

## 15-826: Multimedia Databases and Data Mining

Lecture #24: Compression - JPEG,  
MPEG, fractal  
*C. Faloutsos*

---

---

---


---

---

---

---

---



## Must-read Material

- JPEG: Gregory K. Wallace, *The JPEG Still Picture Compression Standard*, [CACM, 34, 4](#), April 1991, pp. 31-44
- MPEG: D. Le Gall, *MPEG: a Video Compression Standard for Multimedia Applications* [CACM, 34, 4](#), April 1991, pp. 46-58
- Fractal compression: M.F. Barnsley and A.D. Sloan, *A Better Way to Compress Images*, [BYTE, Jan. 1988](#), pp. 215-223.

15-826 Copyright: C. Faloutsos (2013) 2

---

---

---

---

---

---

---

---



## Outline

Goal: 'Find similar / interesting things'

- Intro to DB
- ➔ • Indexing - similarity search
- Data Mining

15-826 Copyright: C. Faloutsos (2013) 3

---

---

---

---

---

---

---

---

CMU SCS

## Indexing - Detailed outline

- primary key indexing
- ..
- multimedia
- Digital Signal Processing (DSP) tools
- ➔ Image + video compression
  - JPEG
  - MPEG
  - Fractal compression

15-826 Copyright: C. Faloutsos (2013) 4

---

---

---

---

---

---

---

---

CMU SCS

## Motivation

- Q: Why study (image/video) compression?

15-826 Copyright: C. Faloutsos (2013) 5

---

---

---

---

---

---

---

---

CMU SCS

## Motivation

- Q: Why study (image/video) compression?
- A1: feature extraction, for multimedia data mining
- A2: (lossy) compression = data mining!

15-826 Copyright: C. Faloutsos (2013) 6

---

---

---

---

---

---

---

---

CMU SCS

## JPEG - specs

- (Wallace, CACM April '91)
- Goal: universal method, to compress
  - losslessly / lossily
  - grayscale / color (= multi-channer)
- What would you suggest?

15-826 Copyright: C. Faloutsos (2013) 7

---

---

---

---

---

---

---

---

CMU SCS

## JPEG - grayscale - outline

- step 1) 8x8 blocks (why?)
- step 2) (Fast) DCT (why DCT?)
- step 3) Quantize (fewer bits, lower accuracy) ← loss
- step 4) encoding
  - DC: delta from neighbors
  - AC: in a zig-zag fashion, + Huffman encoding

Result: 0.75-1.5 bits per pixel (8:1 compression) - sufficient quality for most apps

15-826 Copyright: C. Faloutsos (2013) 8

---

---

---

---

---

---

---

---

CMU SCS SKIP

## JPEG - grayscale - lossless

- Predictive coding:
 

C	B	
A	X	

$$X = f(A, B, C)$$
 eg.  $X = (A+B)/2$ , or?
- Then, encode prediction errors

Result: typically, 2:1 compression

15-826 Copyright: C. Faloutsos (2013) 9

---

---

---

---

---

---

---

---

CMU SCS SKIP

### JPEG - color/multi-channel

- apps?
- image components = color bands = spectral bands = channels
- components are interleaved (why?)

15-826 Copyright: C. Faloutsos (2013) 10

---

---

---

---

---

---


---

---


CMU SCS SKIP

### JPEG - color/multi-channel


- apps?
- image components = color bands = spectral bands = channels
- components are interleaved (why?)
  - to pipeline decompression with display



8x8 'red' block



8x8 'green' block



8x8 'blue' block

15-826 Copyright: C. Faloutsos (2013) 11

---

---

---

---

---

---

---

---

CMU SCS SKIP

### JPEG - color/multi-channel

- tricky issues, if the sampling rates differ
- Also, hierarchical mode of operation: pyramidal structure
  - sub-sample by 2
  - interpolate
  - compress the diff. from the predictions

