JPEG2000

Compression standard for interactive imaging

David Taubman

The University of New South Wales

Image Compression Preliminaries

- Array of sample intensities (e.g. 8 bits)
  - Colour: red, green and blue samples
  - 64 x 64 (small thumbnail)
  - 1024 x 1024 (small photographic)
  - 64k x 64k (typical GIS/military image)
- Lossless compression
  - Only ~2:1 compression (photographic)
- Lossy compression
  - Allows up to ~100:1 compression
  - Can’t recover original samples exactly

Development of JPEG2000

- Developed by “JPEG” Working Group
  - ISO/IEC JTC1/SC29/WG1
  - Major international effort!
- 1996: work item proposed by Ricoh
- Nov 1997: 24 submissions evaluated
  - WTCQ selected after extensive testing
- Nov 1998: major algorithm change
  - EBCOT proposal adopted
- Jan 2001: part 1 became IS 15444-1
- Parts 1-6 complete; 8-11 ongoing

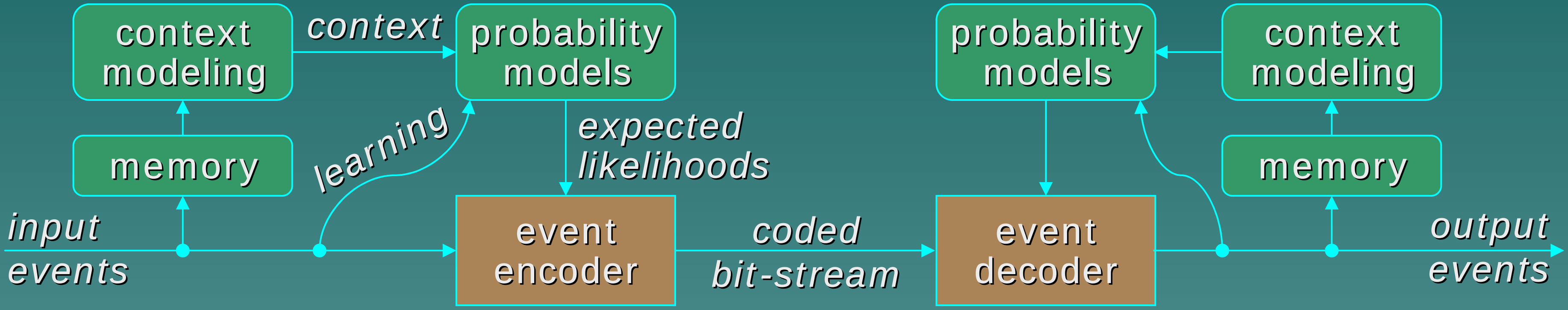
Some Benefits of JPEG2000

- Compress once – decompress many ways
  - one codestream, many applications (example)
- Flexible information progressions
  - by quality, by resolution (size), top down, ...
- State-of-the-art compression efficiency
  - smallest compressed size, for given quality
- Compressed domain editing
  - crop images without accumulation of errors
- Content not limited by rendering devices
  - Limited hardware still handles complex images

Principles: Statistical Redundancy (1)

- Importance of prior expectations
  - Consider 4 events (e.g., snow, hail, rain, fine)
    - If we know it is summer, snow & hail highly unlikely
    - If we know precipitation is rare, only “fine” is likely
  - No need to spend 2 bits indicating the outcome
    - e.g., send 0 if fine; 10 if rain; 110 if snow; 111 if hail
    - expect to send only 1 bit (0=fine) most of the time
- Lessons to learn
  - Average bit-rate depends on prior likelihoods
  - Likelihoods depend on context
    - e.g., “it is summer” or “precipitation in region is rare”

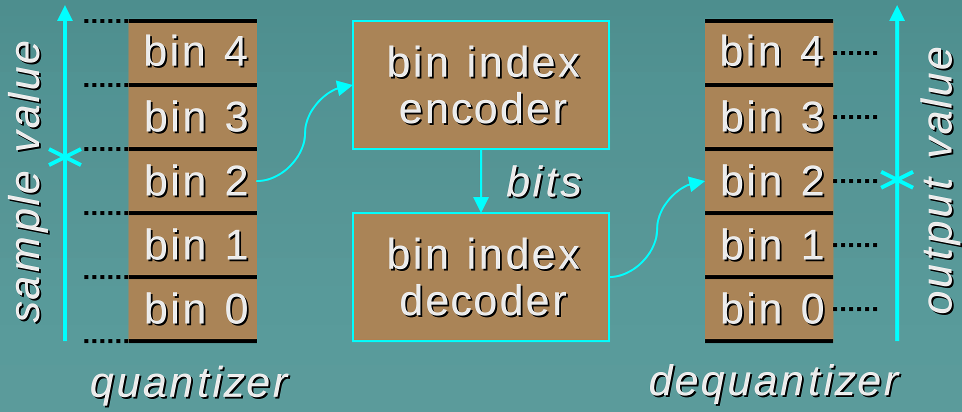
Principles: Statistical Redundancy (2)



- Context modeling
  - – exploit knowledge of events already encoded
    - knowledge is also available at decoder
- Probability modeling
  - – learn likelihoods of events in each context
    - use previously coded events in same context

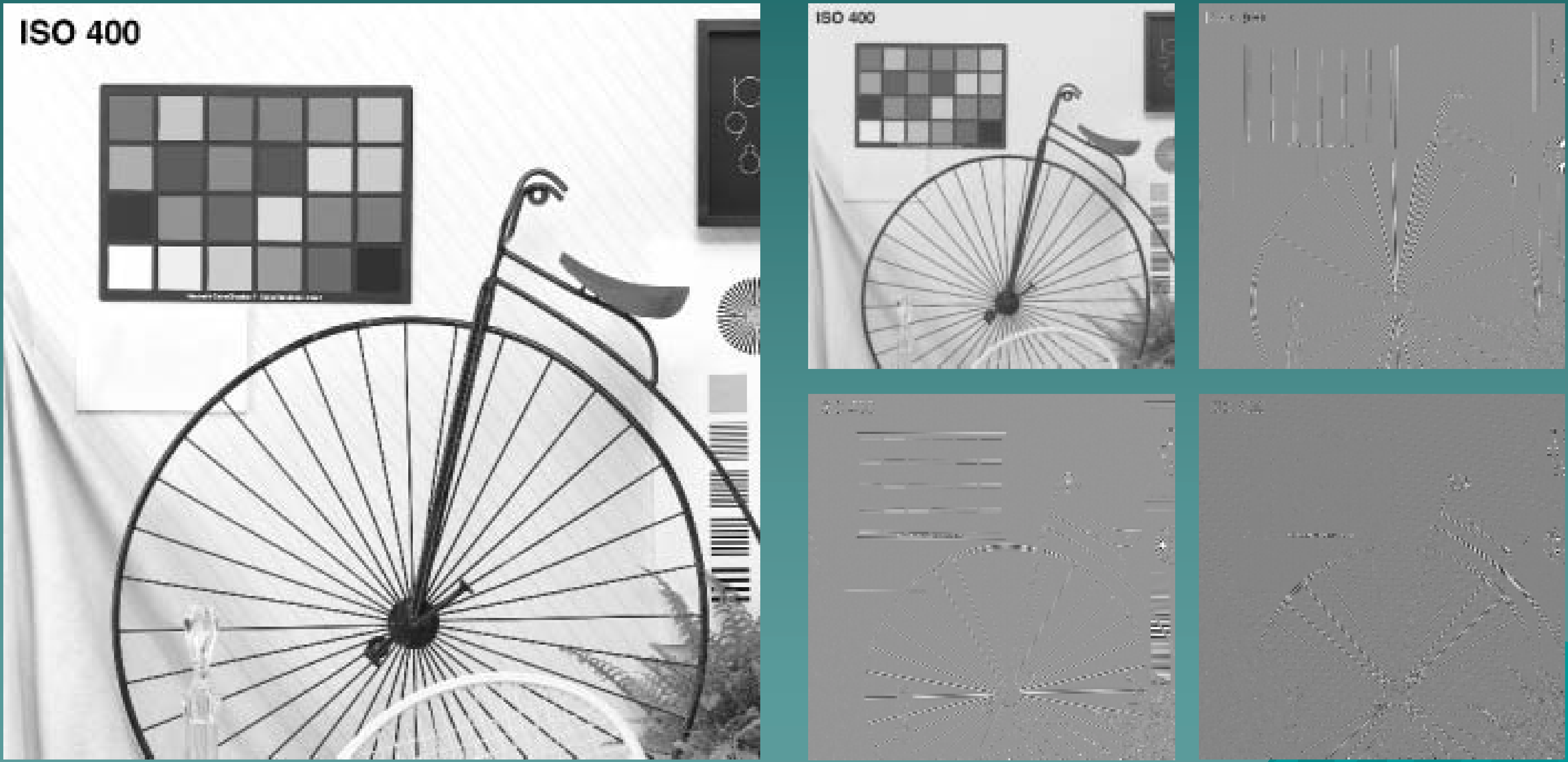
Need for Quantization

- Coded *events* can be image sample values
  - Code 8 bit/sample (bps) image with < 8 bps
  - Process is fully reversible (lossless)
  - But compression ratio not usually large – ~2:1
- Need to quantize sample values into bins

