THE SOURC$_{\text{E}}$

ISSUE 8
Six, Sixty-six

Cory Arcangel

2015

Arcangel Surfware

www.arcangelsurfware.biz

Six, Sixty—six[1]

Cory Arcangel, 2004

http://www.coryarcangel.com

Requirements: Osx, perl, the "lame" mp3 encoder/decoder, the "normailze" binary
Cory Arcangel (special thx to the Radical Software Group)

Script to compress[2] an mp3 recursively (666) times.

"....." — Jazz

```
for ($count=1; $count<=666; $count++)
{
 if ($count == 1)
 {

  `lame --decode Iron_Maiden_The_Number_of_the_Beast_original.mp3
      Iron_Maiden_The_Number_of_the_Beast.aif`;

 }

 `lame -h Iron_Maiden_The_Number_of_the_Beast.aif
    Iron_Maiden_The_Number_of_the_Beast_$count.mp3`;

 `lame --decode Iron_Maiden_The_Number_of_the_Beast_$count.mp3
    Iron_Maiden_The_Number_of_the_Beast.aif`;

 `normalize Iron_Maiden_The_Number_of_the_Beast.aif`;
}
```

[1] Actually, the official title of this work is, "Iron Maiden's The Number of the Beast compressed over and over as an MP3 666 times", but, I don't know, I thought I would re-name it RN for this zine.

2

ON COMPRESSION

Cory Arcangel
2k7 – 2k8

ABSTRACT: JPEGs look the way they do because of quantization and their use of the Discrete Cosine Transform (DCT). The DCT is a technique for converting a signal into elementary frequency components. It is widely used in image compression. Here we will go through some examples to explain how the DCT works. The text is kind of a summary, and if you want to bring the noise, all the math is in the end notes.

1. LOSSY VS. LOSSLESS

The whole point of digital image compression is to be able to reconstruct an image without having to send all the data. This is because data, especially in large amounts, is expensive and slow to transport. Either over cable lines, phone lines, or wirelessly, it is all slow. To this day, the most efficient and cheapest way to transport large amounts of data is by mailing a hard-drive to the destination, and I don't mean emailing, I mean the kind of mailing that involves the post office. So compression is valuable because the less we need to send the cheaper and faster it is. There are two kinds of compression. One is called Lossy, and the other is called Lossless. Lossless compression does not lose any information from the original source. How can this be? Well, let's say we wanted to send this: 'a a a a a a a a a b a' and we were going to send it over the phone by voice. As opposed to having to send all the information by reading out each letter one at a time, we could just tell someone '9a's, one b, and one a' and they would know we meant 'a a a a a a a a a b a' and we have saved ourselves a bit of breath. In computer language it means we have stored all the information using less space. To generalize a bit, if you have ever opened a 'zip' file, your computer has seen '9a's, one b, and one a' and translated it to 'a a a a a a a a a a b a'. This is Lossless compression. On the other hand, Lossy compression actually loses data. Lossy compression, therefore, can not be used for text, or any application where all the information must remain intact. It is used for images, music, and video. This is because, believe it or not, our eyes and ears are pretty crap, and we don't usually notice missing bits here and there. Lossy compression works by getting rid of the information which isn't so important to us. To generalize a bit again, if we tried to send 'a a a a a a a a a b a' using Lossy compression over the phone, we would just get lazy and say '11a's'. In this article, we are going to focus on the Discrete Cosine Transform, aka the DCT, a math formula used in Lossy compression. The reason I'm interested in this Transform is because, when used with quantization, it is what gives JPEGs that 'JPEG look'. By 'JPEG look' I mean those crappy compressed blocky images you need to squint your eyes to understand that are all over the internet. And in case you haven't noticed, this look is everywhere else as well (ads, digital cameras, digital video, etc.) If the '80s gave us 'hot' colors and 'rad' graphics, and the '90s gave us slick vector design, then

the 00's are giving us compressed blocky images.

JPEGs are everywhere today because they have become a standard, or a universally agreed upon set of rules. Today JPEG is a nickname for a file type, but JPEG originated as a shorthand for the group that proposed the standard, the Joint Photographic Experts Group. This standard was created in Geneva in 1992 when members of the CCITT and the ISO/IEC (now together known as JPEG) got together in Geneva and released the technical document ISO/IEC IS 10918-1 / ITU-T Recommendation T.81. This paper recommended 'REQUIREMENTS AND GUIDELINES' of the 'DIGITAL COMPRESSION AND CODING OF CONTINUOUS-TONE STILL IMAGES'. These guidelines, through third party development, eventually became known as JPEG files.

2. THE GUY BEHIND THE GUY BEHIND THE GUY

As mentioned earlier, the heart of JPEG is the DCT formula, and the DCT relies on cosines. The easiest way to think of cosines is to imagine yourself walking counterclockwise around a circle. This circle is centered on the X and Y axis, and has a radius of 1. Radius is the length from the center of the circle to the edge. A cosine of the angle in respect to the positive horizontal axis (aka, the length in our case because we are on a unit circle (radius = 1)) is our position on the x axis as we walk around the circle if we started at X = 1. We must also remember that the length around a circle with a radius of 1 is 2pi. So, cos(0) is 1, because we haven't gone anywhere; we are still standing at the beginning, at X = 1, cos(pi) is -1, because we have travelled halfway around the circle to X = -1, and cos(2pi) is 1, because since we have travelled all the way around our circle, we have ended up back at the beginning (Figure 1). It is this cyclical pattern which is useful in compression (Figure 2). To see the DCT in action we will start with the 1D DCT (Figure 3) formula and use it to compress the input 3,2,1 (Figure 4). DCT-based compression has four steps. First the DCT formula creates basis functions, then it compares the input data to those basis functions, creating what are called DCT coefficients, then those coefficients are quantized, and the last step is decompression, where all this is done in reverse to recreate our data. The first step of our process can be seen in Figure 5.[1] These are the basis functions for our input, which