15-826 Copyright: C. Faloutsos (2013) 12

---

---

---

---

---

---

---

---

CMU SCS

### JPEG - conclusions

- grayscale, lossy: 8x8 blocks; DCT; quantization and encoding
- grayscale, lossless: predictions
- color (lossy/lossless): interleave bands

15-826 Copyright: C. Faloutsos (2013) 13

---

---

---

---

---

---

---

---

CMU SCS

### Indexing - Detailed outline

- primary key indexing
- ..
- multimedia
- Digital Signal Processing (DSP) tools
- Image + video compression
  - JPEG
  - ➔ - MPEG
  - Fractal compression

15-826 Copyright: C. Faloutsos (2013) 14

---

---

---

---

---

---

---

---

CMU SCS

### MPEG

- (LeGall, CACM April '91)
- Video: many, still images
- Q: why not JPEG on each of them?

15-826 Copyright: C. Faloutsos (2013) 15

---

---

---

---

---

---

---

---

CMU SCS

## MPEG

- (LeGall, CACM April '91)
- Video: many, still images
- Q: why not JPEG on each of them?
- A: too similar - we can do better! (~3-fold)

15-826 Copyright: C. Faloutsos (2013) 16

---

---

---

---

---

---

---

---

CMU SCS

SKIP

## MPEG - specs

- ??

15-826 Copyright: C. Faloutsos (2013) 17

---

---

---

---

---

---

---

---

CMU SCS

SKIP

## MPEG - specs

- acceptable quality
- asymmetric/symmetric apps (#compressions vs #decompressions)
- Random access (FF, reverse)
- audio + visual sync
- error tolerance
- variable delay / quality
- editability

15-826 Copyright: C. Faloutsos (2013) 18

---

---

---

---

---

---

---

---

CMU SCS SKIP

## MPEG - approach

- main idea: balance between inter-frame compression and random access
- thus: compress *some* frames with JPEG (*I-frames*)
  - rest: prediction from motion, and interpolation
  - P-frames (predicted pictures, from I- or P-frames)
  - B-frames (interpolated pictures - never used as reference)

15-826 Copyright: C. Faloutsos (2013) 19

---

---

---

---

---

---

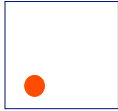
---

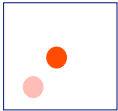
---

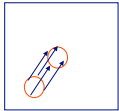
CMU SCS SKIP

## MPEG - approach

- useful concept: 'motion field'

f1  


f2  




15-826 Copyright: C. Faloutsos (2013) 20

---

---

---

---

---

---

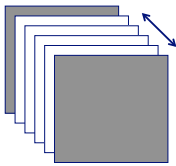
---

---

CMU SCS

## MPEG - conclusions

- with the I-frames, we have a balance between
  - compression and
  - random access



I-frame

P/B-frames

I-frame

15-826 Copyright: C. Faloutsos (2013) 21

---

---

---

---

---

---

---

---

CMU SCS

## Indexing - Detailed outline

- primary key indexing
- ..
- multimedia
- Digital Signal Processing (DSP) tools
- Image + video compression
  - JPEG
  - MPEG
  - ➔ - Fractal compression

15-826 Copyright: C. Faloutsos (2013) 22

---

---

---

---

---

---

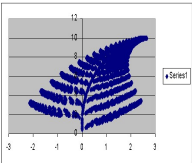
---

---

CMU SCS

## Fractal compression

- ‘Iterated Function systems’ (IFS)
- (Barnsley and Sloane, BYTE Jan. 88)
- Idea: real objects may be self-similar, eg., fern leaf



15-826 Copyright: C. Faloutsos (2013) 23

---

---

---

---

---

---

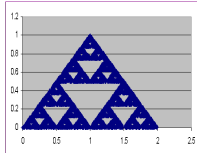
---

---

CMU SCS

## Fractal compression

- simpler example: Sierpinski triangle.
  - has details at every scale -> DFT/DCT: not good
  - but is easy to describe (in English)
- There should be a way to compress it very well!
- Q: How??



