Error Detection and Correction in H.263 coded video over wireless network

Ekram Khan, Hiroshi Gunji, S. Lehmann and Mohammed Ghanbari
Audio and Video Networking Laboratory
Department of Electronic Systems Engineering
University of Essex, Colchester CO4 3SQ
UK

ABSTRACT:

Due to the use of variable length code (VLC), a single bit error in the DCT-based coded bitstream such as H.263 may propagate up to the end of the group of block (GOB) or slice. In this paper, we propose an iterative error detection and correction algorithm for the slice mode of the H.263 bitstreams. The visibly erroneous macroblocks (MB) in the decoded frames are detected by checking a set of error detection conditions derived from the redundant information (such as neighbouring MB and inner DCT block similarity measure) inherent within the frame. An iterative re-decoding based correction algorithm is then applied to the erroneous slices. The proposed technique limits the error into a few MBs only, which can easily be concealed by any error concealment technique. The simulation results demonstrate that our scheme can recover the corrupted frames under the bit error rates up to 0.5% over binary symmetric channel (BSC), and improve the concealed picture quality by up to 8 dB over the conventional methods.

1. INTRODUCTION

Most of the standard video codecs like H.263 are based on motion compensation – discrete cosine transform (MC-DCT) coding scheme, which use the variable length codes (VLC's), such as Huffman for further compression. The use of VLC in the erroneous compressed data would not allow even the non-corrupted parts to be correctly decoded until after the synchronisation point, i.e. start of the following group of blocks (GOB) or slice. Moreover, due to out-of-synchronisation between the encoder and decoder states, the error may also propagate into the temporal domain. Due to these reasons, the emerging video coding techniques include provisions for error resilience particularly in H.263 [1-3] and MPEG [4]. To limit the effect of error propagation and hence to improve the video quality against the transmission error, several techniques have been proposed in the literature. These can be grouped into four categories: feedback channel or retransmission approach [5-7], forward error correction or channel coding approach [8], error resilience approach [9-10] and error detection and correction approach [11-13]. However, these techniques suffer from some of the basic problems. For example, the use of feedback channel introduces additional transmission delay and complexity. The channel coding [8] and use of markers [11-12] increases the data rate and may not be suitable for low bit rate applications. The error concealment methods, although don't increase the transmission bandwidth, but yield poor performance under high channel error rate. The error detection and correction at image level [11-12] is achieved by the frequent use of restart markers. The reversible variable length codes (RVLC) [13] are best suited to detect the errors when bits from erroneous bit onward are not decodable or when invalid codes occur. However, in general all the erroneous bits may not result in invalid codes and bitstreams may be decoded (although erroneously) even in the presence of errors. In addition to all these problems, some errors in the bitstream have almost no or very little impact on the picture quality and it would be waste of resources to correct them.

In this paper, we propose a posteriori approach aimed at improving the perceptual video quality after erroneous H.263 coded bitstreams are received. Our error detection algorithm exploits the inherent redundancies within and outside a macroblock to detect and locate the erroneously decoded macroblocks. The redundancies are measured in terms of a set of parameters (based on MB and inner DCT blocks similarities). After the error detection, an iterative re-decoding based correction algorithm is applied to the erroneous slices. The proposed error detection and correction scheme is applied iteratively until no further corrupted macroblock is detected. Finally spatial (for I frame) and temporal (for P frames) concealment techniques are used to conceal the corrupted macroblocks.

This paper is organised as follows. The modified H.263 video coder with bitstream syntax modification is described in Section 2. The proposed error detection and correction scheme is discussed in Section 3. The simulation results are included in Section 4, followed by concluding remarks in Section 5.

2. MODIFIED H.263 VDEO CODER

The standard video coding systems like H.263 [3] and MPEG-4 [4] envisage various methods to improve their resilience towards channel errors. For example the Annex-K of H.263+ [1] supports the slice-structured mode where all macroblocks of one slice can be decoded independent of the content of other slices by preventing the prediction of motion vectors to cross the slice boundaries. There is however, a need for extra information to decode a slice, because information conveyed in the picture header is not repeated in the slice header. In this work, we have used slice-structured mode with slight modification in the bitstream syntax to be discussed in the next section. Our proposed error detection and correction technique is incorporated in the modified decoder shown in Fig. 1.