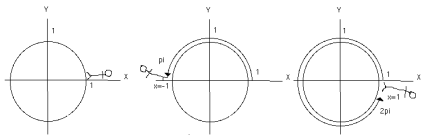


Figure 1: cosine of 0, pi, and 2pi

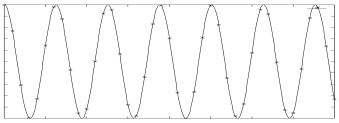


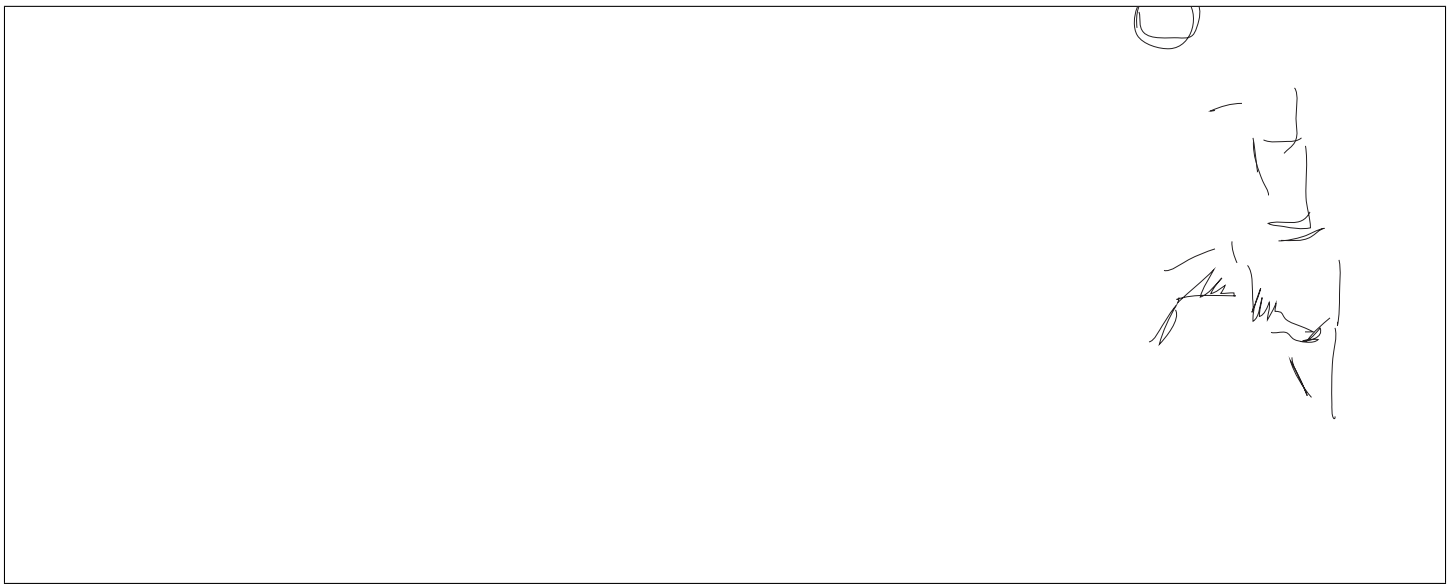
Figure 2: Cosine curve of x = 0 to 40

consist of cosine curves of increasing frequency. They can be thought of as building blocks. Every combination of 3 digits can be recreated by adding these blocks together in different proportions. The second step in the process is comparing our input to our three basis functions to generate three DCT coefficients. Our DCT coefficients represent how much of each basis function is present in our input. In our example, our three DCT coefficients generated by the 1D DTC are 3.46410, 1.41421,

$$F(u) = \sum_{x=0}^{N-1} w(i) f[x] cos \frac{(2x+1)u\pi}{2N}$$

if i = 0, $w = \sqrt{1/N}$, and if i $\neq$ 0, w = $\sqrt{2/N}$

Figure 3: 1D DCT formula. w(i) is a weighting factor fyi



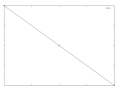
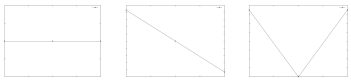


Figure 4: Our 1D DCT example input



(a) Basis function 1 (b) Basis function 2 (c) Basis function 3

Figure 5: Basis Functions for a DCT of length 3

and 0.[2] From these values we can see that our input contains elements of our first and second basis function, but none of the third. This should make sense since our input numbers are a straight line, so they do not contain any data which is similar to the curve in the third basis function. If our DCT formula in this example takes in 3 digits as input and we end up with 3 digits as output, how does this help us save space? The third step, quantization, is the key to this question. Quantization is basically a way to discard DCT coefficients. In this case we would discard our third DCT coefficient because it doesn't help us describe our input. So when we get to the last step in the process which is reversing all of this in order to reconstruct our original input, we will use only 2 DCT coefficients to do this.[3] The same information now takes only two-thirds of the space!

