Enhanced Low Complex Cost Function for H.264/AVC Intra Mode Decision

Mohammed Golam Sarwer, and Q. M. Jonathan Wu
Department of Electrical and Computer Engineering
University of Windsor
Windsor, ON, Canada
sarwer@uwindsor.ca, jwu@uwindsor.ca

***Abstract*—In order to reduce the computation of rate-distortion optimization (RDO) module of H.264/AVC intra encoder, this paper proposes an enhanced low complexity cost function. The enhanced cost function uses sum of absolute Hadamard-transformed differences (SATD) and mean absolute deviation of the residual block to estimate distortion part of the cost function. A threshold based large coefficients count is also used for estimating the bit-rate part. The proposed method improves the rate-distortion performance of the conventional fast cost functions while maintaining low complexity requirement. As the results, the encoding process can be significantly accelerated with use of the proposed cost function. Simulation results confirm that the proposed method reduces about 85% of computation of original encoder with negligible rate-distortion performance degradation.**

Keywords- H.264/AVC; intra prediction; rate-distortion optimization; sum of absolute transform differences

I. INTRODUCTION

H.264/AVC [1] video coding standard is the latest international standard developed by ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group, which provides gains in compression efficiency up to 50% over a wide range of bit rates and video resolutions compared to previous standards [2]. The rate-distortion optimization (RDO) is one of the essential parts of the H.264/AVC encoder to achieve the much better coding performance. However, the computational complexity of the RDO technique is extremely high.

In order to reduce the computation of rate-distortion (RD) optimization several rate distortion models are proposed in literature [3-6]. To reduce the computation of RDO, an improved cost function for intra 4x4 mode decisions was proposed in [5]. In this cost function, sum of absolute integer transform differences (SAITD) is used in distortion part and a rate prediction algorithm is used in rate part. The major drawback of this cost function is that it requires performing the true integer transform. Even though fast transformation algorithm was proposed to perform SAITD but the overall complexity is still quite high. To reduce the complexity of rate-distortion cost computation, a fast bit rate estimation technique is proposed to avoid the entropy coding method during intra and inter mode decision of H.264/AVC [6]. But this cost function still need to calculate computation expensive sum of squared differences (SSD). In this paper, an enhanced sum of absolute Hadamard transform differences (SATD) based cost function is proposed which use the mean absolute deviation of the residual block in addition to SATD for the distortion part and a threshold is used to determine the large coefficients of the Hadamard transformed block for rate estimation.

The remainder of this paper is organized as follows. Section II provides the review of rate distortion optimized cost functions of H.264/AVC intra 4x4 mode decision technique. In section III, cause of sum of squared difference (SSD) is analyzed. Section IV introduces the proposed enhanced SATD based cost function. The simulation results of the proposed method are presented in section V. Finally, section VI concludes the paper.

II. COST FUNCTIONS OF H.264/AVC INTRA PREDICTION

H.264/AVC intra prediction uses 9 prediction modes for a 4x4 luma block. To take the full advantages of all modes, the H.264/AVC encoder can determine the mode that meets the best rate-distortion (RD) tradeoff using rate-distortion optimization (RDO) mode decision scheme. The best mode is the one having minimum RD cost and this cost is expressed as

$$J_{RD} = SSD + \lambda \cdot R \tag{1}$$

where the SSD is the sum of squared difference between the original blocks **S** and the reconstructed block **C**, and it is expressed by

$$SSD = \sum_{i=1}^{4}\sum_{j=1}^{4}(S_{ij} - C_{ij})^2 \tag{2}$$

where S_{ij} and C_{ij} are the *(i, j)*th elements of the current original block **S** and the reconstructed block **C**. In (1), the R is the true bits needed to encode the block and λ is an exponential function of the quantization parameter (QP) and defined as

$$\lambda = 0.85 \times 2^{(QP-12)/3} \tag{3}$$

The rate-distortion function J_{RD} introduces a lot of computation in real encoding as it requires the following computations:

