Dithered Index

An Experiment in Dithering and Texture Compression

Diana Arrieta

August 28, 2013

Outline

Introduction

Texture Compression

- Intro
    - What is it?

Introduction

Texture Compression

- Intro
  - What is it?
- Motivation

Introduction

Texture Compression

- Intro
  - What is it?
- Motivation
  - Smaller: Decreased Storage Requirements

Introduction

Texture Compression

- Intro
  - What is it?
- Motivation
  - Smaller: Decreased Storage Requirements
  - Faster: Decreased Bandwidth Requirements

Introduction

Texture Compression

- Intro
    - What is it?
- Motivation
    - Smaller: Decreased Storage Requirements
    - Faster: Decreased Bandwidth Requirements
    - Better: More Information Can Be Stored

Introduction

Texture Compression

- Intro
  - What is it?
- Motivation
  - Smaller: Decreased Storage Requirements
  - Faster: Decreased Bandwidth Requirements
  - Better: More Information Can Be Stored
- Applications

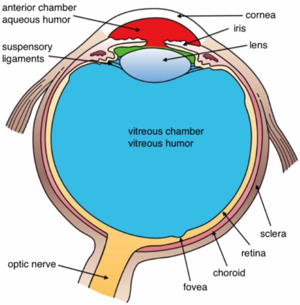
Introduction

Texture Compression

- Intro
  - What is it?
- Motivation
  - Smaller: Decreased Storage Requirements
  - Faster: Decreased Bandwidth Requirements
  - Better: More Information Can Be Stored
- Applications
  - Video Games
  - Simulations
  - Rendering
  - Medical Settings, etc.

Human Vision

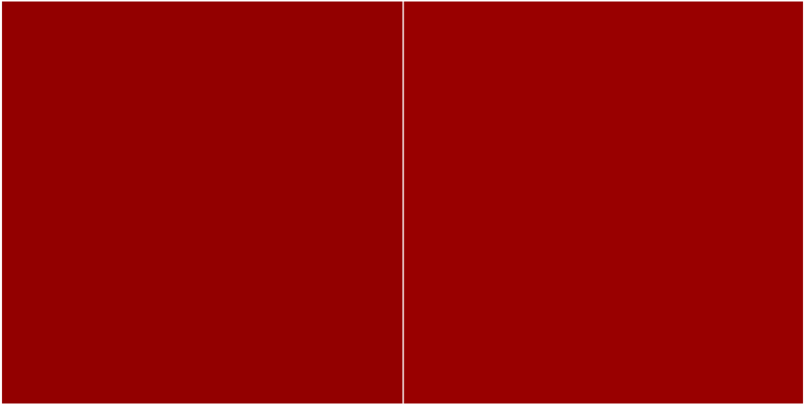
The human eye...



- Can perceive about 10 million colors
  - Compare that to 16.7 million colors on your 24-bit LCD screen!

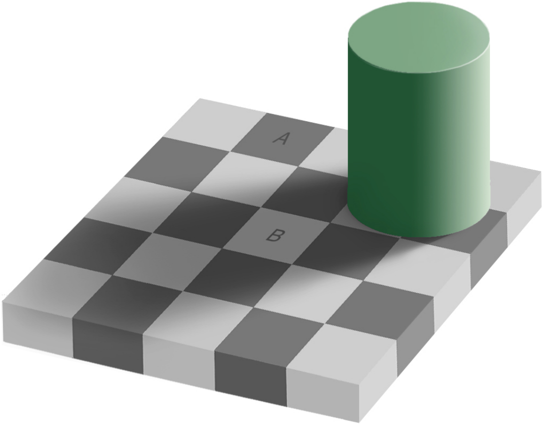
A visual exploit...

- Difficult to differentiate between similar colors.



A different visual exploit...

- Perceived colors are subject to lighting conditions and context.