3. 2D

To work on an image as opposed to a string of input, we need to use the 2D DCT formula (Figure 6). This is basically the same as the 1D formula, except it works on a matrix. The input we'll compress in this example is in Figure 7. So again our first step is generating the

$$F(u,v) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1} w(i,j) f[x,y] cos\frac{(2x+1)u\pi}{2N} cos\frac{(2y+1)u\pi}{2N}$$

if i or j = 0, $w = \sqrt{1/N}$, and if i or j $\neq$ 0, w = $\sqrt{2/N}$

Figure 6: 2D DCT formula

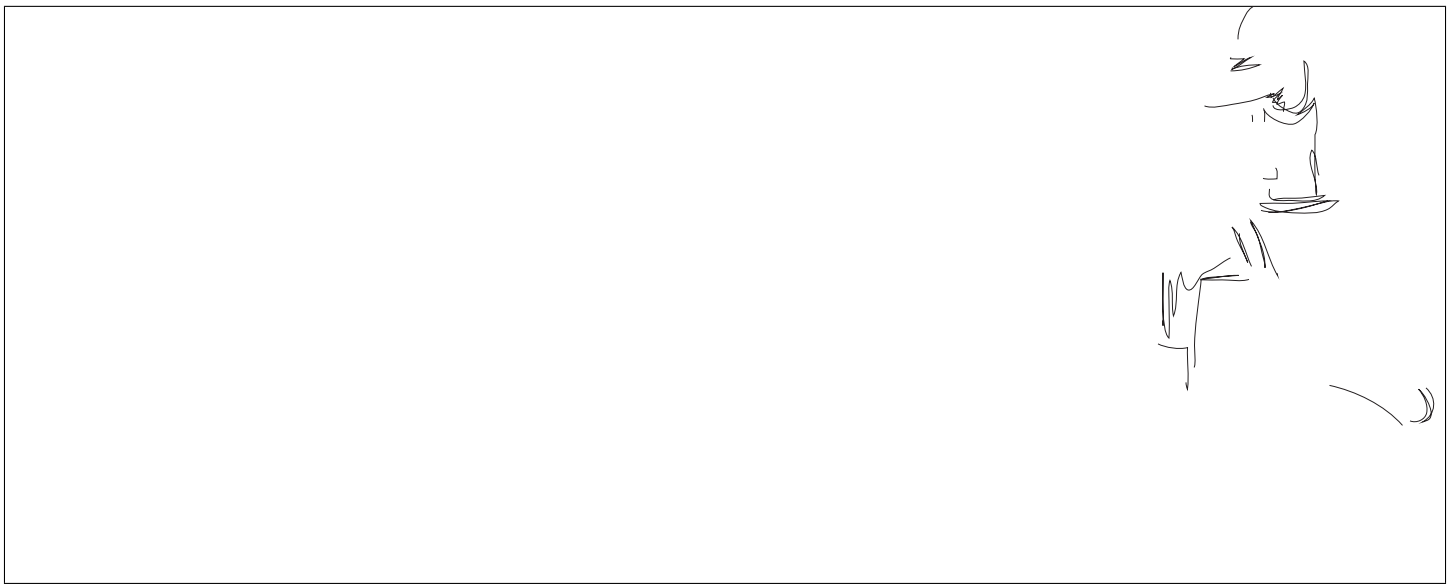
(a) Input Image

$$\begin{bmatrix} 255 & 255 & 255 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

(b) Input matrix

Figure 7: Our input matrix

basis functions (Figure 8).[4] As in Figure 8, our basis functions with the lower cosine frequencies are on the top left and the basis functions with the higher cosine frequencies are on the bottom right. Next, we compare our input image to our basis functions to generate our DCT coefficients. Then, we're left with 9 DCT coefficients (Figure 9).[5] These numbers tell us our input only contains three of our nine basis functions, and one can see the graphic similarities between the basis functions on the left side of Figure 8, and our input in Figure 7. All the other basis functions do not relate. The third step is quantization. This happens by taking the DCT coefficient matrix (Figure 9) and dividing it by a quantization matrix, then rounding to the nearest integer. An example matrix is used in Figure 10. The result, when reversed (Figure 11), gets rid of one of our DCT coefficients. If we complete step four by using the quantized coefficients to reconstruct our input, we clearly have quite a big difference (Figure 12).[6] Where did that grey bar come from? EXACTLY!! We have saved a ton of space, because now we only need to transmit '250, 250, and 7 0s' to recreate our input, but our image no longer looks how it was supposed to! This is because we have discarded the high frequency basis functions, so we can no longer create sharp contrasts. But it's similar, we get the idea, and this is probably good enough.



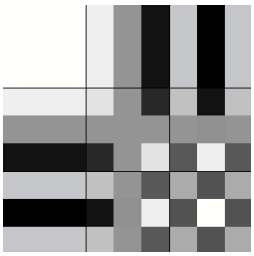


Figure 8: The 9 DCT basis functions for a 3 by 3 matrix

$$\begin{bmatrix} 255.00000 & 0 & 0 \\ 312.30994 & 0 & 0 \\ 180.31223 & 0 & 0 \end{bmatrix}$$

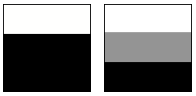
Figure 9: Our 9 DCT coefficients

$$\begin{bmatrix} 255.00000 & 0 & 0 \\ 312.30994 & 0 & 0 \\ 180.31223 & 0 & 0 \end{bmatrix} / \begin{bmatrix} 10 & 50 & 400 \\ 50 & 50 & 400 \\ 400 & 400 & 400 \end{bmatrix} = \begin{bmatrix} 25 & 0 & 0 \\ 6 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Figure 10: Quantization table

$$\begin{bmatrix} 25 & 0 & 0 \\ 6 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} * \begin{bmatrix} 10 & 50 & 400 \\ 50 & 50 & 400 \\ 400 & 400 & 400 \end{bmatrix} = \begin{bmatrix} 250 & 0 & 0 \\ 250 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Figure 11: Reverse Quantization



