Image and Video Applications Using TI DSPs

Edward J. Delp

Video and Image Processing Laboratory (*VIPER)*
School of Electrical and Computer Engineering
Purdue University
West Lafayette, Indiana, USA

email: ace@ecn.purdue.edu
http://www.ece.purdue.edu/~ace

Overview

- Introduction
- System Definition
- Discrete Wavelet Transform
- Still Image Watermarking
- Real-Time Error Concealment in Digital Video Streams
- SAMCoW and Video Streaming
- Future Research

Acknowledgement - Students

- Hyung Kim
- Dan Hintz
- Eugene Lin
- Eduardo Asbun

Introduction

- **Target Processors:** ***TMS320C5410*** **and TMS320C6201**
- **Project Includes:**
  - **Wavelet based video compression**
  - **Still image watermarking**
  - **Error concealment for digital video streams**
  - **Internet video streaming**

System Definition

- **Software Versions**
- ***TMS320C5410* platform**
- **TMS320C6201 hardware platform**
- **Development software**

Software Versions

- **Floating-point PC Code**
  - **Debugging and performance evaluation**
- **Fixed-point PC Code**
  - **Investigate the effects of fixed-point arithmetic**
- ***'C5410* DSP Code**
  - **Code for the *'C5410* DSP Simulator**
- ***'C6201* DSP Code**
  - **Code for the *'C6201* DSP EVM**
- **eXpressDSP compliance**

TMS320C5410 Platform

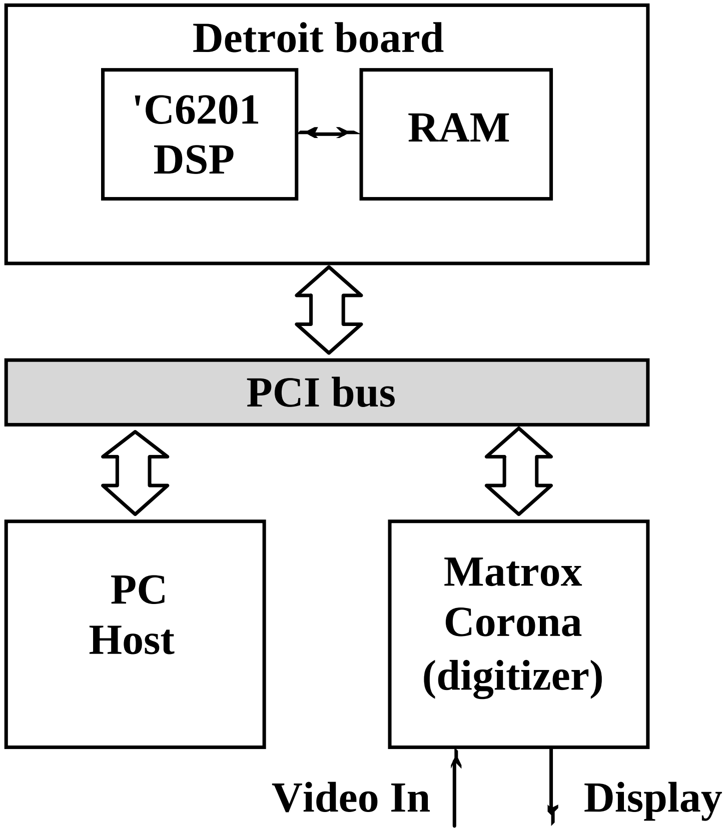
- ***‘C5410*** **Software Simulator**
- **Targeted to the Digital Still Camera**
- **Interface to memory controller is simulated in software**

Texas Instruments *TMS320C548*

- **Fixed-point DSP, 50 MHz clock cycle**
- **40-bit ALU and two independent 40-bit accumulators**
- **17-bit x 17-bit parallel multiplier**
- **192k words x 16-bit addressable memory space**

TMS320C6201 Platform

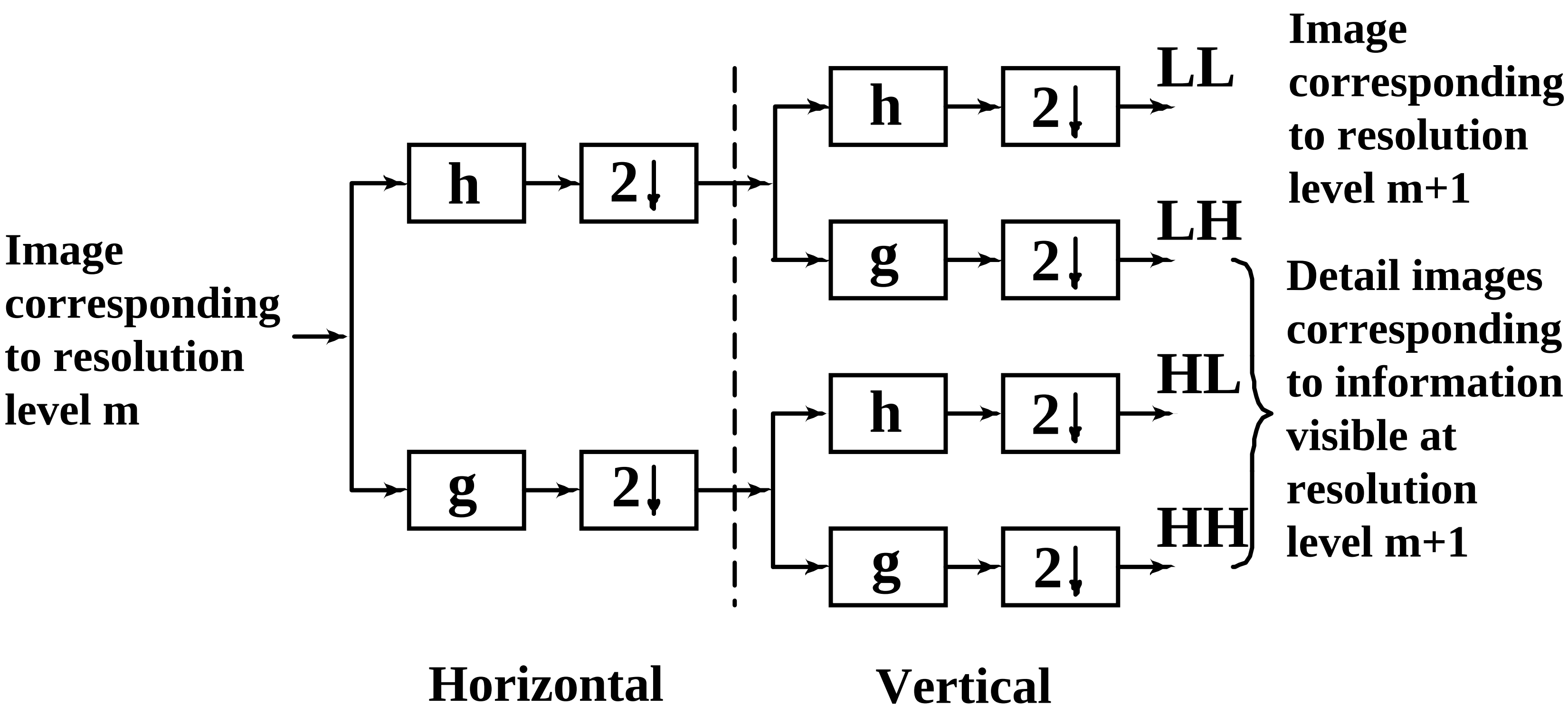
- **Rev. 2.1 'C6201 Detroit Board from Spectrum Signal**
- **Matrox Corona Video Capture Card**



