large video databases," *Storage and Retrieval for Image and Video Databases*, vol. SPIE-1908, pp. 2–12, 1993.

[4] Y. Nakajuma, "A video browsing using fast scene cut detection for an efficient networked video database access," *IEICE Trans. Inform. Syst.*, vol. E77-D, no. 12, pp. 1355–1364, Dec. 1994.

[5] F. Arman, A. Hsu, and M. Y. Chiu, "Image processing on compressed data for large video databases," in *Proc. 1st ACM Int. Conf. Multimedia*, Aug. 1993, pp. 267–272.

[6] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Syst.*, vol. 1, pp. 10–28, Jul. 1993.

[7] H. J. Zhang, C. Y. Low, and S. W. Smoliar, "Video parsing and browsing using compressed data," *Multimedia Tools Applicat.*, vol. 1, no. 1, pp. 89–111, Mar. 1995.

[8] B.-L. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Trans. Circuits Syst. Video Technol*, vol. 5, no. 8, pp. 533–544, Dec. 1995.

[9] K. Otsuji and Y. Tonomura, "Projection detecting filter for video cut detection," in *Proc. 1st ACM Int. Conf. Multimedia*, Aug. 1993, pp. 251–257.

[10] S. F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE J. Select. Areas Commun.*, vol. 13, no. 1, pp. 1–11, Jan. 1995.

[11] N. V. Patel and I. K. Sethi, "Compressed video processing for cut detection," *Proc. Inst. Elect. Eng.—Vis., Image, Signal Process.*, vol. 143, no. 5, pp. 315–323, Oct. 1996.

[12] ——, "Video shot detection and characterization for video databases," *Pattern Recognit.*, vol. 30, no. 4, pp. 583–592, Mar. 1997.

[13] B.-L. Yeo and B. Liu, "Visual content highlighting via automatic extraction of embedded captions on MPEG compressed video," in *Proc. SPIE, Digital Video Compression: Algorithms Technology.*, vol. 2668, 1996, pp. 47–58.

[14] J. Meng, Y. Juan, and S.-F. Chang, "Scene change detection in a MPEG compressed video sequence," in *Proc. SPIE, Digital Video Compression: Algorithms Technol.*, vol. 2419, 1995, pp. 14–25.

[15] Q. Wei, H. Zhang, and Y. Zhong, "A robust approach to video segmentation using compressed data," in *Proc. SPIE, Storage and Retrieval for Still Image and Video Database*, vol. 3022, 1997, pp. 448–456.

[16] L. A. Rowe, S. Smoot, K. Patel, B. Smith, K. Gong, E. Hung, D. Banks, S. T.-S. Fung, D. Brown, and D. Wallach. (1995, Aug.) Berkeley MPEG Tools. Version. 1.0, Rel. 2. [Online] Available: ftp://mm-ftp.cs.berkeley.edu/pub/multimedia/mpeg/ bmt1r1.tar.gz

[17] A. M. Alattar, "Wipe scene change detector for use with video compression algorithms and MPEG-7," *IEEE Trans. Consumer Electron.*, vol. 44, no. 1, pp. 43–51, Feb. 1998.

[18] W. A. C. Fernando, C. N. Canagarajah, and D. R. Bull, "Wipe scene change detection in video sequence," *Proc. ICASP*, pp. 294–298, 1999.

[19] M. Wu, W. Wolf, and B. Liu, "An algorithm for wipe detection," *Proc. ICIP*, pp. 893–897, 1998.

[20] S. J. Golin, "New metric to detect wipes and other gradual transitions in video," *SPIE*, vol. 3653, pp. 1464–1474, 1998.

[21] M. Sugano, Y. Nakajima, H. Yanagihara, and A. Yoneyama, "A fast scene change detection on MPEG coding parameter domain," *Proc. ICIP*, pp. 888–892, 1998.

[22] C. W. Ngo, T. C. Pong, and R. T. Chin, "Camera break detection by partitioning of 2D spatio-temporal images in MPEG domain," in *Multimedia Computing and Systems, 1999: IEEE Int. Conf., Vol. 1*, 1999, pp. 750–755.

[23] S.-C. Pei and Y.-Z. Chou, "Efficient MPEG compressed video analysis using macroblock type information," *IEEE Trans. Multimedia*, vol. 1, no. 4, pp. 321–333, Dec. 1999.

