DjVu: A compression Technique and Software Platform for Publishing Scanned Documents, Digital Documents, and High-resolution Images on the Web

Yann LeCun, Leon Bottou, Patrick Haffner
AT&T Labs-Research, Middletown, NJ
http://yann.lecun.com

Motivation

Much of the world's knowledge is still paper.

Converting to a web format is expensive and imperfect.
What about image compression techniques ?

TIFF/MMR/G4 falls short for bitonal

50–100K per page at 300 dpi

Continuous–tone techniques are inadequate (e.g. jpeg).

Large files (150K–1M), fuzzy text,
large memory requirements, no document structure.s

PDF for scanned documents

is merely an encapsulation for TIFF/G4 or JPEG

New/emerging standards

are getting there, but are not tuned for web vieweing,
and need to work on efficiency (JBIG2, JPEG–2K, T.44)

What is DjVu

A compression technique, a file format, and a software platform for distribution documents on the web (scanned or digitally produced, in color, gray, or b&w.)

- Scanned bitonal 300 dpi : 5–40K per page
  - (3–10 times better than TIFF/G4)
- Scanned color 300 dpi : 30–100K per page
  - (5–10 times better than JPEG or PDF)
- Photos : 2dB better than JPEG
  - (quality similar to JPEG–2000, but faster)
- Digital documents 300 dpi : half the size of ps.gz or pdf.
- High quality text, good enough for printing.
- Progressive decoder/renderer (text appears first.)
- Seamless panning and zooming.
- Good integration with web browsers.
- Available today (already through its 3rd revision)

Summary

1– Examples

2– Entropy coding with the Z–Coder

3– Bitonal images (DjVuBitonal, JB2)

4– Continuous tone images (DjVuPhoto, IW44)

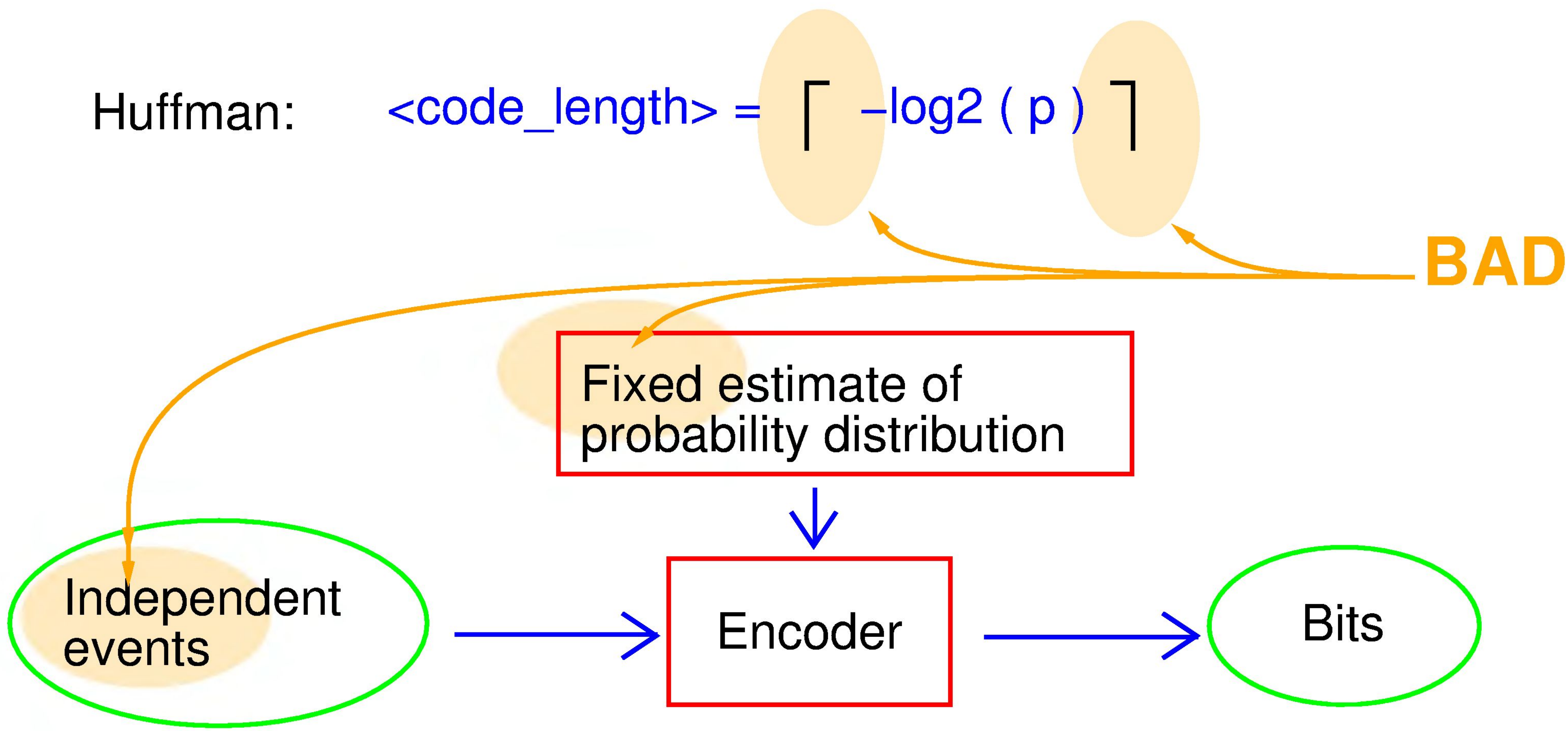
5– Color documents (DjVuDocument)

6– Foreground / background segmentation
- digital documents
- scanned documents

7– Efficient browsing

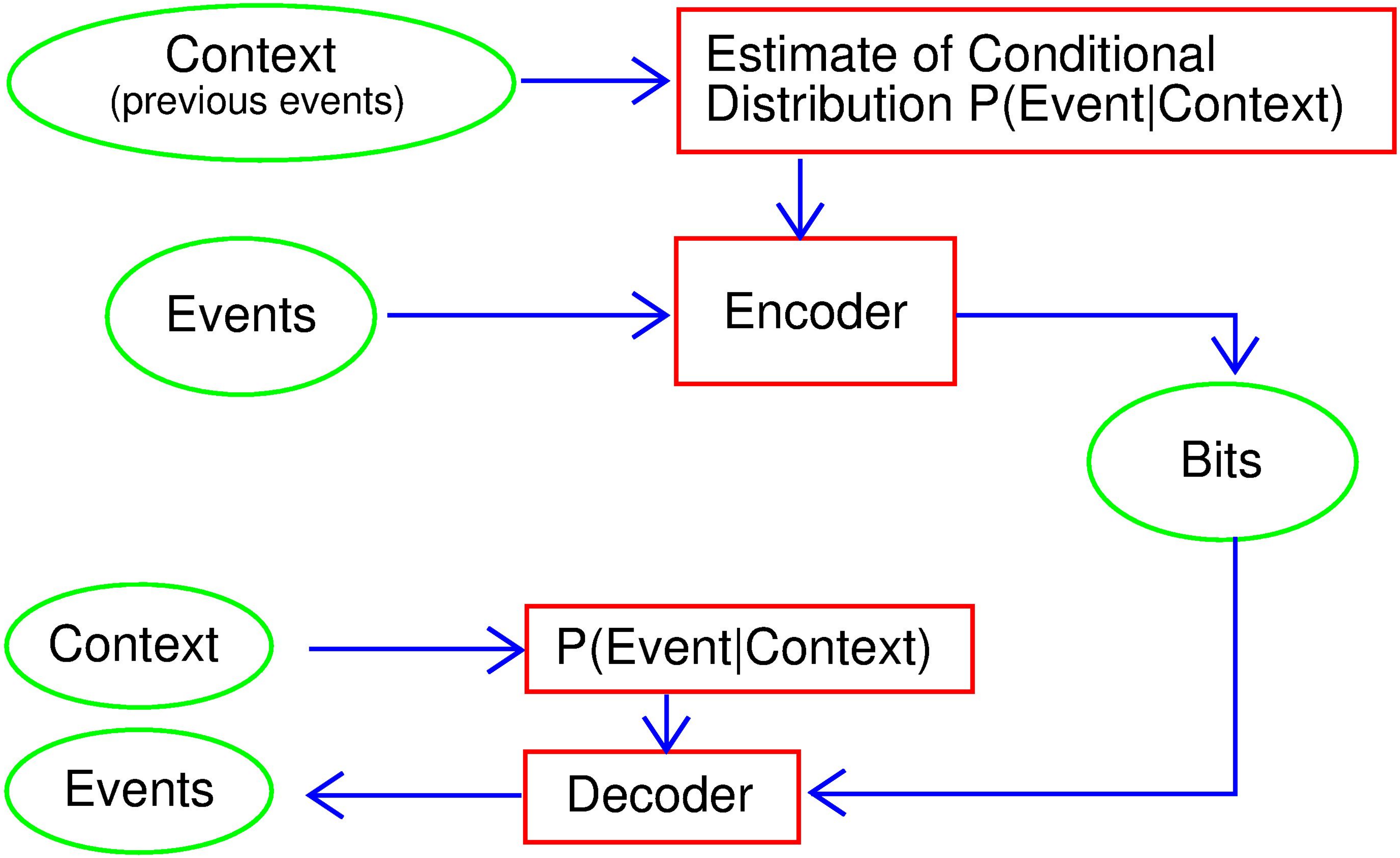
DjVu : Z–Coder : Entropy Coding

– short codes for high probability events
– long codes for low probability events



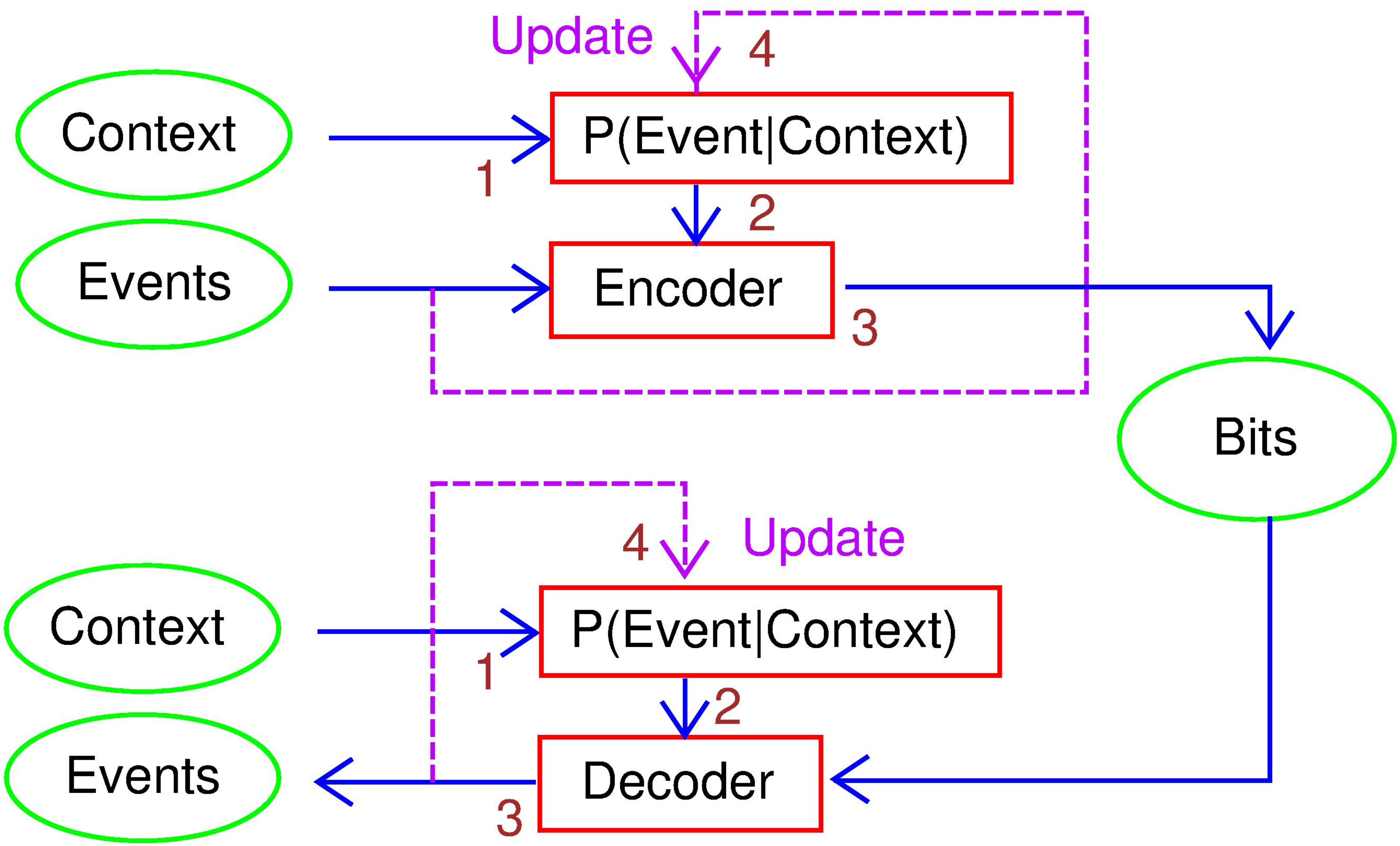
DjVu : Z–Coder : Conditional Entropy Coding

– use context to lower entropy



DjVu : Z–Coder : Adaptive Entropy Coding

– learn the conditional distributions
– encoder and decoder remain in sync.