A
B

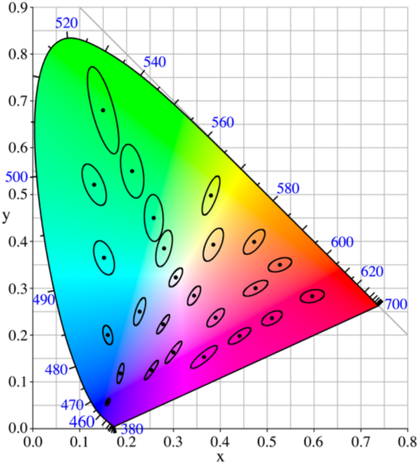


Figure 1 : MacAdam ellipses shown ten times their actual size on the CIE 1931 XYZ color space. Colors inside an ellipse visually match the color in the center.

Dithering

- An optical illusion of color depth in where colors not available in the provided palette are approximated by placing similar colors in a pattern to achieve the desired color.

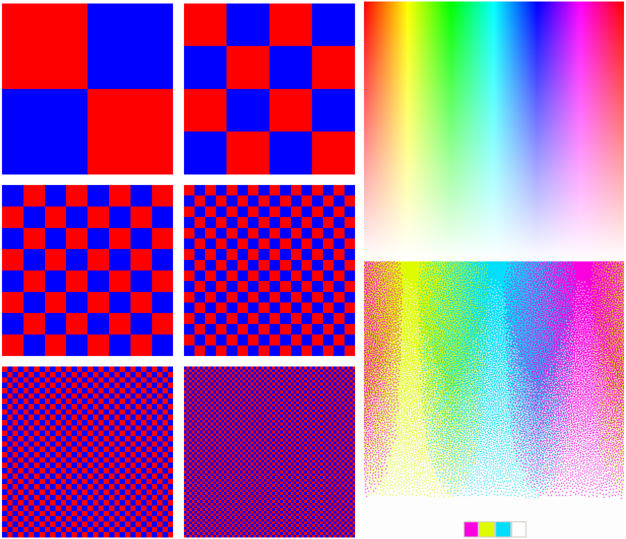


Figure 2 : Using dithering, more colors can be represented using a reduced color palette.

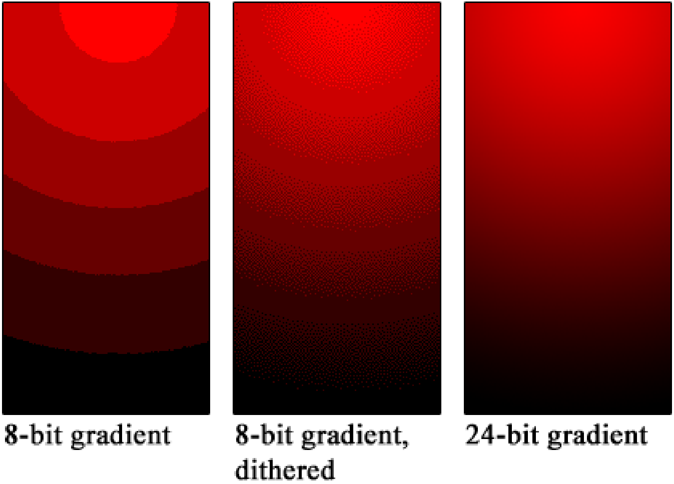
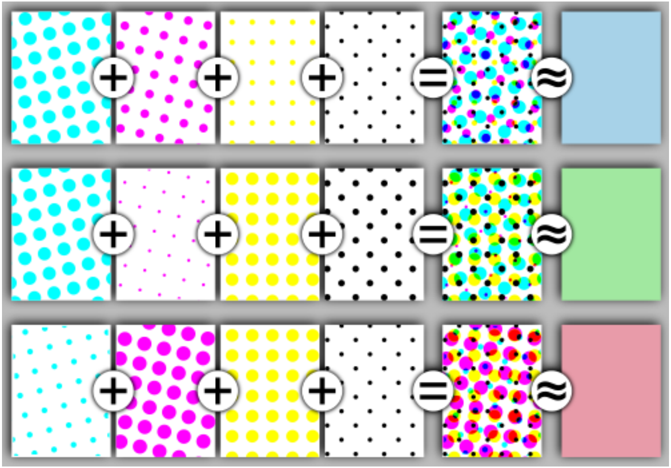


Figure 3 : Banding in a gradient. Banding is reduced when dithering is applied.