Since in the slice structured mode of H.263 bitstream, the quantization information and motion vectors are encoded differentially, if one or more macroblocks in the middle of the slice are omitted, then it is impossible to decode the quantization information and motion vectors of the following macroblocks. In order to avoid this problem, the bitstream syntax is slightly modified from the conventional slice structure mode. The modifications are made only in the slice and macroblock layer, whereas picture and block layers remain unchanged [1]. In the slice layer, SSTUF is modified and LQUANT (last quantizer information) and LMVV (last motion vector value) are added. In the macroblock layer MVD (motion vector difference) prediction is modified. Due to these modifications, for QCIF at 64 kbit/sec the overall bit rate increases by 1.2%.

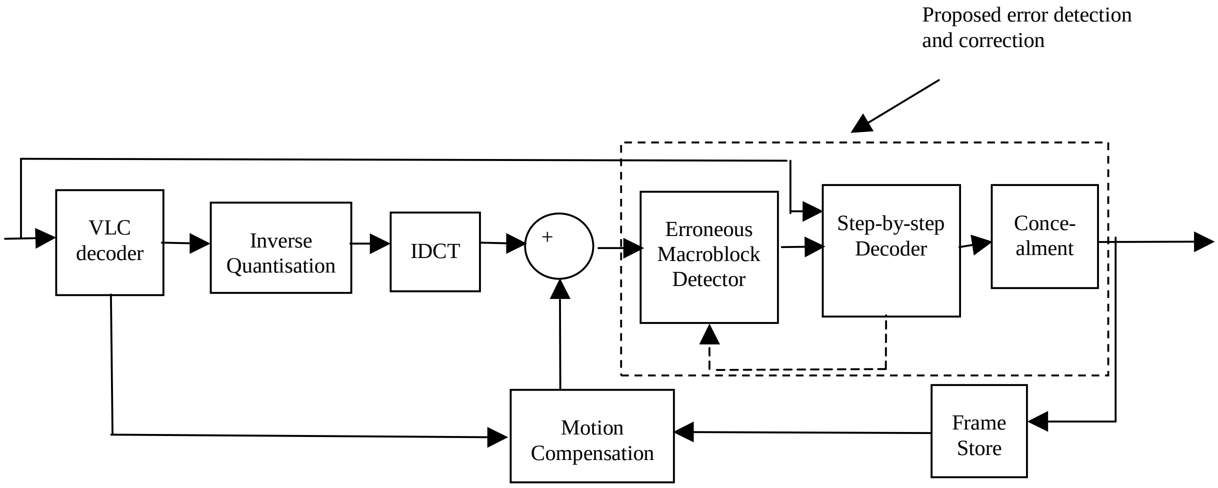


Fig. 1: Modification to the H.263 decoder for error detection, correction and concealment.

3. ERROR DETECTION AND CORRECTION

3.1 Error Detection:

In the presence of transmission errors, a decoded frame may contain three types of macroblocks, (1) non-decodable macroblock (fatal error), (2) correctly decodable macroblock and (3) decodable but erroneous macroblock. In our scheme, the first type of macroblock is detected during decoding a slice but the other two types are detected after the decoding.

While decoding a corrupted bitstream, the H.263 decoder may detect the slice as erroneous if any of the following conditions occurs: (i) Invalid VLC code is detected. (ii) Quantizer information goes out of range. (iii) Invalid INTRA DC code is detected. (iv) Escaped TCOEF with level 0 is detected. (v) Coefficient overrun occurred. (vi) A motion vector refers out of picture. (vii) The number of bits used in decoding of the slice is different from that in the slice. (viii) The quantizer information in the last macroblock is different from LQUANT. (ix) The motion vector of the last coded MB is different from LMVV. If any of the conditions (i) – (vi) occurs, the decoder will detect a fatal error, and the MBs beyond that point are non-decodable and all pixels of these MBs are replaced with some gray values (say 128). However, if in any slice, condition (vii) – (ix) occurred, the decoder will decode that slice erroneously, but it can not determine the location of the error.