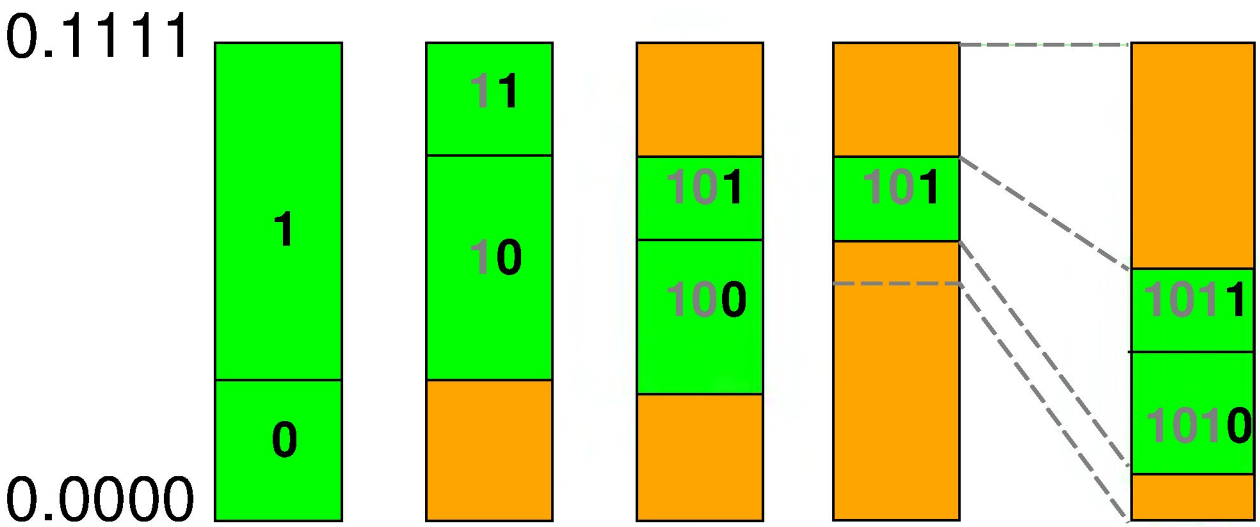


DjVu : Z–Coder : Arithmetic [Rissanen]

– Arithmetic coders

The string of code bits is viewed a number in [0,1)

Events are coded by maintaining an
interval of possible code strings



– Binary arithmetic coders (efficient – adaptive)

Q–Coder [ibm] MQ–Coder [+mel] QM–Coder [+lucent]

DjVu : Z–Coder : Golomb

– Golomb Coder

Assume 1 is more probable than 0.
Chances are that one codes words like 11111110

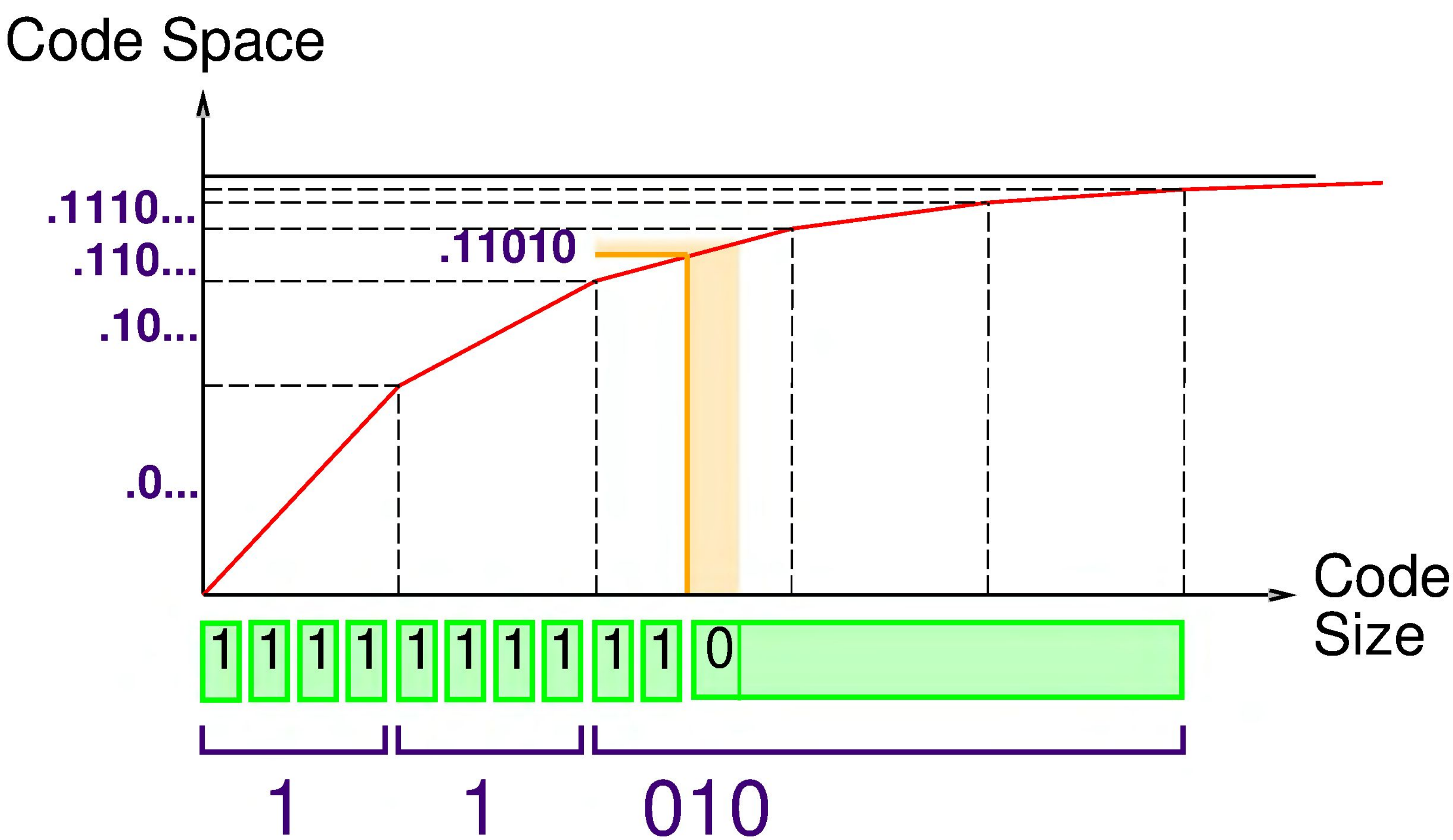
Example: Golomb coding with M=4

Events	1 1 1 1	1 1 1 1	1 1 0
Bits	1	1	0 10

–1 group of 4 ones
–1 group of 4 ones
–1 zero preceded by two ones.

	P(1)/P(0) large	P(1)/P(0) near 1
Events	1111111111111111110	1110
M=16 M=2	100010 **< better** 1111111100	00011 101 **< better**

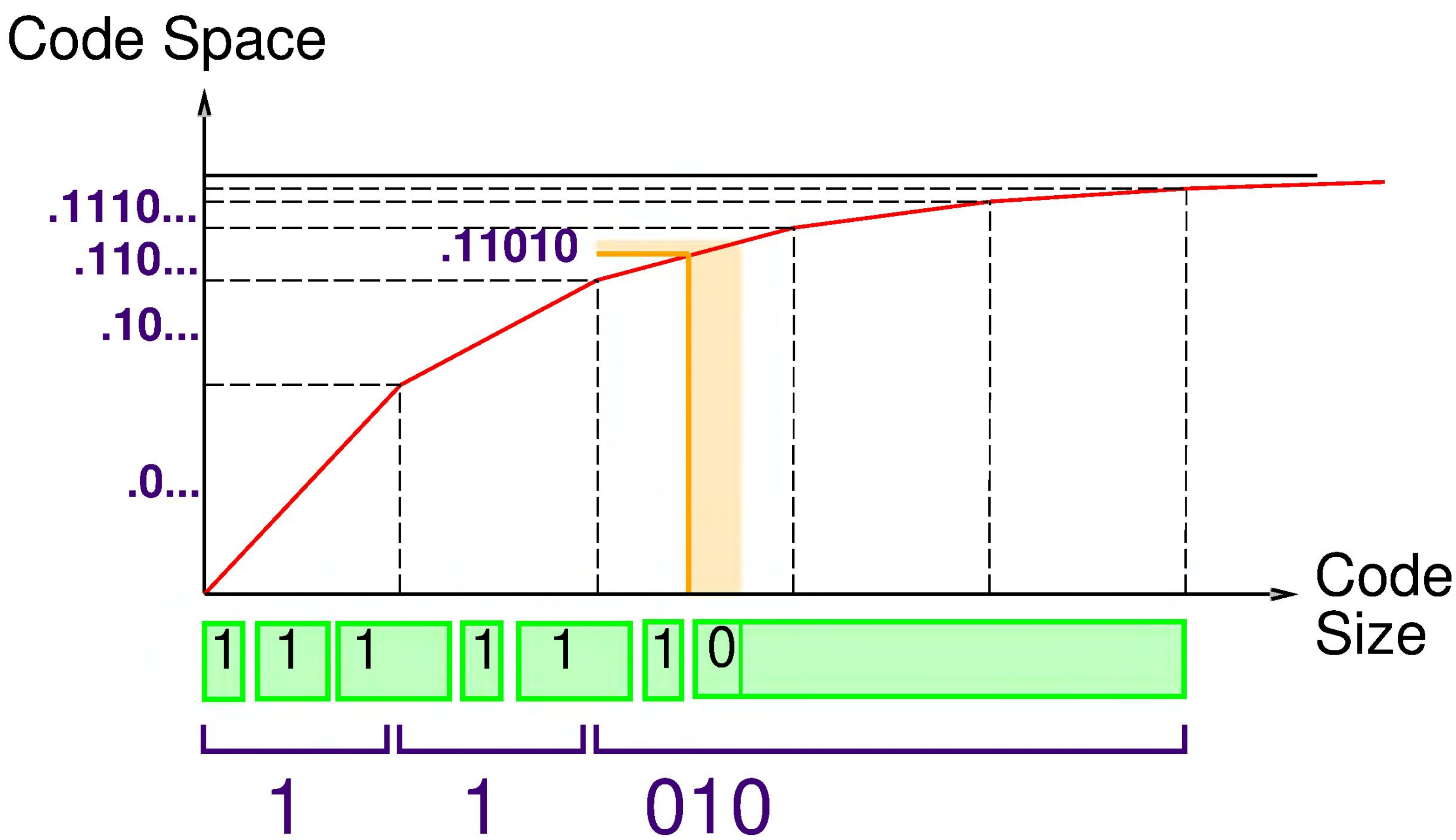
– Arithmetic Interpretation of Golomb Coders



DjVu : Z–Coder : Principle

– Z–Coder

Each 1 takes an arbitrary code bit fraction
that depends of its estimated probability.



Z–Coder Mathematics

Analytic relation determines best
bit fraction for a given probability.

Provably within 0.25% of Shannon bound.