1. Compute the predicted block: **P**
2. Compute the residual block: **E** = **S** – **P**

978-0-7695-4356-7/11 $26.00 © 2011 IEEE
DOI 10.1109/CMSP.2011.17

3. Discrete Cosine Transform (DCT) of the residual block: **F** = DCT(**E**)
4. Quantize the transformed residual block: **F**" = Q(**F**)
5. Entropy coding of the quantized and transformed residual block to determine the bit-rate for encoding the block: R = EC(**F**")
6. Inverse quantize the quantized and transformed residual block: **F**' = Q^{-1}(**F**")
7. Inverse Discrete Cosine Transform (DCT) of the de-quantized block: **E'** = DCT^{-1}(**F'**)
8. Compute the reconstructed image block: **C** = **E**' + **P**
9. Compute SSD between **S** and **C** by using (2)
10. Calculate the cost function : $J_{RD} = SSD + \lambda \cdot R$

The H.264/AVC encoder computes this rate-distortion optimization process for every macroblock with all possible modes. All of these processing explains the high computational complexity of J_{RD} cost calculation. Hence, the cost function will make H.264/AVC impossible to be realized in real-time applications.

To accelerate the coding process, H.264/AVC provides a fast SAD-based cost function:

$$J_{SAD} = SAD + \lambda_1 \cdot 4P \quad (4)$$

where SAD is sum of absolute difference between the original block **S** and the predicted block **P**. The λ_1 is also approximate exponential function of the QP which is almost the square of λ, and the P equal to 0 for the most probable mode and 1 for the other modes. The SAD is expressed by

$$SAD = \sum_{i=1}^{4}\sum_{j=1}^{4}\left|S_{ij} - P_{ij}\right| \quad (5)$$

where S_{ij} and P_{ij} are the *(i, j)*th elements of the current original block **S** and the predicted block **P**, respectively. This SAD-based cost function could save a lot of computations as the distortion part is based on the differences between the original block and the predicted block instead of the reconstructed block. Thus, the processes of image block transformation, quantization, inverse quantization, inverse transformation and reconstruction of image block can all be saved. In addition, the rate part is pre-defined by constants either equal 4 or 0. Thus, the Context-adaptive variable-length coding (CAVLC) or Context-adaptive binary arithmetic coding (CABAC) can also be saved. However, the expense of the computation reduction usually comes with quite significant degradation of coding efficiency.

To achieve better rate-distortion performance, H.264/AVC also provided an alternative SATD-based cost function:

$$J_{SATD} = SATD + \lambda_1 \cdot 4P \quad (6)$$

where SATD is sum of absolute Hadamard-transformed difference between the original block **S** and the predicted block **P**, which is given by

$$SATD = \sum_{i=1}^{4}\sum_{j=1}^{4}\left|h_{ij}\right| \quad (7)$$

where h_{ij} are the (*i, j*)th element of the Hadamard transformed image block **H**. The Hadamard transformed block **H** is defined as

$$\mathbf{H} = \mathbf{T}_H(\mathbf{S} - \mathbf{P})\mathbf{T}_H^T = \mathbf{T}_H(\mathbf{E})\mathbf{T}_H^T \quad (8)$$

$$\text{with, } \mathbf{T}_H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad (9)$$

Experimental results show that the J_{SATD} can achieve better rate-distortion performance than the J_{SAD}, but it requires more computation due to the Hadamard transformation. If the fast Hadamard transform (FHT) is used, the computational requirement of the J_{SATD} can reduce half of the computation. However, the overall rate-distortion performance degradation is still not acceptable.

III. THE CAUSE OF SUM OF SQUARE DIFFERENCES (SSD)

Before we propose the new cost function, we first present the major cause of the sum square differences (SSD) between the original block and reconstruction block in the rate-distortion cost function. Mathematically, the original block **S** and reconstructed block **C** can be expressed as

$$\mathbf{S} = \mathbf{E} + \mathbf{P} \quad (10)$$
$$\mathbf{C} = \mathbf{E}' + \mathbf{P} \quad (11)$$

where the **P** is the predicted block, **E** is the residual block and **E**' is the reconstructed residual block. Then, the SSD can be expressed as [7]