Texas Instruments *TMS320C6201*

- **32-bit fixed-point DSP, 200 MHz clock cycle**
- **Based on VelociTI architecture**
  - **VLIW architecture**
  - **Increased instruction-level parallelism**
  - **Can issue up to 8 instructions per clock cycle**
- **8/16/32-bit data support (important for video and imaging)**

Wavelet Decomposition

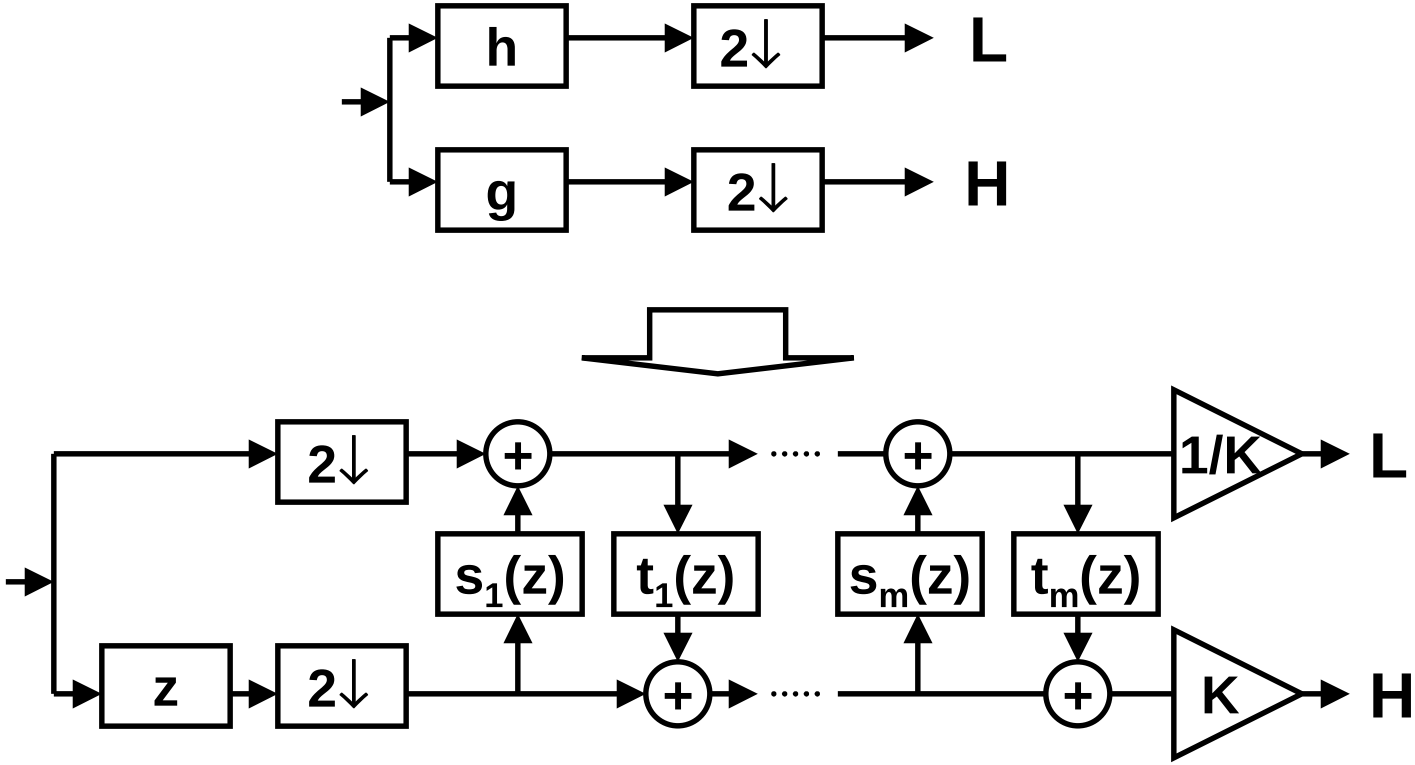


Wavelet Transform on the *'C6201*

- **Using Daubechies (9,7) wavelet filter pair**
- **To obtain perfect reconstruction, Whole-Sample Symmetric (WSS) extension is used at the image boundaries**
- **WSS also avoids coding artifacts**
- **Implemented using the lifting scheme to reduce computational complexity (approximately by half)**