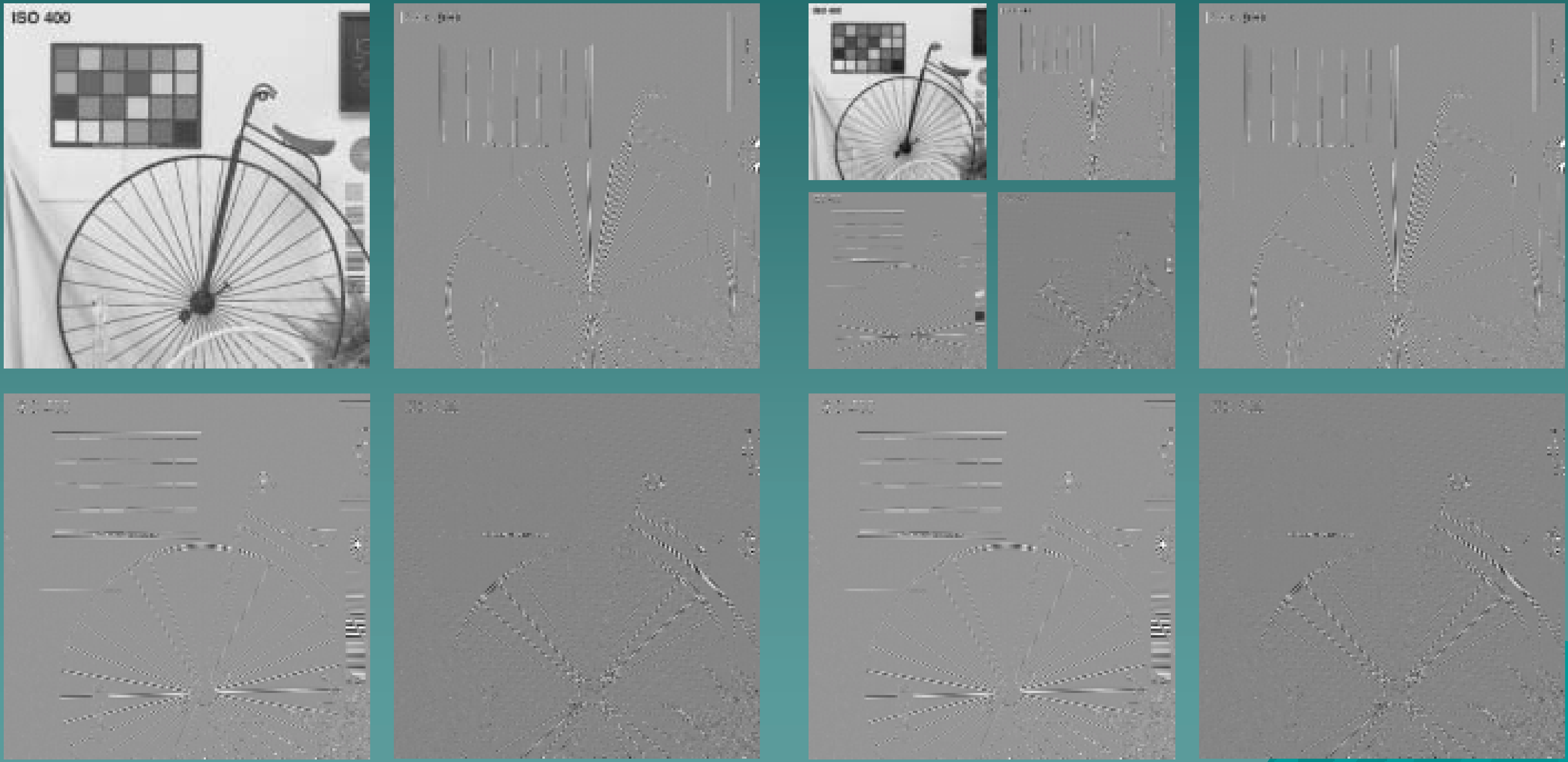


  - Reduce coded bits
  - Cost is distortion
  - Click for example
    - 3,1.5,0.8,0.5 bps

Principles: Wavelet Transform (1)

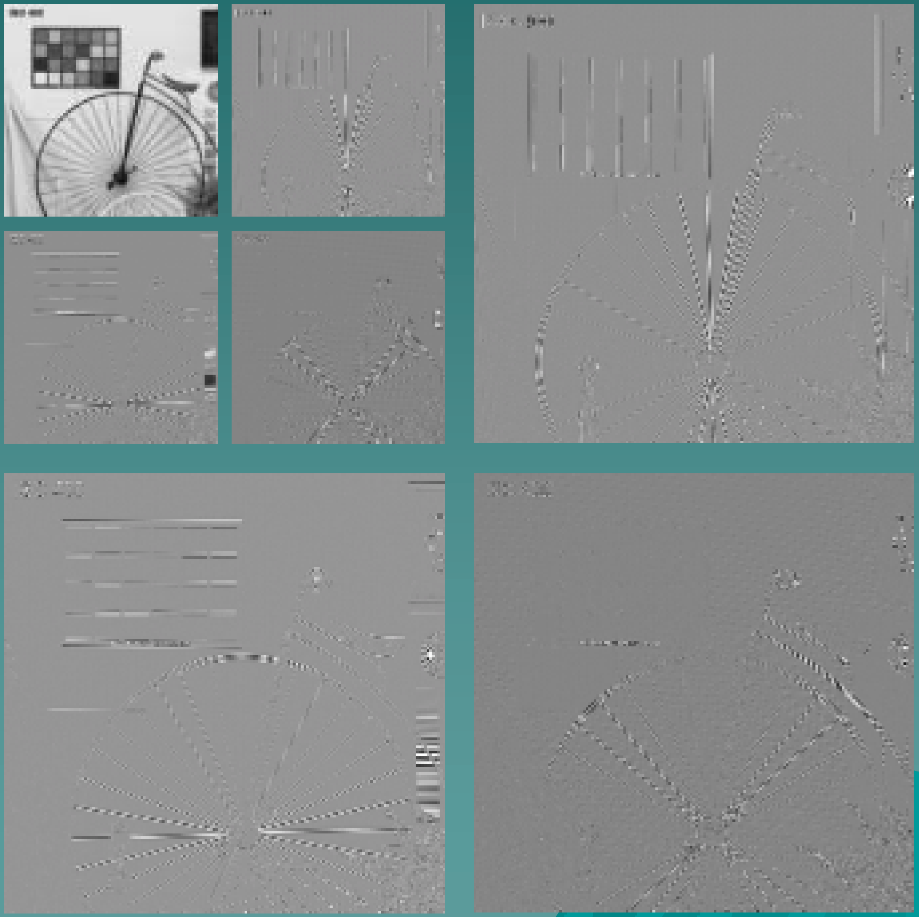


Principles: Wavelet Transform (2)

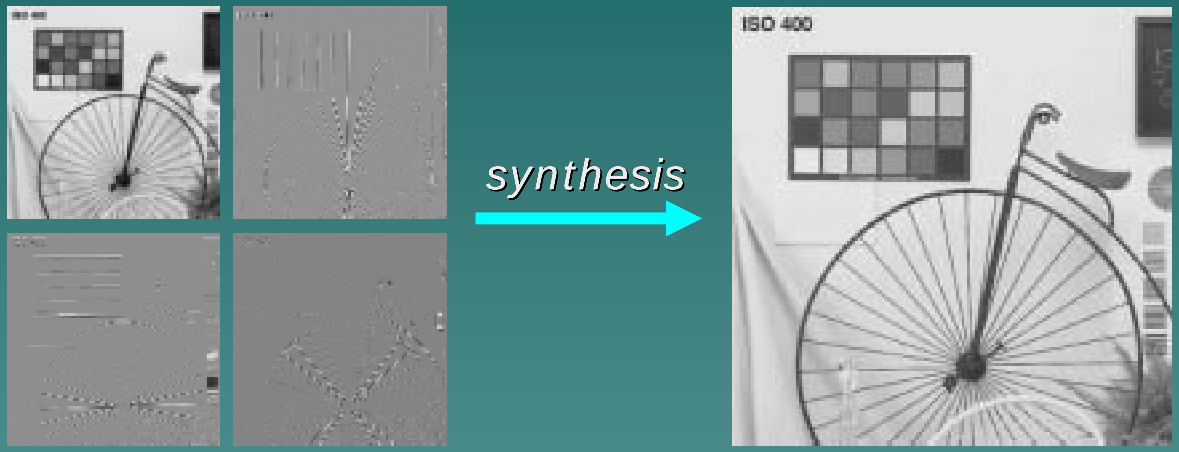


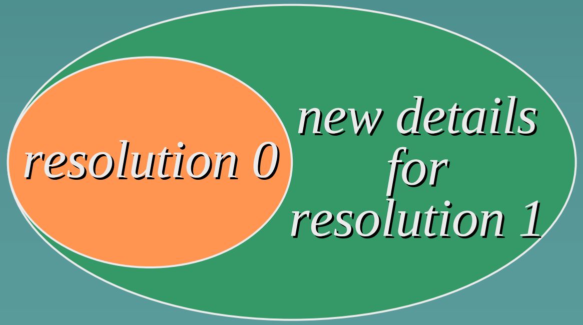
Principles: Wavelet Transform (3)

- Subband images from
  - low- & high-pass filtering
  - plus sub-sampling
- Num samples preserved
  - same as original image
- Transform is invertible
  - get original image back from subband images
- Quantize subbands
  - Click for example
    - 1,.5,.25,.12,.06,.03 bps



Principles:
Resolution Scalability (1)