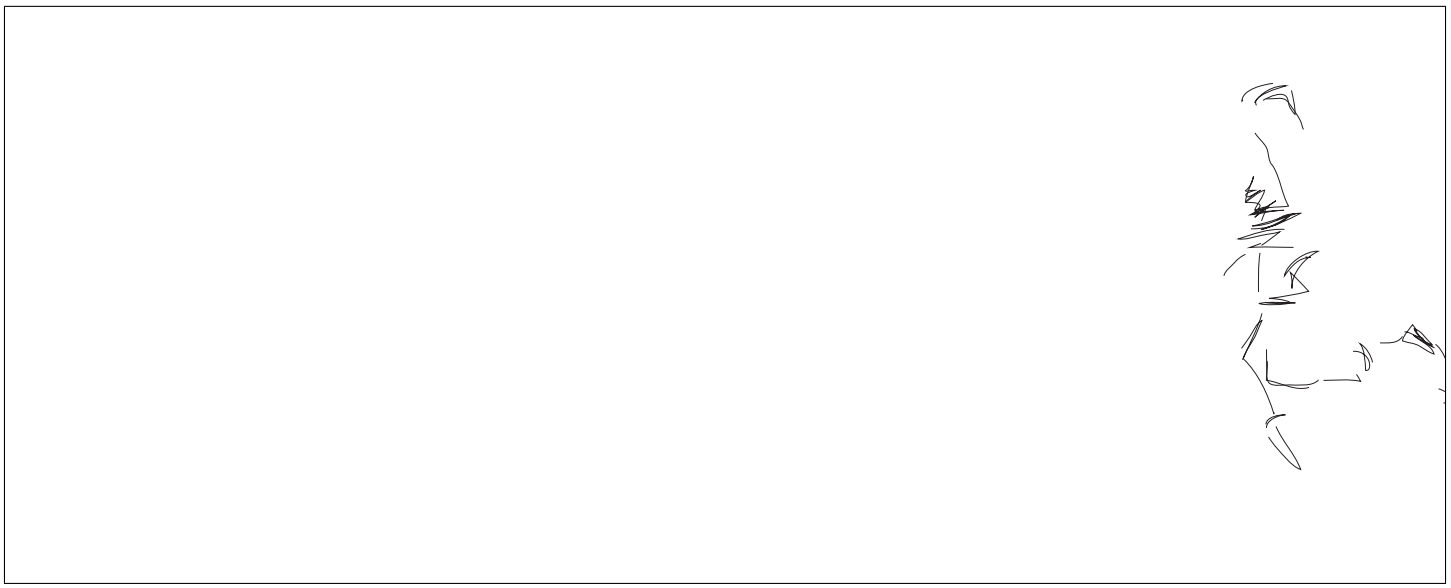
(a) Our input

(b) Our input reconstructed by using the quantization table in Figure 10

Figure 12: Reconstructing our input for our 2D 3 x 3 matrix

4. THE JOINT PHOTOGRAPHIC EXPERTS

The only difference between what we just did and a JPEG, is that a JPEG always splits the image into 8 x 8 blocks and then these 8 x 8 blocks are run through the 2D DCT. 8 x 8 blocks are used because they are small enough to have consistent spatial qualities. Even at high rates of compression, we can still make out the original image. The basis functions for a JPEG are shown in Figure 13. Also, JPEGs don't specify what quantization matrix is used. Photoshop's quantization matrix is different from Canon, etc. etc., so actually one has very little control of the discarded information. Awesome. In Figure 14 and Figure 15 we can see a sample JPEG compressed with a sample quantization matrix. Take a close look — we are recreating the image only using the top left basis function of Figure 13. Hopefully you can see now that heavily compressed JPEGs are really a bunch of 8 by 8 squares composed of only the first few low frequency basis functions of an 8 x 8 2D DCT (Figure 13). We get a 90 percent reduction in file size because we only need to send a few DCT coefficients down the line, but we get an image which is only a shadow of its former self. Welcome to the future.



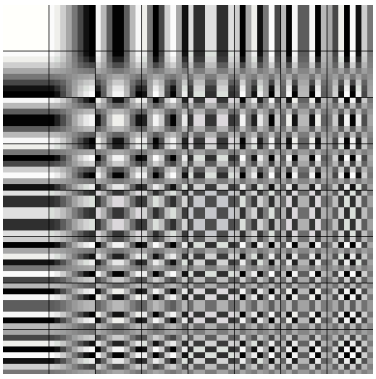


Figure 13: Our Basis functions for a JPEG

Figure 14: Our JPEG input

$$\begin{bmatrix} 51 & 101 & 151 & 201 & 251 & 301 & 351 & 401 \\ 101 & 151 & 201 & 251 & 301 & 351 & 401 & 451 \\ 151 & 201 & 251 & 301 & 351 & 401 & 451 & 501 \\ 201 & 251 & 301 & 351 & 401 & 451 & 501 & 551 \\ 251 & 301 & 351 & 401 & 451 & 501 & 551 & 601 \\ 301 & 351 & 401 & 451 & 501 & 551 & 601 & 651 \\ 351 & 401 & 451 & 501 & 551 & 601 & 651 & 701 \\ 401 & 451 & 501 & 551 & 601 & 651 & 701 & 751 \end{bmatrix}$$