Lifting Scheme

- **Wavelet transform is decomposed into multiple lifting steps**



Lifting Scheme

- Wavelet transform is decomposed into multiple lifting steps using the Euclidean algorithm

$$\begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} = \prod_{i=1}^{m} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix}$$

$$h_e(z) = \sum_k h_{2k} z^{-k} \qquad g_e(z) = \sum_k g_{2k} z^{-k}$$

$$h_o(z) = \sum_k h_{2k+1} z^{-k} \qquad g_o(z) = \sum_k g_{2k+1} z^{-k}$$

m=n/2+1(n is the filter length)

Implementation Issues

- **Floating-point arithmetic performed in a fixed-point DSP**
- **Using 16-bit arithmetic (Q6 notation)**
- **Memory management is critical (need to minimize access to external memory)**

Results: Wavelet Transform

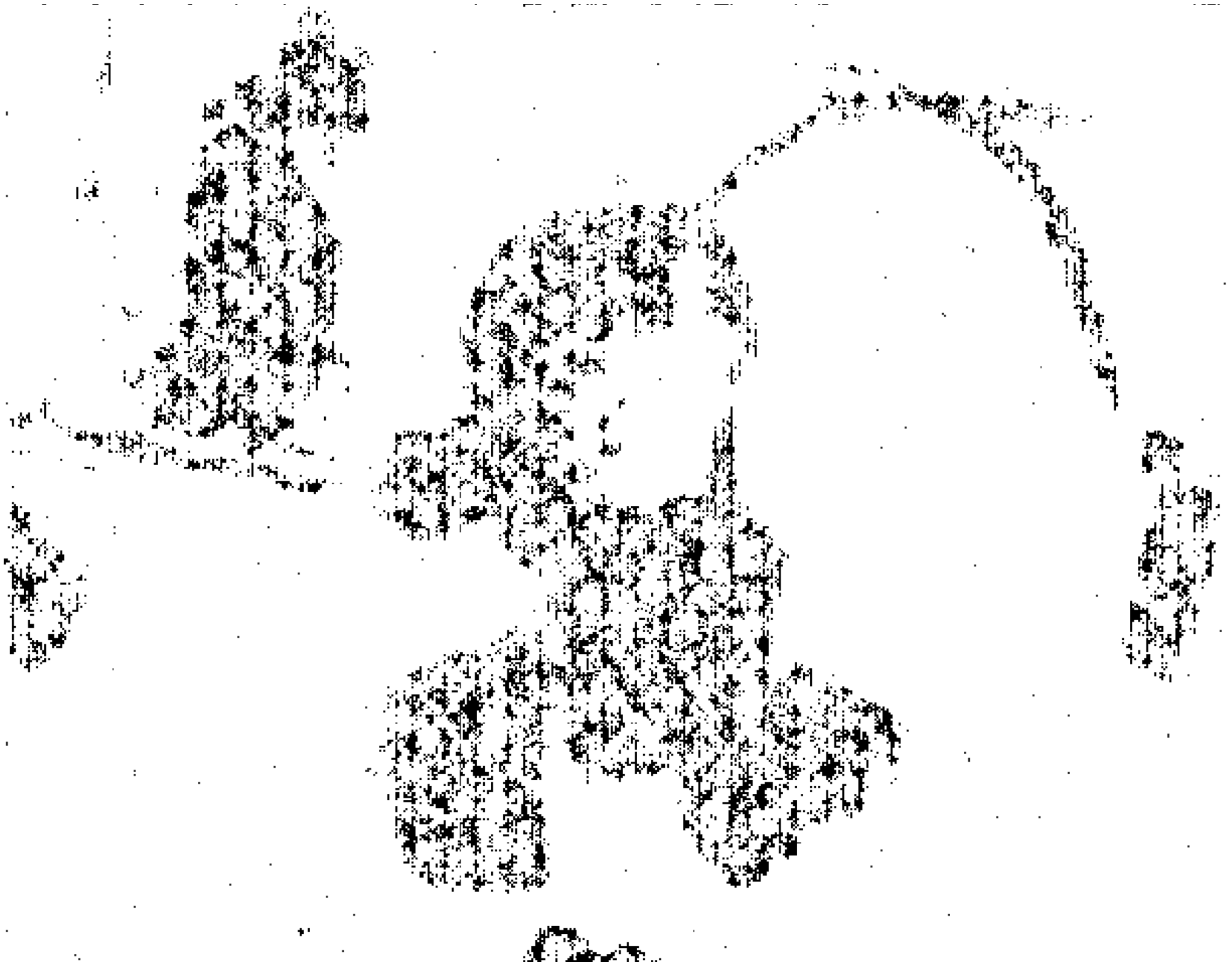
- **Performed on the *'C6201* fixed-point DSP**
- **3-level decomposition, 512x512 grayscale image**
- **Timing results**

	Cycles (millions)	Time (sec) (200 MHz *'C6201*)
Decomposition	**Unoptimized: 57.3** **Optimized, lifting: 16.6**	**~ 0.30** **~ 0.08**
Reconstruction	**Unoptimized: 99.7** **Optimized, lifting: 15.6**	**~ 0.50** **~ 0.08**

Results (Original Image)

Results (Reconstructed Image)