$$\begin{aligned}\mathbf{SSD} &= \|\mathbf{S} - \mathbf{C}\|_F^2 = \|\mathbf{E} + \mathbf{P} - \mathbf{E}' - \mathbf{P}\|_F^2 \\ &= \|\mathbf{E} - \mathbf{E}'\|_F^2 = \|\mathbf{F} - \mathbf{F}'\|_F^2\end{aligned} \quad (12)$$

where $\| \ \|_F$ is the Frobenius norm. and **F** and **F**' are the transformed residual block and inverse quantized-transformed residual block. That means the spatial-domain SSD is equivalent to DCT domain SSD. Based on this relationship, we can calculate the R-D cost in transform domain with $J_{RD} = SSD(\boldsymbol{F}, \mathbf{F}') + \lambda \cdot R$ for saving the computation of inverse DCT transform of image block . However, this computational reduction is very limited. The quantization is applied in the transformed coefficients of **F**.

Thus, the cause of the SSD is due to the quantization errors in the DCT transformed residual block **F'**. This formation explains why the SATD can perform better than the SAD for the distortion part estimation for the rate-distortion cost function. It is because SATD uses the sum of the absolute coefficient values of **H** to estimate the distortion part, and property of the Hadamard transform is quite close to the Integer Transform used in H.264/AVC, thus the accuracy of SATD is much better than SAD. In this paper, we use Hadamard transfrom coefficients to define an enhanced cost function, which maintain similar complexity of the SATD but with better rate-distortion performance.

IV. ENHANCED SATD BASED COST FUNCTION

It is reasonable to say that a complex block produces a large distortion value compared to a simple block. Therefore, residual block with high detail has larger distortion than homogeneous block. Let us consider two residual blocks E_A and E_B.

$$\boldsymbol{E_A} = \begin{bmatrix} 0 & 10 & 8 & 10 \\ 9 & 7 & 4 & 10 \\ 1 & 10 & 11 & 4 \\ 19 & 6 & 15 & 7 \end{bmatrix} \text{ and } \boldsymbol{E_B} = \begin{bmatrix} 22 & 22 & 22 & 22 \\ 22 & 22 & 22 & 22 \\ 20 & 20 & 20 & 20 \\ 22 & 22 & 22 & 22 \end{bmatrix}$$

Assume *QP=24*. If we compute SSD, it is found that $SSD_A = 173$ and $SSD_B = 48$. That means distortion of block **A** is higher than that of block **B**. This is understandable because block **A** contains larger detail than block **B**. But if we calculate SATD of A and B by (7), we found that both of the residual blocks produce same SATD value $SATD_A = SATD_B = 368$. In order to address this issue, the distortion part of the proposed cost function is calculated as follows

$$ESATD = SATD + \alpha \times \sigma(\boldsymbol{E}) \tag{13}$$

where *ESATD* is enhanced SATD and α is a constant value and $\sigma(\boldsymbol{E})$ is the absolute deviation of the residual block **E** which represent the variation of pixel values. $\sigma(\boldsymbol{E})$ is defined as

$$\sigma(\boldsymbol{E}) = \frac{1}{16}\sum_{i=1}^{4}\sum_{j=1}^{4}\left|E_{ij} - \mu\right| \tag{14}$$

$$\text{with } \mu = \frac{1}{16}\sum_{i=1}^{4}\sum_{j=1}^{4}\left|E_{ij}\right| = \frac{h_{11}}{16} = (h_{11} >> 4)$$

Where h_{11} is the (1,1)th co-efficient of Hadamard transformed matrix **H** which is already computed for SATD calculation. The proposed cost function also uses a rate-predictor for estimating the rate for encoding the residual block instead of just using a penalty cost (4P) for the unfeasible modes. The rate-predictor is based on the property of the context-based adaptive variable length coding (CAVLC) entropy coder. To avoid DCT, quantization and CAVLC encoding processes, the proposed rate-predictor only uses the total number of the non-zero quantized Hadamard Transform coefficients (T_{bc}) which can be obtained by perform simple threshold and counter operations. Based on the property of the CAVLS's VLC tables, the larger T_{bc} will produce more encoded bits. With the penalty cost (4P) from J_{SAD}, the overall estimated rate R_e is defined as

$$R_e = 3T_{bc} + 4P \tag{15}$$

The total number of the non-zero quantized Hadamard Transform coefficients (T_{bc}) is calculated as follows:

Step1: $T_{bc} = 0$;
Step 2: For i=1 to 4
  For j=1 to 4
    If ($|h_{ij}| \geq Q_{step}$) then $T_{bc} = T_{bc} + 1$
  End for
End for

Q_{step} is the quantization step size used in H.264/AVC encoder. Values of Q_{step} with 7 different QPs are given in Table I. Q_{step} doubles in size for every increment of 6 in QP [8]. Therefore the proposed cost function is defined as

$$J_{ESATD} = ESATD + \lambda_1 R_e \tag{16}$$

TABLE I. QUANTIZATION STEP SIZES Q_{step} IN H.264/AVC CODEC

QP	Q_{step}
0	0.625
1	0.6875
2	0.8125
3	0.875
4	1
5	1.125
6	1.25

By putting value of ESATD and R_e, the cost function becomes

$$J_{ESATD} = SATD + \alpha \times \sigma(\boldsymbol{E}) + \lambda_1(3T_{bc} + 4P) \tag{17}$$

In order to find the value of α , we have done some simulations with different types of sequences with different QPs and better results were found at $\alpha = 1.25$. As explained in the previous section, the cause of the SSD is due to the quantization error of the integer transform coefficients of the residual block **E**, which can be estimated by sum of absolute coefficients of **H** in SATD. It is well understandable that high frequency coefficients of Hadamard transform coefficients are insignificant and bears low values.

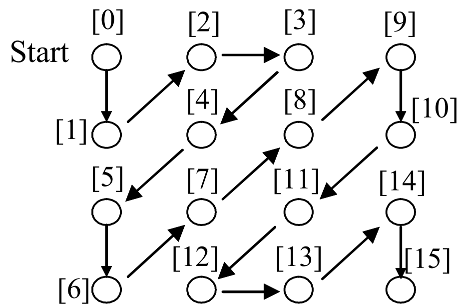

Figure 1. Zig-zag scan and corresponding frequency of **H**

Fig. 1 shows the zig-zag scan and corresponding value of frequency of Hadamard Transform matrix **H. H** can be redefined in terms of frequency as follows.

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} = \begin{bmatrix} I_0 & I_2 & I_3 & I_9 \\ I_1 & I_4 & I_8 & I_{10} \\ I_5 & I_7 & I_{11} & I_{14} \\ I_6 & I_{12} & I_{13} & I_{15} \end{bmatrix} \quad (18)$$

where I_f is the Hadamard Transform coefficient of frequency *f*. After Hadamard transform, the high frequency coefficient usually has small energy. For this reason, the proposed cost function calculates only 10 low frequency Hadamard co-efficients instead of calculating all 16 co-efficient. Therefore the proposed cost function becomes,

$$J_{ESATD} = SATD' + \alpha \times \sigma(\boldsymbol{E}) + \lambda_1(3T'_{bc} + 4P) \quad (19)$$

with, $SATD' = \sum_{f=0}^{9} |I_f|$

The step by step computation of proposed cost function is provided below:

1. Compute the predicted block: **P**
2. Compute the residual block: **E** = **S** – **P**
3. Compute only 10 low frequency coefficients of Hadamard transform matrix of the residual block: **H** = HT(**E**)
4. Calculate the $SATD'$ and T'_{bc}
   Set $SATD' = 0$ and $T'_{bc} = 0$

   For f=0 to 9
   $SATD' = SATD' + |I_f|$;
   If ($|I_f| \geq Q_{step}$) $T'_{bc} = T'_{bc} + 1$;
   End for
5. Calculate $\sigma(\boldsymbol{E})$ by (14)
6. Calculate
   $J_{ESATD} = SATD' + \alpha \times \sigma(\boldsymbol{E}) + \lambda_1(3T'_{bc} + 4P)$

V. Simulation Results

The performance of proposed cost function was tested using the first 100 frames from nine different video sequences (*Akiyo*, *Claire*, *News*, *Container*, *Foreman*, *Stefan*, *Mother_daughter*, *Silent*, and *Hall*). The experiment was carried out in the JVT JM 12.4 [9] encoder and the test parameters are (a) CAVLC is enabled; (b) Frame rate is 30; (c) All frames are I frame; (d) Intra 4x4 prediction only; (e) QPs are 30/36/42/48; and (f) Number of frames: 100.