After a frame is decoded, the visibly erroneous macroblocks will be detected using redundant information inherent in the neighbouring macroblocks. Since a channel error most often affects either 8x8 pixels DCT blocks or 16x16 pixel macroblocks (containing four luminance and one each of the chrominance DCT blocks), in the proposed error detection scheme, the emphasis is given on detecting erroneous macroblocks by exploiting either the characteristics of the neighbouring macroblocks or DCT blocks within a macroblock. Furthermore, it is very likely that a macroblock following an erroneous one is itself affected by the error, the main challenge is to detect the first erroneous macroblock in a slice. In order to detect corrupted MBs, the following similarity measures are exploited:

a) Macroblock (MB) Characteristics
   1) Macroblock boundary
   2) Macroblock mean pixel value
b) Inner DCT block characteristics
   1) Inner boundary
   2) Individual DCT block boundaries
   3) DCT block’s mean pixel value

For each of these similarity measures, a parameter is calculated and compared with either an absolute or a relative threshold or both. For a macroblock under consideration, if the parameter has a value greater than an absolute threshold, the macroblock is considered as erroneous. However, if the value of the parameter is greater than a relative threshold, it is considered as a candidate of being erroneous (subject to other criteria to be fulfilled).

3.2 Error Correction:

Assuming that the decoder knows the location of the first erroneous macroblock (if any) in the slice, the step-by-step decoding shown in Fig. 2 works as follows. The part of the slice preceding the first erroneous macroblock is conventionally decoded and a pointer is initialised at the second bit of the first erroneous macroblock, i.e. skipping the first bit. The rest of the bitstream from that point is checked to see whether it is decodable or not. If the bitstream is non decodable, the pointer is incremented by one bit. This process is continued until correctly decodable bitstream is found and decoded. We call this part of the slice as step-by-step decoded part. If the number of decoded macroblocks in the slice is less than the actual number of macroblocks (which is known), the step-by-step decoded part is right aligned and any missing macroblocks are replaced by gray values (say 128). This process is repeated for each erroneous slice (found while decoding the frame) and thus the decoded frame is generated. Although it is expected that the step-by-step decodable part of the slice is obtained when pointer is pointing to the bit of a macroblock boundary, however, in practice it is possible that rest of the slice is decoded from the point different from MB boundary. It is also likely that decodable bitstream for the rest of the slice also contains some erroneous bits. In such cases the decoded frame may still contain erroneous macroblocks. For this reason the error detection and step-by-step decoding is repeatedly performed on the decoded frame until the error detector does not detect any more corrupted MBs, except the gray valued MBs, which are concealed later. A pseudo code for the proposed error detection and correction algorithm is given in Fig. 3.

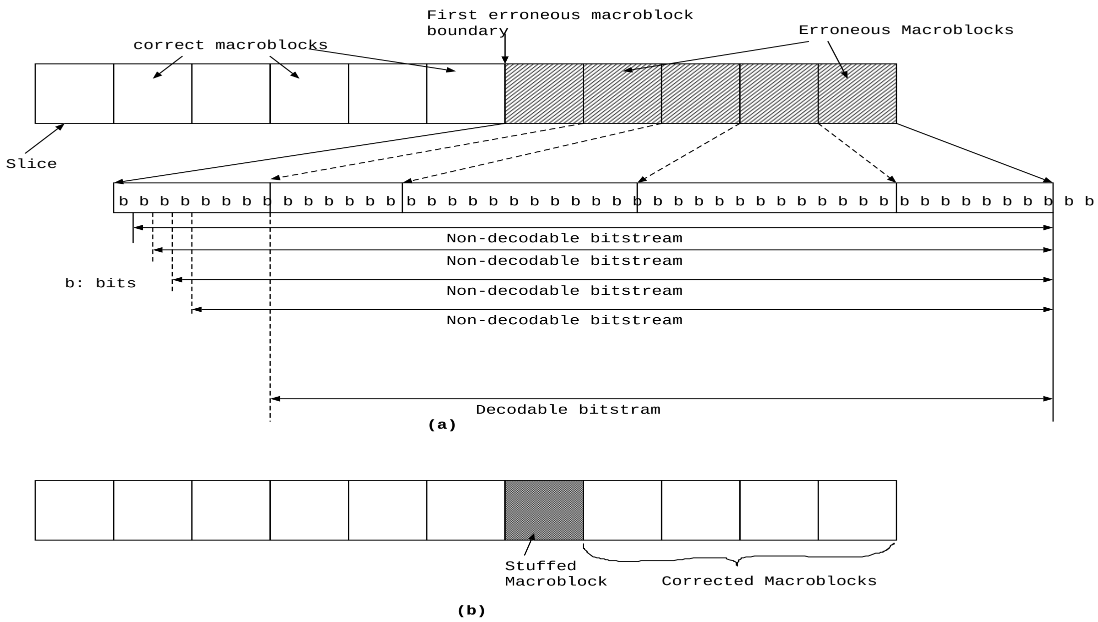


Fig. 2: Principle of step by step decoding (a) Slice before the correction and sequential bit skipping (b) Slice after correction