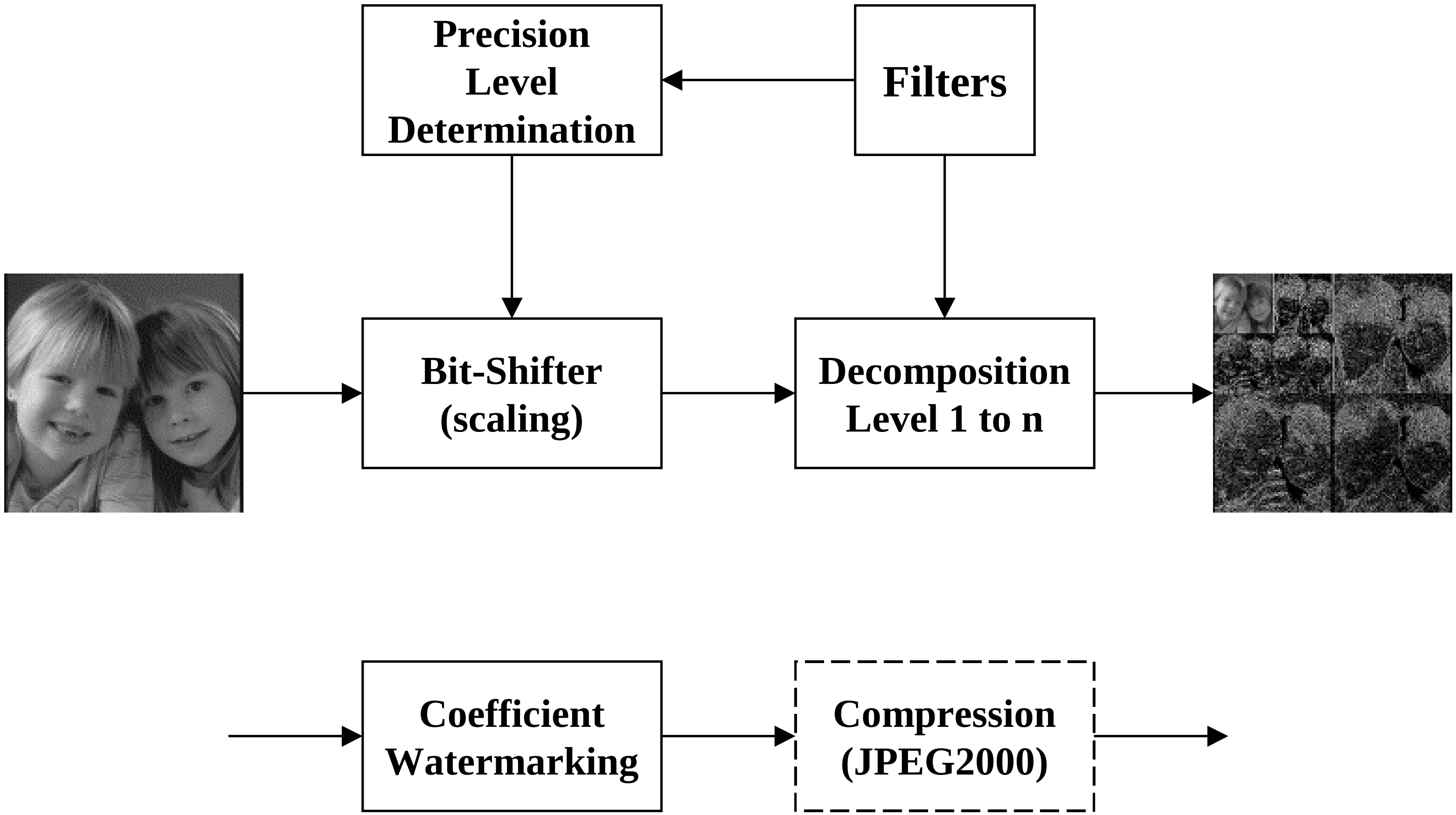
Results (Difference Image)



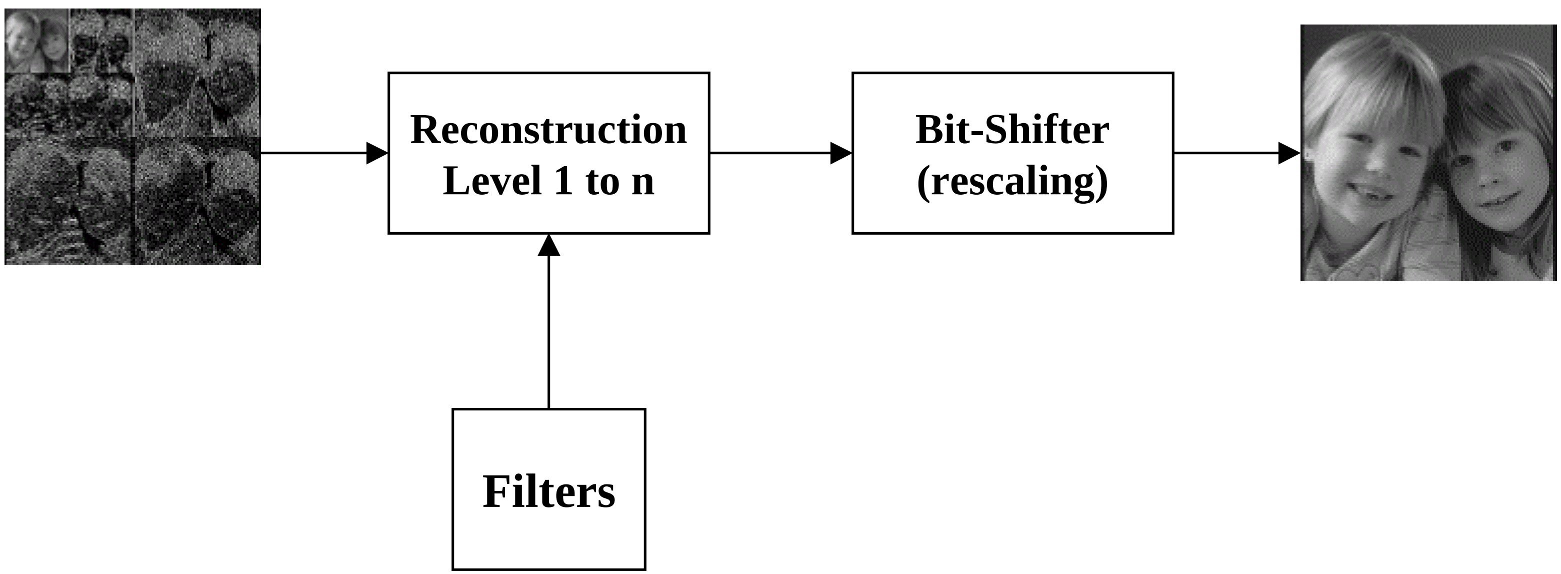
Still Image Watermarking

- **Aids in protecting intellectual property rights**
- **Offers forgery detection**
- **Chain-of-custody determination**

