JPEG XR: A New Coding Standard for Digital Photography

Presented by Gary J. Sullivan
Direct contact: garysull@microsoft.com
Deck co-authored with
Sridhar Srinivasan, Chengjie Tu, and Shankar L. Regunathan
Microsoft Corporation
Group contact: hdphoto@microsoft.com

18 February 2009
JPEG XR Camera Raw AHG Meeting
Tokyo, Japan

What is JPEG XR?

- JPEG XR is a new draft standard specifying a compressed format for images
  - Feature rich to support existing and emerging use scenarios
  - Achieves high quality compression with low complexity
  - Based on technology developed by Microsoft known as HD Photo and Windows Media Photo
  - Coding spec now at FDIS (Final Draft International Standard) status in JPEG committee as ISO/IEC 29199-2 and Draft ITU-T Rec. T.832
    - Compressed codestream & decoding process specified in main body
    - TIFF-like file storage specified in Annex A
- JPEG XR supports the following features
  - Lossy and lossless compression in the same signal flow path
  - Up to 24 bits/sample lossless, 32 bits/sample lossy
    - Unsigned int, Fixed-point signed, Half, Float and RGBE radiance image data formats
  - Bit-exact specification with 16/32 bit integer arithmetic
    - Even to compress floating point data

What is JPEG XR?

- … Features
  - Low complexity, low memory footprint
  - Random access and large image sizes (gigapixels) supported
  - Progressive decoding
  - Embedded thumbnails
  - Spatially-dominant or frequency-dominant codestream layout
  - Alpha channel support
    - Planar or interleaved
  - Multiple color format & sampling choices
    - 4:2:0, 4:2:2, 4:4:4
    - CMYK
    - n-Component
  - Compressed-domain manipulation support
  - Rich metadata support (e.g. ICC profiles, Adobe XMP, JEITA EXIF)

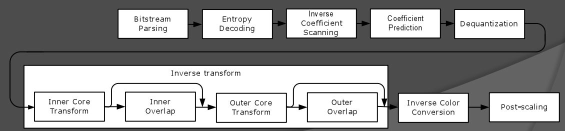
Starting Points

1. Good compression alone is not a deal maker
   - But poor compression is a deal breaker
     - Generally, codecs in the ±10-15% range of quality are considered equivalent
   - Complexity and feature set are also key to determining success
2. Block transforms are back in the game
   - Used in H.264 / MPEG-4 AVC and VC-1 for video compression
     - Excellent compression ratios leading to widespread acceptance
   - Well understood and appreciated by both h/w and s/w designers
     - Offers reduced complexity and memory benefits
3. Feature set should be chosen carefully
   - Rich features place a burden on algorithmic and runtime complexity
     - Also complicate conformance and test processes
   - Approach: Limit the number of bitstream layouts and hierarchies
     - Find simple and acceptable alternatives for addressing other required features

JPEG XR Technology Overview

- Design is built around two central innovations
  1. Reversible lapped biorthogonal transform (LBT)
  2. Advanced coefficient coding
- Structure is rather "traditional"
  - Color conversion
  - Transform
  - Quantization
  - Coefficient prediction
  - Coefficient scanning
  - Entropy coding

Decoder shown below



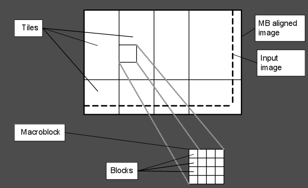
JPEG XR Design Choices

1. Multi-resolution representation (wavelet-like concept)
   - Only steps of 1:1, 1:4 and 1:16 in each dimension
     - 1:4 rather than Dyadic (1:2) is due to core 4x4 transform structure
     - 1:16 is usually sufficient for a large thumbnail for typical photographic images
     - Additional steps would cause increased range expansion in transforms
     - Resolutions below 1:16 (i.e. 1:256 spatially) may be stored as independent images with little overhead
2. Sub-region decoding: Supported through use of "tiles"
   - Tiles are regular partitions of the image into rectangular regions
   - Tiles are independently decodable
   - Tile boundaries can be seamless visually ("soft") or can be "hard" boundaries
   - Tile repartitioning for soft tiles is possible in compressed domain without introducing distortion
3. Quantization control
   - Multiple degrees of freedom with low overhead
     - Q values can be shared across color planes, tiles and/or frequency bands
     - Q values can be specified at the macroblock level by indexing
   - Lossless coding is achieved when Q=1