Figure 15: Our JPEG quantization table

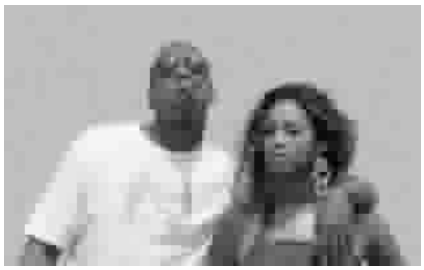
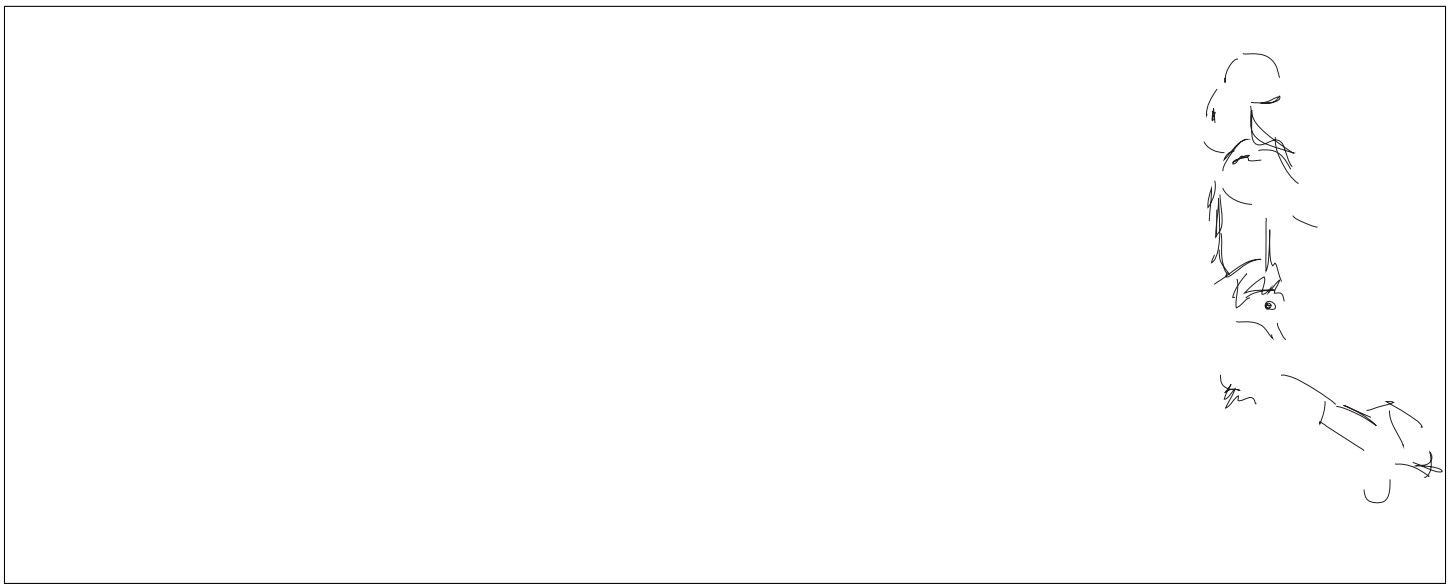


Figure 16: Our compressed JPEG using the above input and quantization table (Figure 14 and 15)



NOTES

[1] $\begin{bmatrix} cos\frac{(2*0+1)0\pi}{2*3} & cos\frac{(2*1+1)0\pi}{2*3} & cos\frac{(2*2+1)0\pi}{2*3} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$
$\begin{bmatrix} cos\frac{(2*0+1)1\pi}{2*3} & cos\frac{(2*1+1)1\pi}{2*3} & cos\frac{(2*2+1)1\pi}{2*3} \\ cos\frac{(2*0+1)2\pi}{2*3} & cos\frac{(2*1+1)2\pi}{2*3} & cos\frac{(2*2+1)2\pi}{2*3} \end{bmatrix} = \begin{bmatrix} .86603 & 6.1232e-17 & -.86603 \\ .5 & -1 & .5 \end{bmatrix}$

[2] $3.46410 = \sum_{x=0}^{2} \sqrt{1/3} * \begin{bmatrix} 3 & 2 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$
$1.41421 = \sum_{x=0}^{2} \sqrt{2/3} * \begin{bmatrix} 3 & 2 & 1 \end{bmatrix} * \begin{bmatrix} .86603 & 6.1232e-17 & -.86603 \end{bmatrix}$
$0.00000 = \sum_{x=0}^{2} \sqrt{2/3} * \begin{bmatrix} 3 & 2 & 1 \end{bmatrix} * \begin{bmatrix} .5 & -1 & .5 \end{bmatrix}$
FYI: $3.46410 = (\sqrt{1/3} * 3 * 1) + (\sqrt{1/3} * 2 * 1) + (\sqrt{1/3} * 1 * 1)$

[3] $\sqrt{1/3} * \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} * 3.46410 + \sqrt{2/3} * \begin{bmatrix} .86603 & 6.1232e-17 & -.86603 \end{bmatrix} * 1.41421 = \begin{bmatrix} 3 & 2 & 1 \end{bmatrix}$