3.3 Concealment:

After detecting and correcting most of the erroneous macroblocks, the erroneous slices may still have some erroneous macroblocks (at least one in which the error actually occurred) which were filled in with gray values during the step by step decoding. These macroblocks can now be efficiently concealed with any standard error concealment technique. There are two basic approaches for error concealment: spatial and temporal. In the spatial interpolation, which is used for intra coded frames, the pixels of missing macroblocks are reconstructed as the median value of the corresponding pixels in eight surrounding macroblocks. For inter coded frames, we use simple motion compensated temporal error concealment, in which the motion vectors of the missing macroblock is set to the median value of the motion vectors of the eight neighbouring macroblocks. If the surrounding macroblocks are not available, the corresponding motion vectors are set to zero. Then, the macroblock from the previous frame at the spatial location specified by this motion vector is copied to the location of the missing macroblock in the current frame.

```
• Decode the frame from received bit stream.
• Check for erroneous macroblocks.
• If (frame contains any erroneous macroblock) do:
  {
      for each slice do:
    {
        if(contains no erroneous MB)
              decode as with conventional decoder
        else
        {
              • Initialise pointer at first bit location of first erroneous MB;
              • decoder part of the slice from start to pointer location conventionally;
              • do until decodable bitstream is found
                {
                • increment the pointer by one bit location;
                • decode remaining  part of the slice starting from current pointer address;
                • If (part is decodable) break;
                }
              • Decode this part of the slice with right alignment;
              • Stuff the gray valued macroblocks between two decoded parts, if  needed;
        }
    }
      • Reconstruct the frame, and check again for error.
      • If (no corrupted MB except grayed MBs) break;
  }
```

Fig. 3: Error detection and correction algorithm

4. SIMULATION RESULTS

In order to evaluate the performance of our error detection and correction technique, we have considered one frame of the ‘Salesman’ sequence decoded from a 64kbps H.263 coded bitstream, shown in Fig. 4(a). The errors are artificially introduced in the bitstream, and the erroneously decoded frame is shown in Fig. 4 (b). The error detection and correction algorithm is recursively applied to this erroneous frame, and decoded images at each step are given in Fig. 4(c)-(f). By comparing these frames, it can be seen that applying error detection and correction each time improved the quality of the decoded frame. After four iterations, the detector no longer detects any additional erroneous macroblocks except the inserted gray macroblocks, and hence program will terminate. As can be seen from Fig. 4(f) each erroneous slice has now only one erroneous block, which can be concealed better than that of Fig. 4(b). Thus our proposed technique is highly efficient in restricting the erroneous area.

To test the performance of our technique over some lossy channels, we have considered simple binary symmetric channel (BSC). This is because for a given bit error rate (BER) the BSC model is actually the worst model yielding the highest probability of error in the slice as compared to channels with correlated errors (burst errors). For our experiment we have considered *Salesman* (QCIF, 176x144) video sequences in 4:2:0 YUV format at temporal resolution of 10 frame/sec, coded at 64 kbps and transmitted over BSC with different BER. It is assumed here that the picture and slice headers including LQUANT and LMVV are not corrupted. To provide statistically significant results, all the experiments are performed for 100 independent channel conditions. The overall PSNR of a frame is measured as:

$$PSNR = 10\log_{10}\frac{255^2}{\left(MSE(Y)+MSE(U)+MSE(V)\right)/3} \tag{1}$$

where MSE(X), X $\in$ {Y, U, V} is the mean-squared-error of component X averaged over all independent channel realizations. The overall video quality is also measured in terms of PSNR, with each MSE is averaged over all frames and all channel conditions. In all theses experiments, the first frame is coded as I-frame and the subsequent frames are coded as P-frames.