JPEG XR Design Choices

4. Bitstream truncation
   - Limited support through frequency partitioning and use of *flexbits*
     - Flexbits are adaptive fixed length coded (FLC) values
     - Flexbits may form a large portion of the bitstream, especially at high bit rates and with high dynamic range content
     - Syntax allows flexbits to be degraded dynamically to support bitstream truncation
     - Flexbits provide a lower complexity than bitplane coding
5. Bitdepth support
   - 1 bit through 32 bits per channel
     - 32 bit integer signal processing
     - Float, Half float and RGBE radiance are "cast" to 32/16/16 bit respectively
     - Loss may happen beyond 24 bit input in the lower bits (user controllable)
6. Color format & channel support
   - Grayscale, RGB, n-Channel
   - YUV 4:2:0, YUV 4:2:2, YUV 4:4:4
     - RGB internally converted to one of above
   - Alpha channel support for many formats

JPEG XR Data Hierarchy



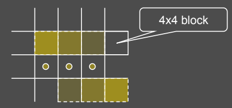
- Spatial hierarchy
  - Image composed of one or more *tiles*
  - Tile composed of integer number of 16x16 size *macroblocks*
  - Macroblock composed of 16 4x4 size *blocks* across all color planes
    - Special cases defined for 4:2:0 and 4:2:2
- Frequency hierarchy
  - Each structure divided into 3 frequency bands (*DC*, *lowpass*, *highpass*)
  - Bands correspond to 1:16, 1:4 and 1:1 resolutions

JPEG XR Transform Principles

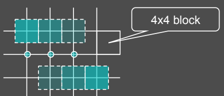
- Lapped Biorthogonal Transform
  - Lapped Orthogonal Transform developed by Malvar
  - Better compression observed with biorthogonality
  - Equivalence of overlap operators in spatial and frequency domain
- JPEG XR transform builds on the above and adds
  - Flexibility
    - Spatial domain lapped operator design allows for overlapping to be switchable
    - Core transform and overlap operator are separated out
  - Reversibility
    - Efficient reversible transform (i.e. determinant = 1) is a hard problem
  - Computational efficiency
    - Non-separable transform for reversibility has surprising complexity benefits
    - Data mappings to address SIMD parallelism
- Transform built using “lifting” technique
  - Concepts borrowed from wavelet literature
  - Provides lossless/reversibility capability
  - Also helps minimize processing dynamic range

JPEG XR Transform Structure

- The basic transform consists of two building blocks
  - A core transform which is similar to a 4x4 DCT
    - Core transform is applied to 4x4 blocks in a regular pattern within macroblocks / tiles



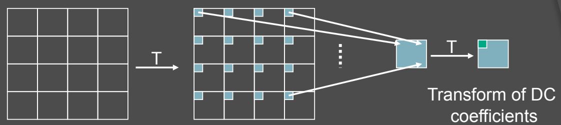
  - An overlap operator
    - Overlap operator is applied to 4x4 areas in a staggered pattern among blocks



    - Overlap operator may be switched on/off (constrained, signaled in bitstream)
    - Special cases for 4:2:2 and 4:2:0 chroma sub-sampling and image edges

JPEG XR Transform Hierarchy

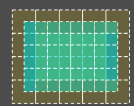
- Hierarchical transform
  - Two transform stages – second stage operates on DC of 4x4 blocks of first stage



- Three overlap modes supported
  - Mode 0 or no overlap, which reduces to 2 stages of 4x4 core (block) transform
    - Lowest complexity mode
    - Good for lossless operation and “light” degrees of compression
  - Mode 1 – overlap operator applied only to full resolution, and not to 2nd stage
    - Best R-D performance for general-purpose use
  - Mode 2 – overlap operator applied at both resolutions
    - Best visual quality for very highly compressed images
- All modes support all features, including lossless coding

Transform Memory Issues

- Transform operations
  - Inverse transform steps



  - 1 extra stored macroblock row is sufficient for all overlap modes
  - Further cache reduction can actually be achieved
    - No extra cache necessary for overlap Mode 0
    - 2 sample row cache necessary for overlap Mode 1
    - 10 sample row cache sufficient for overlap Mode 2
      - Smaller caches can be used for Mode 2 with tricks