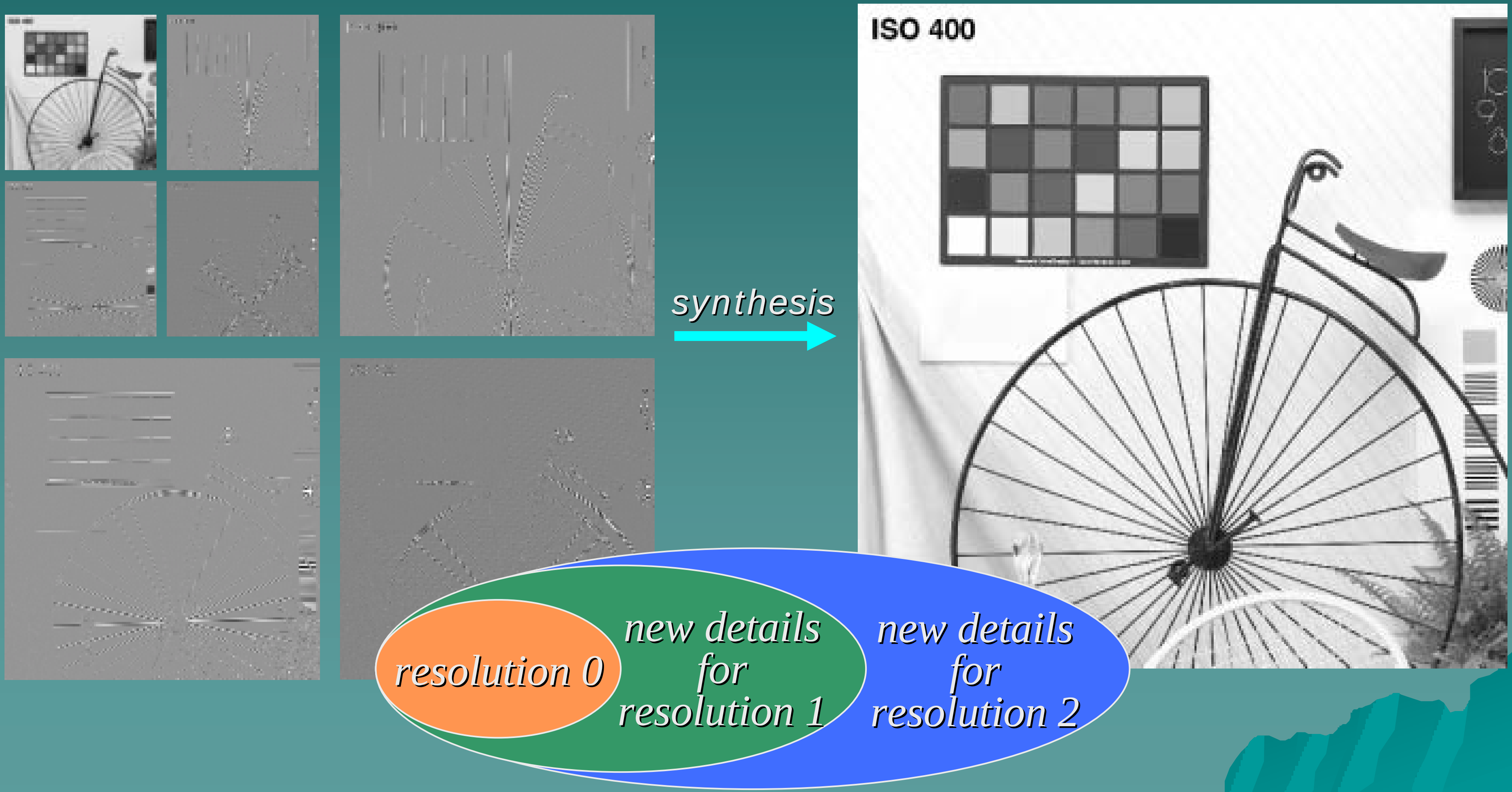


resolution 0

new details for resolution 1

Embedded codestream

Principles:
Resolution Scalability (2)



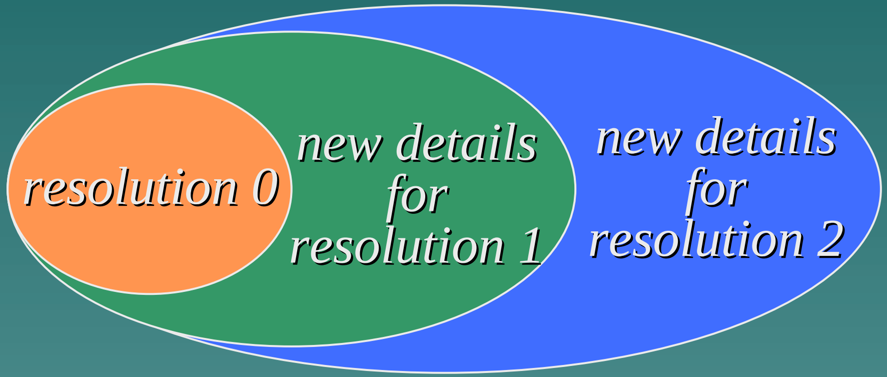
synthesis

resolution 0

new details for resolution 1

new details for resolution 2

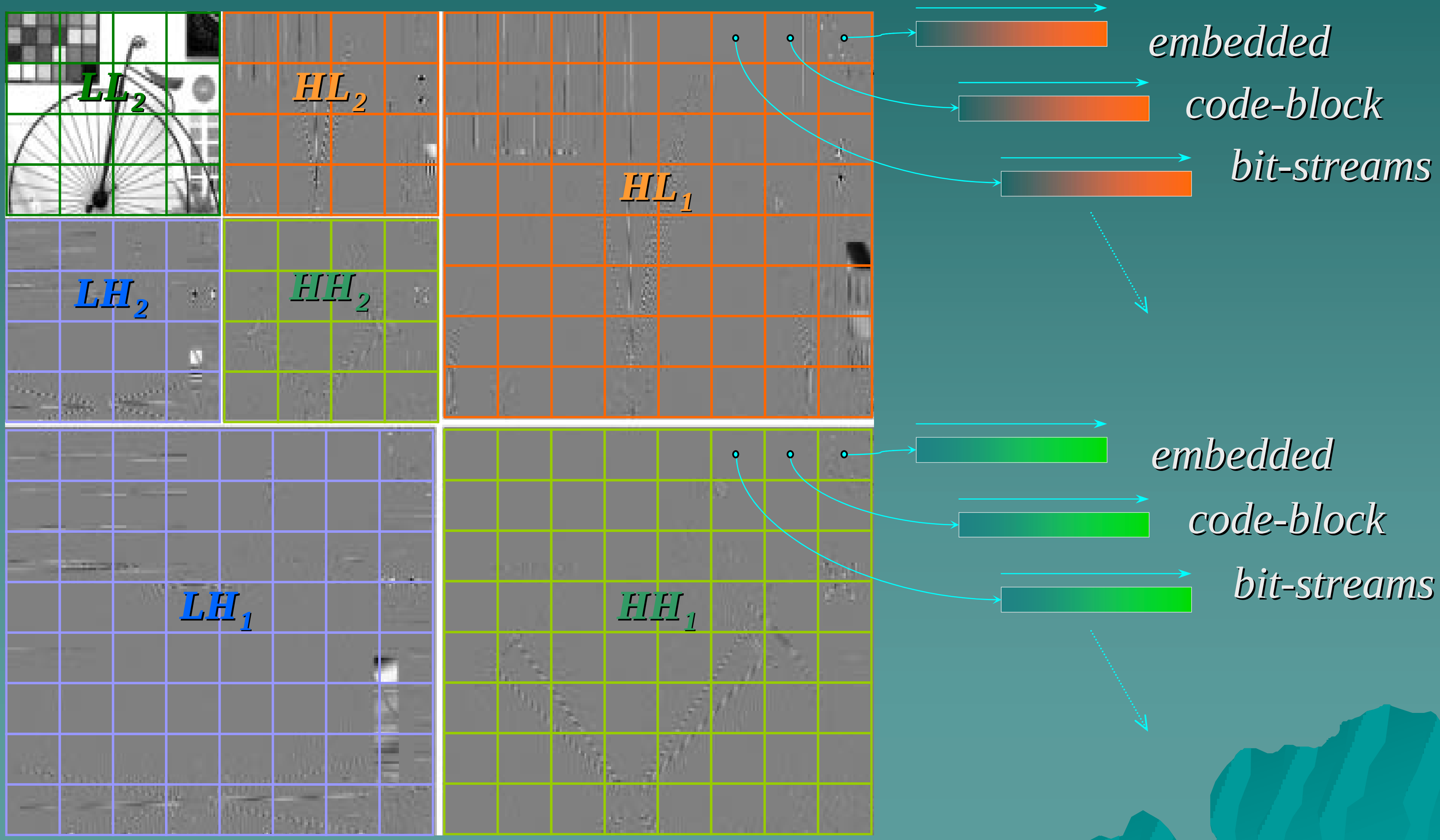
Principles: Resolution Scalability (3)



Embedded codestream

- Can choose image size
  - – after compression
  - – prior to decompression
  - – during transmission

