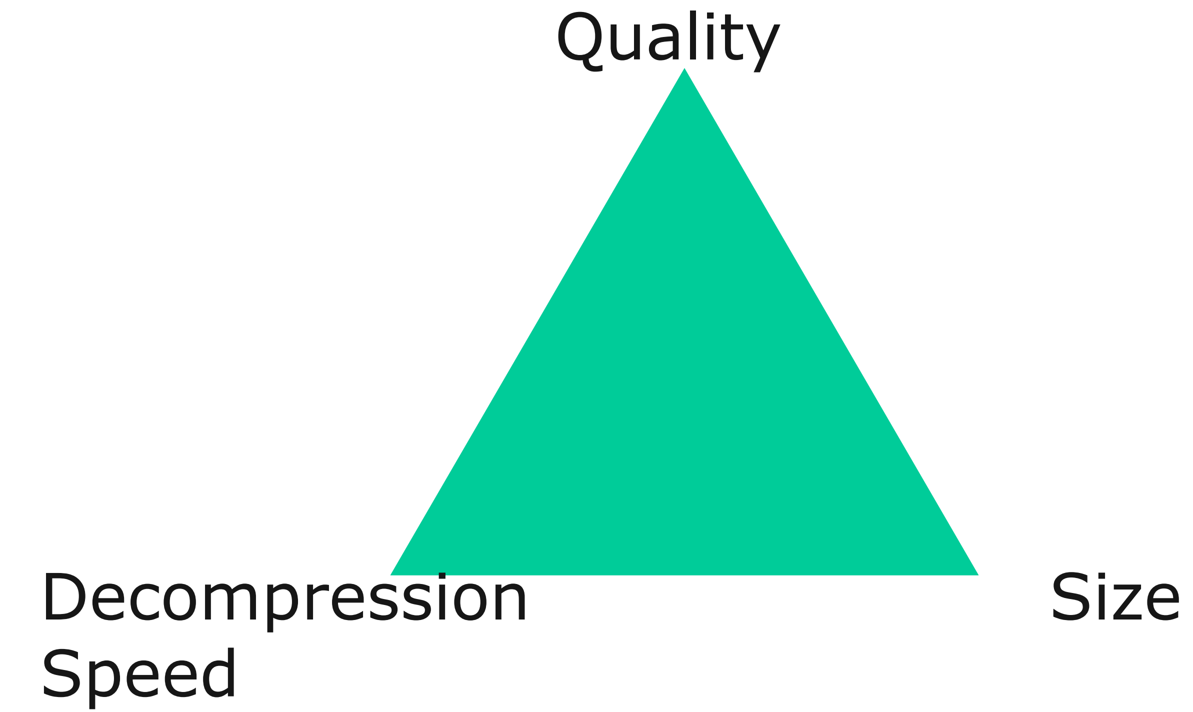
DXT is NOT ENOUGH

Colt "MainRoach" McAnlis
Developer Advocate at Google

GAME DEVELOPERS CONFERENCE
SAN FRANCISCO, CA
MARCH 5-9, 2012
EXPO DATES: MARCH 7-9
2012

Gathered here today...

- Texture footprint matters for games
- Retail moving to 17GB of DVDs
- Not OK for digital distrib & consumers!


Quality
Decompression Speed
Size

The way it's done

- Most people simply zip their DXT data
  - In archive w/ other data
- Memcpy right to the GPU

Why do I care?

- You should not keep your full zip archive in memory.
    - You should only keep around what’s streamed
- Tough to bin-sort all your assets into proper archives
    - So instaed, we leave textures hyper compressed.