System Overview (Decomposition)



System Overview (Reconstruction)



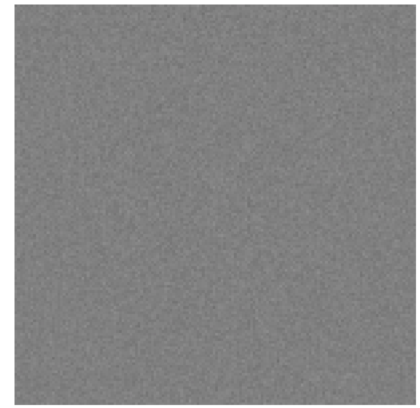
Precision Level for DSPs

- **Determined off-line (remember you have only 16 bits)**
- **Must determine the magnitude of the largest and smallest wavelet coefficients**
  - **determines the number of integer bits used**

Watermarked Images

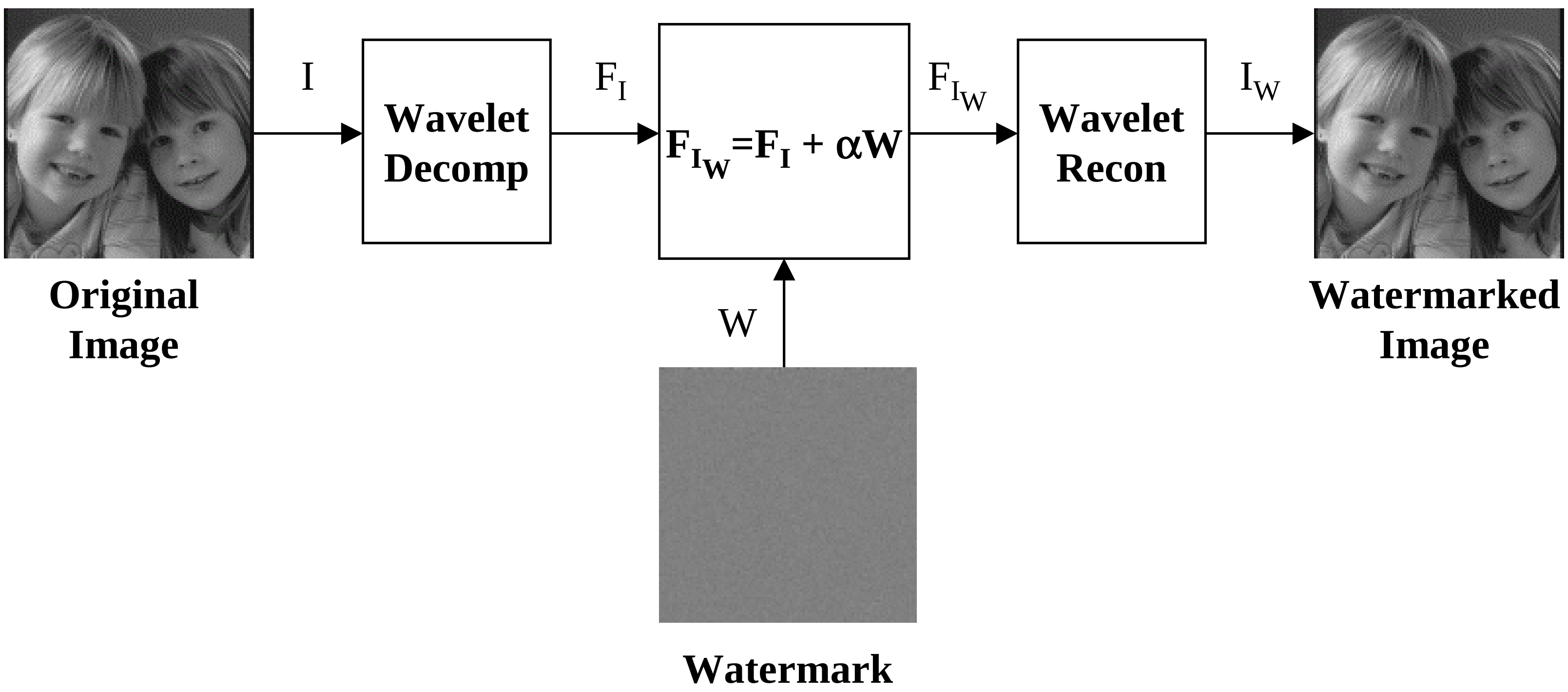
Original image

Watermarked image

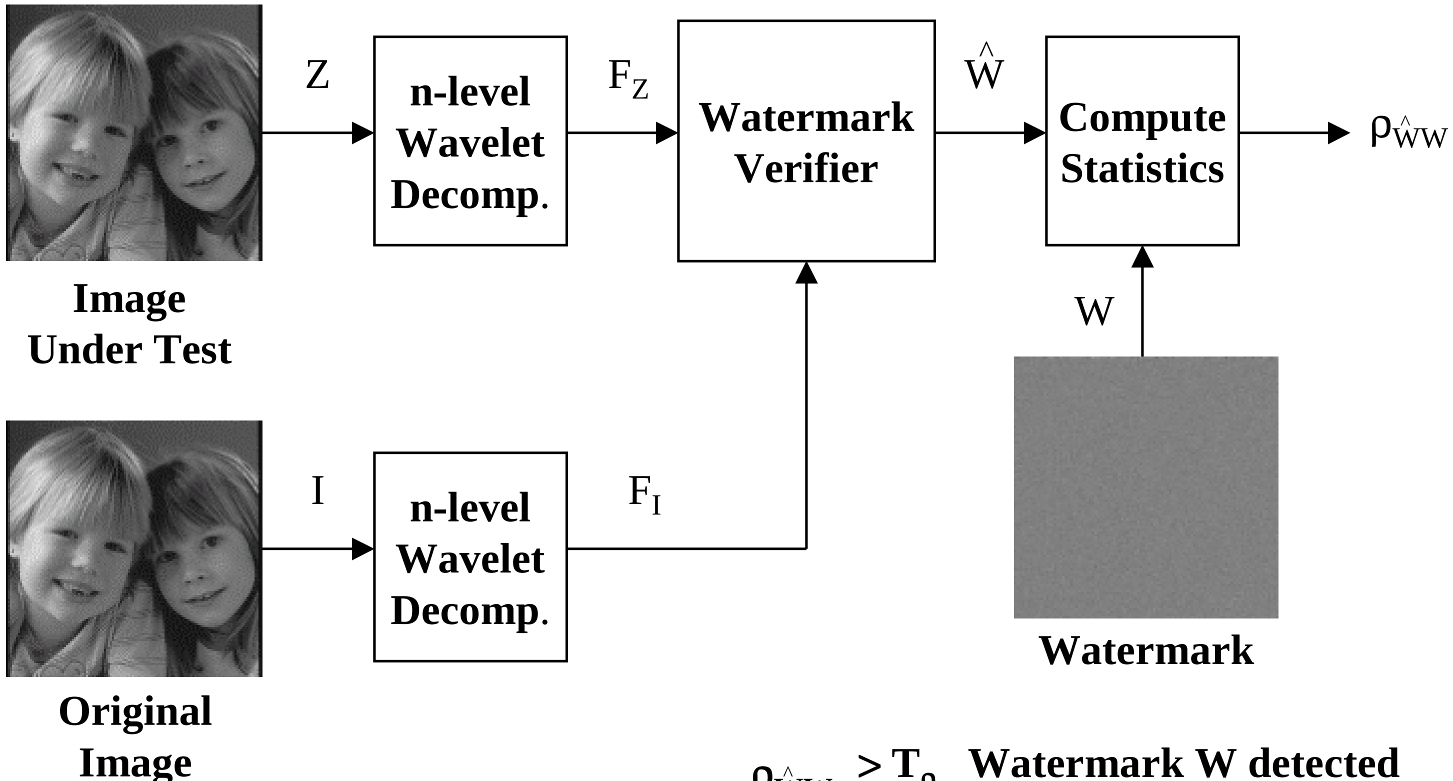


Watermark

Watermarking Technique Overview



Watermark Verification

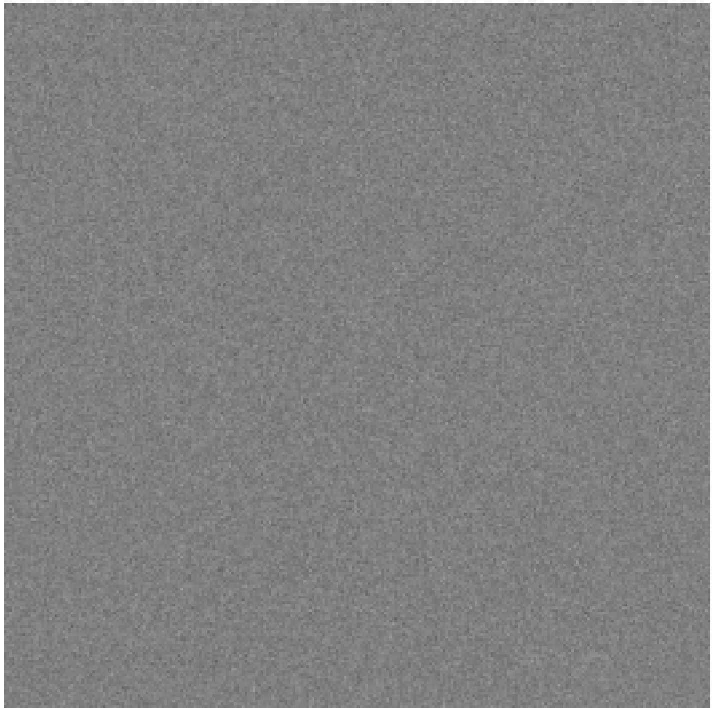


$$\rho_{\hat{W}W} \begin{cases} > T_\rho & \textbf{Watermark W detected} \\ \le T_\rho & \textbf{Watermark W not detected} \end{cases}$$

Watermark Creation

- **The Watermark consists of Gaussian distributed random numbers**
  - **Created off-line**
- **Watermark inserted into image**

Watermark Used



Gaussian Watermark

"Purdue" watermark

Experimental Results

Watermark Used	α	Embedder Version	Detector Version	ρ
Gaussian	0.1	Fixed point	Fixed point	0.539224
Gaussian	0.15	Fixed point	Fixed point	0.715037
Gaussian	0.2	Fixed point	Fixed point	0.807437
Gaussian	0.3	Fixed point	Fixed point	0.899317
Gaussian	0.4	Fixed point	Fixed point	0.939558
Gaussian	0.5	Fixed point	Fixed point	0.96038
Gaussian	1	Fixed point	Fixed point	0.989667
Gaussian	0.1	Floating point	Fixed point	0.541586
Gaussian	0.5	Floating point	Fixed point	0.960564
Gaussian	1	Floating point	Fixed point	0.989711
Purdue	0.1	Fixed point	Fixed point	0.038289
Purdue	0.5	Fixed point	Fixed point	0.269927
Purdue	1	Fixed point	Fixed point	0.658033
Purdue	0.1	Floating point	Fixed point	0.038289
Purdue	0.5	Floating point	Fixed point	0.27164
Purdue	1	Floating point	Fixed point	0.67358