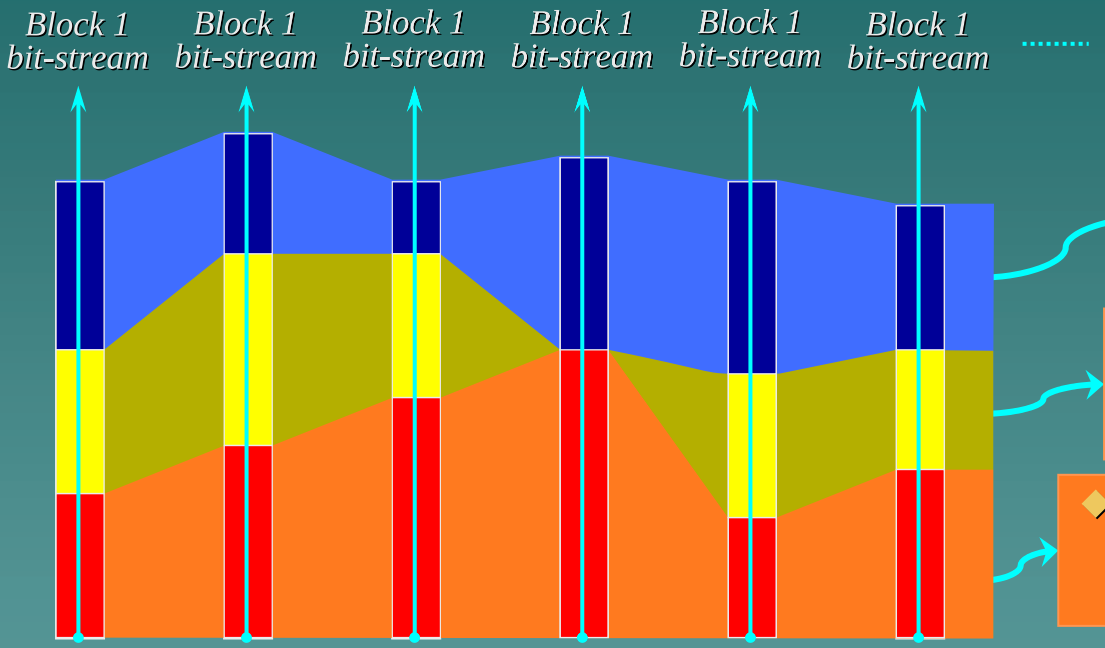
Principles: Embedded Block Coding (1)



Principles:
Embedded Block Coding (2)

- Each subband image divided into blocks
  - typically 32x32 or 64x64 samples per block
  - each block coded independently
- Embedded block coding algorithm
  - sample accuracy progressively refined
    - quantizer bin size effectively shrinks as bit-stream progresses
    - ultimate representation lossless (no quantization)
  - highly efficient representation
    - sophisticated context modeling within each block
    - adaptive probability modeling within each block

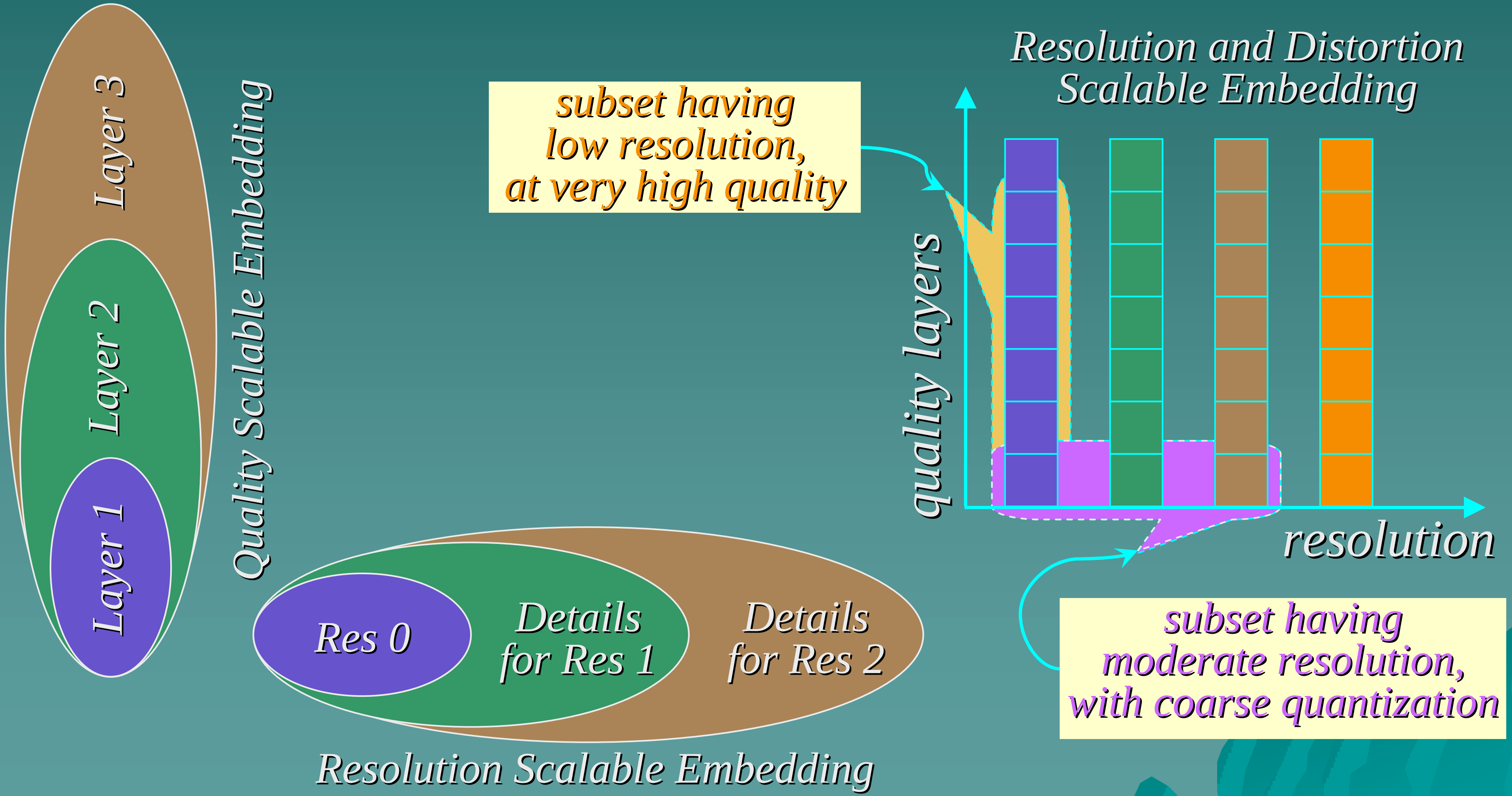
Principles: Abstract Quality Layers



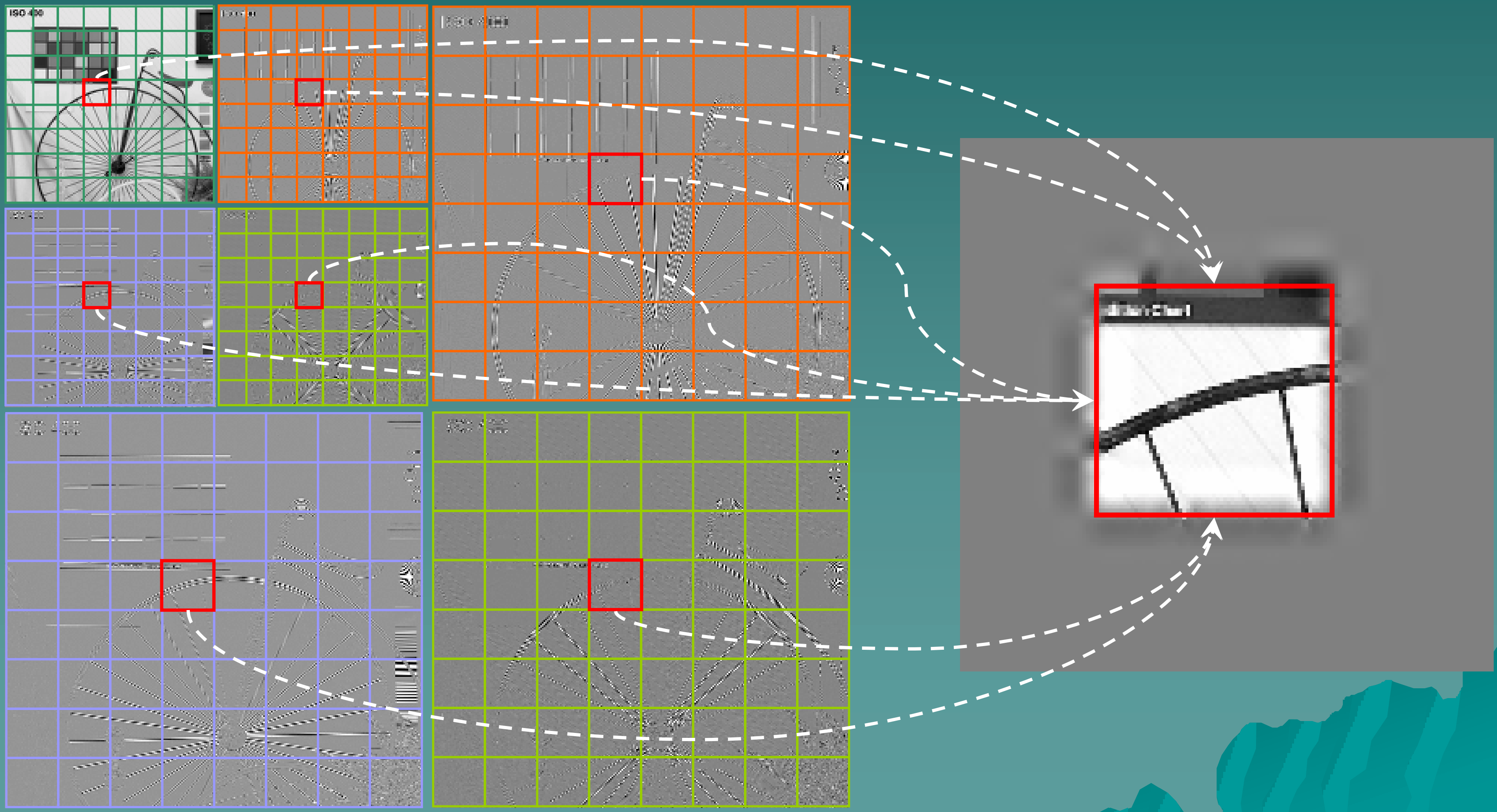
- Layer 3
  - finest quantization
  - best imagery
- Layer 2
  - finer quantization
  - better imagery
- Layer 1
  - coarse quantization
  - low quality imagery