Z–Coder Speed

Coding one (typical) event :

1 ADD + 1 COMPARE

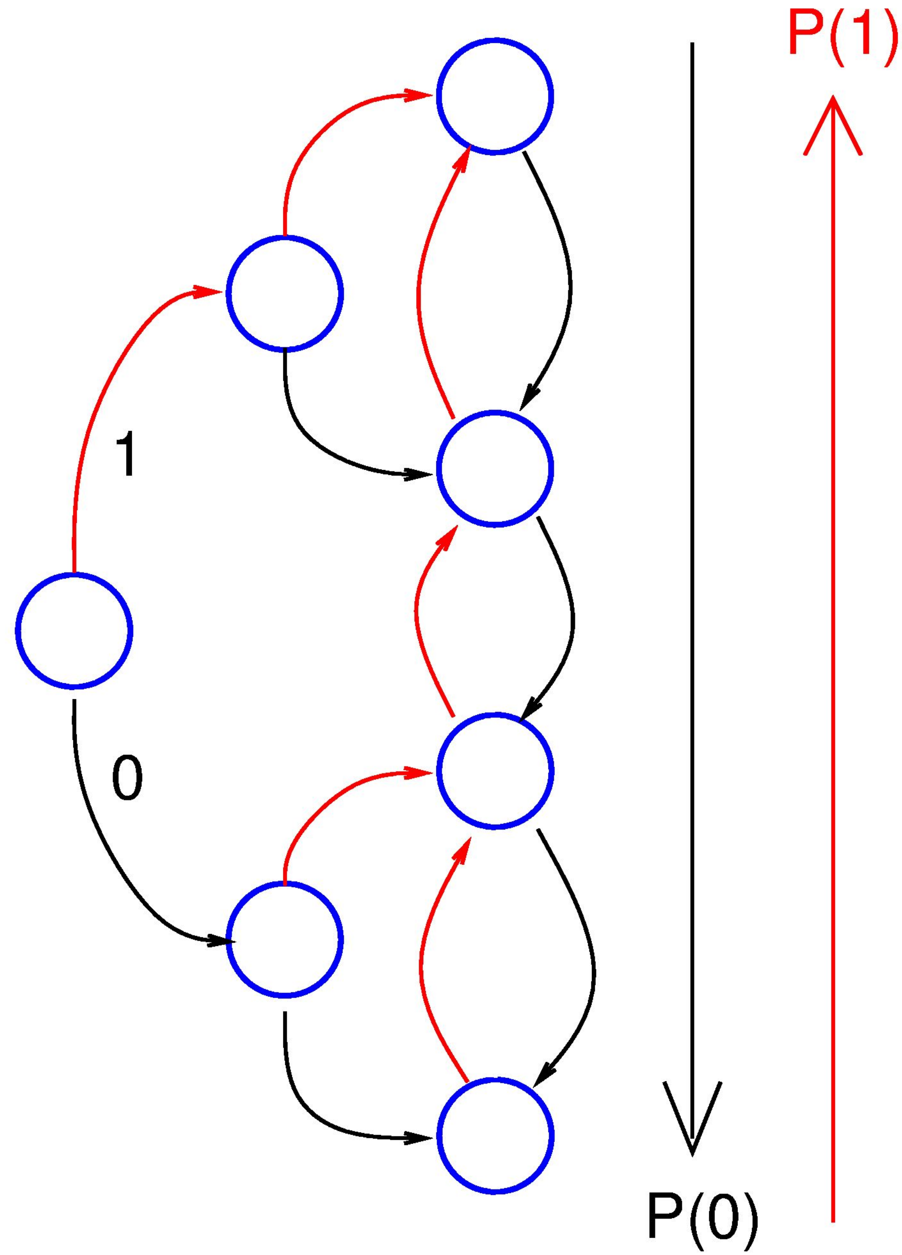
Z–Coder Adaptation

Finite state machine.

Each context is represented by a small integer (state).

Coding 1 or 0 causes transitions in a state machine

Adaptation noise increases bitrate by about 3%



DjVu : Z–Coder : Conclusion

- Z–Coder is a binary adaptive entropic coder.
  It supports multiple contexts.
  It learns probabilities on the fly.
- Z–Coder typically performs 1 ADD + 1 COMPARE per decoded event
- Compressed bitrates are 3–5% above the theoretical lower bound (Shannon)

DjVu : DjVuBitonal (a.k.a. JB2)

Documents contain multiple repetitions of nearly identical shapes (e.g. characters)

JB2 data = Sequence of encoded records that
- describe shapes
- display selected shapes at specified positions

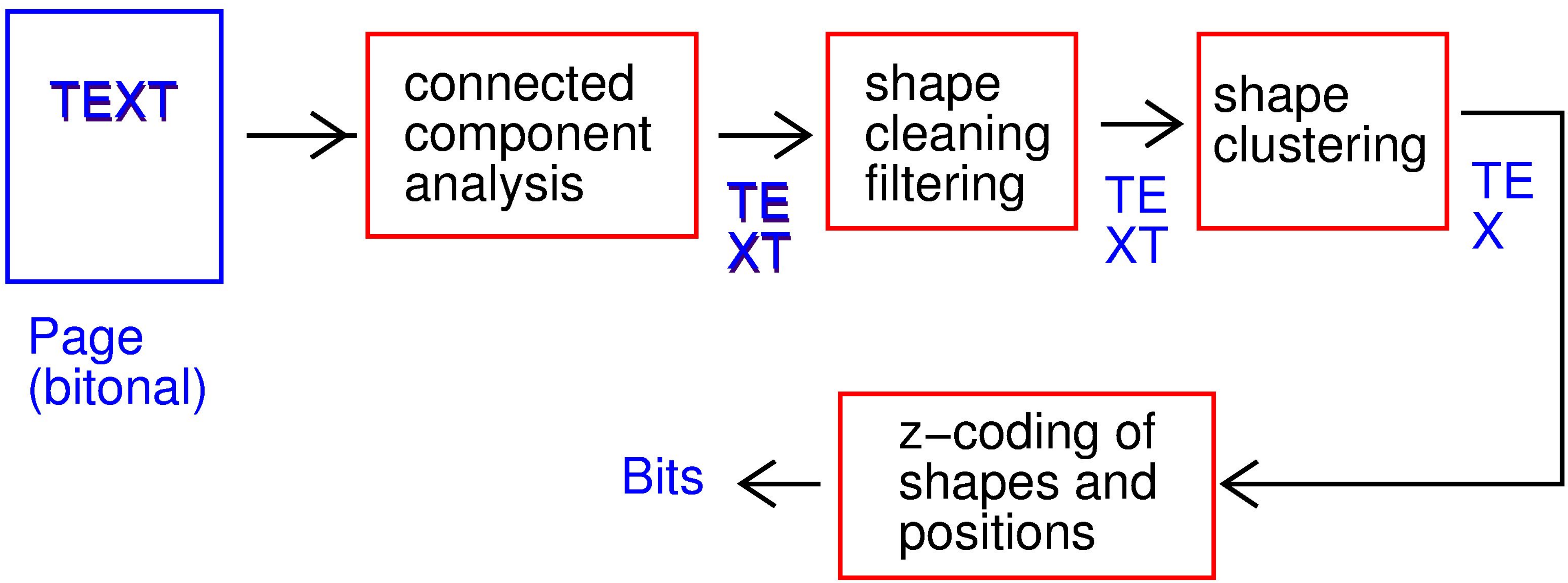
Cross–Coding: Coding shapes by comparing them with a previously coded shape. JB2

Soft Pattern Matching: Small differences between shapes can be safely ignored. JB2

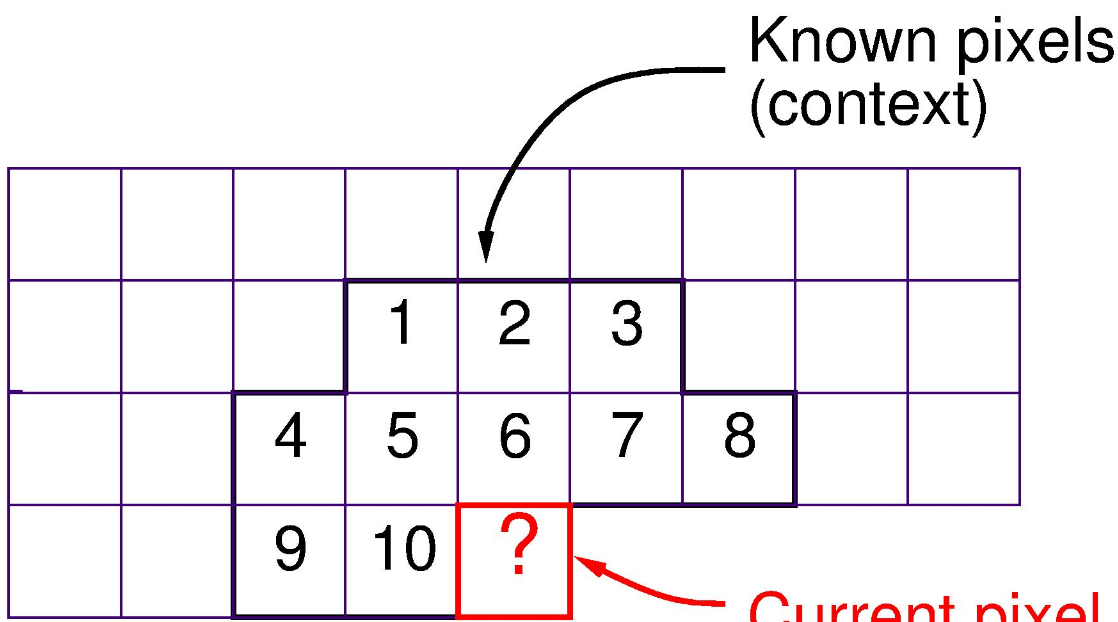
ZP–Coder: Fast adaptive arithmetic coder that squeezes JB2 data (almost) to the theoretical limit. DjVu1

Shared Shapes: Shape description can be shared by several pages of a document. DjVu3

DjVu : DjVuBitonal : Schema (a.k.a. JB2)



DjVu : DjVuBitonal : Direct

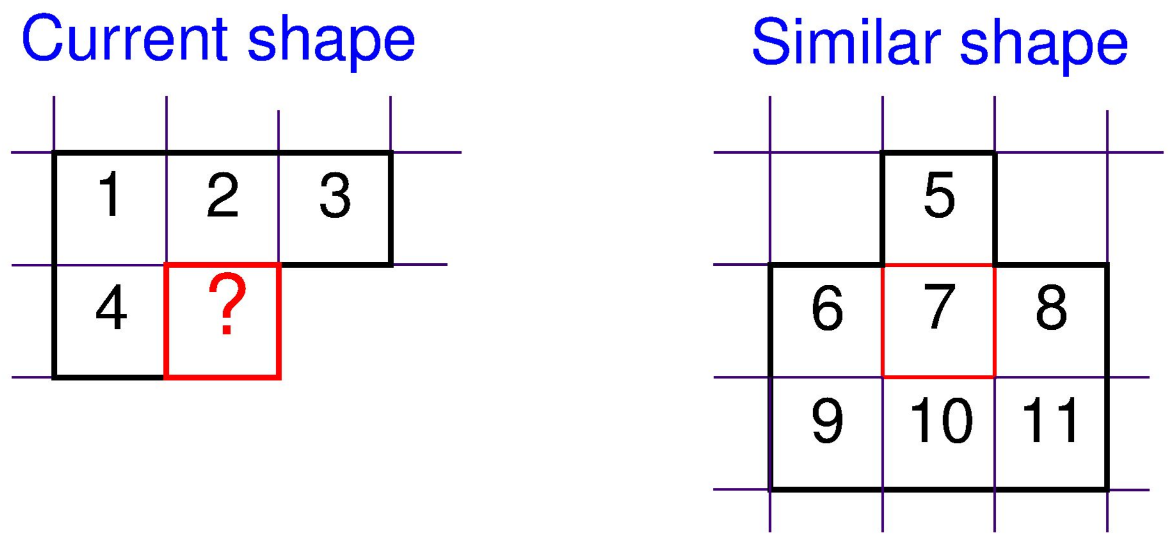


1024 different contexts

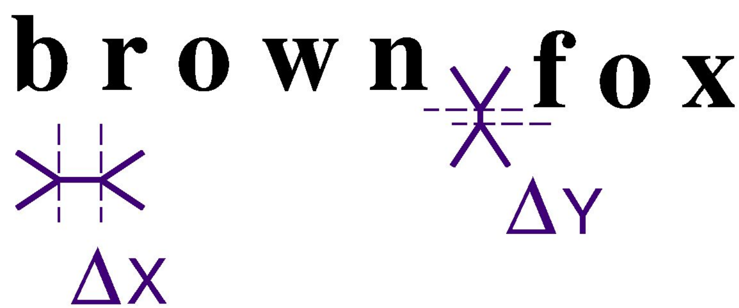
Z–Coder learns P(current pixel | known pixels)