Real-Time Error Concealment

- In data networks, channel errors or congestion can cause cell or packet loss
- When MPEG compressed video is transmitted, cell loss causes macroblocks and motion vectors to be removed from compressed data streams
- Goal of error concealment: Exploit redundant information in a sequence to recover missing data

Error Concealment

Original frame

Damaged frame

Approaches for Error Concealment

- **Two approaches for error concealment:**
  - **Active concealment: Use of error control coding techniques and retransmission**
    - **unequal error protection**
  - **Passive concealment: The video stream is post-processed to reconstruct missing data**
- **Passive concealment is necessary:**
  - **where active concealment cannot be used due to compliance with video transmission standards**
  - **when active concealment fails**

Use of Error Concealment

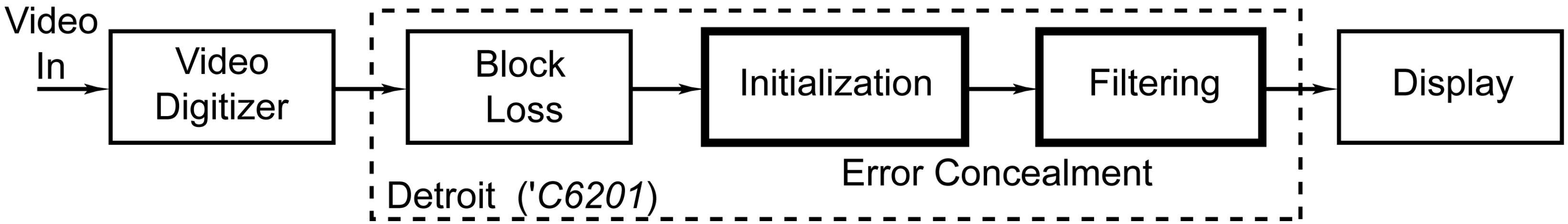
- **All video decoders that will be used in consumer applications, such as set-top decoder boxes must implement some form of passive error concealment**

- **This problem is interesting in that <u>it absolutely requires real-time implementation</u>**

- **Digital signal processors (DSPs) are well suited for the demands of real-time processing**

Texas Instruments *TMS320C6201*

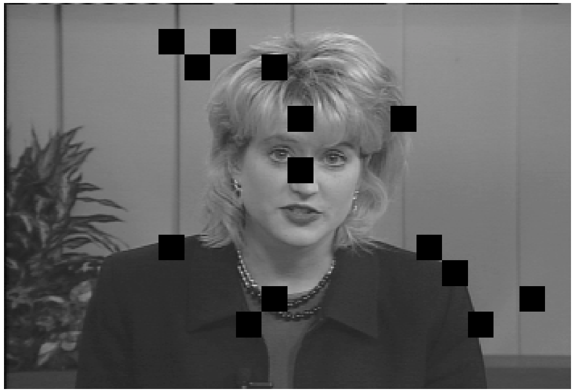
- **Fixed-point DSP, 200 MHz clock cycle**
- **Based on VelociTI architecture**
  - **VLIW architecture**
  - **Can issue up to 8 instructions per clock cycle**



Block Diagram of Spatial Error Concealment System

Results

- **At this time, the overall system is running in real-time (30 frames/second) on 320x240 size video sequences**



Damaged frame

Recovered errors

Implementation Issues

- **Error concealment is a module of an MPEG decoder**
- **Spatial technique, or modified temporal technique, used in damaged I frames**
- **Temporal technique used for damaged P and B frames**
- **Access to frame buffer in Motion Compensation module required for temporal technique**
- **Use of internal data memory of *'C6201* necessary to avoid high penalty of accessing external memory**

Implementation Issues

- **A single 200 MHz *'C6201* is needed for the implementation of the spatial technique**
- **In spatial concealment, each pixel is median filtered twice**
- **Capable to process up to 89 frames/sec**
- **Computational requirements of temporal technique are lower than spatial technique**

To Read More About It

E. Asbun and E. J. Delp, "Real-Time Error Concealment in Compressed Digital Video Streams," Proceedings of the Picture Coding Symposium 1999, April 21-23, 1999, Portland, Oregon, pp. 345-348.

Video Streaming

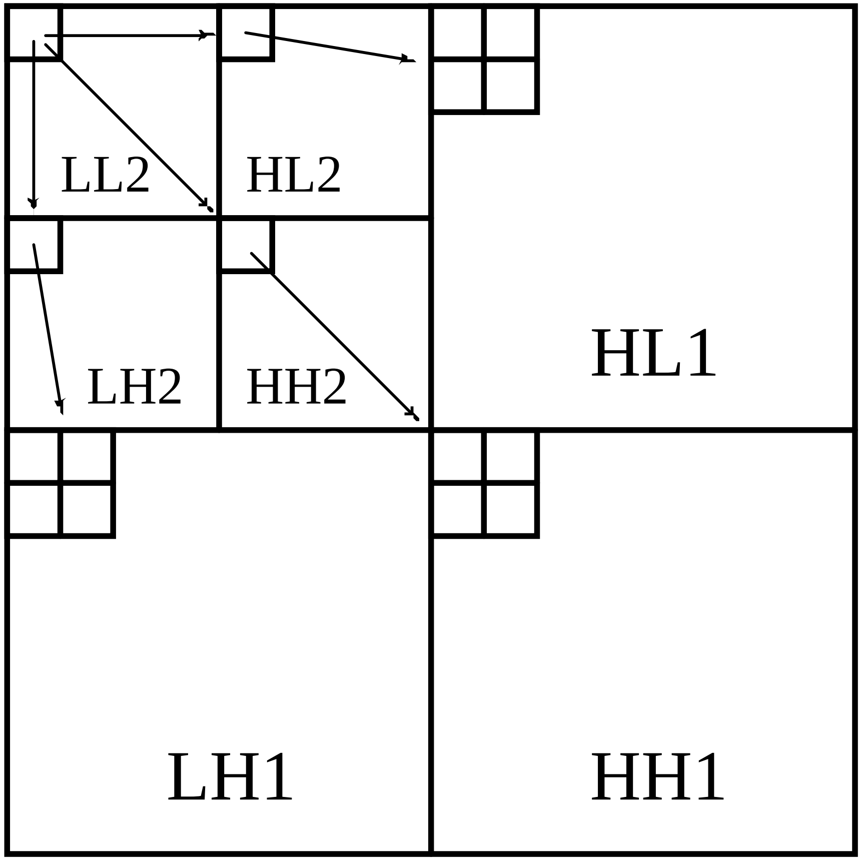
- **Examine rate scalable video compression**
- **Rate scalability is one of the most important scalability modes for streaming video over packet networks**
- **Scalable Adaptive Motion Compensated Wavelet (*SAMCoW*) rate scalable video compression algorithm**
- ***SAMCoW* uses the Color Embedded Zerotree Wavelet (*CEZW*) rate scalable, color image compression algorithm**