- What's the difference between Dithering and Half-toning?

- What's the difference between Dithering and Half-toning?
    - Half-toning uses CMYK colors, varying dot sizes, and overlapping techniques to achieve the color desired.

- What's the difference between Dithering and Half-toning?
  - Half-toning uses CMYK colors, varying dot sizes, and overlapping techniques to achieve the color desired.



Compression

Involves encoding information into fewer bits than would otherwise be occupied by the original source.

Lossless Compression Methods

Definition: A data encoding method that removes redundancies and uses fewer bits to represent the same data. (Best uses: Text or Archiving.)

Huffman Encoding

- An entropy encoder that substitutes more common characters with fewer bits, while infrequent characters are encoded with more bits.

Huffman Encoding

- An entropy encoder that substitutes more common characters with fewer bits, while infrequent characters are encoded with more bits.
- Average Compression: 2.3 to 2.9 bits per character (8 bits)
- Cons: Requires two passes to build variable length codes.
  - You can use pre-built trees to encode rather than building one.

Letter	Frequency	Letter	Frequency
e	0.12702	w	0.02360
t	0.09056	f	0.02228
a	0.08167	g	0.02015
o	0.07507	y	0.01974
i	0.06966	p	0.01929
n	0.06749	b	0.01492
s	0.06327	v	0.00978
h	0.06094	k	0.00772
r	0.05987	j	0.00153
d	0.04253	x	0.00150
l	0.04025	q	0.00095
c	0.02782	z	0.00074
u	0.02758		
m	0.02406		

Figure 4 : Frequency for common letters in the English language.

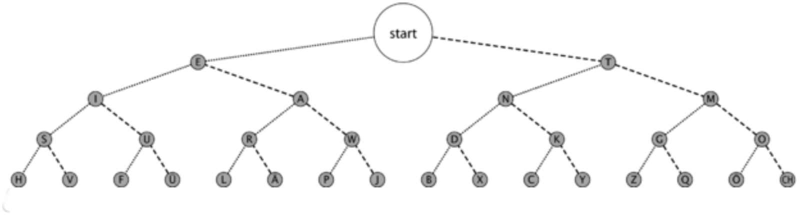


Figure 5 : A Huffman binary tree. This is the tree also used for Morse code.

Lempel-Ziv-Welch

- As the algorithm encounters patterns that it has seen before, it substitutes these with a shorter representation provided that it is already in the dictionary. If not, it creates one on the spot and outputs the corresponding code.

Lempel-Ziv-Welch

- As the algorithm encounters patterns that it has seen before, it substitutes these with a shorter representation provided that it is already in the dictionary. If not, it creates one on the spot and outputs the corresponding code.
- Average Compression: 2 bits per character (8 bits)

Lempel-Ziv-Welch

- As the algorithm encounters patterns that it has seen before, it substitutes these with a shorter representation provided that it is already in the dictionary. If not, it creates one on the spot and outputs the corresponding code.
- Average Compression: 2 bits per character (8 bits)
- Different files construct different dictionaries which do not need to be stored.

Lempel-Ziv-Welch

- As the algorithm encounters patterns that it has seen before, it substitutes these with a shorter representation provided that it is already in the dictionary. If not, it creates one on the spot and outputs the corresponding code.
- Average Compression: 2 bits per character (8 bits)
- Different files construct different dictionaries which do not need to be stored.
- Can create substitutions for entire strings rather than single characters.

Lempel-Ziv-Welch

- As the algorithm encounters patterns that it has seen before, it substitutes these with a shorter representation provided that it is already in the dictionary. If not, it creates one on the spot and outputs the corresponding code.
- Average Compression: 2 bits per character (8 bits)
- Different files construct different dictionaries which do not need to be stored.
- Can create substitutions for entire strings rather than single characters.
- In longer files, a better compression ratio can often be achieved since a large dictionary has a better chance of finding matches.