- Block contributions to each layer optimized
  - Each block bit-stream optimally truncated
  - Permits custom interpretation of image quality

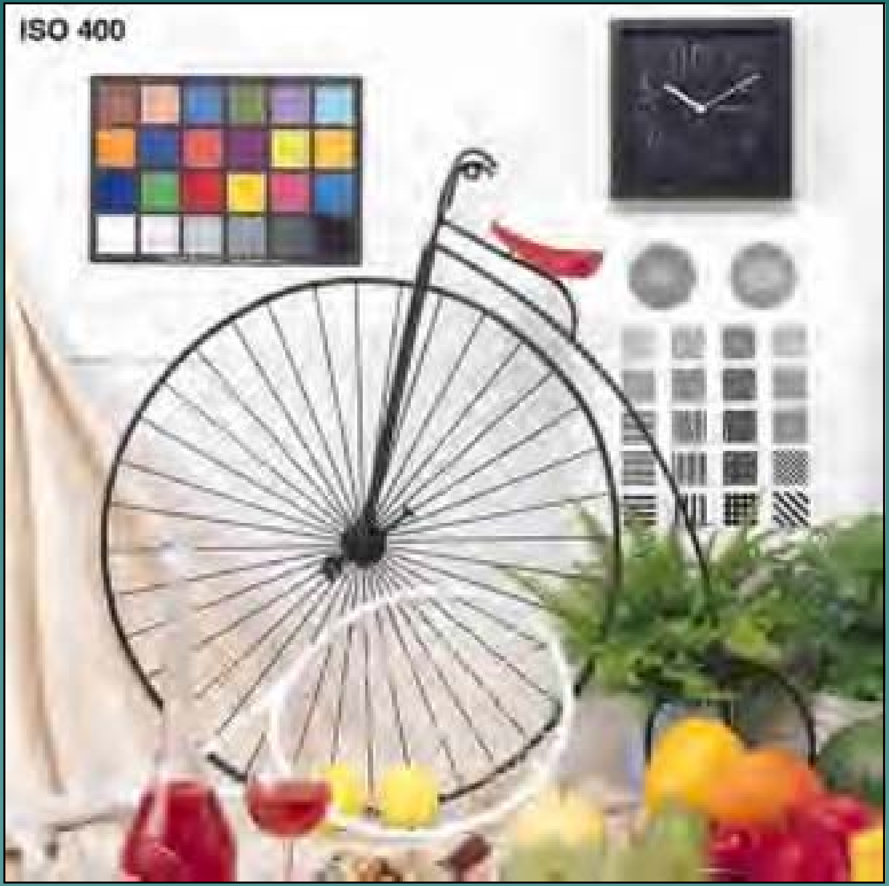
Principles: Dimensions of Scalability

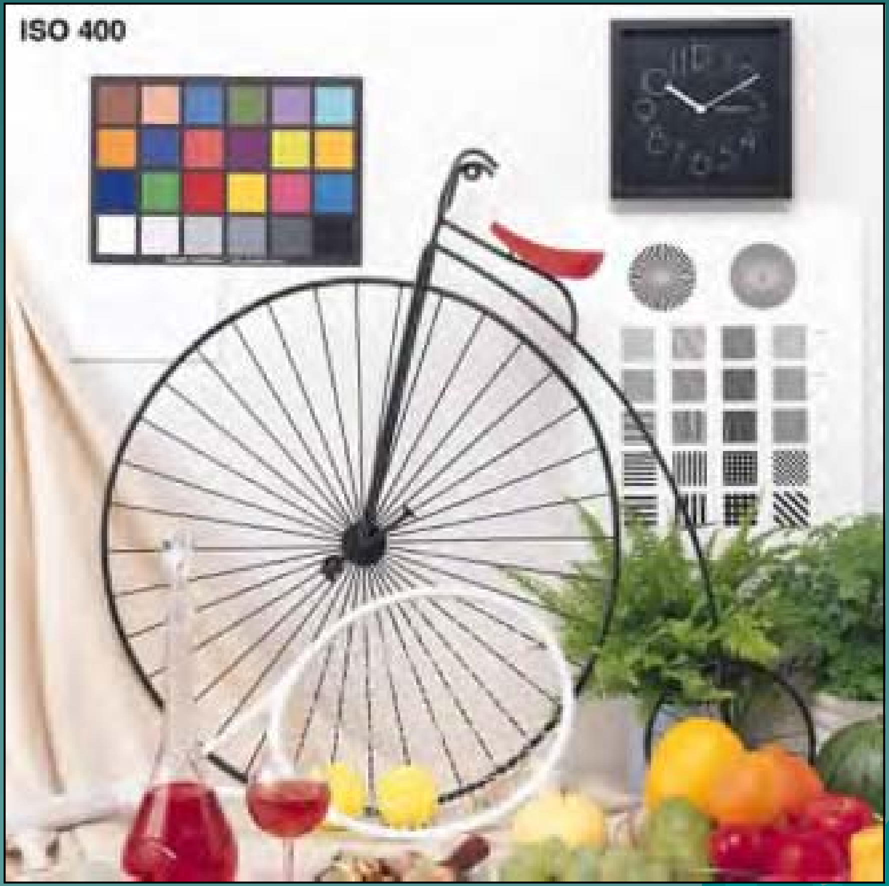

Layer 3
Layer 2
Layer 1
Quality Scalable Embedding
Res 0
Details for Res 1
Details for Res 2
Resolution Scalable Embedding
Resolution and Distortion Scalable Embedding
quality layers
resolution
subset having low resolution, at very high quality
subset having moderate resolution, with coarse quantization

Principles:
Spatial Access with Code-Blocks

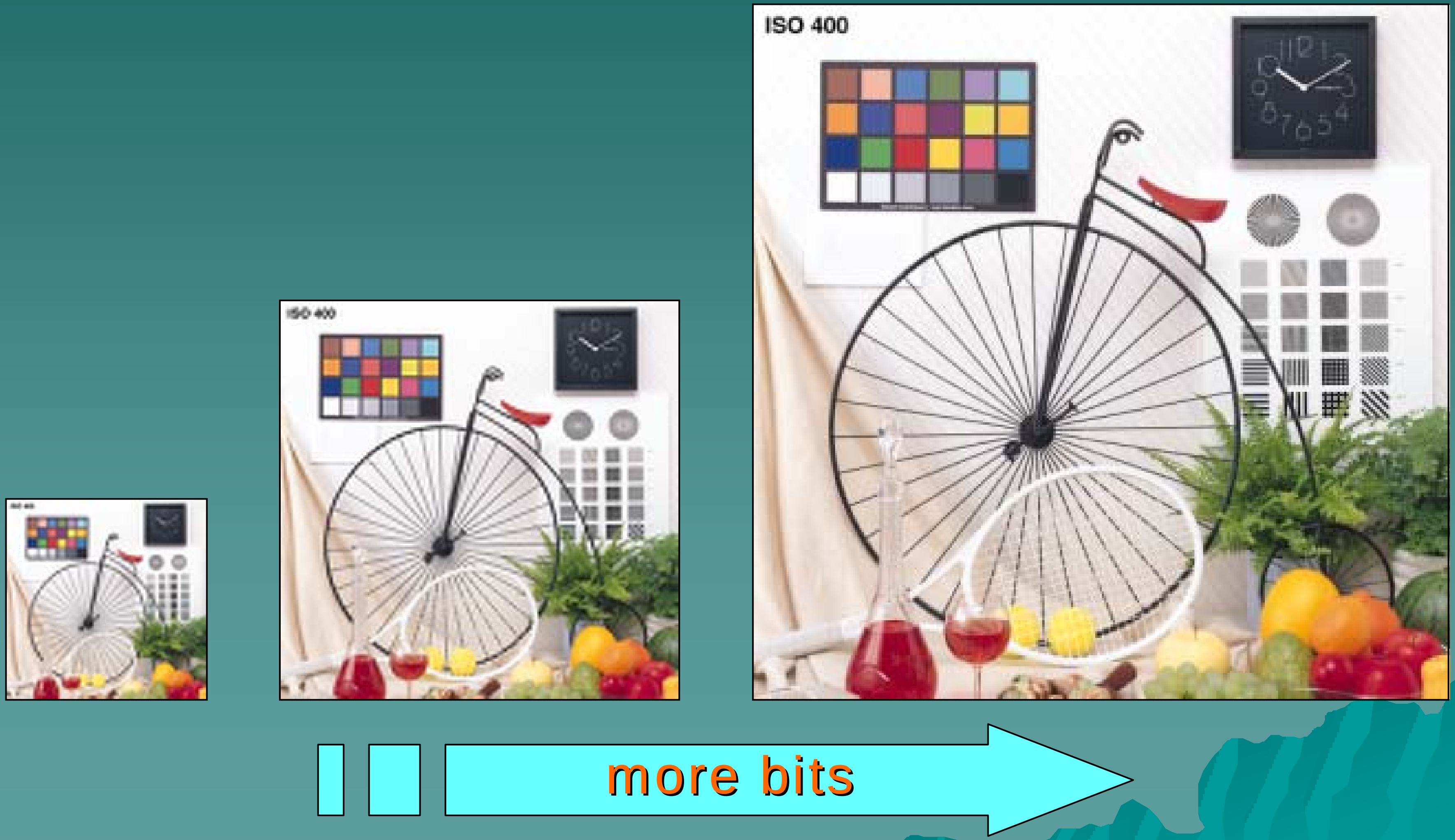


Information Sequencing: Layer Dominant (LRCP)