[4] $\begin{bmatrix} cos\frac{(2*0+1)0\pi}{2*3} * cos\frac{(2*0+1)0\pi}{2*3} & cos\frac{(2*0+1)0\pi}{2*3} * cos\frac{(2*1+1)0\pi}{2*3} & cos\frac{(2*0+1)0\pi}{2*3} * cos\frac{(2*2+1)0\pi}{2*3} \\ cos\frac{(2*1+1)0\pi}{2*3} * cos\frac{(2*0+1)0\pi}{2*3} & cos\frac{(2*1+1)0\pi}{2*3} * cos\frac{(2*1+1)0\pi}{2*3} & cos\frac{(2*1+1)0\pi}{2*3} * cos\frac{(2*2+1)0\pi}{2*3} \\ cos\frac{(2*2+1)0\pi}{2*3} * cos\frac{(2*0+1)0\pi}{2*3} & cos\frac{(2*2+1)0\pi}{2*3} * cos\frac{(2*1+1)0\pi}{2*3} & cos\frac{(2*2+1)0\pi}{2*3} * cos\frac{(2*2+1)0\pi}{2*3} \end{bmatrix}$
$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
$\begin{bmatrix} cos\frac{(2*0+1)0\pi}{2*3} * cos\frac{(2*0+1)1\pi}{2*3} & cos\frac{(2*0+1)0\pi}{2*3} * cos\frac{(2*1+1)1\pi}{2*3} & cos\frac{(2*0+1)0\pi}{2*3} * cos\frac{(2*2+1)1\pi}{2*3} \\ cos\frac{(2*1+1)0\pi}{2*3} * cos\frac{(2*0+1)1\pi}{2*3} & cos\frac{(2*1+1)0\pi}{2*3} * cos\frac{(2*1+1)1\pi}{2*3} & cos\frac{(2*1+1)0\pi}{2*3} * cos\frac{(2*2+1)1\pi}{2*3} \\ cos\frac{(2*2+1)0\pi}{2*3} * cos\frac{(2*0+1)1\pi}{2*3} & cos\frac{(2*2+1)0\pi}{2*3} * cos\frac{(2*1+1)1\pi}{2*3} & cos\frac{(2*2+1)0\pi}{2*3} * cos\frac{(2*2+1)1\pi}{2*3} \end{bmatrix} =$
$\begin{bmatrix} .86603 & 6.1232e-17 & -.86603 \\ .86603 & 6.1232e-17 & -.86603 \\ .86603 & 6.1232e-17 & -.86603 \end{bmatrix}$
$\begin{bmatrix} cos\frac{(2*0+1)0\pi}{2*3} * cos\frac{(2*0+1)2\pi}{2*3} & cos\frac{(2*0+1)0\pi}{2*3} * cos\frac{(2*1+1)2\pi}{2*3} & cos\frac{(2*0+1)0\pi}{2*3} * cos\frac{(2*2+1)2\pi}{2*3} \\ cos\frac{(2*1+1)0\pi}{2*3} * cos\frac{(2*0+1)2\pi}{2*3} & cos\frac{(2*1+1)0\pi}{2*3} * cos\frac{(2*1+1)2\pi}{2*3} & cos\frac{(2*1+1)0\pi}{2*3} * cos\frac{(2*2+1)2\pi}{2*3} \\ cos\frac{(2*2+1)0\pi}{2*3} * cos\frac{(2*0+1)2\pi}{2*3} & cos\frac{(2*2+1)0\pi}{2*3} * cos\frac{(2*1+1)2\pi}{2*3} & cos\frac{(2*2+1)0\pi}{2*3} * cos\frac{(2*2+1)2\pi}{2*3} \end{bmatrix} =$
$\begin{bmatrix} .5 & -1 & .5 \\ .5 & -1 & .5 \\ .5 & -1 & .5 \end{bmatrix}$
$\begin{bmatrix} cos\frac{(2*0+1)1\pi}{2*3} * cos\frac{(2*0+1)0\pi}{2*3} & cos\frac{(2*0+1)1\pi}{2*3} * cos\frac{(2*1+1)0\pi}{2*3} & cos\frac{(2*0+1)1\pi}{2*3} * cos\frac{(2*2+1)0\pi}{2*3} \\ cos\frac{(2*1+1)1\pi}{2*3} * cos\frac{(2*0+1)0\pi}{2*3} & cos\frac{(2*1+1)1\pi}{2*3} * cos\frac{(2*1+1)0\pi}{2*3} & cos\frac{(2*1+1)1\pi}{2*3} * cos\frac{(2*2+1)0\pi}{2*3} \\ cos\frac{(2*2+1)1\pi}{2*3} * cos\frac{(2*0+1)0\pi}{2*3} & cos\frac{(2*2+1)1\pi}{2*3} * cos\frac{(2*1+1)0\pi}{2*3} & cos\frac{(2*2+1)1\pi}{2*3} * cos\frac{(2*2+1)0\pi}{2*3} \end{bmatrix} =$
$\begin{bmatrix} 8.6603e-01 & 8.6603e-01 & 8.6603e-01 \\ 6.1232e-17 & 6.1232e-17 & 6.1232e-17 \\ -8.6603e-01 & -8.6603e-01 & -8.6603e-01 \end{bmatrix}$
$\begin{bmatrix} cos\frac{(2*0+1)1\pi}{2*3} * cos\frac{(2*0+1)1\pi}{2*3} & cos\frac{(2*0+1)1\pi}{2*3} * cos\frac{(2*1+1)1\pi}{2*3} & cos\frac{(2*0+1)1\pi}{2*3} * cos\frac{(2*2+1)1\pi}{2*3} \\ cos\frac{(2*1+1)1\pi}{2*3} * cos\frac{(2*0+1)1\pi}{2*3} & cos\frac{(2*1+1)1\pi}{2*3} * cos\frac{(2*1+1)1\pi}{2*3} & cos\frac{(2*1+1)1\pi}{2*3} * cos\frac{(2*2+1)1\pi}{2*3} \\ cos\frac{(2*2+1)1\pi}{2*3} * cos\frac{(2*0+1)1\pi}{2*3} & cos\frac{(2*2+1)1\pi}{2*3} * cos\frac{(2*1+1)1\pi}{2*3} & cos\frac{(2*2+1)1\pi}{2*3} * cos\frac{(2*2+1)1\pi}{2*3} \end{bmatrix} =$

$$\begin{bmatrix} 7.5000e-01 & 5.3029e-17 & -7.5000e-01 \\ 5.3029e-17 & 3.7494e-33 & -5.3029e-17 \\ -7.5000e-01 & -5.3029e-17 & 7.5000e-01 \end{bmatrix}$$