Lempel-Ziv-Welch

- As the algorithm encounters patterns that it has seen before, it substitutes these with a shorter representation provided that it is already in the dictionary. If not, it creates one on the spot and outputs the corresponding code.
- Average Compression: 2 bits per character (8 bits)
- Different files construct different dictionaries which do not need to be stored.
- Can create substitutions for entire strings rather than single characters.
- In longer files, a better compression ratio can often be achieved since a large dictionary has a better chance of finding matches.
- Tradeoff: A large dictionary can find more matches at the cost of processing time.

```
w = NIL;
   while ( read a character k )
   {
      if wk exists in the dictionary
         w = wk;
      else
         add wk to the dictionary;
      output the code for w;
      w = k;
   }
```

Figure 6 : Pseudo-code for LZW (compression).

```
read a character k;
   output k;
   w = k;
   while ( read a character k )
   /* k could be a character or a code. */
   {
      if k exists in the dictionary
         entry = dictionary entry for k;
         output entry;
         add w + entry[0] to dictionary;
         w = entry;
      else
         output entry = w + firstCharacterOf(w);
      add entry to dictionary;
      w = entry;
   }
```

Figure 7 : Pseudo-code for LZW (decompression).

Lossy Compression Methods

Definition: A data encoding method that discards some data in order to achieve compression. The resulting data is similar enough to the original data. (Best uses: Images, Audio, Video.)

Lossy: Wavelet Transform

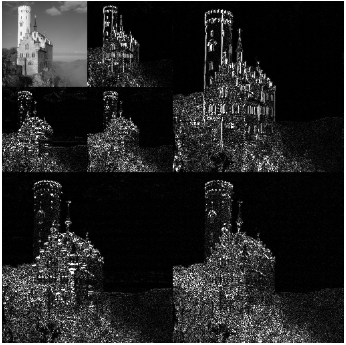
- Becomes lossy when coefficients that don't meet the threshold are reduced to zero.
- Is first performed over rows, then columns.
- Most of the resulting data is discarded, with high level data in the upper left, and smaller details to the lower right.

Lossy: Wavelet Transform

```
7   1   6   6   3   -5   4   2

Averages:
(7 +  1) / 2 =  4
(6 +  6) / 2 =  6
(3 + -5) / 2 = -1
(4 +  2) / 2 =  3

Differences:
(7 -  4) = ( 4 -  1) = 3
(6 -  6) = ( 6 -  6) = 0
(3 - -1) = (-1 - -5) = 4
(4 -  3) = ( 3 -  2) = 1

Resulting array:
4   6   -1   3   3   0   4   1
```

Lossy: Wavelet Transform

Lossy: Wavelet Transform



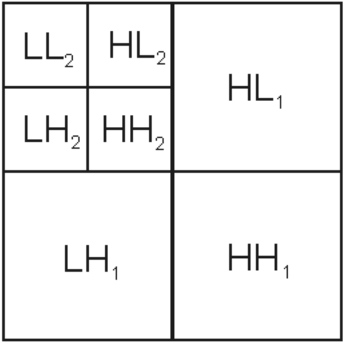


Figure 8 : Image processed with a wavelet transform. Each block contains coefficients to reconstruct the original image.

Lossy: Motion Compensation

Compensates for differences between subsequent frames.

- MPEG-2
  - I-Frame: Initial Frame.

Lossy: Motion Compensation

Compensates for differences between subsequent frames.

- MPEG-2
  - I-Frame: Initial Frame.
  - P-Frame: Predicted Frame or Delta Frame. Holds changes that occur between frames. Holds less data than a full frame.

Lossy: Motion Compensation

Compensates for differences between subsequent frames.

- MPEG-2
    - I-Frame: Initial Frame.
    - P-Frame: Predicted Frame or Delta Frame. Holds changes that occur between frames. Holds less data than a full frame.
    - B-Frame: Bi-directional frame. Contains forward and backward prediction of the closest I-Frame and P-frame.