Transform Complexity (page 1 of 2)

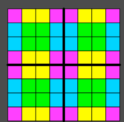
- Trivial and non-trivial operations
  - Trivial operation: sum/difference; bit shift by 1 bit
  - Non-trivial operation: a = a ± (3 * b + r) >> k
  - e.g. a += (b>>1) counts as 2 trivial operations, whereas a += (3*b+4)>>3 counts as 1 non-trivial operation
- Non-trivial operations can always be decomposed as
  - A multiplier of exactly 3 (can implement as shift+add)
  - A small constant offset
  - A shift by a small amount
  - An add/subtract
  - Summary: 1 non-trivial operation <=5 trivial operations
- Core transform: 91 trivial + 11 non-trivial per 4x4 block (plus 2nd level transform at 1/16th resolution)
- Overlap: 156 trivial + 15 non-trivial per 4x4 block (applied or not applied at 1st and 2nd level)

Transform Complexity (page 2 of 2)

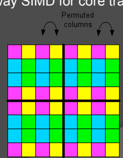
- Complexity analysis
  - Assuming nontrivial operation = 3 trivial operations
    - Overlap mode 0: **8.23** ops / image sample (124/16 * 17/16)
    - Overlap mode 1: **20.80** ops / image sample (8.23 + 201/16)
    - Overlap mode 2: **21.58** ops / image sample (8.23 + 201/16 * 17/16)
  - Observations
    - Core transform alone is very low complexity
    - Overlap operator is substantially more expensive than core transform
    - Second level costs are small (diminished by a factor of 16)
  - Low memory requirements
    - Needs only 1 extra macroblock row of data for encode or decode

Transform Design

- Implementation issues
  - Two-stage transform increases dynamic range by only 5 bits
    - Therefore, 8 bit input leads to 13 bit dynamic range of transform coefficients
    - Additional precision of 3 bits can be used to minimize rounding errors
    - 16 bit arithmetic is sufficient for 8 bit data
  - Parallelization
    - Most operations can be parallelized with SIMD
    - All operations can be parallelized with MIMD
  - SIMD
    - 16 bit word SIMD can be used for 8 bit data
    - Good mappings exist to exploit 2-way and 4-way SIMD for core transform and overlap operators (2 way mapping shown below)



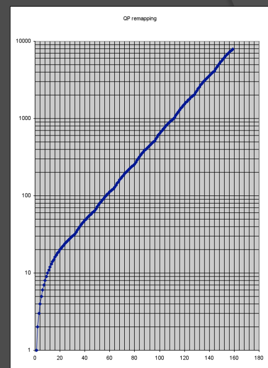
Original (natural) ordering



Permuted ordering

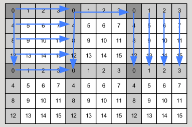
JPEG XR Quantization

- "Harmonic" quantization scale
  - Defined by the set
    {Q} U {(Q+16) 2^k}, for Q = 0…15
    - Easy to implement division for quantization by multiply and shift, due to limited number of "mantissas"
    - Inverse quantization is a simple multiply
  - Harmonic scale has advantages
    - Quantizer steps are between 3 and 6% spacing – sufficiently fine quantizers for good rate control
    - Large quantizers allowed for high bit depth data or high compression ratios
- Transform coefficients are equal norm by design
  - No additional scaling required during (de)quantization to compensate for unequal norms

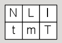


JPEG XR Coefficient Prediction

- JPEG XR uses DC & AC prediction
  - Predicts first “row” or “column” of transform block coefficients from causal blocks
    - Extends strong linear features in horizontal and vertical dimensions
  - Overlap operator extracts some extended spatial redundancy
    - Thus, DC & AC prediction is only used when orientation is strong and dominant
  - Three levels of prediction
    1. Prediction of DC values of second stage transform (DC subband)
    2. Prediction of DC & AC values of second stage transform (lowpass subband)
    3. Prediction of DC & AC values of first stage transform (highpass subband)



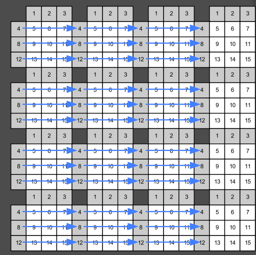
Predicted DC (dark grey) and DCAC values (light grey) of 2nd stage transform