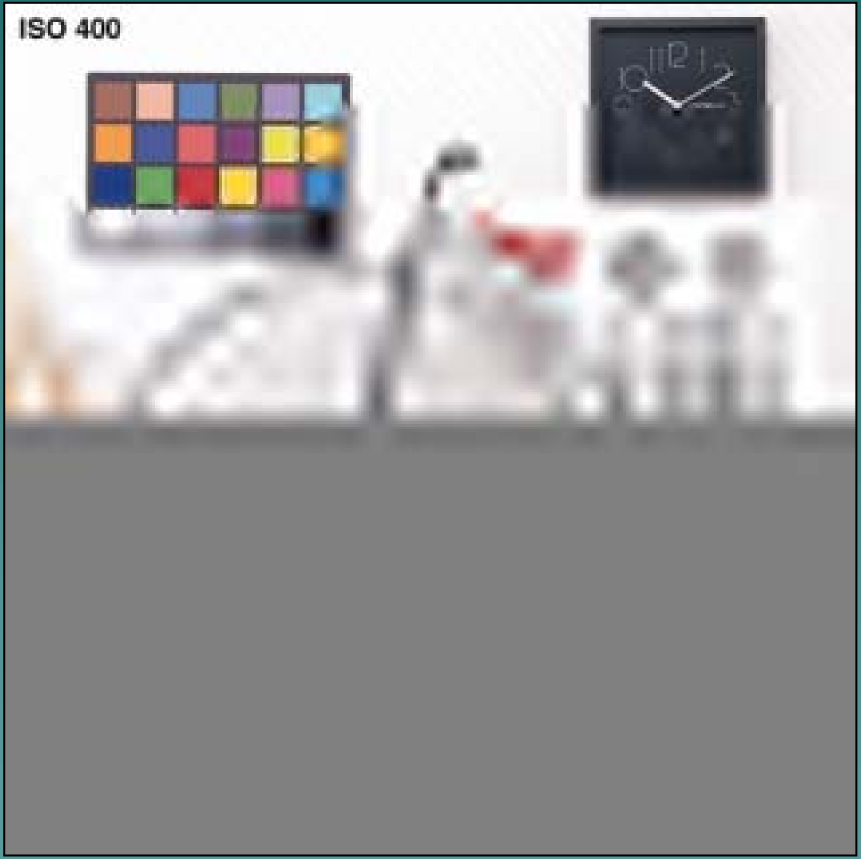


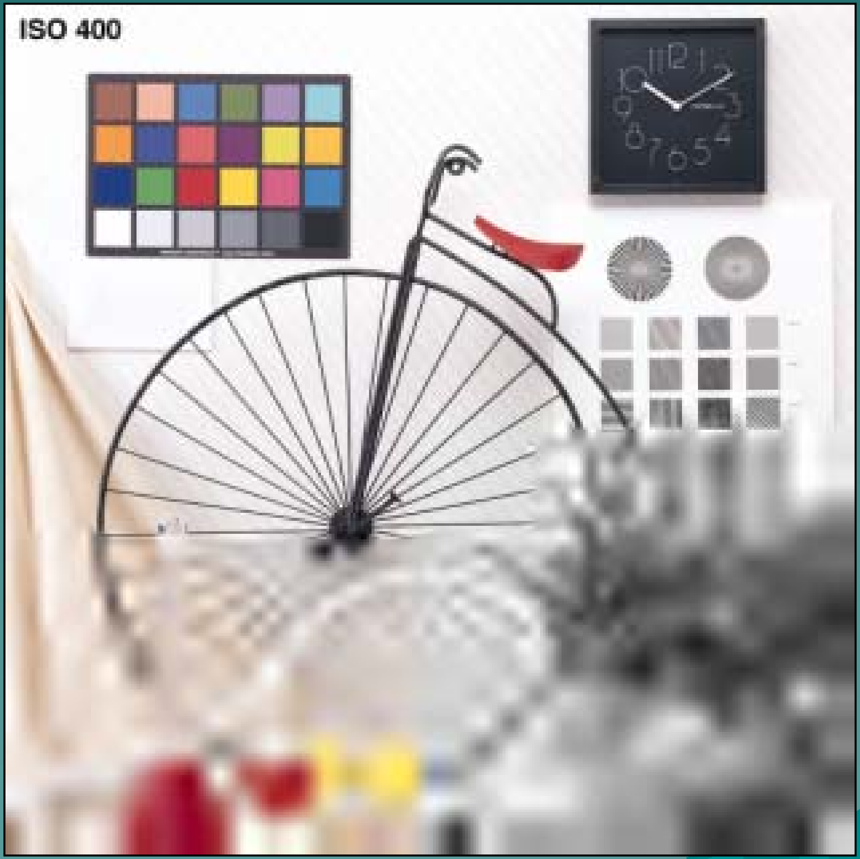


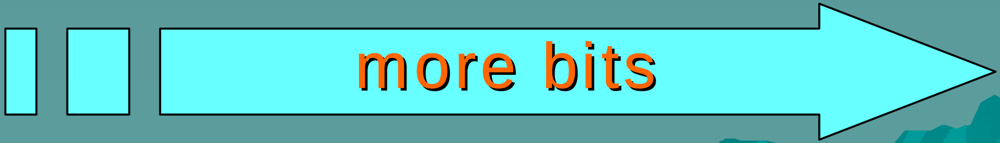
Information Sequencing: Resolution Dominant (RLCP)



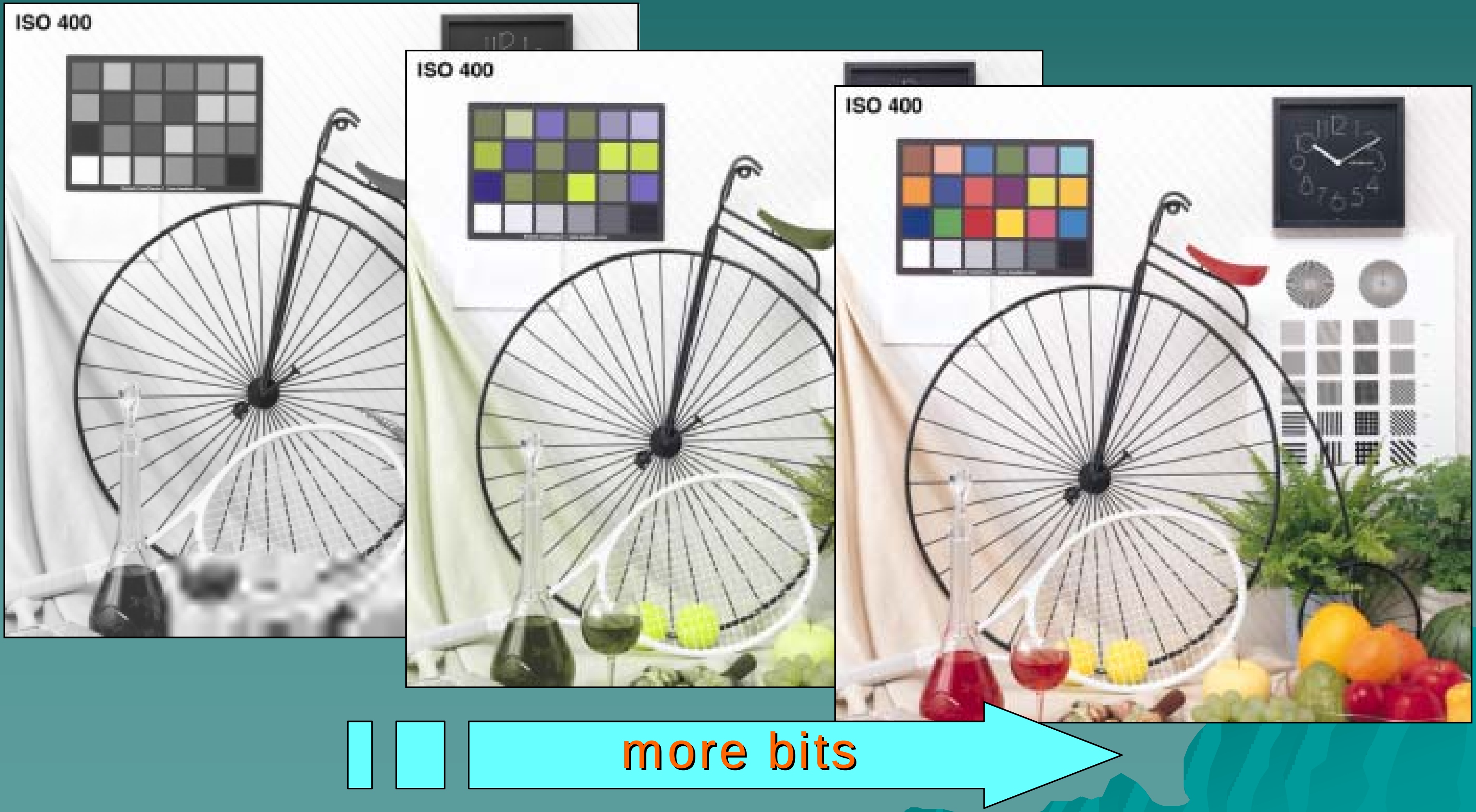
Information Sequencing: Position Dominant (PCRL)