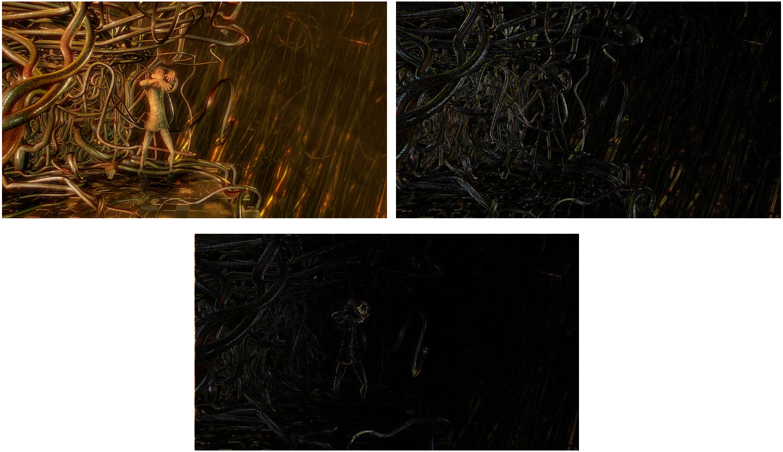


Figure 9 : The original frame, the difference frame, and the motion compensated frame.

Textures

Images that are intended to be mapped to a surface. They are highly compressed using fixed-rate compression algorithms, and are also quick to decompress, either as a whole or just a section.

- Can contain:

Textures

Images that are intended to be mapped to a surface. They are highly compressed using fixed-rate compression algorithms, and are also quick to decompress, either as a whole or just a section.

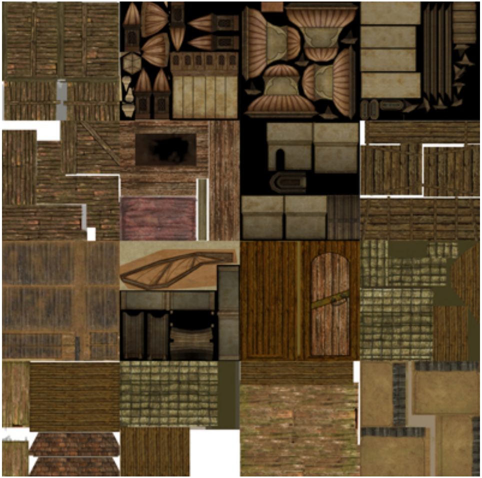
- Can contain:
  - Terrain, usually grass, concrete, etc. (Typically flat)
  - Object surfaces, such as boxes, furniture, etc. (Can be irregular)
  - Lighting information, dictating how light interacts with an object.

Texture Atlas

Textures atlases contain a group of images...

Texture Atlas

Textures atlases contain a group of images...

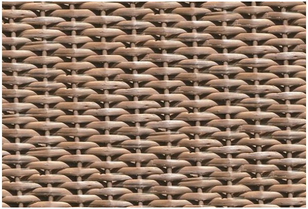
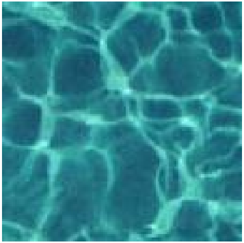


Texture Synthesis

While synthesised textures are a single image that has been stitched to itself repeatedly to a larger resolution.

Texture Synthesis

While synthesised textures are a single image that has been stitched to itself repeatedly to a larger resolution.


Texture Compression

- Has Constraints:

Texture Compression

- Has Constraints:
  - Compressed ratio must be 1 to 4 or higher.

Texture Compression

- Has Constraints:
  - Compressed ratio must be 1 to 4 or higher.
  - Decompression time must be lower than compression time.

Texture Compression

- Has Constraints:
  - Compressed ratio must be 1 to 4 or higher.
  - Decompression time must be lower than compression time.
  - Allow Random Access to individual blocks (fixed-rate compression).

Texture Compression

- Has Constraints:
  - Compressed ratio must be 1 to 4 or higher.
  - Decompression time must be lower than compression time.
  - Allow Random Access to individual blocks (fixed-rate compression).
  - Resulting decompressed images must look relatively similar.

Current Standard: Direct X

- The current standard under Fixed-Rate Block Compression patent (S3TC).