15-826 Copyright: C. Faloutsos (2013) 24

---

---

---

---

---

---

---

---



CMU SCS

## Fractal compression

- simpler example: Sierpinski triangle.
  - has details at every scale -> DFT/DCT: not good
  - but is easy to describe (in English)
- There should be a way to compress it very well!
- Q: How??
- A: several, affine transformations
- Q: how many coeff. we need for a (2-d) affine transformation?

15-826 Copyright: C. Faloutsos (2013) 25

---

---

---

---

---

---


---

---

CMU SCS

## Fractal compression

- A: 6 (4 for the rotation/scaling matrix, 2 for the translation)
- $(x,y) \rightarrow w((x,y)) = (x', y')$ 
  - $x' = a x + b y + e$
  - $y' = c x + d y + f$
- for the Sierpinski triangle: 3 such transformations - which ones?



15-826 Copyright: C. Faloutsos (2013) 26

---

---

---

---

---

---

---

---



CMU SCS

## Fractal compression

- A:
 

	a	b	c	d	e	f	p
w1	0.5	0	0	0.5	0	0	1/3
w2	0.5	0	0	0.5	1	0	1/3
w3	0.5	0	0	0.5	0.5	0.5	1/3

prob (~ fraction of ink)
 

15-826 Copyright: C. Faloutsos (2013) 27

---

---

---

---

---

---

---

---

CMU SCS

## Fractal compression

- The above transformations ‘describe’ the Sierpinski triangle - is it the only one?
- ie., how to de-compress?

15-826 Copyright: C. Faloutsos (2013) 28

---

---

---

---

---

---

---

---

CMU SCS

## Fractal compression

- The above transformations ‘describe’ the Sierpinski triangle - is it the only one?
- A: YES!!!
- ie., how to de-compress?
- A1: Iterated functions (expensive)
- A2: Randomized (surprisingly, it works!)

15-826 Copyright: C. Faloutsos (2013) 29

---

---

---

---

---

---

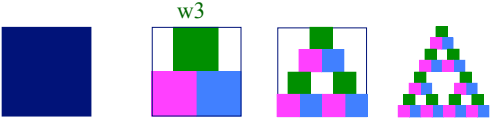
---

---

CMU SCS

## Fractal compression

- Sierpinski triangle: is the ONLY fixed point of the above 3 transformations:



15-826 Copyright: C. Faloutsos (2013) 30

---

---

---

---

---

---

---

---

CMU SCS

## Fractal compression

- We'll get the Sierpinski triangle, NO MATTER what image we start from! (as long as it has at least one black pixel!)
- thus, (one, slow) decompression algorithm:
  - start from a random image
  - apply the given transformations
  - union them and
  - repeat recursively
- drawback?

15-826 Copyright: C. Faloutsos (2013) 31

---

---

---

---

---

---

---

---

CMU SCS

## Fractal compression

- A: Exponential explosion: with 3 transformations, we need  $3^k$  sub-images, after  $k$  steps
- Q: what to do?

15-826 Copyright: C. Faloutsos (2013) 32

---

---

---

---

---

---

---

---

CMU SCS

## Fractal compression

- A: PROBABILISTIC algorithm:
  - pick a random point  $(x_0, y_0)$
  - choose one of the 3 transformations with prob.  $p_1/p_2/p_3$
  - generate point  $(x_1, y_1)$
  - repeat
  - [ignore the first 30-50 points - why??]
- Q: why on earth does this work?
- A: the point  $(x_n, y_n)$  gets closer and closer to Sierpinski points  $(n=1, 2, \dots)$ , ie:

15-826 Copyright: C. Faloutsos (2013) 33

---

---

---

---

---

---

---

---

CMU SCS

## Fractal compression

... points outside the Sierpinski triangle have no chance of attracting our 'random' point  $(x_n, y_n)$

Q: how to compress a real (b/w) image?  
 A: 'Collage' theorem (informally: find portions of the image that are miniature versions, and that cover it completely)

Drills:

15-826 Copyright: C. Faloutsos (2013) 34

---

---

---

---

---

---


---

---

CMU SCS

## Fractal compression

Drill#1: compress the unit square - which transformations?