A. Rate-distortion performace comparison

In these experiments, four cost functions (J_{RD}, J_{SAD}, J_{SATD}, and J_{ESATD}) are simulated. The PSNR and bit rate comparisons of the proposed cost function are tabulated in Table II. PSNR and bit rate differences ($\Delta R\%$) are calculated according to the numerical averages between RD curves derived from the original and proposed algorithm, respectively. The detail procedure to calculate these differences can be found in [10]. The positive values mean increments whereas negative values mean decrements. It is shown that RD performance degradation for both J_{SAD} and J_{SATD} are significant. In case of J_{SATD} the average PSNR reduction is about 0.31 dB and average bit rate increment is about 7.10%. Whereas in our proposed method, the average PSNR reduction is about 0.13 dB and average bit rate increment is about 3.62%.

TABLE II. PSNR and Bit Rate Comparison

Sequence	$\Delta PSNR$ in dB			$\Delta R\%$		
	J_{SAD}	J_{SATD}	J_{ESATD}	J_{SAD}	J_{SATD}	J_{ESATD}
Akiyo QCIF	-0.39	-0.37	-0.12	7.49	6.86	2.71
Foreman QCIF	-0.37	-0.21	-0.07	8.46	4.76	2.34
Container QCIF	-0.36	-0.30	-0.18	8.83	7.26	4.29
Claire QCIF	-0.33	-0.31	-0.01	5.96	5.64	1.52
Stefan QCIF	-0.60	-0.47	-0.27	16.84	12.99	7.43
News QCIF	-0.49	-0.42	-0.19	10.79	9.11	4.82
Mother_daughter QCIF	-0.29	-0.22	-0.07	6.90	4.86	2.13
Silent CIF	-0.28	-0.21	-0.12	8.90	6.20	4.82
Hall CIF	-0.36	-0.35	-0.14	6.41	6.29	2.59
Average	-0.38	-0.31	-0.13	8.95	7.10	3.62

The coding results of proposed method are very similar to actual RDO method. It is clear that our proposed cost function always perform better than J_{SATD} and J_{SAD}. The worst case is encoding of *Stefan* video sequence. For *Stefan*, the proposed method increase the bit rate of about 7.43% whereas J_{SATD} generate around 12.99% of bit rate increment.

B. Complexity Comparison

In order to evaluate complexity reduction of proposed method as compared to original JM encoder, ΔT_1 (%) is defined as follows

$$\Delta T_1 = \frac{T_{original} - T_{proposed}}{T_{original}} \times 100\% \qquad (20)$$

where, $T_{original}$ and $T_{proposed}$ denote the total encoding time of the JM 12.4 encoder with J_{RD} and with proposed cost function, respectively.

TABLE III. COMPLEXITY COMPARISON

Sequence	ΔT_1 %		
	J_{SAD}	J_{SATD}	J_{ESATD}
Akiyo (QCIF)	86.76	85.92	85.76
Foreman (QCIF)	87.33	86.40	86.17
Container QCIF	85.49	84.65	83.99
Claire QCIF	84.03	83.15	82.49
Stefan QCIF	88.88	88.17	87.36
News QCIF	85.66	84.99	84.30
Mother_daughter QCIF	84.17	83.38	82.57
Silent CIF	87.43	86.99	86.67
Hall CIF	90.80	86.39	85.78
Average	86.73	85.56	85.01

TABLE IV. COMPARISON WITH J_{SAITD} [5]

Sequence	$\Delta PSNR$	$\Delta R\%$	ΔT_2 %
Akiyo (QCIF)	0.10	-1.93	16.24
Foreman (QCIF)	0.04	-0.89	18.69
Container QCIF	0.05	-1.48	14.23
Claire QCIF	0.11	-1.85	12.77
Stefan QCIF	0.08	-1.99	11.60
News QCIF	0.12	-2.66	13.62
Mother_daughter QCIF	0.09	-1.87	12.98
Silent CIF	0.04	-1.22	21.35
Hall CIF	0.06	-1.15	22.59
Average	0.10	-1.67	16.00