DjVu : DjVuBitonal : More Coding

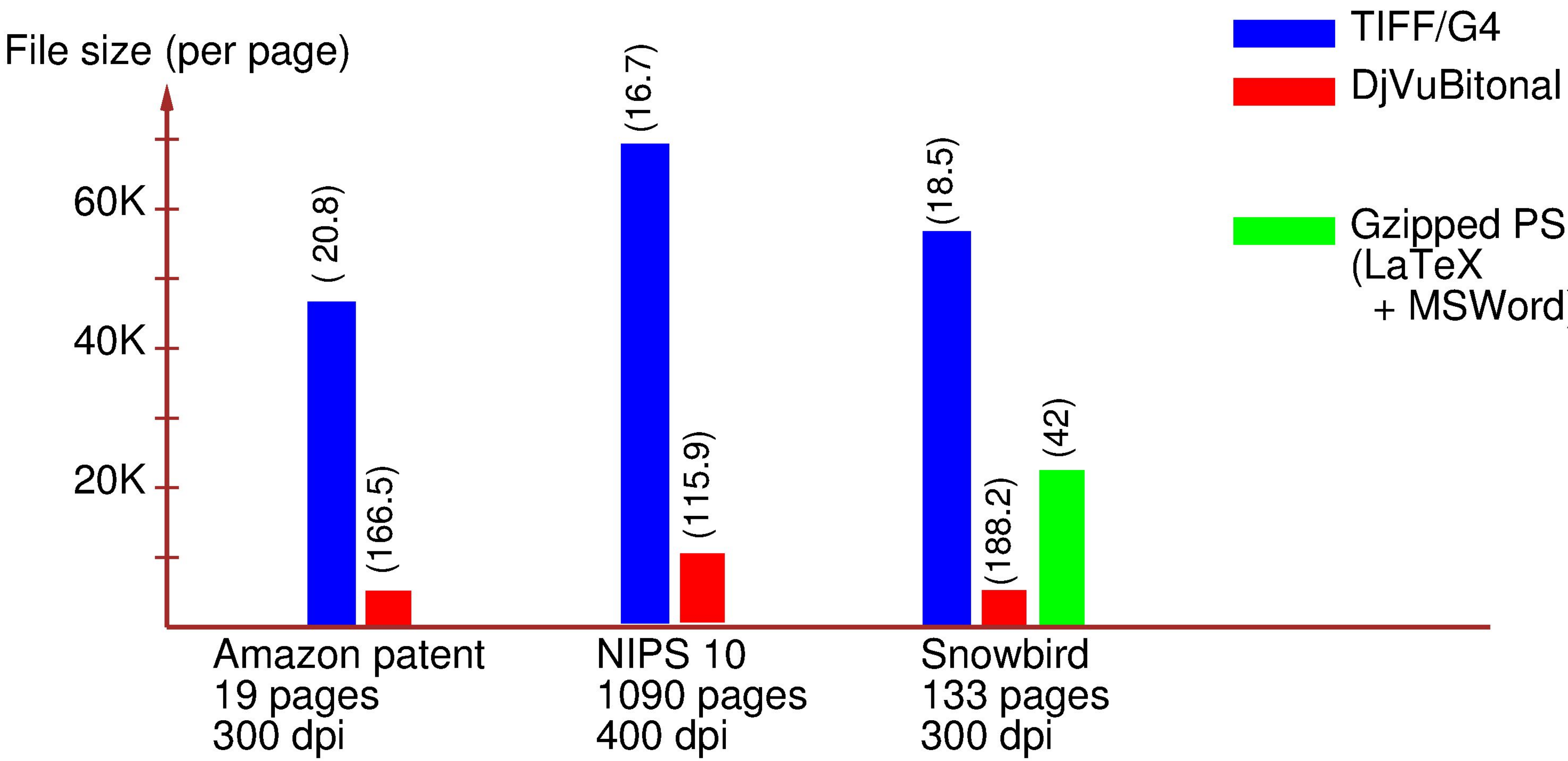
Cross Coding



Position Coding



DjVu : DjVuBitonal : Results vs TIFF/G4

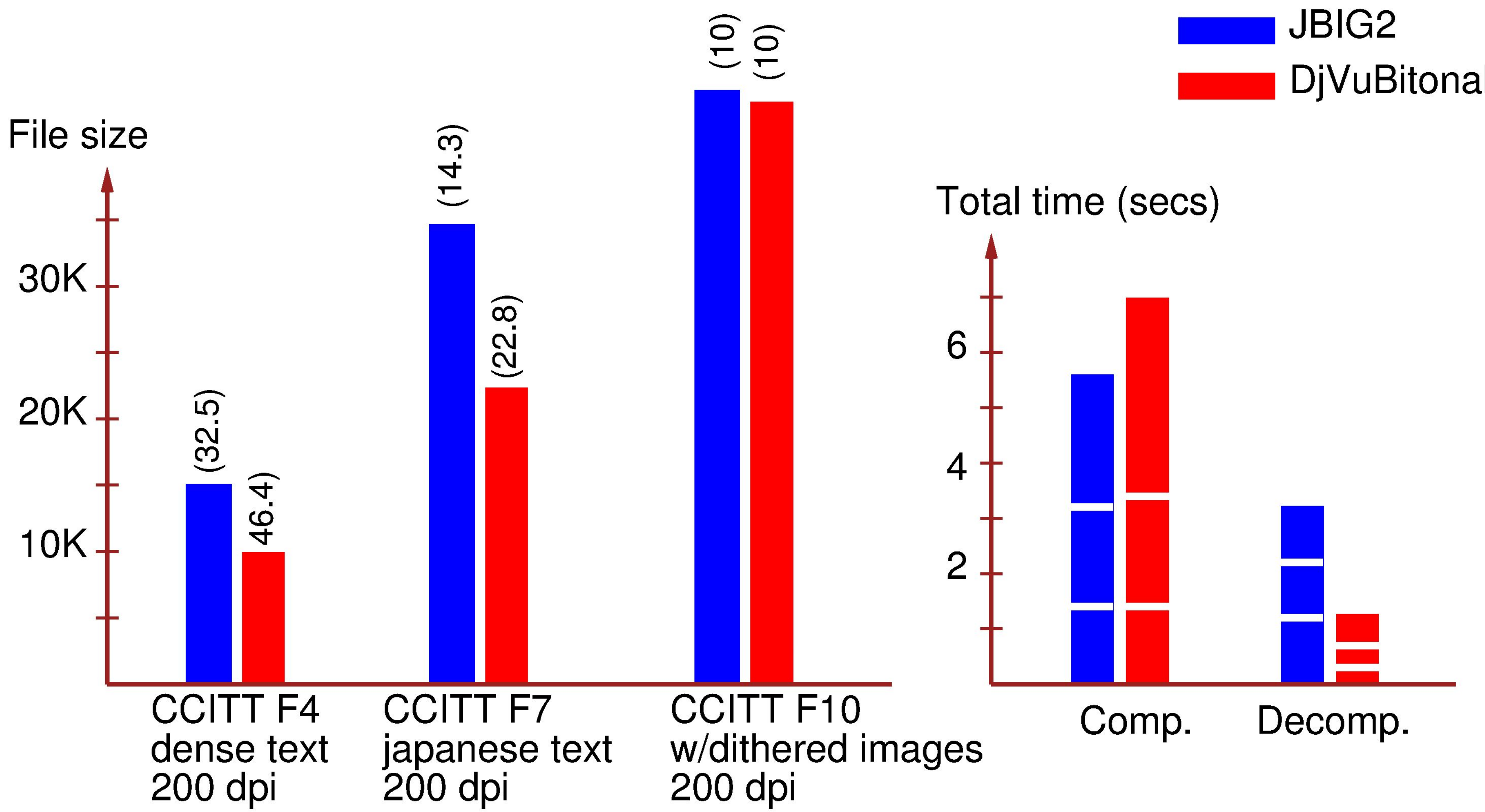


Notes :

PDF encodes scanned bitonal documents using TIFF/G4 compression.

More...

DjVu : DjVuBitonal : Results vs JBIG2



Notes :

All files contain a single page (no multipage encoding)

– JBIG2 (CSM,MQ), PIII–500, (ImagePower encoder)

– DjVuBitonal, PII–400

DjVu : DjVuBitonal : Conclusion

- Lossy JB2 is typically 3–10 times smaller than MMR/G4 on text and 2–4 times smaller on line art.
- Multipage compression with shared shape dictionay between pages boosts compression by 1.4 – 2.0.
- Lossy JB2 is significantly smaller than arithmetically coded lossy JBIG2 on mostly textual images, and about the same on line art and halftones
- JB2 compression / decompression is faster than JBIG2.
- Lossless JB2 compression is 2 times smaller than MMR/G4 and 1.4 times smaller than JBIG1 [Inglis, 1999]

DjVu : DjVuPhoto (a.k.a. IW44)

Wavelet Image Compression with specific features :

Fast lifting transform (without multiplications)
Enables fast panning in viewer.

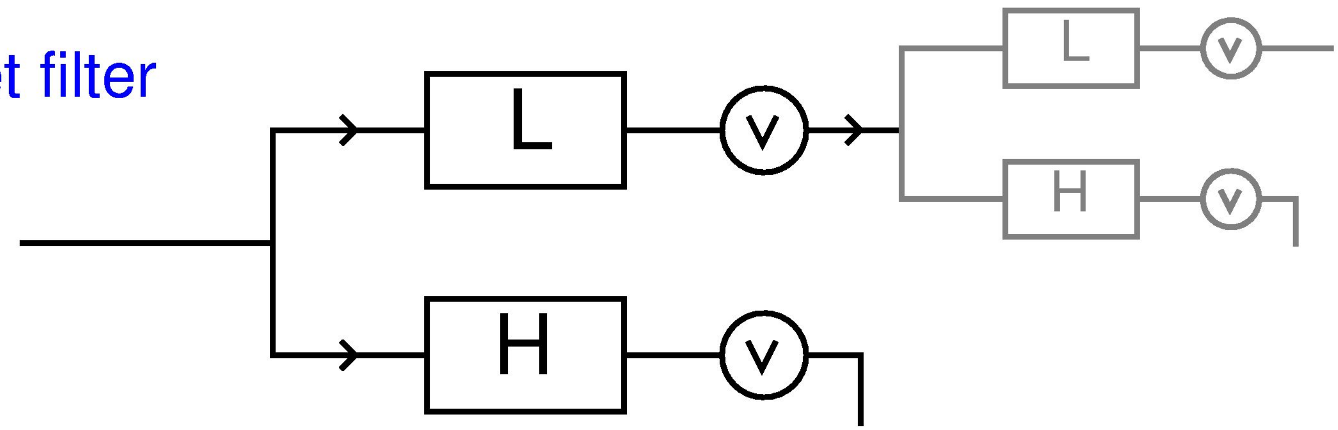
Progressive images
Browser displays successive refinements.

Decoder has very small memory footprint
Even for large progressive images.

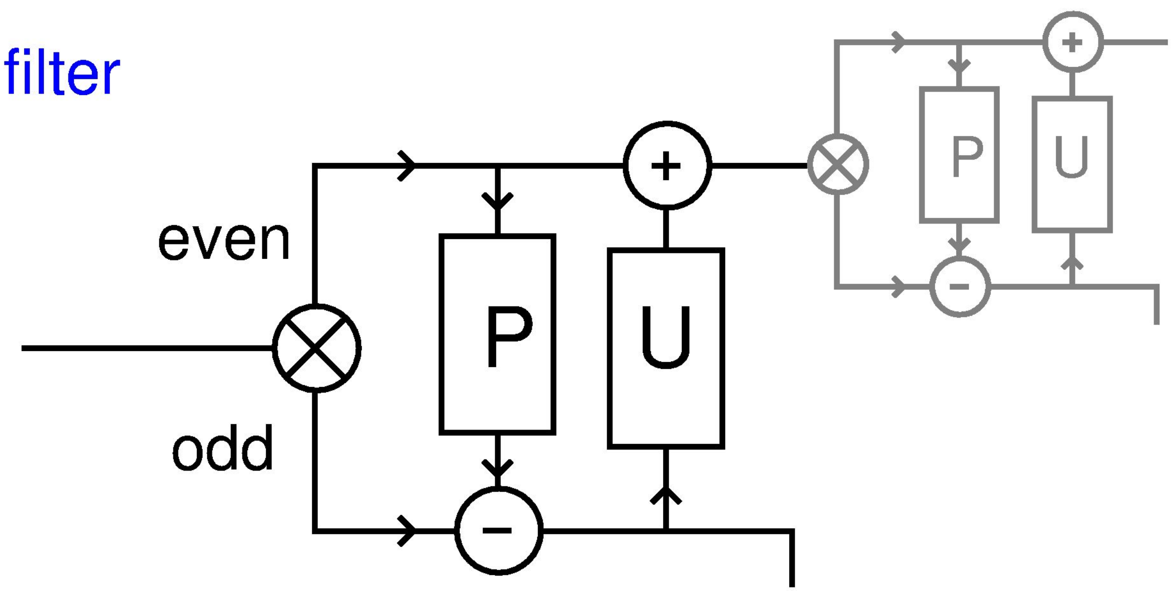
One can specify “don’t care" pixels while encoding
– non rectangular images
– partially masked images

DjVu : DjVuPhoto : Lifting (a.k.a. IW44)