[24] N. Gamaz, X. Huang, and S. Panchanathan, "Scene change detection in MPEG domain," in *Proc. IEEE Southwest Symp. Image Analysis and Interpretation*, 1998, pp. 12–17.

[25] U. Gargi, R. Kasturi, and S. H. Strayer, "Performance chararterization of video-shot-change detection methods," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 10, no. 1, pp. 1–13, Feb. 2000.

[26] K. Tse, J. Wei, and S. Pancanathan, "A scene change detection algorithm for MPEG compressed video sequences," in *Proc. Canadian Conf. Electrical and Computer Engineering*, vol. 2, 1995, pp. 827–830.

[27] J. Meng, Y. Juan, and S.-F. Chang, "Scene change detection in a MPEG compressed video sequence," in *IS&T/SPIE Symp. Proc.*, vol. 2419, Feb. 1995.

[28] T. Shin, J.-G. Kim, H. Lee, and J. Kim, "Hierarchical scene change detection in an Mpeg-2 compressed video sequence," in *Proc. 1998 IEEE Int. Symp. Circuits and Systems*, vol. 4, pp. 253–256.

[29] G.-S. Yu, M. M.-K. Liu, and M. W. Marcellin, "POCS-based error concealment for packet video using multiframe overlap information," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 4, pp. 422–434, Aug. 1998.

[30] Y. J. Chung, J. W. Kim, and C.-C. J. Kuo, "Real-time streaming video with adaptive bandwidth control and DCT-based error concealment," *IEEE Trans. Circuits Syst. II*, vol. 46, no. 7, pp. 951–956, Jul. 1999.

[31] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projections onto convex sets," *IEEE Trans. Image Process.*, vol. 4, no. 4, pp. 470–477, Apr. 1995.

[32] V. Parthasarathy, J. W. Modestino, and K. S. Vastola, "Design of a transport coding scheme for high-quality video over ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 2, pp. 358–376, Apr. 1997.

[33] W. Keck, "A method for robust decoding of erroneous MPEG-2 video bitstreams," *IEEE Trans. Consumer Electron.*, vol. 42, no. 3, pp. 411–421, Aug. 1996.

[34] J. Feng, K.-T. Lo, and H. Mehrpour, "Error concealment for MPEG video transmissions," *IEEE Trans. Consumer Electron.*, vol. 43, no. 2, pp. 183–187, May 1997.

[35] W. Zhu, Y. Wang, and Q.-F. Zhu, "Second-order derivative-based smoothness measure for error concealment in DCT-based codecs," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 6, pp. 713–718, Oct. 1998.

[36] J. F. Arnold, M. R. Frater, and J. Zhang, "Error resilience in the MPEG-2 video coding standard for cell based networks—a review," *Signal Processing: Image Commun.*, vol. 14, pp. 607–633, 1999.

[37] S. Kaiser and K. Fazel, "Comparison of error concealment techniques for an MPEG-2 video decoder in terrestrial TV-broadcasting," *Signal Processing: Image Commun.*, vol. 14, pp. 655–676, 1999.

[38] H.-C. Shyu and J.-J. Leou, "Detection and concealment of transmission errors in MPEG-2 images—a genetic algorithm approach," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 6, pp. 937–948, Sep. 1999.

[39] J. M. Boyce, "Packet loss resilient transmission of MPEG video over the internet," *Signal Processing: Image Commun.*, vol. 15, pp. 7–24, 1999.

[40] D. W. Redmill, D. R. Bull, J. T. Chung-How, and N. G. Kingsbury, "Error-resilient image and video coding for wireless communication systems," *Electron. Commun. Eng. J.*, Aug. 1998.

[41] M. Bystrom, V. Parthasarathy, and J. W. Modestino, "Hybrid error concealment schemes for broadcast video transmission over ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 6, pp. 868–881, Sep. 1999.

[42] P.-J. Lee and M.-J. Chen, "Robust error concealment algorithm for video decoder," *IEEE Trans. Consumer Electron.*, vol. 45, no. 3, pp. 851–859, Aug. 1999.

[43] M.-C. Hong, H. Schwab, L. P. Kondi, and A. K. Katsaggelos, "Error concealment algorithms for compressed video," *Signal Processing: Image Commun.*, vol. 14, pp. 473–492, 1999.