$$\begin{bmatrix} cos\frac{(2*0+1)1\pi}{2*3}*cos\frac{(2*0+1)2\pi}{2*3} & cos\frac{(2*0+1)1\pi}{2*3}*cos\frac{(2*1+1)2\pi}{2*3} & cos\frac{(2*0+1)1\pi}{2*3}*cos\frac{(2*2+1)2\pi}{2*3} \\ cos\frac{(2*1+1)1\pi}{2*3}*cos\frac{(2*0+1)2\pi}{2*3} & cos\frac{(2*1+1)1\pi}{2*3}*cos\frac{(2*1+1)2\pi}{2*3} & cos\frac{(2*1+1)1\pi}{2*3}*cos\frac{(2*2+1)2\pi}{2*3} \\ cos\frac{(2*2+1)1\pi}{2*3}*cos\frac{(2*0+1)2\pi}{2*3} & cos\frac{(2*2+1)1\pi}{2*3}*cos\frac{(2*1+1)2\pi}{2*3} & cos\frac{(2*2+1)1\pi}{2*3}*cos\frac{(2*2+1)2\pi}{2*3} \end{bmatrix} =$$

$$\begin{bmatrix} 4.3301e-01 & -8.6603e-01 & 4.3301e-01 \\ 3.0616e-17 & -6.1232e-17 & 3.0616e-17 \\ -4.3301e-01 & 8.6603e-01 & -4.3301e-01 \end{bmatrix}$$

$$\begin{bmatrix} cos\frac{(2*0+1)2\pi}{2*3}*cos\frac{(2*0+1)0\pi}{2*3} & cos\frac{(2*0+1)2\pi}{2*3}*cos\frac{(2*1+1)0\pi}{2*3} & cos\frac{(2*0+1)2\pi}{2*3}*cos\frac{(2*2+1)0\pi}{2*3} \\ cos\frac{(2*1+1)2\pi}{2*3}*cos\frac{(2*0+1)0\pi}{2*3} & cos\frac{(2*1+1)2\pi}{2*3}*cos\frac{(2*1+1)0\pi}{2*3} & cos\frac{(2*1+1)2\pi}{2*3}*cos\frac{(2*2+1)0\pi}{2*3} \\ cos\frac{(2*2+1)2\pi}{2*3}*cos\frac{(2*0+1)0\pi}{2*3} & cos\frac{(2*2+1)2\pi}{2*3}*cos\frac{(2*1+1)0\pi}{2*3} & cos\frac{(2*2+1)2\pi}{2*3}*cos\frac{(2*2+1)0\pi}{2*3} \end{bmatrix} =$$

$$\begin{bmatrix} .5 & .5 & .5 \\ -1 & -1 & -1 \\ .5 & .5 & .5 \end{bmatrix}$$

$$\begin{bmatrix} cos\frac{(2*0+1)2\pi}{2*3}*cos\frac{(2*0+1)1\pi}{2*3} & cos\frac{(2*0+1)2\pi}{2*3}*cos\frac{(2*1+1)1\pi}{2*3} & cos\frac{(2*0+1)2\pi}{2*3}*cos\frac{(2*2+1)1\pi}{2*3} \\ cos\frac{(2*1+1)2\pi}{2*3}*cos\frac{(2*0+1)1\pi}{2*3} & cos\frac{(2*1+1)2\pi}{2*3}*cos\frac{(2*1+1)1\pi}{2*3} & cos\frac{(2*1+1)2\pi}{2*3}*cos\frac{(2*2+1)1\pi}{2*3} \\ cos\frac{(2*2+1)2\pi}{2*3}*cos\frac{(2*0+1)1\pi}{2*3} & cos\frac{(2*2+1)2\pi}{2*3}*cos\frac{(2*1+1)1\pi}{2*3} & cos\frac{(2*2+1)2\pi}{2*3}*cos\frac{(2*2+1)1\pi}{2*3} \end{bmatrix} =$$

$$\begin{bmatrix} 4.3301e-01 & 3.0616e-17 & -4.3301e-01 \\ -8.6603e-01 & -6.1232e-17 & 8.6603e-01 \\ 4.3301e-01 & 3.0616e-17 & -4.3301e-01 \end{bmatrix}$$

$$\begin{bmatrix} cos\frac{(2*0+1)2\pi}{2*3}*cos\frac{(2*0+1)2\pi}{2*3} & cos\frac{(2*0+1)2\pi}{2*3}*cos\frac{(2*1+1)2\pi}{2*3} & cos\frac{(2*0+1)2\pi}{2*3}*cos\frac{(2*2+1)2\pi}{2*3} \\ cos\frac{(2*1+1)2\pi}{2*3}*cos\frac{(2*0+1)2\pi}{2*3} & cos\frac{(2*1+1)2\pi}{2*3}*cos\frac{(2*1+1)2\pi}{2*3} & cos\frac{(2*1+1)2\pi}{2*3}*cos\frac{(2*2+1)2\pi}{2*3} \\ cos\frac{(2*2+1)2\pi}{2*3}*cos\frac{(2*0+1)2\pi}{2*3} & cos\frac{(2*2+1)2\pi}{2*3}*cos\frac{(2*1+1)2\pi}{2*3} & cos\frac{(2*2+1)2\pi}{2*3}*cos\frac{(2*2+1)2\pi}{2*3} \end{bmatrix} =$$