IDtech5

- RAGE had different requirements
  - Tons more texture data
- Stored textures as a hyper compressed
- Converted to DXT @ runtime
- 112 MP/sec on dual core

Down-sides

- Very processor intensive
- Introduces 2x noise
- DXT color quality is very low

Different Idea

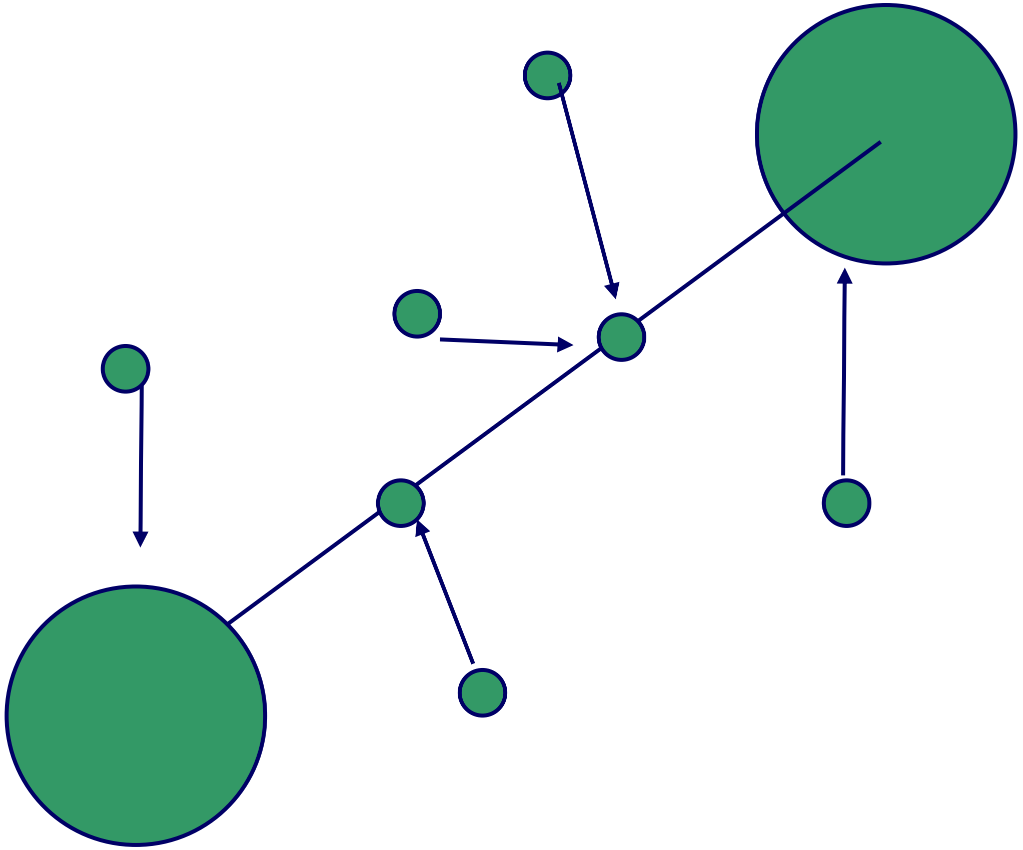
- What if we post-compress the DXT data?
  - No error introduced
  - Can store in memory hyper compressed

Data set

- Random collection of images
- Some from games (source imgs)
- Some from public (lena)
- Some from img libraries(kodak)
- All numbers include DDS headers! (128b)
- All %s are *amount of reduction*

DXT

hiColor : 5:6:5			
loColor : 5:6:5			
11	01	00	10
11	01	10	10
00	10	01	00
00	10	01	00



DXT

- Orig 37mb
- Dxt1 – 7.63mb
- Dxt1 + zip – 4.82mb (36.83%)
- Dxt1 + zip (indv) – 5.1mb

Can we beat this?