Standard wavelet filter



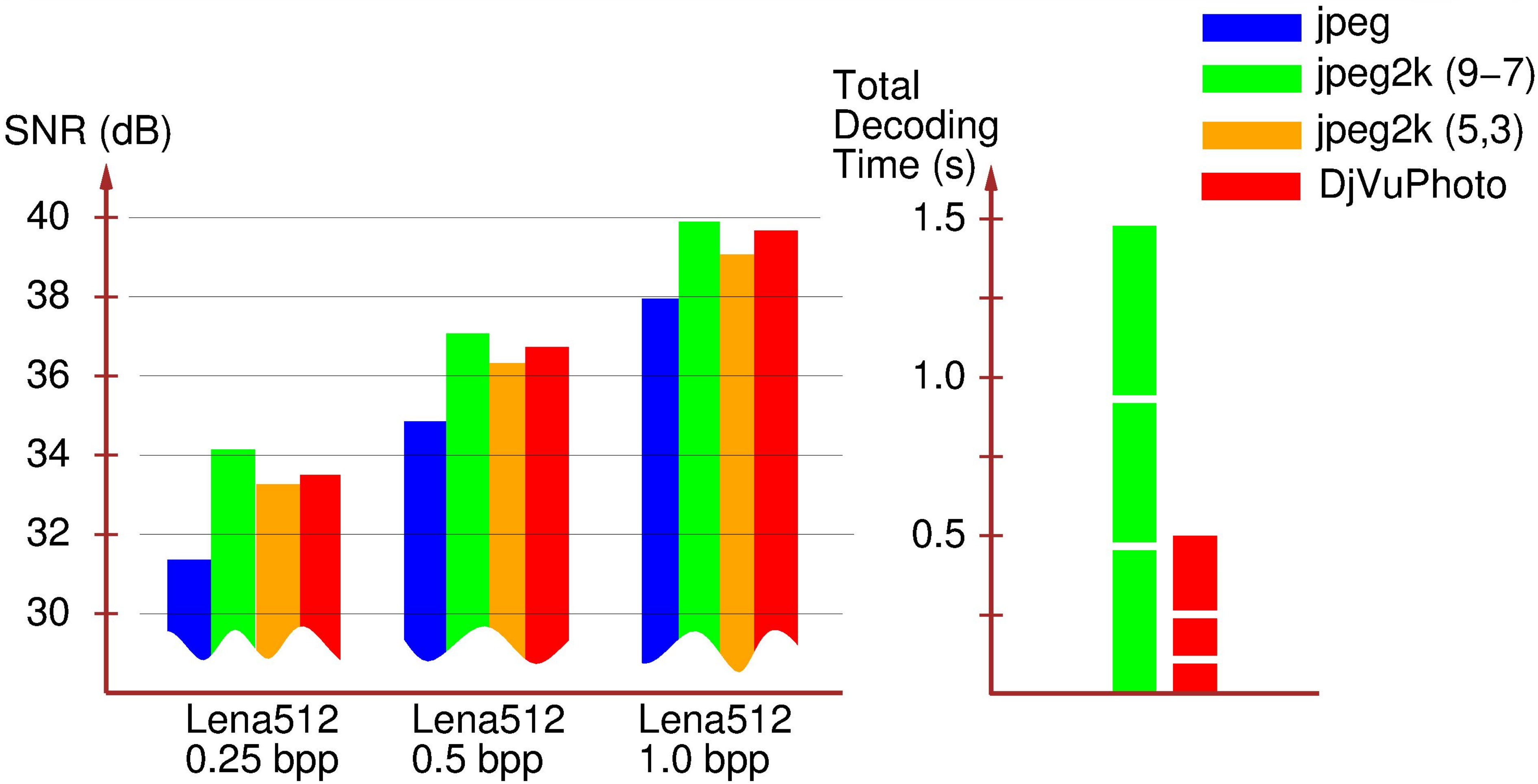
Lifting wavelet filter



Easily reversible –> integer transforms –> faster implementations
Integer transforms usually confined to lossless compression.

IW44 uses fixed point lifting for lossy image compression.

DjVu : DjVuPhoto : Results

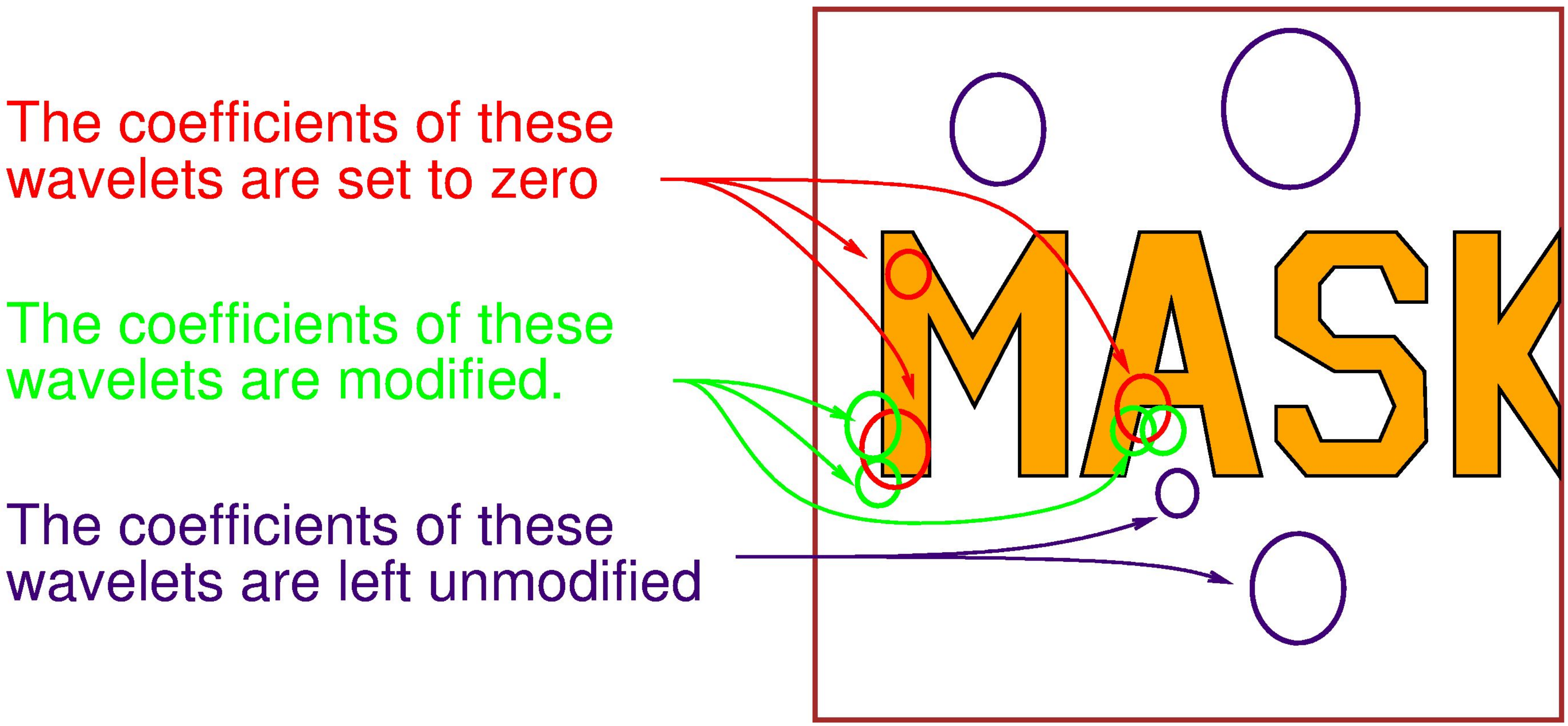


DjVuPhoto
- 2dB better than JPEG
- Between JPEG2K (5,3) and (9–7) transforms
- Decodes 3 times faster than JPEG–2000

JPEG–2000 results using JasPer Software (9–7 transform)
Decoding times measured on a R10K–195

DjVu : DjVuPhoto : Masking

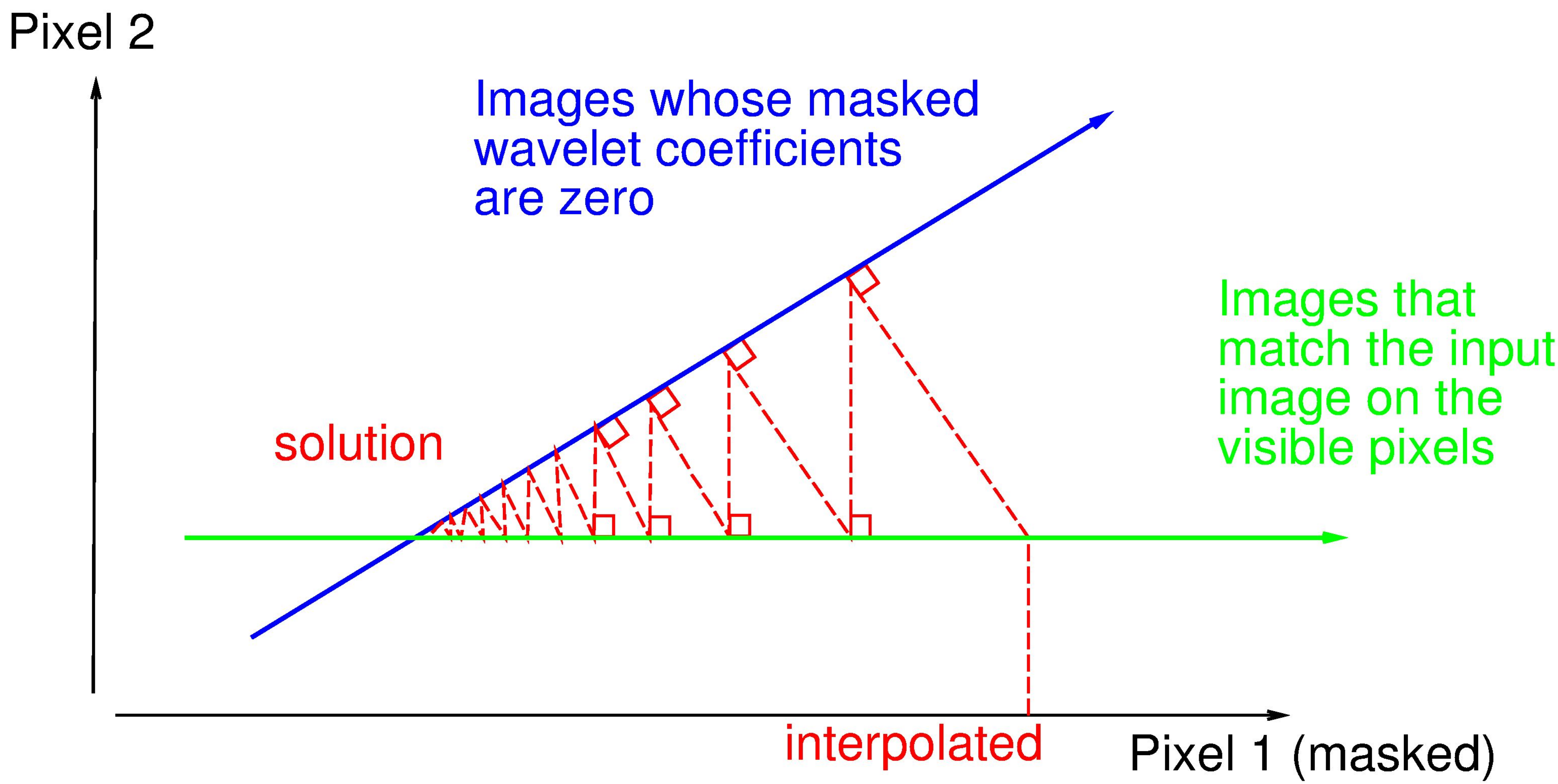
We do not want to spend bits on pixels located below foreground objects.