15-826 Copyright: C. Faloutsos (2013) 35

---

---

---

---

---

---


---

---

CMU SCS

## Fractal compression

Drill#1: compress the unit square - which transformations?



15-826 Copyright: C. Faloutsos (2013) 36

---

---

---

---

---

---


---

---

CMU SCS

### Fractal compression

Drill#2: compress the diagonal line:



15-826 Copyright: C. Faloutsos (2013) 37

---

---

---

---

---

---

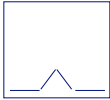
---

---

CMU SCS

### Fractal compression

Drill#3: compress the 'Koch snowflake':



15-826 Copyright: C. Faloutsos (2013) 38

---

---

---

---

---

---

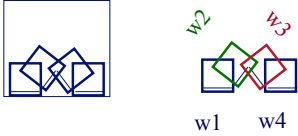
---

---

CMU SCS

### Fractal compression

Drill#3: compress the 'Koch snowflake': (we can rotate, too!)



15-826 Copyright: C. Faloutsos (2013) 39

---

---

---

---

---

---

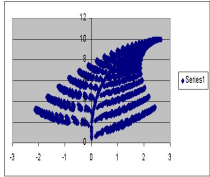
---

---

CMU SCS

### Fractal compression

Drill#4: compress the fern leaf:



15-826 Copyright: C. Faloutsos (2013) 40

---

---

---

---

---

---

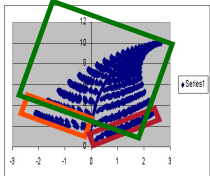
---

---

CMU SCS

### Fractal compression

Drill#4: compress the fern leaf: (rotation + diff.  $p_i$ )



PS: actually, we need one more transf., for the stem

15-826 Copyright: C. Faloutsos (2013) 41

---

---

---

---

---

---

---

---

CMU SCS

### Fractal compression

- How to find self-similar pieces automatically?
- A: [Peitgen+]: eg., quad-tree-like decomposition

15-826 Copyright: C. Faloutsos (2013) 42

---

---

---

---

---

---

---

---

CMU SCS

## Fractal compression

- Observations
  - may be lossy (although we can store deltas)
  - can be used for color images, too
  - can ‘focus’ or ‘enlarge’ a given region, without JPEG’s ‘blockiness’

15-826 Copyright: C. Faloutsos (2013) 43

---

---

---

---

---

---

---

---

CMU SCS

## Conclusions

- JPEG: DCT for images
- MPEG: I-frames; interpolation, for video
- IFS: surprising compression method

15-826 Copyright: C. Faloutsos (2013) 44

---

---

---

---

---

---

---

---

CMU SCS

## Resources/ References

- IFS code: [www.cs.cmu.edu/~christos/SRC/ifs.tar](http://www.cs.cmu.edu/~christos/SRC/ifs.tar)
- Gregory K. Wallace, *The JPEG Still Picture Compression Standard*, CACM, 34, 4, April 1991, pp. 31-44

15-826 Copyright: C. Faloutsos (2013) 45

---

---

---


---

---

---

---

---

 CMU SCS

### References

- D. Le Gall, *MPEG: a Video Compression Standard for Multimedia Applications* CACM, 34, 4, April 1991, pp. 46-58
- M.F. Barnsley and A.D. Sloan, *A Better Way to Compress Images*, BYTE, Jan. 1988, pp. 215-223
- Heinz-Otto Peitgen, Hartmut Juergens, Dietmar Saupe: *Chaos and Fractals: New Frontiers of Science*, Springer-Verlag, 1992

15-826 Copyright: C. Faloutsos (2013) 46

---

---

---

---

---

---

---

---