All %s are *amount of savings*

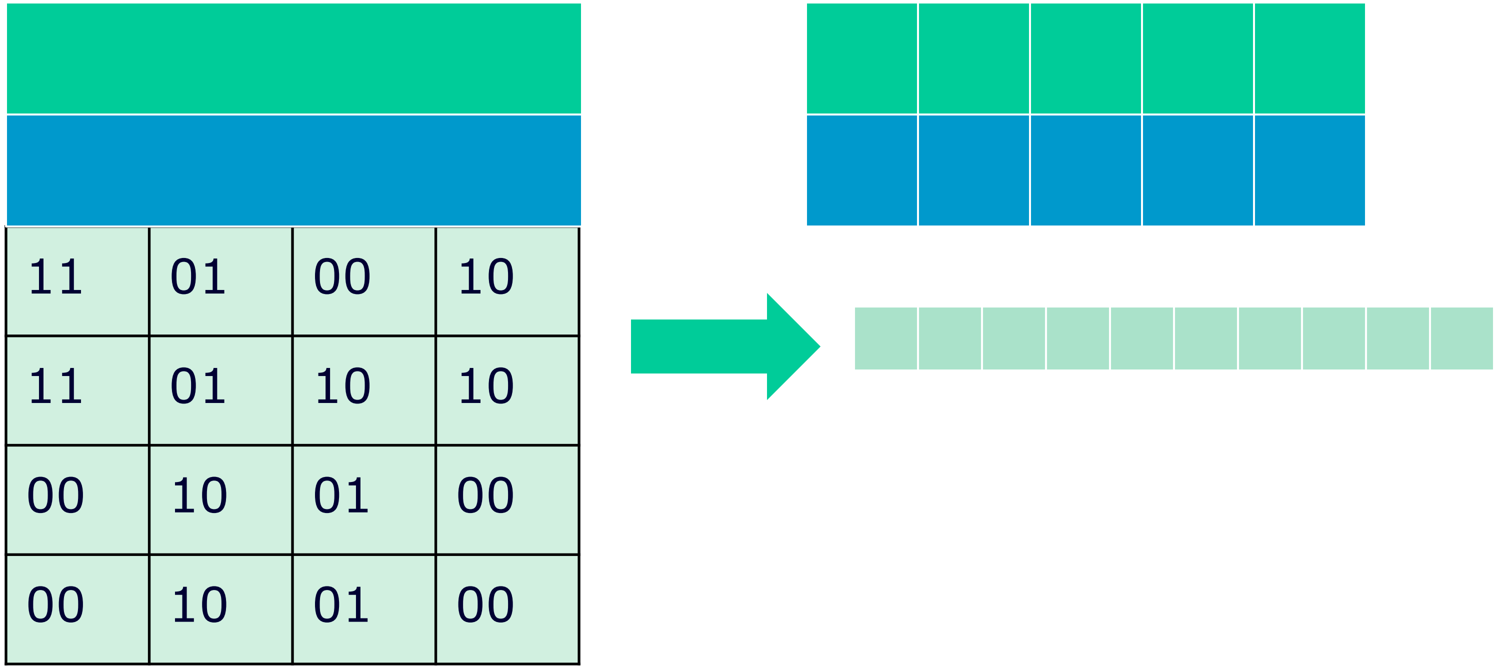
Bag of tricks - lossless

- De-interleaving
- Huffman compression
- Delta encoding
- Codebooks

Back of tricks lossy

- Expanding blocks / ROI

De-interleaving



DXTi (De-interleaving)

- Dxt1 – 7.63mb
- Dxt1i – 7.63mb (0%)
- Dxt1i + zip– 4.33mb (43.25%)

All %s are *amount of reduction*

Huffman compression

- Dictionary system
  - Creates a dictionary of input symbols
  - Replaces symbols in stream with minimum bit-codes (like Morse code)
- AAAABBC (56 bits)
- 0000 11 10 (8 bits)