Information Sequencing: Component Dominant (CPRL)

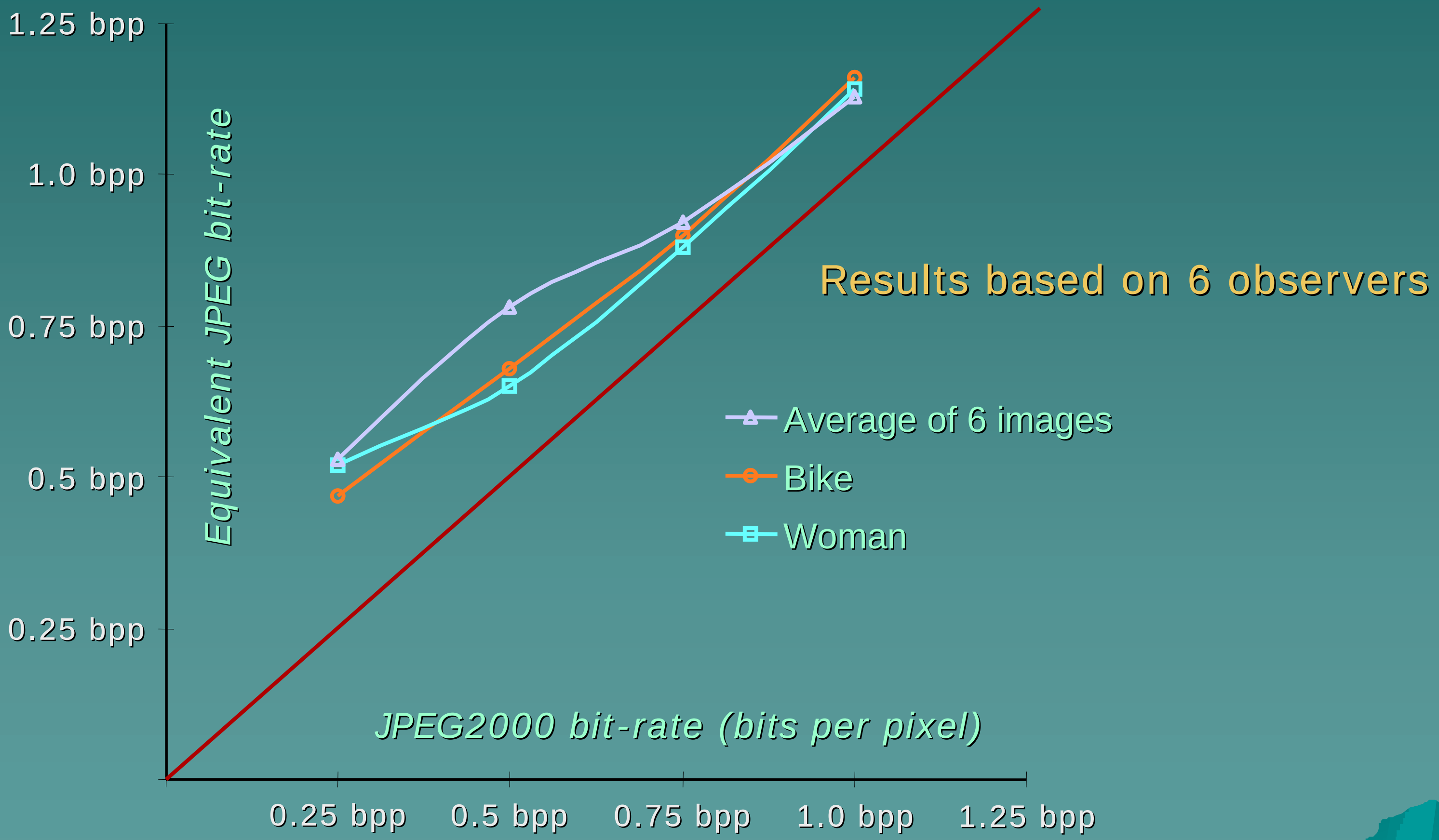


more bits

Information Sequencing Principles

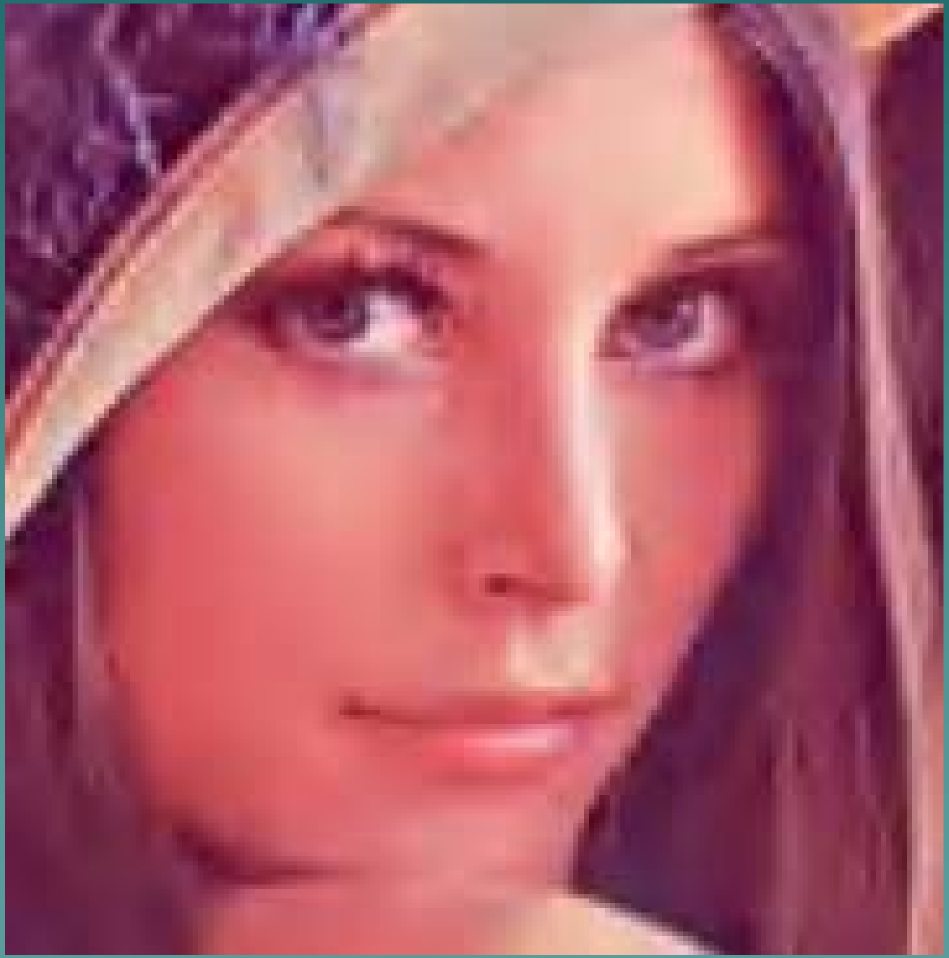
- Fundamental mechanism
  - sequencing of code-block layer contributions
- Signalling mechanisms
  - code-blocks partitioned into precincts
    - 1 spatial region of 1 resolution, in 1 component
  - precincts divided into packets
    - 1 layer of 1 precinct
  - packet progression follows selected scheme
- Can always rearrange packet sequence
  - choose sequence best suited to compressor
  - later rearrange to suit consumer applications