Current Standard: Direct X

- The current standard under Fixed-Rate Block Compression patent (S3TC).
    - Breaks images down to 4x4 blocks.

Current Standard: Direct X

- The current standard under Fixed-Rate Block Compression patent (S3TC).
  - Breaks images down to 4x4 blocks.
  - Finds a best fit line for a set of pixels.

Current Standard: Direct X

- The current standard under Fixed-Rate Block Compression patent (S3TC).
    - Breaks images down to 4x4 blocks.
    - Finds a best fit line for a set of pixels.
    - Stores indexes of pixels along the line.

Current Standard: Direct X

- The current standard under Fixed-Rate Block Compression patent (S3TC).
  - Breaks images down to 4x4 blocks.
  - Finds a best fit line for a set of pixels.
  - Stores indexes of pixels along the line.
- Pros: Fast to decode and results in good compression.
- Con: If a value doesn't have a good fit to the line, it will not be represented accurately.

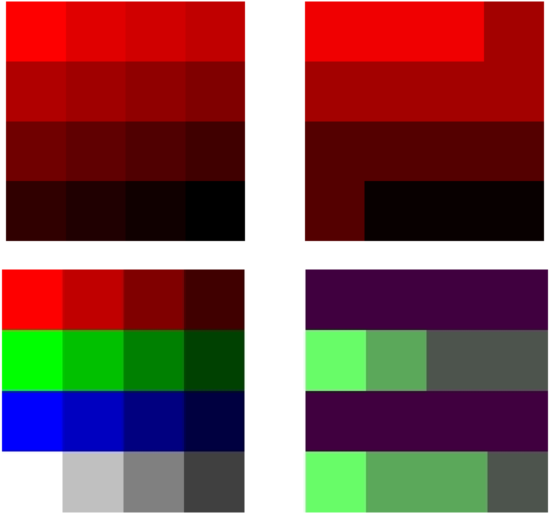


Figure 10 : The errors for gradient textures is shown in the top set of images. The right shows the result of a reduced color palette, which cannot interpolate the original colors at all.

Current Standard: Ericsson

Current standard used for Mobile Phones (notably Android)

- Starts with a 4x4 block...
  - Which is then broken down into a 4x2 or 2x4 block.
  - Each part is given a singel base color (could be 4/4/4 or 5/5/5 RGB).
  - Remaining bits are used to indicate the table used luminance values.

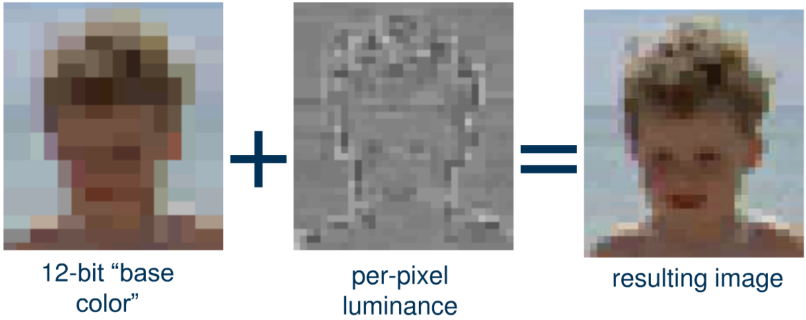


Figure 11 : The base colors are shown for each block on the left image, while luminance modulation is shown in the middle. The final image is the decompressed image.

Project Method

- Relevance Dither
- Relevance Compression

Relevance Dither

```
//Floyd-Steinberg
//          X   7
//      3   5   1

//Jarvis-Judice-Ninke
//          X   7   5
// 3    5   7   5   3
// 1    3   5   3   1

//Mine!
//          X   4  1
//          4   2
//          1
```

Relevance Compression

- Start with a 2x2 block.
- Convert pixel color to indexed color. (Keep count of this!)
- Find the most common indexes.
- Pixels that are not part of the most common indexes are changed to color that is part of this group.
- The best match is found using RGB as a distance coordinate, to find the next closest color.
- Save the most common indexes in the header.