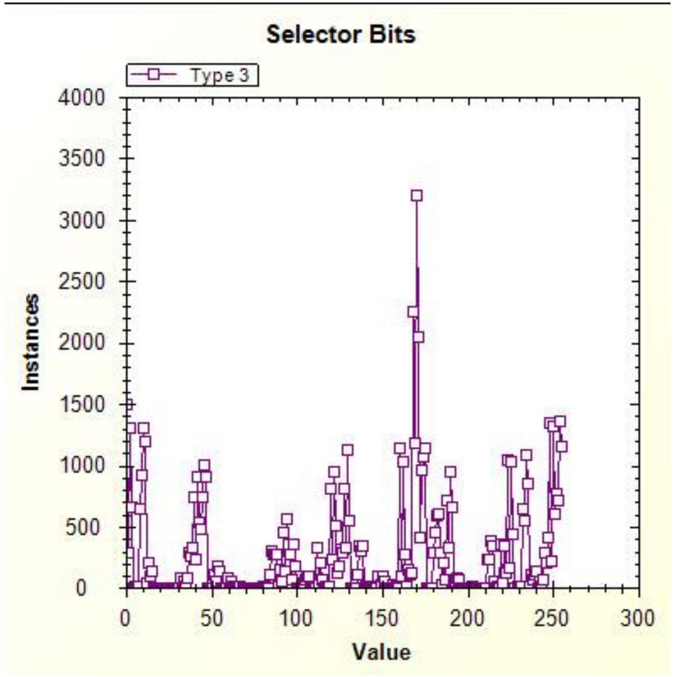
DXTih (+ huffman)

16b colors, 8b sel

- Dxt1 – 7.63mb
- Dxt1dih – 4.56mb (40.23%)
- Dxt1dih+zip – 4.27mb (44.04%)

All %s are *amount of reduction*

Better selector selection.



http://www.sebbylive.com/projects/texture-compression/improving-dxt-compression-file-sizes/

Delta encoding

- Creates duplicate symbols for easier compression

- 155,156,157,157,157,221,222,225
- 155,1,1,0,0,64,1,3

DXT1ihd (+ delta encoding)

- Dxt1 – 7.63mb
- Dxt1ihd – 4.48mb (41%)
- Dxt1ihd + zip – 4.17mb (45%)

All %s are *amount of reduction*

Code books

- Create codebook of colors (unique)
  - Delta encode them
- In Block stream, store 256 bit index into codebook
- Use sliding window approach to ensure that you'll always have a 256 bit index
  - NOTE, makes codebook base bigger..

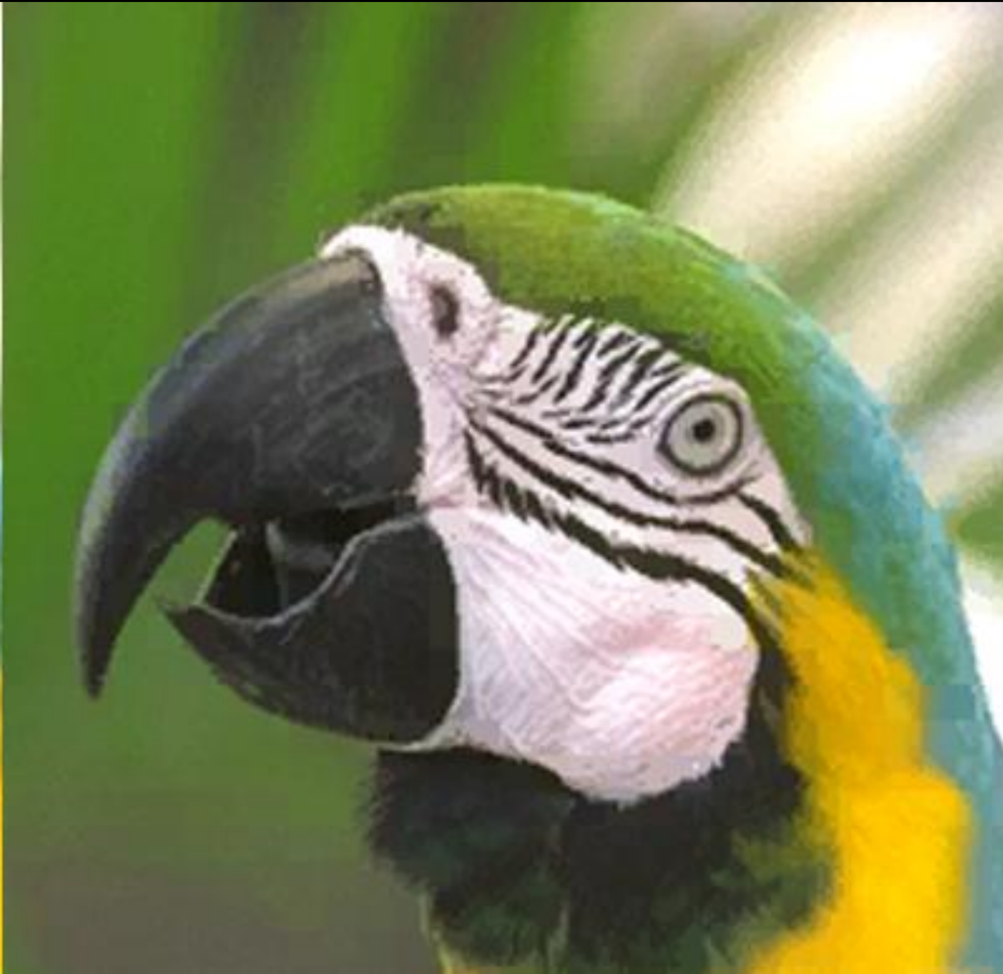
DXT1ihdc (+ code book)