DjVu : DjVuPhoto : Masking : Projections

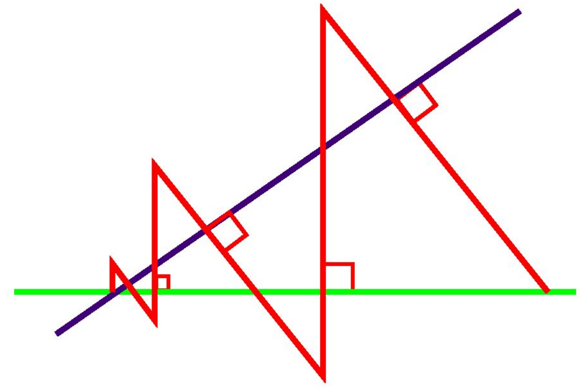
[Bottou–Pigeon–98]

- An image is a vector
- A wavelet transform is a linear operator

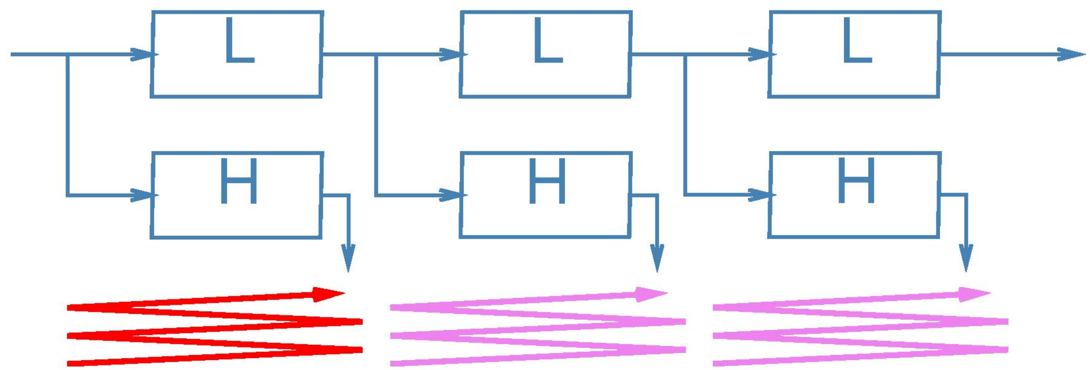


DjVu : DjVuPhoto : Masking : Speedup

(1) over–relaxation



(2) each scale is treated separately



Overall cost = 3 x ordinary wavelet

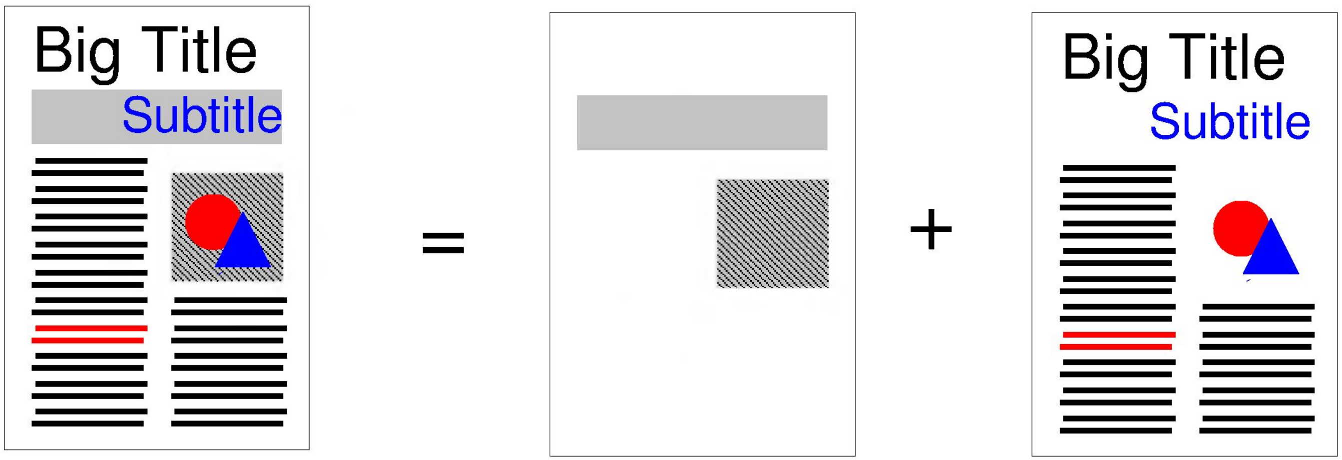
DjVu : DjVuPhoto : Masking : Results

Image			Compressed File Size		
image	size	%masked	reg.	interp.	masked
hobby1	825x1074	19%	131K	51K	40K
hobby2	825x1074	19%	140K	65K	52K
metric	744x1074	26%	170K	35K	26K
missel	610x429	40%	64K	30K	19K
plugin	757x1035	32%	189K	41K	31K

DjVu : DjVuPhoto : Conclusion

- State–of–the–art wavelet compression
  (i.e. far superior to jpeg)
- Fast lifting wavelet transform
  (even for lossy compression)
- Unique masking technique
  avoids wasting bits on masked pixels.

DjVu : DjVuDocument



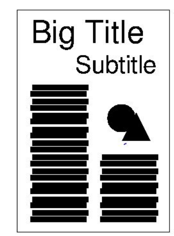
Page Image

Background

Subsampled
DjVuPhoto

Foreground

Tranparency Mask
DjVuBitonal

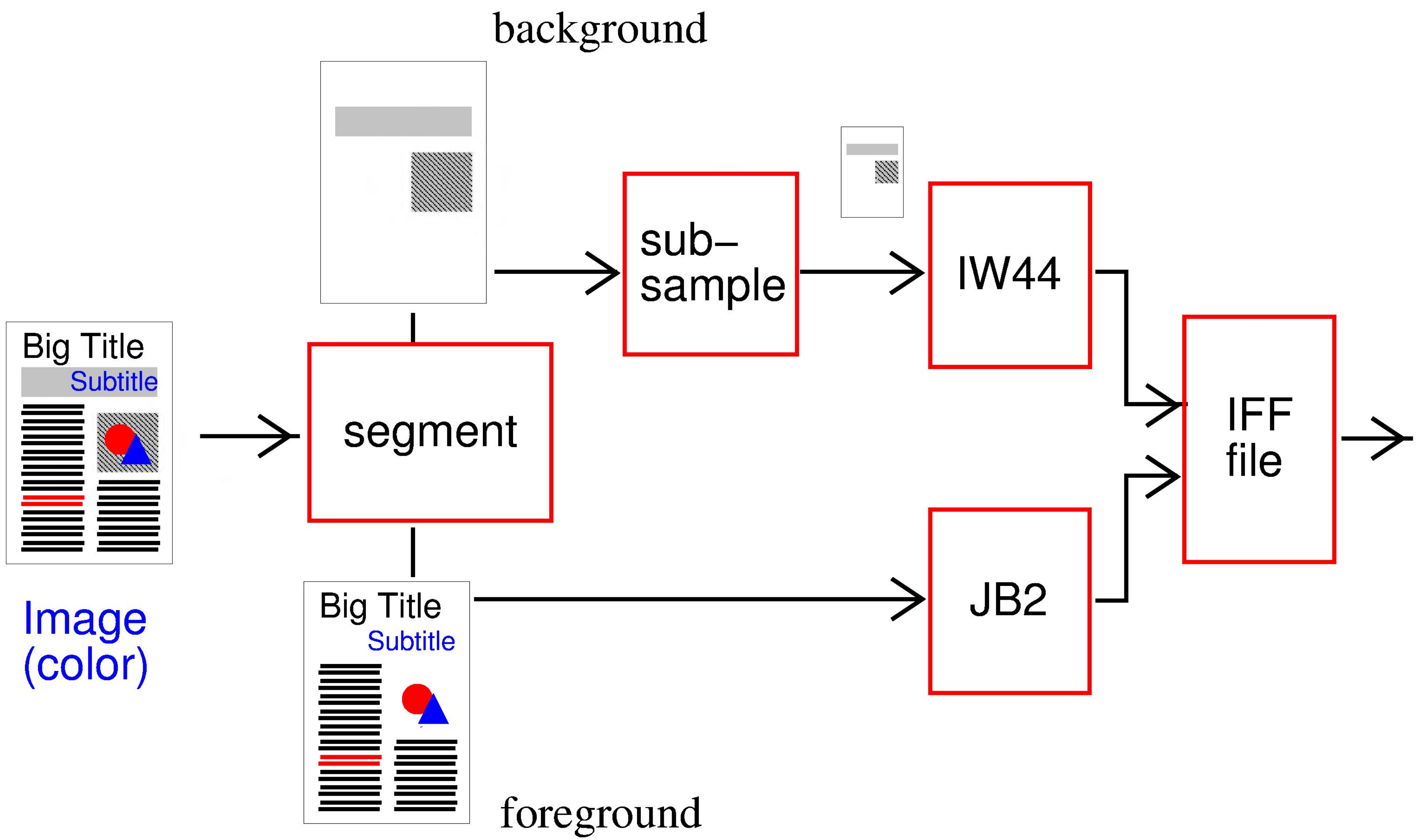


Examples

Colors encoded either:
– as one solid color per shape.
– as a lores color image

Size of average DjVu 300 dpi color page
= Size of average web page (html + images)

DjVu : DjVuDocument : Schema



Segmentation

- **For scanned documents**

  We start with a high resolution RGB image.

  Example: PPM image
  (22MB for a 8’x11’ page, 300 dpi, 24 bpp)

  Critical for image size & quality.
  Basically a “recognition” problem.

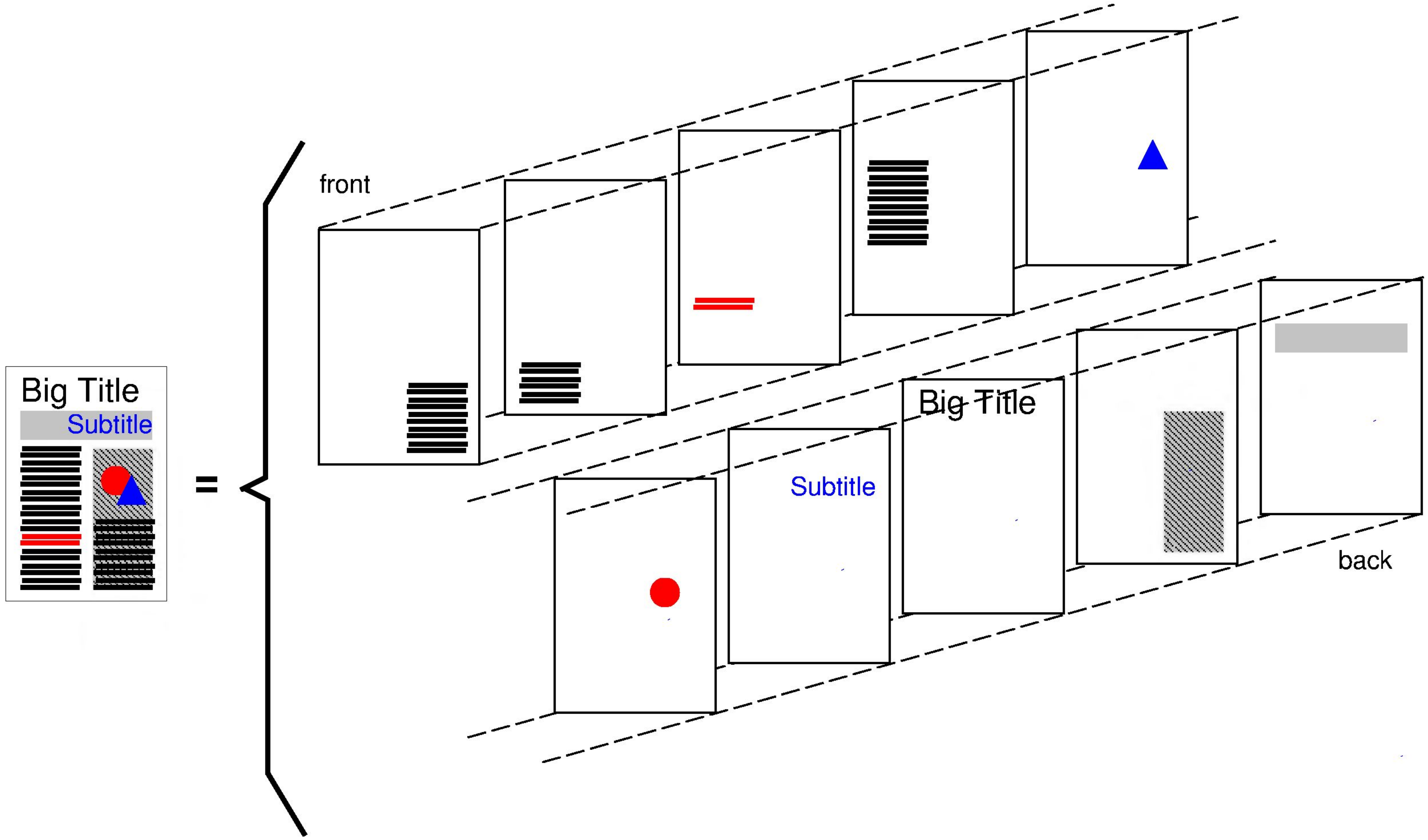
- **For digital documents**

  We start with an electronical document.

  Application level: XML/HTML, MSOffice, ...
  Printer level: PDF PostScript

  Apparently easier.
  Expectations are much higher.