Prediction direction – lower case indicates only DC is predicted, caps indicate DC and DCAC are predicted

JPEG XR Coefficient Prediction

- Coefficient prediction rules
  - DC prediction
    - Null, top, left and mixed (mean of top and left) allowed
    - Only within-tile prediction
  - Lowpass prediction
    - Null, top and left allowed
    - Top and left need dominant edge signatures to be picked
  - Highpass prediction
    - Null, top and left allowed
    - Dominant direction indicated by lowpass values
    - Only within macroblock prediction



Left prediction of highpass DCAC showing within macroblock prediction

JPEG XR Coefficient Scanning

- The process of converting the 2D transform into a linear encodable list
  - Also referred to as *zigzag scan*
- Scan order is adaptive
  - Changes as data is traversed
  - Simple rule based on one step of bubble sort
  - Tracks incidence of coefficients
  - In the event of an inversion (current coefficient occurs more frequently than preceding coefficient in scan), swaps the scan orders of current and preceding coefficient for the future
  - Periodic and frequent resets of counters
  - Scan orders captured in tile context
- Around 3% savings in bits over fixed scan orders

Order	4	8	5	1	12	9	6	2	13	3	15	7	10	14	11
Totals	60	54	55	48	40	40	36	33	26	24	17	16	12	5	3

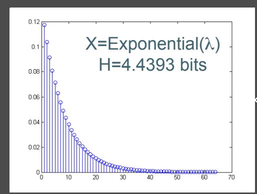
Inverse Vertical scan order just prior to an exchange operation occurring between shaded coefficients.

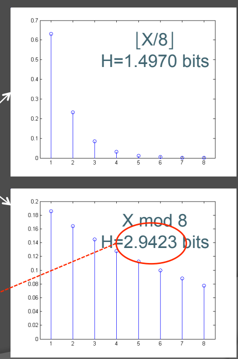
Order	4	5	8	1	12	9	6	2	13	3	15	7	10	14	11
Totals	60	55	54	48	40	40	36	33	26	24	17	16	12	5	3

Inverse Vertical scan order and Totals array subsequent to exchange operation.

JPEG XR Entropy Coding

- Adaptive coefficient normalization
  - Handles high-variance transform coefficient data
  - Key observation:

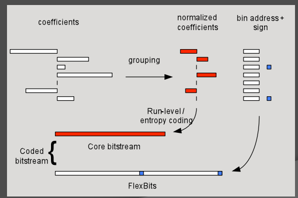




Why not use fixed length codes?
Alphabet of 8 ⇒ H=3 bits
Overall entropy = 4.4970 bits

JPEG XR Entropy Coding

- Adaptive coefficient normalization
  - Triggered when nonzero transform coefficients happen frequently
    - Often for lossless and high quality lossy compression
    - Happens more often when > 8 bit data is compressed
  - When triggered, additional bit stream layer “Flexbits” is generated
    - Flexbits is sent “raw”, i.e. uncoded
    - For lossless 8 bit compression, Flexbits may account for more than 50% of the total bits
    - Flexbits forms an enhancement layer which may be omitted or truncated



JPEG XR Entropy Coding

- Coding of normalized transform coefficients in HD Photo
  - Uses Huffman codes and other variable length codes (VLCs)
  - Symbols are remapped to a reduced alphabet set (which may have between 2 and 12 symbols)
    - Very small set of code tables
- Small number of contexts are defined for level-run coding
  - Runs and zeros are binned
  - Joint coding signals “nonzero run” and “non-one level” events
  - Run > 0 and |level| > 1 are sent independently
- 3½D-2½D coding
  - Subsequent symbols jointly code level with run of subsequent zeros or whether coefficient is last nonzero in block (2½D symbols)
  - First symbol codes in addition first run (3½D symbol)
    - More code table choices for first symbol
- Code tables are switched adaptively
  - Choice of 2 to 5 statically defined tables
  - Switch is based on discriminant counting incidences (state machine)
  - All-zero blocks are signaled through Coded Block Pattern symbols