- Dxt1 – 7.63mb
- Dxt1ihdc – 4.21mb (46%)
- Dxt1ihdc + zip – 3.87mb (49%)

All %s are *amount of reduction*

Expanding blocks

- Adjacent cells often share color profiles
- Use 8x8 cells
  - 1 hi 1 lo color per 8x8
  - 64 2b selectors



DXTihc8 (+ 8x8 blocks)

BOOM

- Dxt1 – 7.36mb
- Dxt1ihc8 – 2.46mb (67.7%)
- Dxt1ihc8 + zip – 2.46mb (67.7%)

All %s are *amount of reduction*

Timings

- Dxt1_ihdc8 –
  - CS101 style huffman & delta encoding
    - (ie not optimized at all)
- ~67.759% compression savings
- ~73.28 MP/sec

1.32 bpp!

YMMV

- Normal Textures - dxt1_ihdc8
  - ~70.33% reduction
- AO textures – dxt1_ihdc16
  - ~82.94% reduction

Big reveal

- Variable block (4-16)
- De interleaved, delta encoded, huffman

- ~80% reduction @ 93MP/s (diffuse texs)

0.8 bpp!

Bigger reveal

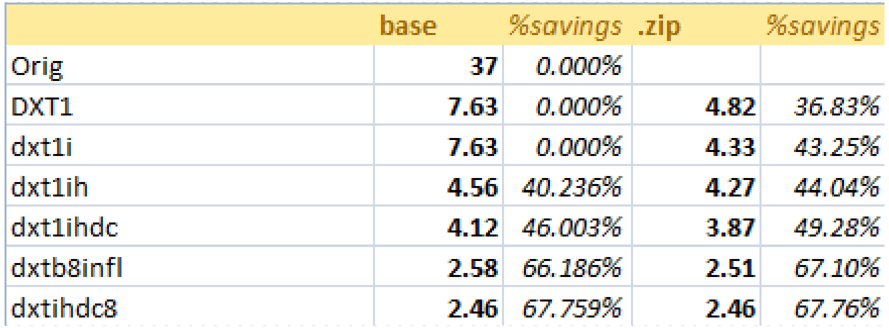
- CRUNCH codec
- 256mt/sec
- ~0.1 bpp

Take away

- Easy to get savings with simple algorithms
- YMMV for texture types
- Spend time offline doing best compression

THANK YOU!

Special thanks:
Rich Geldreich, John Brooks, Ken Adams



	base	%savings	.zip	%savings
Orig	37	0.000%		
DXT1	7.63	0.000%	4.82	36.83%
dxt1i	7.63	0.000%	4.33	43.25%
dxt1ih	4.56	40.236%	4.27	44.04%
dxt1ihdc	4.12	46.003%	3.87	49.28%
dxtb8infl	2.58	66.186%	2.51	67.10%
dxtihdc8	2.46	67.759%	2.46	67.76%

Colt “MainRoach” McAnlis | colton@google.com