EZW

- **Embedded Zerotree Wavelet (EZW) is a wavelet-based, rate scalable image compression technique**
- **EZW exploits interdependence between subbands of wavelet decomposition**
- **Coefficients are encode via significance maps using a hierarchical tree**
- **Quantize and encode the subband data via successive approximation**

Embedded Zerotree Wavelet (EZW)

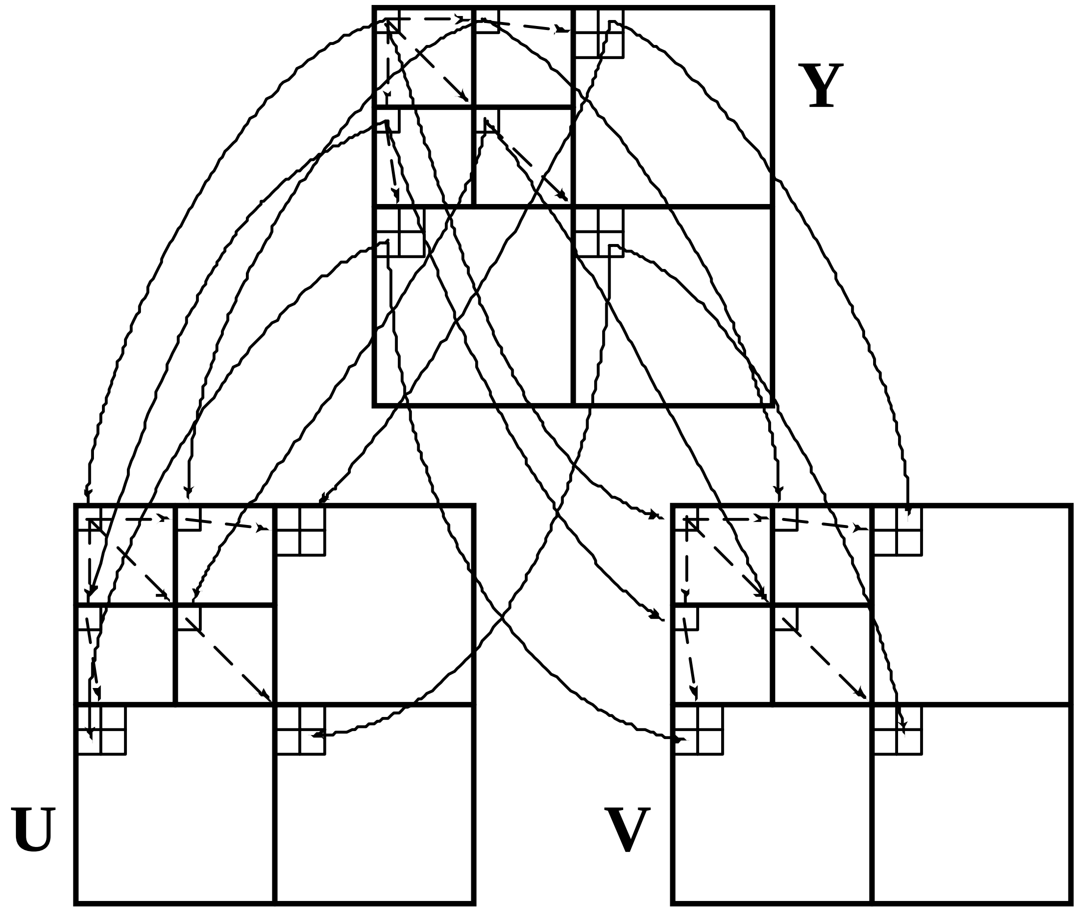
- **EZW was developed for grayscale images**



Color Embedded Zerotree Wavelet (*CEZW*)

- **For color images, EZW is applied on each color component independently**
- **A unique spatial orientation tree in the YUV color space is used**
- ***CEZW* exploits the interdependence between color components to achieve a higher degree of compression**

Color Embedded Zerotree Wavelet (*CEZW*)



Arithmetic Coding

- **Lossless Coding of EZW symbols**
- **Adaptive arithmetic coding to incorporate learning**

Results: EZW

- **EZW obtained on the *'C6201* fixed-point DSP**
- **3-level decomposition, 176x144 grayscale image**
- **Timing results**

	Cycles (millions)	Time (sec) (200 MHz *'C6201*)
Encode	**42.6**	**0.23**
Decode	**35.9**	**0.18**

Results EZW

Original

Reconstruction

Future Work

- Further investigate watermarking techniques
- Continue efforts on Internet streaming