JPEG XR Entropy Coding

SYMBOL	Code 0	Code 1	Code 2	Code 3	Code 4
0	0000 1	0010	11	001	010
1	00 0001	0 0010	001	11	1
2	000 0000	00 0000	000 0000	000 0000	000 0001
3	000 0001	00 0001	000 0001	0 0001	0001
4	0 0100	0011	0 0001	0 0010	000 0010
5	010	010	010	010	011
6	0 0101	0 0011	000 0010	000 0001	0000 0000
7	1	11	011	011	0010
8	0 0110	011	100	0 0011	000 0011
9	0001	100	101	100	0011
10	0 0111	0 0001	000 0011	00 0001	0000 0001
11	011	101	0001	101	0 0001

SYMBOL	Code 0	Code 1	Code 2	Code 3
0	1	01	0000	0 0000
1	0 0000	0000	0001	0 0001
2	001	10	01	01
3	0 0001	0001	10	1
4	01	11	11	0001
5	0001	001	001	001

SYMBOL	Code 0
0	1
1	01
2	001
3	000

All code tables used in JPEG XR for coefficient and coded block pattern coding, enumerated in binary

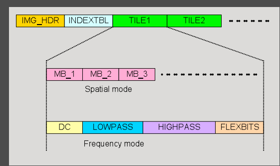
SYMBOL	Code 0	Code 1
0	010	1
1	0 0000	001
2	0010	010
3	0 0001	0001
4	0 0010	00 0001
5	1	011
6	011	0 0001
7	0 0011	000 0000
8	0011	000 0001

SYMBOL	Code 0	Code 1
0	01	1
1	10	01
2	11	001
3	001	0001
4	0001	0 0001
5	0 0000	00 0000
6	0 0001	00 0001

SYMBOL	Code 0	Code 1
0	1	1
1	01	000
2	001	001
3	0000	010
4	0001	011

SYMBOL	Code 0
0	10
1	001
2	0 0001
3	0001
4	11
5	010
6	0 0000
7	011

JPEG XR Bitstream Layout



- Bitstream consists of header, index table and tile payloads
  - Index table points to start of tile payloads
  - Index entries can be up to 64 bits
    - Supports gigantic images
- Bitstream laid out in *spatial* or *frequency* mode
  - All tile payloads in an image are in the same mode
  - In spatial mode, macroblock data is serialized left to right, top to bottom
  - In frequency mode, tile payload is separated into four bands – DC, lowpass, highpass and flexbits
    - Within each band, macroblock data is serialized in raster scan order

Other Features

- Color conversion uses YCoCg color space
  - Concept as proposed by Malvar & Sullivan to H.264 / MPEG-4 AVC (adopted there in 2004)
  - Extended to CMYK (and Bayer pattern camera Raw)
- Interleaved alpha channel supported
  - Restricted to have same tiling structure as main image
- Floating point support
  - Normal/denormal casting for smooth mapping across zero for float
  - User specified base allows discarding of redundant mantissa bits
- Compressed domain operations supported include
  - Mirror flips, 90 degree rotates (all combinations)
    - Fully reversible without loss, i.e. flip of flip is exactly same as original
    - Actual rotation, not merely flipping a flag in header
  - Region extraction
    - No need for macroblock alignment of extracted region
  - Arbitrary restructuring of tile boundary locations

Design Enhancements Included

- A few recent design improvements were included during the JPEG XR standardization process:
- [WG 1 N 4660] Enhanced overlap operators
- [WG 1 N 4792] Hard tiles and related signal processing
  - Hard tile boundary support
  - Improved image edge handling with DC gain matching
  - Enhanced overlap operators for 4:2:0 and 4:2:2 chroma
- [WG 1 N 4680] Small feature capability enhancements
  - Greater flexibility of file-level metadata support
  - YCC and CMYK support with bypass of internal color transform
  - Simplified color indicator support (e.g., for video)

Planned & Probable Further Work

- Reference software finalization
- Conformance testing suite finalization
- Motion format for sequences of images
- Technical report on general usage and integration in systems
- “Box” file storage format
- Camera Raw support
- JPIP interactive protocol support
- JPSec security use support

JPEG XR Summary

- In summary, JPEG XR is a format for compressed image data that
  - Is feature rich
  - Offers low complexity
  - Provides high compression performance
- JPEG XR is based on sound technology
  - Several years of R&D have been invested in developing the technology
  - It is derived from a long line of academic and industry research
  - The feature set is based on extensive feedback from a wide range of customers and partners
  - It has been built and tested rigorously as a mass market product
- Standardizing JPEG XR has facilitated and encouraged broad, compatible use within the marketplace