[44] J.-W. Suh and Y.-S. Ho, "Error concealment based on directional interpolation," *IEEE Trans. Consumer Electron.*, vol. 43, no. 3, pp. 295–302, Aug. 1997.

[45] M. Al-Mualla, N. Canagarajah, and D. R. Bull, "Temporal error concealment using motion field interpolation," *Electron. Lett.*, vol. 35, no. 3, pp. 215–217, Feb. 1999.

[46] W.-M. Lam and A. R. Reibman, "An error concealment algorithm for images subject to channel errors," *IEEE Trans. Image Process.*, vol. 4, no. 5, pp. 533–542, May 1995.

[47] W. Kwok and H. Sun, "Multi-directional interpolation for spatial error concealment," *IEEE Trans. Consumer Electron.*, vol. 39, no. 3, pp. 455–460, Aug. 1993.

[48] W. Zeng and B. Liu, "Geometric-structure-based error concealment with novel applications in block-based low-bit-rate coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 4, pp. 648–665, Jun. 1999.

[49] J.-T. Wang and P.-C. Chang, "Error-propagation prevention technique for real-time video transmission over ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 3, pp. 513–523, Apr. 1999.

[50] M.-J. Chen, L.-G. Chen, and R.-M. Weng, "Error concealment of lost motion vectors with overlapped motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 3, pp. 560–563, Jun. 1997.

[51] V. Parthasarathy, J. W. Modestino, and K. S. Vastola, "Reliable transmission of high-quality video over ATM networks," *IEEE Trans. Image Process.*, vol. 8, no. 3, pp. 361–374, Mar. 1999.

[52] Z. Wang, Y. Yu, and D. Zhang, "Best neighborhood matching: An information loss restoration technique for block-based image coding systems," *IEEE Trans. Image Process.*, vol. 7, no. 7, pp. 1056–1061, Jul. 1998.

[53] X. Lee, Y.-Q. Zhang, and A. Leon-Garcia, "Information loss recovery for block-based image coding techniques-a fussy logic approach," *IEEE Trans. Image Proces.*, vol. 4, no. 3, pp. 259–273, Mar. 1995.

[54] J. W. Park and S. U. Lee, "Recovery of corrupted image data based on the NURBS interpolation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 7, pp. 1003–1008, Oct 1999.

Soo-Chang Pei (SM'89–F'00) was born in Soo-Auo, Taiwan, R.O.C., in 1949. He received the B.S.E.E. degree from National Taiwan University (NTU), Taipei, in 1970 and the M.S.E.E. and Ph.D. degrees from the University of California, Santa Barbara (UCSB), in 1972 and 1975, respectively.

He was an Engineering Officer in the Chinese Navy Shipyard from 1970 to 1971. From 1971 to 1975, he was a Research Assistant at UCSB. He was the Professor and Chairman in the Electrical Engineering Department, Tatung Institute of Technology and NTU from 1981 to 1983 and 1995 to 1998, respectively. Presently, he is the Professor in the Electrical Engineering Department at NTU. His research interests include digital signal processing, image processing, optical information processing, and laser holography.

Dr. Pei received the National Sun Yet-Sen Academic Achievement Award in Engineering in 1984, the Distinguished Research Award from the National Science Council, R.O.C., from the Chinese Institute of Electrical Engineering in 1998, the Academic Achievement Award in Engineering from the Ministry of Education in 1998, the Pan Wen-Yuan Distinguished Research Award in 2002; and the National Chair Professor Award from Ministry of Education in 2002. He has been President of the Chinese Image Processing and Pattern Recognition Society in Taiwan from 1996-1998, and is a member of Eta Kappa Nu and the Optical Society of America. He became an IEEE Fellow in 2000 for contributions to the development of digital eigenfilter design, color image coding, and signal compression, and to electrical engineering education in Taiwan.

Yu-Zuong Chou was born in Tainan, Taiwan, R.O.C. He received the B.S. degree from the National Tsing Hua University, Hsinchu, Taiwan, in 1994, and the M.S. and Ph.D. degrees from the National Taiwan University (NTU), Taipei, Taiwan, in 1996 and 2002, respectively, both in electrical engineering.

He is currently an Engineer with Realtek Semiconductor Corporation, Hsinchu. His research interests include video compression, image processing, multimedia application, and SOC design.