

LIS-5362

Multimedia Concepts

Remember:

There is no *MAGIC*..

...but wow it really can look like it.

Remember:

- We're presuming that:
 - - the local computer is fast
 - and
 - File storage is costly
 - - the network is slow
- => thus, small filesizes are better.

File Compression (ZIP)

- Lossless (or “Perfect”) file compression.

To make smaller, so as to be able to store more, or send faster.

But also, to reproduce PERFECTLY.

(it's not magic)

Consider:

- *“Penelope and Robert Jones Smith went to the car and grabbed the bat and the ball and the chair and the Doritos and Penelope's coat and Robert Jones Smith's favorite suit and the directions to the park. Penelope told Robert Jones Smith that they and the other people were going to have a wonderful and fun and lovely day. Robert Jones Smith told Penelope that he agreed. Also, that her name started with P.”*

First step:

- Turn all the ands into &.

- &->and
- *Penelope & Robert Jones Smith went to the car & grabbed the bat & the ball & the chair & the Doritos & Penelope's coat & Robert Jones Smith's favorite suit & the directions to the park. Penelope told Robert Jones Smith that they & the other people were going to have a wonderful & fun & lovely day. Robert Jones Smith told Penelope that he agreed. Also, that her name started with P.*

But this doesn't have to read like English...

- &->and
- #->the

Penelope & Robert Jones Smith went to # car & grabbed # bat & # ball & # chair & # Doritos & Penelope's coat & Robert Jones Smith's favorite suit & # directions to # park. Penelope told Robert Jones Smith that #y & # o#r people were going to have a wonderful & fun & lovely day. Robert Jones Smith told Penelope that he agreed. Also, that her name started with P.

Robert has a long name...

- &->and
- #->the
- RJS->Robert Jones Smith
- *Penelope & RJS went to # car & grabbed # bat & # ball & # chair & # Doritos & Penelope's coat & RJS's favorite suit & # directions to # park.
Penelope told RJS that #y & # o#r people were going to have a wonderful & fun & lovely day.
RJS told Penelope that he agreed. Also, that her name started with P.*

Penelope too, but wait...

- “Also, that her name started with P”



“Also, that her name started with Penelope”

So, then...

- &->and
- #->the
- RJS->Robert Jones Smith
- PP->P
- P->Penelope (UNLESS ITS PP)
-
- P & RJS went to # car & grabbed # bat & # ball & # chair & # Doritos & P's coat & RJS's favorite suit & # directions to # park. P told RJS that #y & # o#r people were going to have a wonderful & fun & lovely day. RJS told P that he agreed. Also, that her name started with PP.

But what if the next file is?

- P P P P
-
- →

PP PP PP PP

The file gets BIGGER.

But what if the next file is?

...file gets bigger..

In other words, this is proof that there is never “magic.”

Specifically, for ANY compression algorithm, there is at least some data you could put in that makes it larger/worse.

To sum up compression

Somewhere, there is “repetition” or a “predictable pattern,” and so you tell the computer to take advantage of that.

(whether or not you do it “perfectly” is the difference between lossless and lossy)

Very interesting implications:

Here's how to sound smart:

Kolmogorov Complexity!!

What is the **minimum** amount of data needed to (fully) store a “concept” or “thing?”

(and how can we hack around it?)

Eg.

What about things like “the value of pi.”

We know that storing pi AS A DECIMAL is infinitely complex. It literally can't be done.

But ..

Complexity

But what about the phrase:

”The ratio of a circle’s circumference to its radius”

That fits in a tweet.

(a fancy mathy way to talk about this is “Kolmogorov Complexity”)

Pictures on the web:

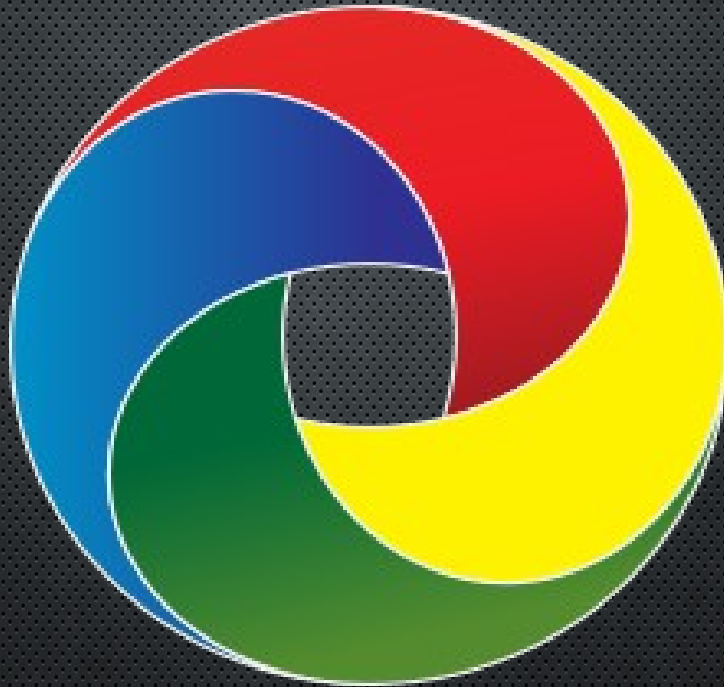
- Basically, two formats:

Raster and Vector.