To evaluate the performance, we have coded the entire video sequence at 64 kbps with first frame as intra (I) and the remaining as P-frames. The I-frame is transmitted error free while all the other frames are transmitted over lossy BSC with BER = {$2\text{x}10^{-4}$, $5\text{x}10^{-4}$, $1\text{x}10^{-3}$, $2\text{x}10^{-3}$, $5\text{x}10^{-3}$}. Fig. 5(a) shows the frame-by-frame comparison of the overall PSNR of our method with the conventional one (both with and without temporal concealment) for the test sequences at the BER = 10^{-3}. From these graphs, it is apparent that our proposed method consistently gives improvement over the conventional decoder. However the relative improvement with concealment is more than that without concealment. This can be attributed to the fact that the isolated grayed (perhaps erroneous) macroblocks in our method can suitably be concealed. Furthermore, it should be noted that apart from

(a) (b)

(c) (d)

(e) (f)

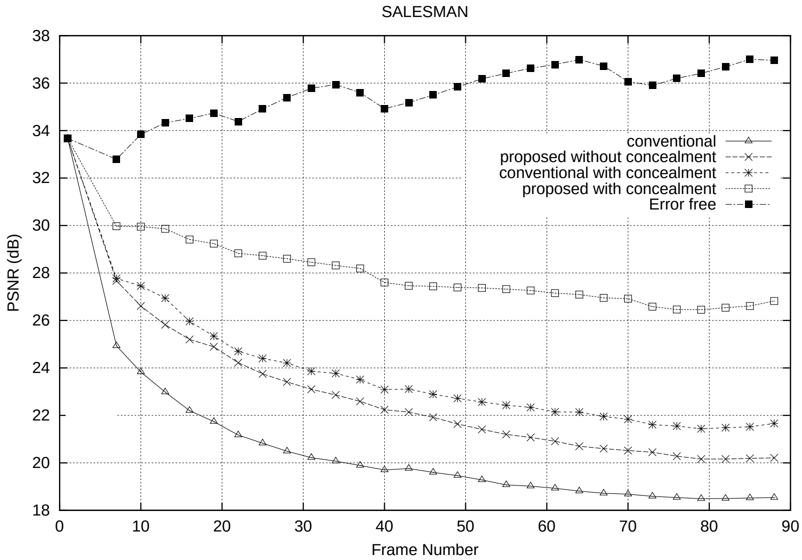
Fig. 4: Results of recursively applying error detection and step-by-step decoding based correction (a) Non-erroneous frame (b) erroneous frame (c-f) reconstructed frame after recusive error detection and correction at four consequtive steps.

iterative decoding, our method does not utilize any extra resources (like bandwidth, channel rate etc), and hence under no error conditions, frames can be reconstructed without any loss in the quality. However under the lossy channel conditions, some of the macroblocks are grayed or concealed, which results in the loss of subjective and objective quality.

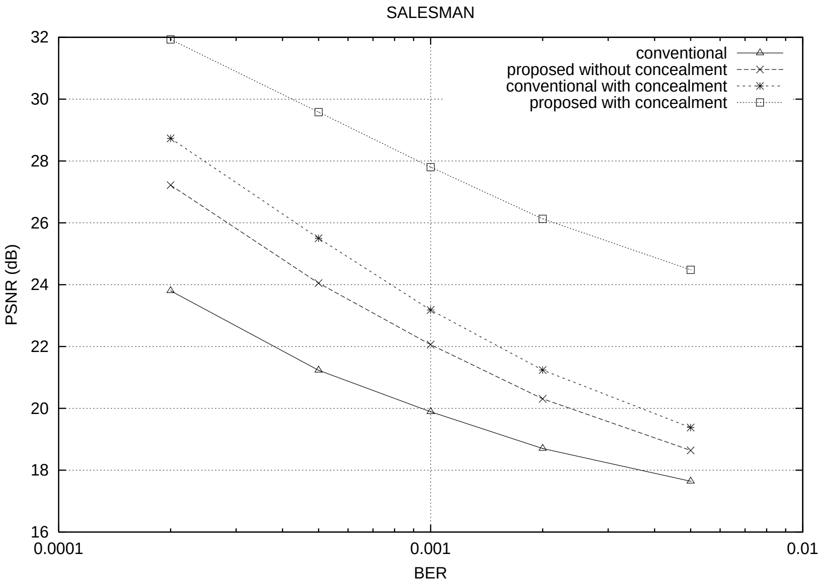
Finally Fig. 5(b) illustrates the comparison of resulting overall video quality for the conventional and proposed methods (with and without concealment) at various BERs. The overall video quality is represented in terms of PSNR defined in eqn. (1), where each MSE(X), $X \in \{Y, U, V\}$ is averaged over all frames and all 100 independent channel conditions. It is worth mentioning here that our method consistently outperforms the conventional one at all bit error rates, but relative improvement is higher at lower BERs than higher BERs. Further to note that the temporal concealment improves the results in both cases but PSNR improvement of our method over conventional method is more with the concealment than without.