Segmentation : Digital : Decomposition



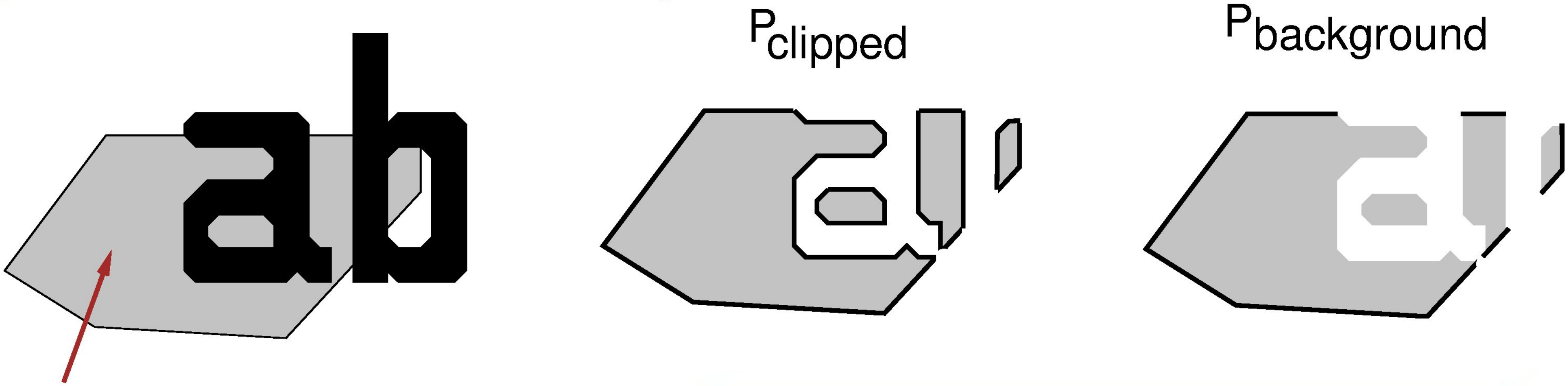
Printing language provides a document decomposition.

Superposed objects can be monochromatic (e.g. text)
polychromatic (e.g. pictures)

Good start ... but simple solutions are not sufficient.

Segmentation : Digital : Rate/Distorsion Criterion

[Bottou–Haffner–LeCun 2001]

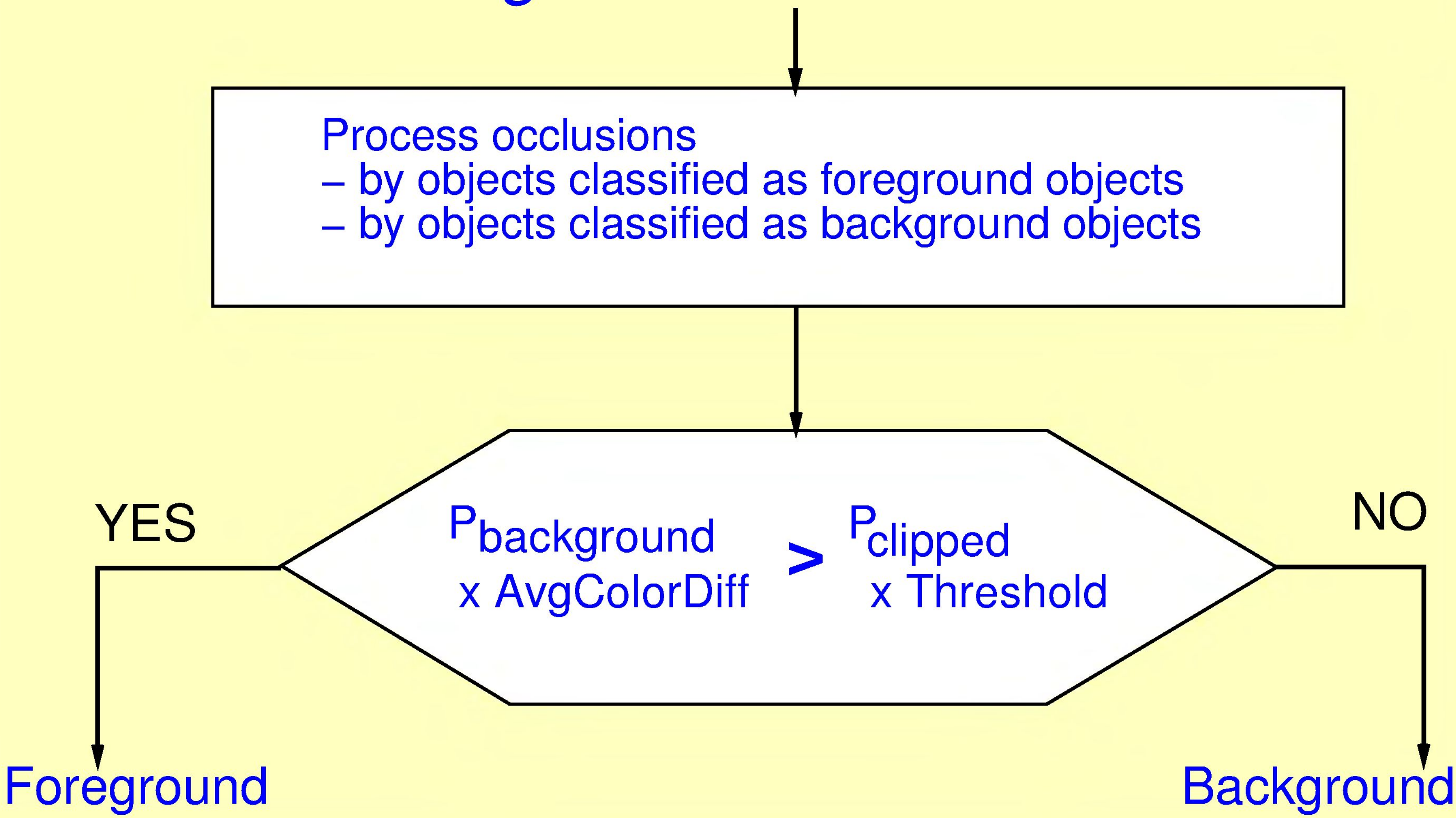


Background or foreground ?

	Coding as foreground (with jb2)	Coding as background (with iw44)
Estimate of the coding cost (bits)	$P_{clipped}$	$P_{background}$ x AvgColorDiff
Estimate of the distorsion cost (bits)	**0**	$P_{background}$ x AvgColorDiff

Segmentation : Digital : Greedy Optimisation

For all monochromatic objects
in reverse drawing order



Segmentation : Digital : Implementation

DjVuDigital (ps2djvu)

- Aladdin GhostScript driver “djvusep”
- Back–end encoders “msepdjvu”
- `http://djvuserver.research.att.com/~leonb/ps2djvu`

Example

- PostScript
- Encoded from PostScript
- Encoded from RGB image

Segmentation : Digital : Results

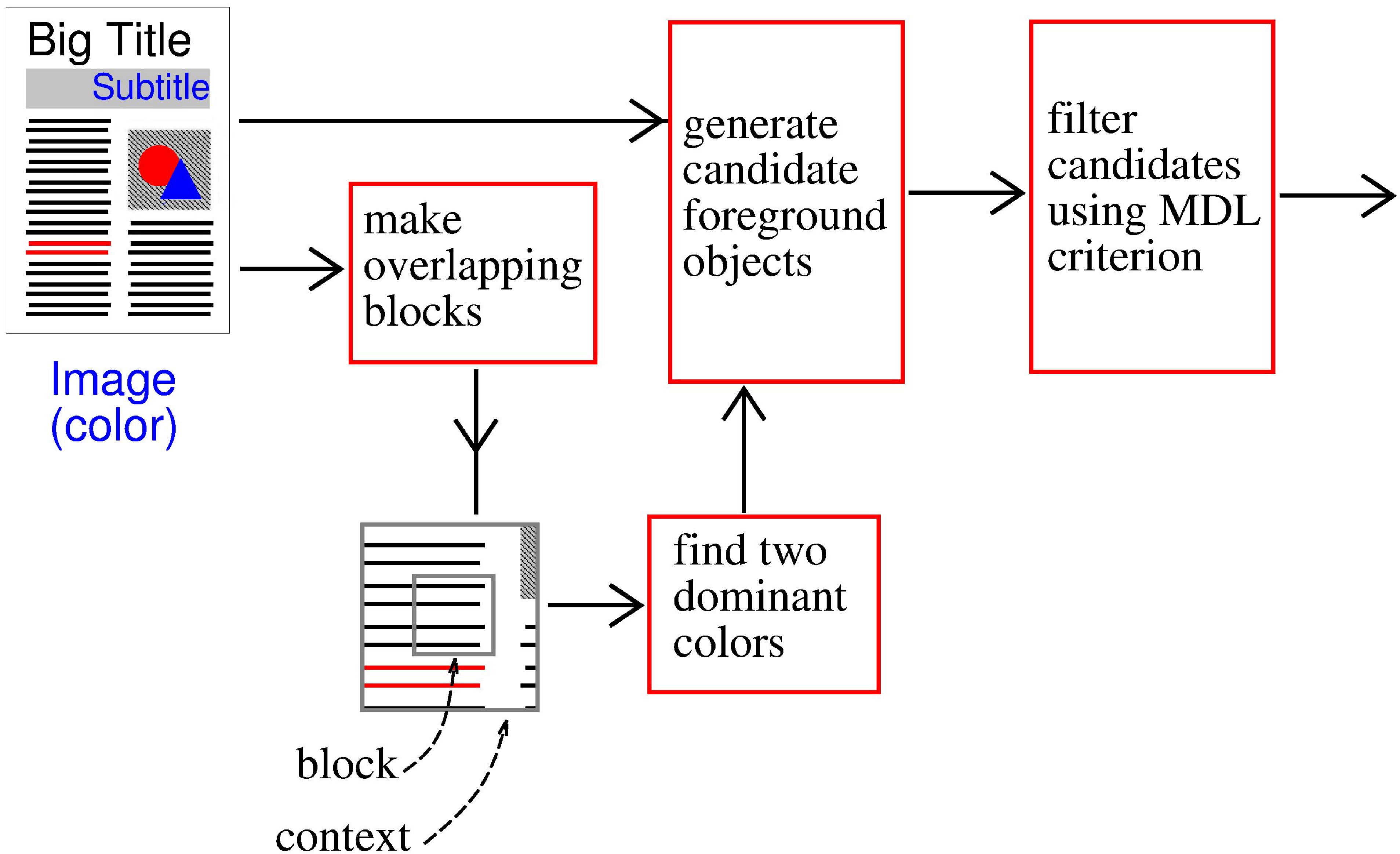
Document	Type	Pages	ps.gz / pdf	ps2djvu
mask.ps.gz	latex	10	400K	78K
paper2web.pdf	book	327	4230K	2925K
sgi.pdf	flyer	4	484K	106K
stanford.pdf	map	1	412K	170K

Total time with ps2djvu 23 + 1230 + 27 + 30 seconds

Total time before* 383 + 7900 + 201 + 123 seconds

* rendering a 300dpi rgb image with GhostScript
and converting this image to DjVu

Segmentation : Scanned : Schema



Browsing : Goal

DjVu browsing goal:
<< Replicating the “paper” experience. >>

Action	Equivalent for Paper Documents	Acceptable Response Time
Zooming/Panning	Move the eyes	Immediate
Next/Prev Page	Turn a page	< 1 second
Go To Page #	Find a page	< 3 seconds

Browsing : Progressivity

Revisiting a well known trick:

DjVu Progressive Decoding.

Chunk order determines what information comes first.

Foreground mask :	1 chunk
Foreground colors :	1 chunk
Background wavelets :	N chunks

text & lineart	fgc	bg1	bg2	bg3	b&w text / color text / text+background / / background sharpens twice.

text & lineart	fgc	bg1	2	3	4	5	6	b&w text / color text / text+background / / background sharpens four times.

bg1	fgc	text & lineart	bg2	bg3	coarse background / / color text over coarse background / background sharpens twice

Browsing : Modeless GUI

Save a split second
by minimizing delay between

User intention and user action!