The complexity reductions of J_{SAD}, J_{SATD}, and J_{ESATD} are tabulated in Table III. The proposed algorithm reduced about 85.01% of total encoding time compared to rate-distortion optimized cost function J_{RD}. The computational reduction of proposed method is almost similar with J_{SATD}. However, the rate-distortion performance of proposed method is much better as compared with $J_{SATD.}$

C. *Comparison with other method*

In this experiment, the proposed method is compared with fast SAITD based cost function (J_{SAITD}) proposed in [5] in terms of rate distortion performance and complexity. Table IV shows the comparison result. In Table IV, PSNR and bit rate performances are calculated based on [10] and complexity reduction is calculated as follows:

$$\Delta T_2 \% = \frac{T_{ref[5]} - T_p}{T_{ref[5]}} \times 100\% \qquad (21)$$

Where $T_{ref[6]}$ and T_p are the total encoding time of the method presented in [5] and proposed method, respectively. From the comparison result, it is shown that our proposed method reduced the bit rate of about 1.67% and increases the PSNR of about 0.10 dB on average. The proposed cost function not only improves the RD performance but also about 16% faster than that of J_{SAITD}.

VI. CONCLUSION

In this paper, a simple and fast cost function based on SATD is proposed for intra mode decision of H.264/AVC. The distortion part of the cost function is estimated based on the SATD and mean absolute deviation of residual block. If mean absolute deviation is higher, the distortion is also higher. Bit rate is predicted based on the number of large Hadamard Transformed coefficients. A block with large number of Hadamard Transformed coefficient produce large bit rate. With the proposed scheme, DCT, quantization, inverse-quantization, inverse DCT operations can be skipped during the mode decision process. The proposed technique reduces encoding time by 85% with acceptable performance degradation. The RD performance and computational complexity of this algorithm is also better than methods stated in [5].

ACKNOWLEDGMENT

The work described in this paper has been supported by National Science and Engineering Research Council (NSERC), Canada.

REFERENCES

[1] ITU-T Rec. H.264 ISO/IEC 14496-10 AVC, Draft Text of Final Draft International Standard for Advanced Video Coding. International Organization of Standardization, Pattaya, 2003.

[2] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, T. Wedi, " Video Coding with H.264/AVC: Tools, Performance, and Complexity", IEEE Circuits and Systems Magazine, vol. 4, 2004, pp. 7-28

[3] Yu-Kuang, Jar-Ferr Yang, and Ming-Ting Sun, "Efficient rate-distortion estimation for H.264/AVC coders." IEEE Trans. Circuits Syst. Video Tech., vol. 16, no. 5, pp.600-611, May 2006.

[4] Yu-Ming Lee, Yu-Ting Sun, and Yinyi Lin "SATD-Based Intra Mode Decision for H.264/AVC Video Coding", IEEE Trans. Circuits Syst. Video Tech., vol. 20, no. 3, pp.463-469, March 2010.

[5] C.H. Tseng, H.M. Wang, and J. F. Yang, "Enhanced Intra 4x4 Mode Decision for H.264/AVC Coders," IEEE Trans. Circuits Syst. Video Tech., vol. 16, no. 8, pp.1027-1032, August 2006.

[6] Mohammed Golam Sarwer, and Lai Man Po "Fast bit rate estimation for mode decision of H.264/AVC" IEEE Transaction on Circuit and System for Video Technology, volume 17, number 10, October 2007, pp 1402-1407.

[7] L. M. Po and K. Guo, "Transform-Domain Fast Sum of the Squared Difference Computation for H.264 Rate-Distortion Optimization," IEEE Trans. on Circuits and Systems for Video Technology, vol. 17, No. 6, pp. 765-773, June 2007.

[8] Iain E. G. Richardson, H.264 and MPEG-4 Video Compression – video coding for next generation multimedia, John Wily & Sons, 2003, pp. 191-192.

[9] Joint Video Team (JVT) reference software version 12.4, available online: http://iphome.hhi.de/suehring/tml/download/old_jm/

[10] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," presented at the 13th VCEG-M33 Meeting, Austin, TX, April 2001.