Results

Results

Pros:

- Uses fewer colors.
- Smaller result than using Direct X.
- Compresses faster than Direct X.

Results

Pros:

- Uses fewer colors.
- Smaller result than using Direct X.
- Compresses faster than Direct X.

Cons:

- Does not outperform Direct X and Ericsson in decompression.
- The index sometimes does not include all the colors necessary to reconstruct the image at decompression time.

Results

Pros:

- Uses fewer colors.
- Smaller result than using Direct X.
- Compresses faster than Direct X.

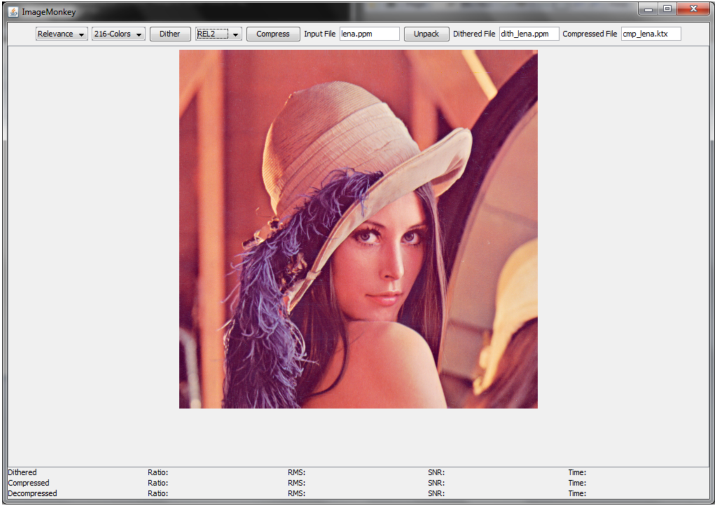
Cons:

- Does not outperform Direct X and Ericsson in decompression.
- The index sometimes does not include all the colors necessary to reconstruct the image at decompression time.

Suggestions:

- Port to a language that doesn't use a virtual machine.
- Index tiles instead of single pixels.

Demo


ImageMonkey
Relevance
216-Colors
Dither
REL2
Compress
Input File lena.ppm
Unpack
Dithered File dith_lena.ppm
Compressed File cmp_lena.ktx
Dithered Ratio: RMS: SNR: Time:
Compressed Ratio: RMS: SNR: Time:
Decompressed Ratio: RMS: SNR: Time:

References I

- Chui, C. K. 1992. *An Introduction to Wavelets*. San Diego, CA: Academic Press Professional, Inc.
- Efros, A. A., and Freeman, W. T. 2001. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, New York, NY: ACM, pages 341–346.
- Gonzalez, R. C., and Woods, R. E. 2006. *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ: Prentice-Hall, Inc.
- Iourcha, K. I., Nayak, K. S., and Hong, Z. System and method for fixed-rate block-based image compression with inferred pixel values. US Patent 5,956,431, filed Oct. 2, 1997, and issued Sep. 21, 1999.

References II

- Knuth, D. E. Digital Halftones by Dot Diffusion. *ACM Transactions on Graphics (TOG)* 6(4)(1987):245–273.
- MacAdam, D. L. Visual Sensitivities to Color Differences in Daylight. *The Journal of the Optical Society of America* 32(5)(1942):247–273, doi:10.1364/JOSA.32.000247.
- Shannon, C. E. A Mathematical Theory of Communication. *ACM SIGMOBILE Mobile Computing and Communications Review* 5(1)(2001):3–55.
- Sherrod, A. 2008. *Game Graphics Programming*. Boston, MA: Course Technology.

References III

- Ström, J., and Akenine-Möller, T. iPACKMAN: High-Quality, Low-Complexity Texture Compression for Mobile Phones. *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware* (2005):63–70.
- Tudor, P. MPEG-2 Video Compression. *Electronics & Communication Engineering Journal* 7(6)(1995):257–264.
- Ziv, J., and Lempel, A. Compression of Individual Sequences via Variable-Rate Coding. *IEEE Transactions on Information Theory* 24(5)(1978):530–536.