References (page 1 of 3)

1. ISO/IEC FDIS 29199-2 | Draft ITU-T Rec. T.832, "Information technology – JPEG XR image coding system – Image coding specification", JPEG document WG 1 N 4918, Feb. 2009.
2. S. Srinivasan, C. Tu, S. L. Regunathan, and G. J. Sullivan, "HD Photo: A new image coding technology for digital photography", SPIE Appl. of Dig. Image Proc. XXX, vol. 6696, Aug., 2007.
3. C. Tu, S. Srinivasan, G. J. Sullivan, S. L. Regunathan, and H. S. Malvar, "Low-complexity hierarchical lapped transform for lossy-to-lossless image coding in JPEG XR / HD Photo, SPIE Appl. of Dig. Image Proc. XXXI, vol. 7073, paper 7073-12, Aug., 2008.
4. H. S. Malvar, G. J. Sullivan, and S. Srinivasan, "Lifting-based reversible color transformations for image compression", SPIE Appl. of Dig. Image Proc. XXXI, vol. 7073, paper 7073-07, Aug., 2008.
5. Microsoft Corp., *HD Photo Specification and HD Photo Device Porting Kit 1.0*, available on-line at http://www.microsoft.com/windows/windowsmedia/forpros/wmphoto/, Dec. 2006.
6. S. Srinivasan, Z. Zhou, G. J. Sullivan, R. Rossi, S. L. Regunathan, C. Tu, and A. Roy, "Coding of High Dynamic Range Images in JPEG XR / HD Photo", SPIE Appl. of Dig. Image Proc. XXXI, vol. 7073, paper 7073-42, Aug. 2008.

References (page 2 of 3)

7. G. J. Sullivan, S. Sun, S. L. Regunathan, D. Schonberg, C. Tu, and S. Srinivasan, "Image coding design considerations for cascaded encoding-decoding cycles and image editing: analysis of JPEG 1, JPEG 2000, and JPEG XR / HD Photo", SPIE Appl. of Dig. Image Proc. XXXI, vol. 7073, paper 7073-39, Aug. 2008.
8. D. Schonberg, S. Sun, G. J. Sullivan, S. L. Regunathan, Z. Zhou, and S. Srinivasan, "Techniques for enhancing JPEG XR / HD Photo rate-distortion performance for particular fidelity metrics", SPIE Appl. of Dig. Image Proc. XXXI, vol. 7073, paper 7073-41, Aug. 2008.
9. S. L. Regunathan, D. Schonberg, S. Srinivasan, G. J. Sullivan, S. Sun, C. Tu, Z. Zhou, "Improved Overlap Filter for JPEG XR", JPEG contribution WG 1 N 4660, July 2008.
10. G. J. Sullivan, S. Sun, R. Rossi, S. L. Regunathan, and Zhi Zhou, "Image Format Support and Color Handling Aspects for JPEG XR", JPEG contribution WG 1 N 4680, July 2008.
11. D. Schonberg, S. L. Regunathan, G. Sullivan, S. Sun, and Z. Zhou, "Enhanced Tiling and Related Signal Processing Aspects for JPEG XR", JPEG contribution WG 1 N 4792, October 2008.
12. H. S. Malvar and D. H. Staelin, "The LOT: transform coding without blocking effects," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* vol. 37, pp. 553–559, Apr. 1989.

References (page 3 of 3)

13. H. S. Malvar and D. H. Staelin, "Reduction of blocking effects in image coding with a lapped orthogonal transform," *IEEE International Conference on Acoustics, Speech, and Signal Processing,* New York, NY, pp. 781–784, Apr. 1988.
14. H. S. Malvar, "Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts," *IEEE Trans. Signal Processing*, pp. 1043–1053, Apr. 1998.
15. T. D. Tran, J. Liang, and C. Tu, "Lapped transform via time-domain pre- and post-filtering", *IEEE Trans. on Signal Processing*, pp. 1557-1571, June 2003.
16. A. K. Jain, *Fundamentals of Digital Image Processing*, Chapter 5, Prentice Hall, 1989.