JPEG2000 vs. JPEG: Visual Performance

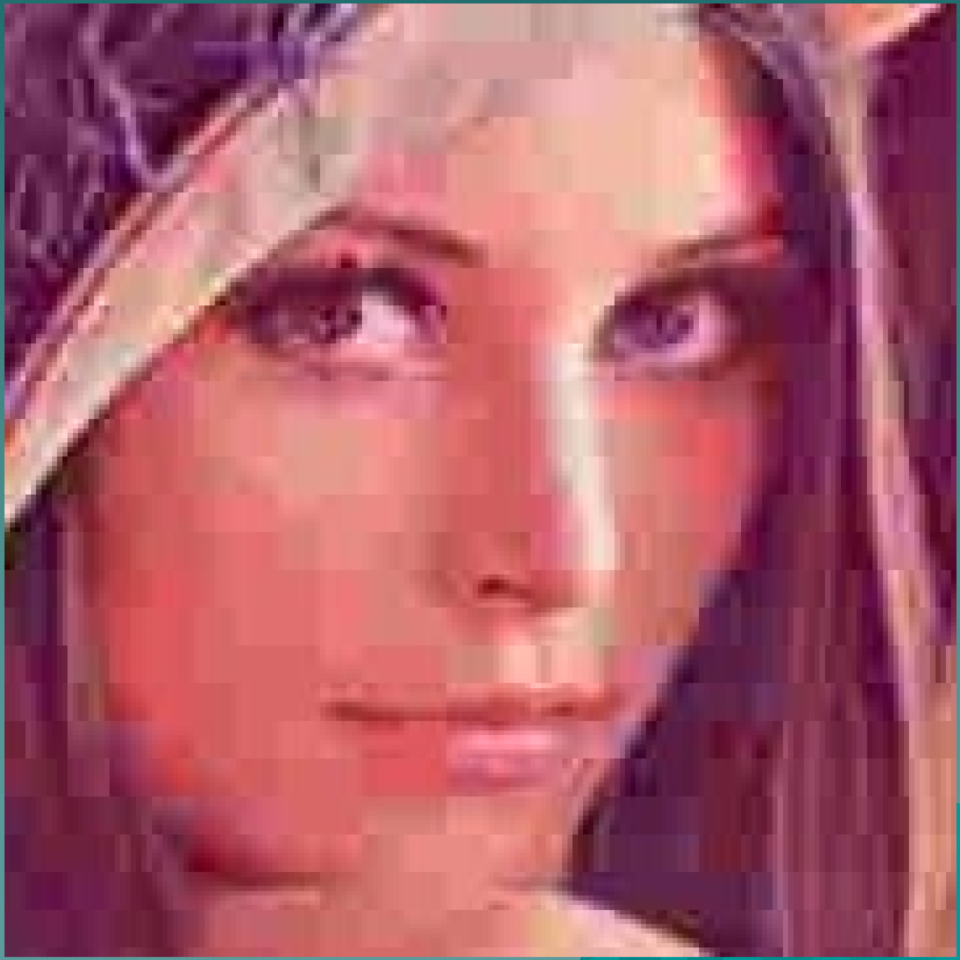


Source: *Chinen & Chien, "Visual evaluation of JPEG2000 color image compression performance," ISO/IEC JTC1/SC29/WG1 Technical Report N1583.*

JPEG2000 vs. JPEG: Blocking Artefacts



JPEG2000 @ 0.25 bits/pixel



JPEG @ 0.25 bits/pixel

JPEG2000 vs. JPEG: Features

- Rich information progressions
  - JPEG: also offers progressive refinement
    - But, done with not inter-operable modes
- Progressive lossy to lossless compression
  - JPEG: no integrated lossy/lossless capability
- Spatial random access
  - JPEG: need to tile image
- Lossless image cropping
  - JPEG: editing ⇒ quantization error build-up
- Exact rate control in one shot

Complexity Issues: Kakadu Speed (2GHz)

- Large Image (24 bpp colour)
  - Compress to 0.5 bits/pel ⇒ 3.0 (4.0) Mpels/s
    - decompress ⇒ 10 Mpels/s
  - Compress to 1.0 bits/pel ⇒ 2.4 (3.2) Mpels/s
    - decompress ⇒ 7.5 Mpels/s
  - Compress to lossless ⇒ > 1 Mpels/s
    - decompress ⇒ 1.3 Mpels/s
- Video (352x240, 4:2:0 colour)
  - Compress to 0.5 bits/pel ⇒ 60 frames/s
    - decompress ⇒ 160 frames/s
  - Compress to 1.0 bits/pel ⇒ 45 frames/s
    - decompress ⇒ 110 frames/s

Complexity Issues: Memory

- Transform and code-block buffering
  - Need 200 to 400 image lines equivalent
  - Does not depend on image height
- Code-block bit-stream buffering
  - Most packet progression orders:
    - About 20% more than final size of entire code-stream
  - Top-down packet progression order:
    - Equivalent to about 1000 lines of compressed data
- Much more memory than JPEG
  - But, no need to buffer entire image

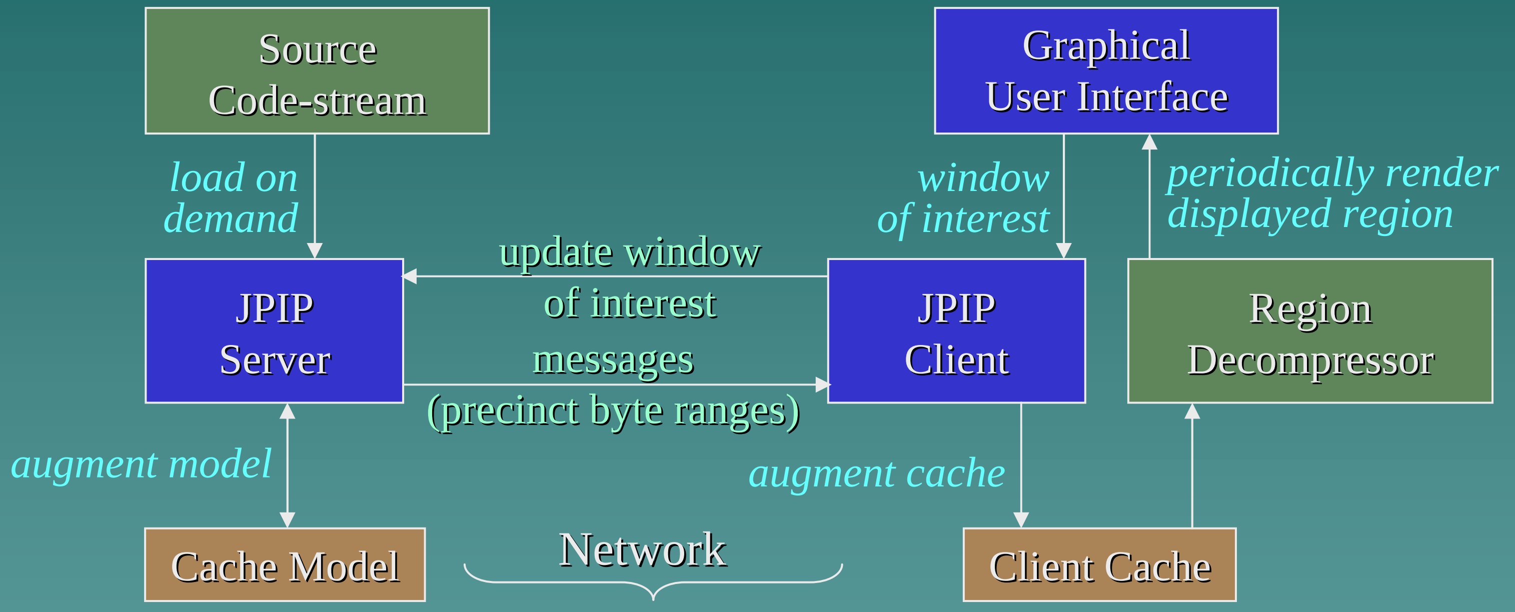
JPEG2000 File Formats

- JPEG2000 algorithm produces code-stream
  - – no interpretation of compressed sample values
- JP2 file format (simple wrapper)
  - – colour rendering info (1/3 colour ICC profiles)
  - – other rendering info (alpha, resolution/size)
- JPX file format (extends JP2)
  - – rich colour descriptions (arbitrary ICC profiles)
  - – multiple code-streams, animation, IPR, XML, ...
- JPM file format (compound documents)
- MJ2 file format (Motion JPEG2000)
  - – one code-stream per frame

Parts to the Standard

- Part 1: core technology (+ JP2 file format)
- Part 2: extended coding tools (+ JPX)
- Part 3 (12 & 13): motion JPEG2000 (MJ2)
- Part 4: conformance testing for Part 1
- Part 5: reference software (JJ2000+Jasper)
- Part 6: compound documents (JPM)
- Part 8: JPSEC (encryption, watermarking)
- Part 9: JPIP (interactive protocols)
- Part 10: JP3D (hyperspectral/volumetric)
- Part 11: JPWL (robust comms for wireless)

Interactive Distribution with JPIP (JPEG2000 Internet Protocol)



- Demonstration
  - jpip://dst-m/wharf3.jp2
  - jpip://dst-m/phoenix.jp2

Applications for JPEG2000

- Internet
- Medical
  - adopted by DICOM: medical imaging standard
- Military/Surveillance/GIS
  - 7 of 8 military image standards to require J2K
- Digital Cameras
  - lossless image cropping; support for internet
- Scanners/printers
  - chips appearing for these markets already
- Mobile Devices
  - efficient interactive browsing of large images
- Digital Cinema and other Video
  - produce/archive/distribute; Yahoo Messanger; etc.

Available Implementations: Software

- JJ2000: Java (Canon/Ericsson/EPFL)
  - free reference software
- Jasper: “C” (Adams/Imagepower/UBC)
  - free reference software
- Kakadu: “C++” (Taubman/UNSW)
  - probably most popular commercial system
- Luratech/AlgoVision
- Aware Inc.
- Ricoh (TRUEW)
- Imagepower

Available Implementations: Hardware

- Analog Devices
  - ADV-JP2000 was first to implement (partially)
- Amphion
  - IP Core claims 60 Msamples/second
  - May do only the wavelet transform part
- Picture Elements Inc.
  - Nearing release of high-performance compressor for scanners
- Ricoh, Imagepower, ...
  - Working on chips; status unknown

Useful References

- Web-site
  - www.jpeg.org is an excellent starting point
  - www.crc.ricoh.com/~gormish/jpeg2000.html
- Book
  - "JPEG2000: image compression fundamentals, standards and practice," Kluwer, by Taubman and Marcellin
- General Articles
  - IEEE Signal Processing Magazine, Sept 2001
  - Taubman & Marcellin, Proc. IEEE, August 2002