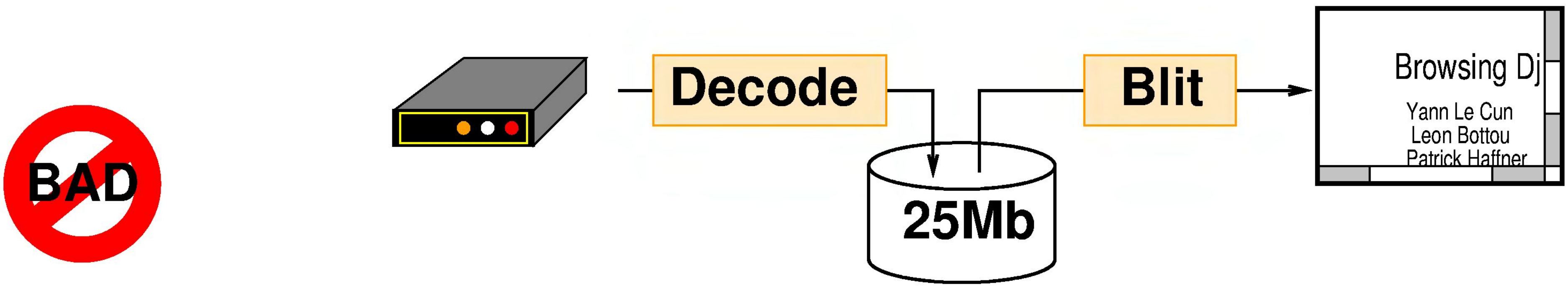
Modal GUI

Select ...	... then act
Panning mode Zooming mode Selecting mode	Point Drag Release

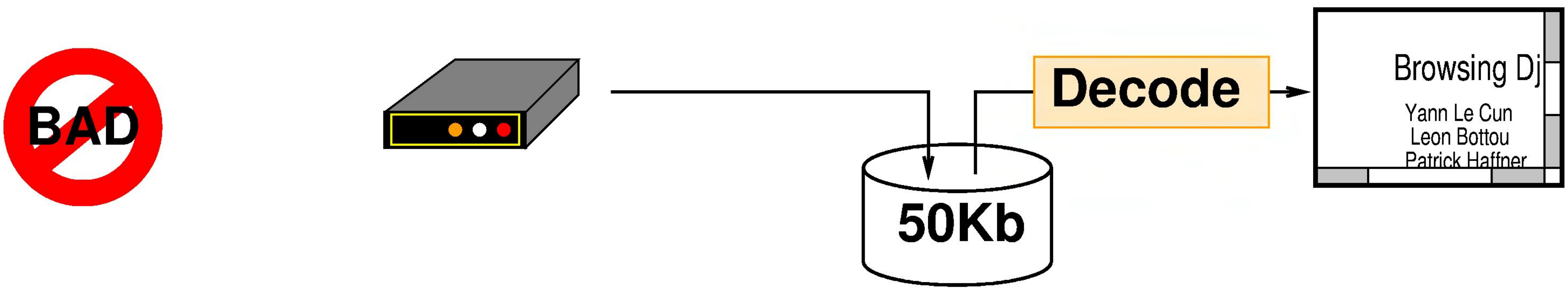
Modeless GUI

Single action
Drag whole image Click zoom button Click hyperlink

Browsing : Panning : Bad



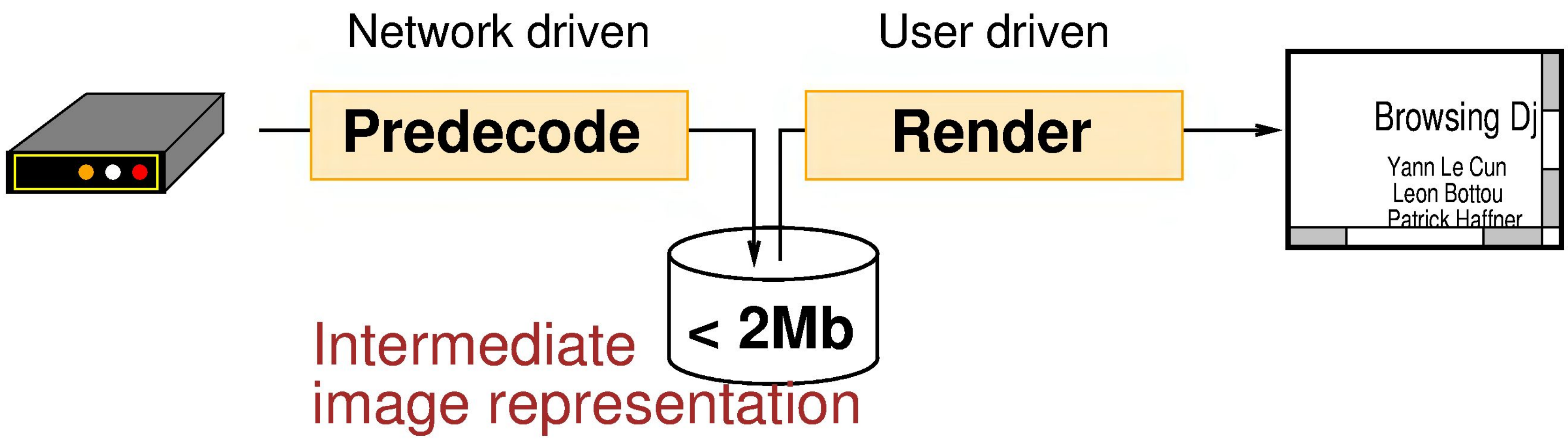
Big memory requirement --> swapping --> slow panning.
Progressive display --> full decoder run after each chunk.



Decoder runs only on exposed parts --> image tiling:
large tiles --> slow panning
small tiles --> poor compression

Browsing : Panning : Intermediate Representation

[Bottou & al. 2000]



- Predecoding time overlaps data transfer time.
- Predecoder incrementally builds the intermediate representation whenever a chunk arrives.
- Intermediate memory representation is optimized for rendering any portion of the image at any resolution.

--> Fast panning.
--> CPU efficient progressive display

Browsing : Panning : Details

Intermediate Image Representation

DjVuBitonal
(JB2)

Run–length–encoded shapes
\+ Array of (shapeno, position) records

DjVuPhoto
(IW44)

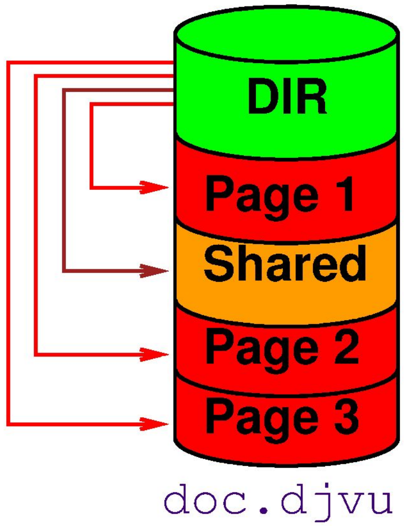
Blocks of 32x32 quantized
wavelet coefficients.

Each block is represented by
a sparse array holding only
the non zero coefficients.

Sparse array
structured as 2–level 4x16x16 tree
that matches the IW44 bitstream.

Browsing : Multipage : Formats

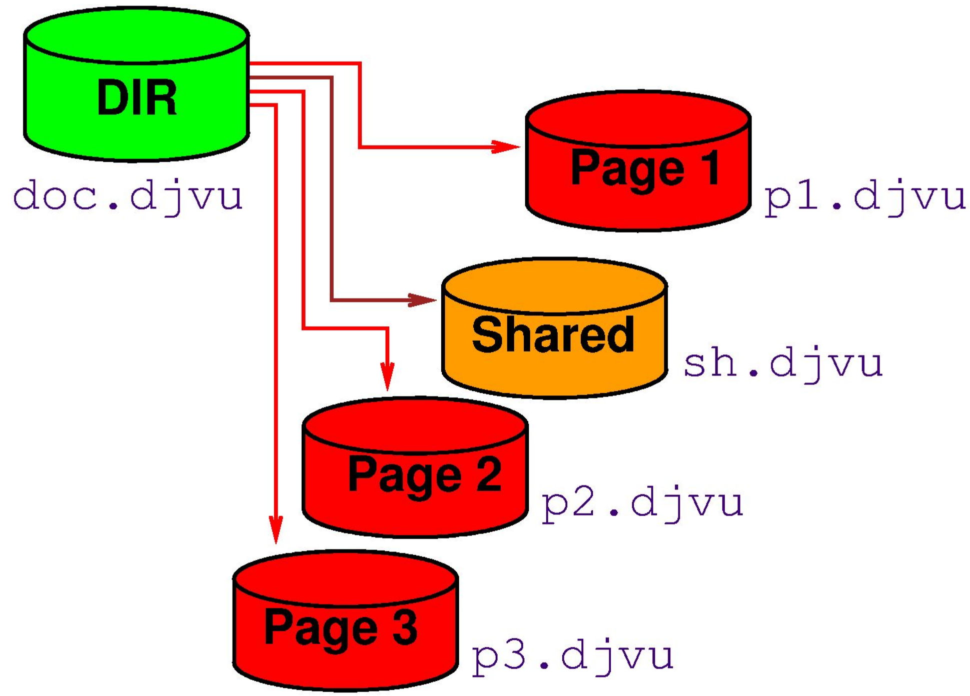
Bundled



– Initial request eventually downloads the whole document
– Efficient random access requires a byte–server.

Convenient for small documents and email attachments

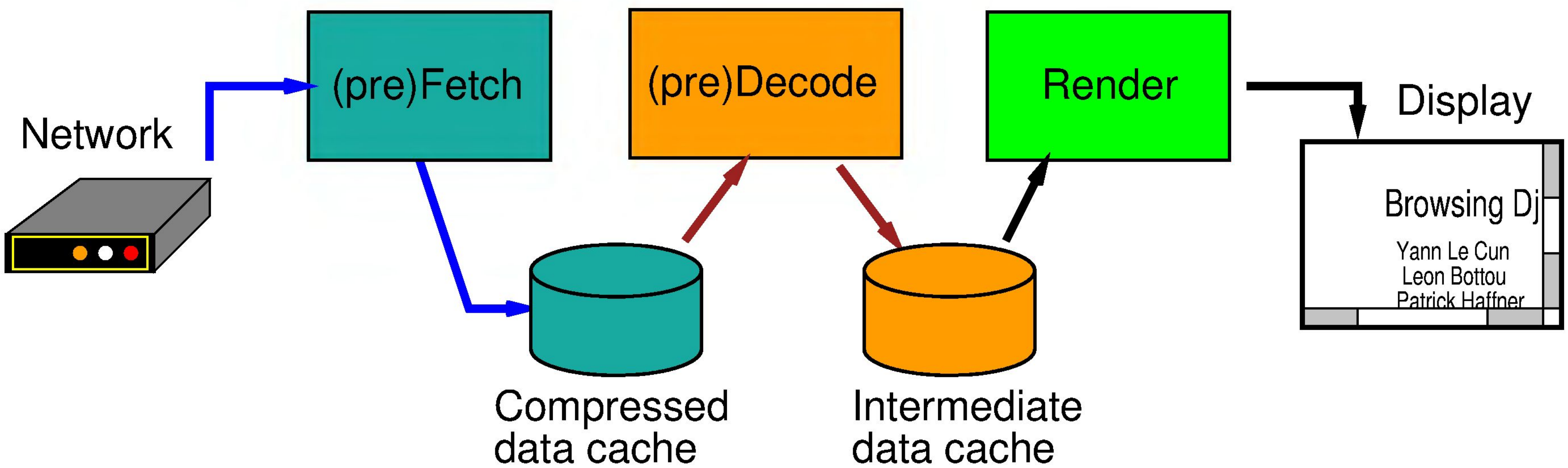
Indirect

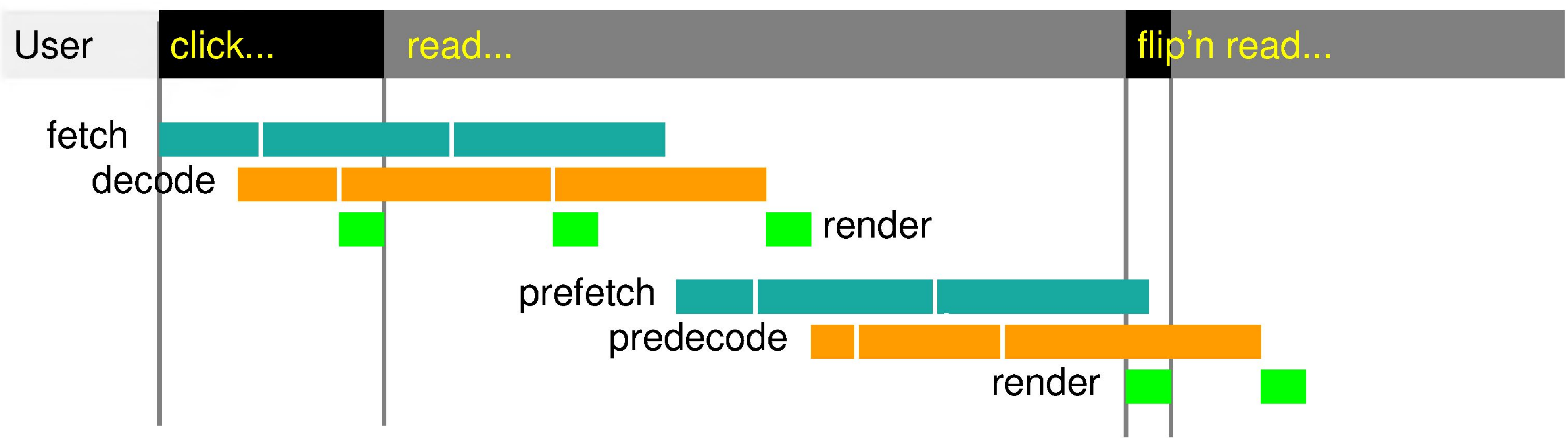


– Fine grain web–serving without a byte–server.
– Supports sophisticated caching and prefetching.
– Initial web request only downloads the directory.

Follows the Web metaphor !

Browsing : Multipage : Caching etc.





Browsing : Web Integration

Annotation Chunks

Annotations can define hyperlinks
Annotations can hilite specified areas
Servers can generate annotations on the fly

Searchable Hidden Text

Generated by OCR or otherwise

CGI–Style URLs

`http://localhost/stanley.djvu?djvuopts&page=39`