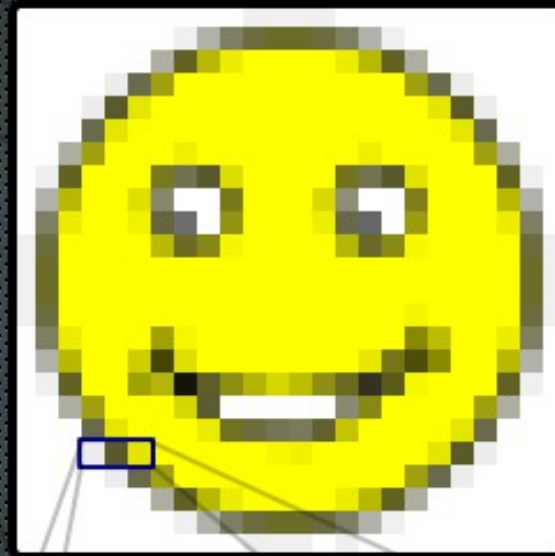
Most of the time you're looking at RASTER,
which is basically a grid*

Vector Graphics

- Use math (specifically, geometry) to tell the computer how to draw the lines and do the colors – then render it somehow



Lossless (RAW) Raster



R 93%	R 35%	R 90%
G 93%	G 35%	G 90%
B 93%	B 16%	B 0%

Lossless (image) compression

- Instead of
“0,0 = white 0,5 = white
- 0,1 = white 0,6 = white
- 0,2 = white 0,7 = white
- 0,3 = white 0,8 = white
- 0,4 = white 0,9 = white”...

Just say..

0,0 through 0,10 = white.

This also works for audio...

1 sec. = silence

2 sec. = silence

3 sec. = silence....etc

just say

1-10 sec. = silence

..and video

“at 0,0 for 10 seconds = white”

Lossy Compression

- As implied, you LOSE some of the original data, by design.

“Goldilocks and the 3 bears” (with lossy compression)

-
-
- Some nosey girl went up in some bears' house, ate their food, fell asleep and freaked out when the bears came back.

Lossy compression = Good for analogue/multimedia

Tiny filesizes; takes advantage of the limitations in human perception.

We cannot perceive individual pixels (or samples) when surrounded by other meaningful pixels/samples. They blur, frequently in predictable patterns.

- “if 0,0 is white, 0,1 is likely to be white, or light gray”

“if 0,1 is light gray, 0,2 is likely to be slightly darker gray...”

- What happens if you compress something losslessly over and over?

What happens if you compress something lossily over and over?

Images

RAW/LOSSLESS: GIF, PNG, RAW, TIFF, BMP

LOSSY: JPG, “webm”

VECTOR (lossless by definition) : SVG

AUDIO

RAW: WAV

LOSSLESS: FLAC

LOSSY: MP3, Ogg Vorbis

(then there's *MIDI*, which is basically “sheet music.”)

A bit of mythbusting...

- 192kb mp3 is probably fine
- Humans DEFINITELY can't distinguish beyond 320k
- Don't get me started on vinyl fetishists..

(Just rip your cd's to 320K..)

(and don't buy monster cables)

Video:

RAW/LOSSLESS: You're pretty much never gonna see it.

LOSSY: MPG, AVI, DIVX, webm, Ogg Theora.

(hint, the “OGGs” are the only “completely open” ones)

Understanding Power

```
10 PRINT "John is AWESOME";  
20 GOTO 10
```


Teaching the robots to escape

- 1) If there's a door in arms-reach, exit – you're done, else
- 2) If you can, take one step forward then goto 1), else
- 3) Rotate to the left until there's not a wall in front of you then goto 1)

(this will get you out of any “regular” empty room)

An almost random bit on recursion

- In computers, it's actually okay to define something with itself.

PSUEDOCODE!

Define function="EscapeFromRoom"{

- 1) If there's a door in arms-reach, exit - you're done, else
 - 2) If you can, take one step forward then EscapeFromRoom, else
 - 3) Rotate to the left until there's not a wall in front of you then EscapeFromRoom
- }

(this will get you out of any "regular" empty room)

Go to the store; if they have 2% lactose free chocolate milk, then get me a carton.

Misusing Power

```
go to the store;
```

```
if [[ they have 2% lactose free  
chocolate milk]]
```

```
then
```

```
    get me a carton.
```


The Magic Genie
Recursion, trees, and “crowdsourcing”

{0) Start with “Is it Batman”?}

1) Ask my (yes/no) questions down the tree

2) If win, “yay”

3) If lose, add/replace new last question to one for which my guess was wrong and her guess was right (*optionally, try to be general or “half-y”?*)

- repeat until genius

The Magic Genie

Now, there are other ways to

”suck up all the data”

and

“give back info based on it.”

(Simple form – database or other search, eg. google)

The Magic Genie

Four score and seven _____

(what will google yield here and why?)

The Magic Genie

Four score and seven _____

(what will google yield here and why?)

“years,” of course, because that’s what came up the most

The Magic Genie

Okay, what about..

Seven Years _____

(ago?)

(in Tibet?)

(war?)

(a slave?)

The Magic Genie

All possible, especially given how you may weight things.

And you can imagine..

War .. pigs .. in a blanket ... pardon,
something...

(but also, you can imagine this might yield a
real sentence if you weight it right..)

MARKOV CHAINS

That's a "Markov Chain."

You look at the last thing, look at a database, pick what comes next based on whatever you find there.

But lets put some stuff together

- Image compression (using math to predict patterns)
- Image denoising (same thing)
- Figuring out “patterns” you might find based on words attached to the pictures, like millions of them
- Markov Chains (step by step, hey what does this look like...)

Latent Diffusion

“Latent” – we, the humans, don’t really tell the computer what “things look like,” we throw pictures at it and tell it to figure it out.

”Diffusion” – basically, spreading of random noise (and then fixing it. I think)