5. CONCLUSIONS:

In this paper, we have proposed a novel error detection and correction technique for H.263 coded video over BSC channels. We have suggested and analysed the criteria for detecting erroneous macroblocks in a decoded frame and found that these criteria are sufficient to detect most of the erroneous macroblocks. The information about each macroblock whether erroneous or not, along with the received bitstreams are then used in a step-by-step decoding based correction. The simulation results show that iterative use of our technique can recover frames from the corresponding corrupted frames without using any reverse channel or channel coding for BER up to 0.005-0.01 over BSC. Further the spatial and temporal concealment can better be utilized to further improve the quality with our method. Since error detection is performed at frame level, the quality of decoded image is a constraint on its performance. Thus the channel BER and transmission rate will affect the error detection. Further, since correction and detection are applied iteratively at slice level, the decoding time is much larger compare to the conventional decoder. The decoding time increases with the increase in channel BER.



(a)



(b)

Fig. 5: Performance comparison of proposed and conventional methods with and without concealment for 'Salesman' sequence (a) Frame-by-frame comparison at BER=10^{-3} (b) performance at different BERs.

REFERENCES

[1] ITU-T Recommendation H.263 version 2, “Video coding for low bit rate communication”, Jan 1998.

[2] G. Cote, B. Erol, M. Gallant and F. Kossentini, “H.263+: Video coding at low bit rates”, IEEE Trans Circuit and Systems for Video Technology, Vol. 8, Nov. 1998.

[3] S. Wenger, G. Knorr, J. Oti and F. Kossentini, “Error resilience support in H.263+”, IEEE Trans Circuit and Systems for Video Technology, Vol. 8, Nov. 1998.

[4] R. Talluri, “Error resilient video coding in the MPEG-4 standard”, IEEE Communication Magzine, Vol. 36, pp. 112-119, June 1998.

[5] E. Steinbach, N. Fraber and B. Girod, “Standard compatible extension of H.263 for robust video transmission in mobile environment”, IEEE Trans. Circuit and Systems for Video Technology, vol. 7, pp. 872-881, Dec. 1997.

[6] M. Budagvi and J. D. Gibson, “Multiframe video coding for improved performance over wireless channels”, IEEE Trans. Image Processing, vol. 10, pp. 252-265, Feb. 2001.

[7] B. Girod and N. Farber, “Feedback-based error control for mobile video transmission”, Proc. IEEE, vol. 97, pp. 1707-1723, Oct. 1999.

[8] K. Stuhlmuller, N. Farber, M. Link and B. Girod, “Analysis of video transmission over lossy channels”, IEEE Journal on Selected Area in Communication, vol. 18, pp. 1012-1032, June 2000.

[9] D. W. Redmill and N. G. Kingsbury, “The EREC: An error resilient technique for coding variable length blocks of data”, IEEE Trans. Image Processing, vol. 5, pp. 565-574, April 1996.

[10] Y. Wang and Q. Zhu, “Error control and concealment for video communication: A review”, Proc. IEEE, vol. 86, pp. 974-996, May 1998.

[11] Y. –H. Han and J. –J. Leou, “Detection and correction of transmission errors in JPEG images”, IEEE Trans. Circuit and Systems for Video Technology, vol. 8, pp. 221-231, Apr. 1998.

[12] M. Abdat et al., “Transmission error detection and concealment in JPEG images”, Signal Processing: Image Communication, vol.13, pp. 45-64, July 1998.

[13] J. Wen and J. D. Villasenor, “Reversible variable-length codes for robust image and video transmission”, Proc. 31st Asilomars Conf., Signal, systems and computers, vol. 2, pp. 973-979, Nov. 1997.