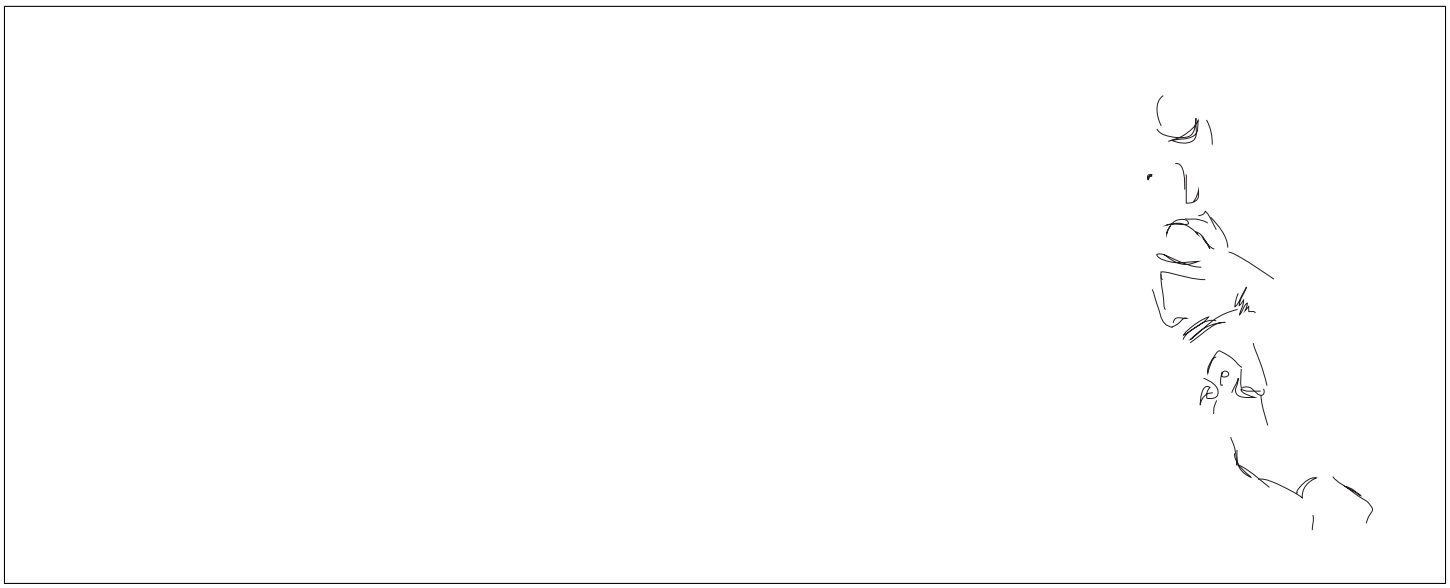
$$\begin{bmatrix} 0.25000 & .5 & 0.25 \\ -.5 & 1 & -.5 \\ 0.25000 & -.5 & 0.25 \end{bmatrix}$$

[5] $255.00000 = \sum_{x=0}^{2}\sum_{y=0}^{2} sqrt(1/3) * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} * sqrt(1/3) * \begin{bmatrix} 255 & 255 & 255 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

$0 = \sum_{z=0}^{2}\sum_{y=0}^{2} sqrt(1/3) * \begin{bmatrix} .86603 & 6.1232e-17 & .86603 \\ .86603 & 6.1232e-17 & .86603 \\ .86603 & 6.1232e-17 & .86603 \end{bmatrix} * sqrt(2/3) * \begin{bmatrix} 255 & 255 & 255 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

$0 = \sum_{z=0}^{2}\sum_{y=0}^{2} sqrt(1/3) * \begin{bmatrix} .5 & -1 & .5 \\ .5 & -1 & .5 \\ .5 & -1 & .5 \end{bmatrix} * sqrt(2/3) * \begin{bmatrix} 255 & 255 & 255 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

$312.30994 = \sum_{z=0}^{2}\sum_{y=0}^{2} sqrt(2/3) * \begin{bmatrix} 8.6603e-01 & 8.6603e-01 & 8.6603e-01 \\ 6.1232e-17 & 6.1232e-17 & 6.1232e-17 \\ -8.6603e-01 & -8.6603e-01 & -8.6603e-01 \end{bmatrix} * sqrt(1/3) * \begin{bmatrix} 255 & 255 & 255 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$



$$0 = \sum_{z=0}^{2} \sum_{y=0}^{2} sqrt(2/3) * \begin{bmatrix} 7.5000e-01 & 5.3029e-17 & -7.5000e-01 \\ 5.3029e-17 & 3.7494e-33 & -5.3029e-17 \\ -7.5000e-01 & -5.3029e-17 & 7.5000e-01 \end{bmatrix} * sqrt(2/3) * \begin{bmatrix} 255 & 255 & 255 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$0 = \sum_{z=0}^{2} \sum_{y=0}^{2} sqrt(2/3) * \begin{bmatrix} 4.3301e-01 & -8.6603e-01 & 4.3301e-01 \\ 3.0616e-17 & -6.1232e-17 & 3.0616e-17 \\ -4.3301e-01 & 8.6603e-01 & -4.3301e-01 \end{bmatrix} * sqrt(2/3) * \begin{bmatrix} 255 & 255 & 255 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$180.31223 = \sum_{z=0}^{2} \sum_{y=0}^{2} sqrt(2/3) * \begin{bmatrix} .5 & .5 & .5 \\ -1 & -1 & -1 \\ .5 & .5 & .5 \end{bmatrix} * sqrt(1/3) * \begin{bmatrix} 255 & 255 & 255 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$0 = \sum_{z=0}^{2} \sum_{y=0}^{2} sqrt(2/3) * \begin{bmatrix} 4.3301e-01 & 3.0616e-17 & -4.3301e-01 \\ -8.6603e-01 & -6.1232e-17 & 8.6603e-01 \\ 4.3301e-01 & 3.0616e-17 & -4.3301e-01 \end{bmatrix} * sqrt(2/3) * \begin{bmatrix} 255 & 255 & 255 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$0 = \sum_{z=0}^{2} \sum_{y=0}^{2} sqrt(2/3) * \begin{bmatrix} 0.25000 & .5 & 0.25000 \\ -.5 & 1 & -.5 \\ 0.25000 & -.5 & 0.25000 \end{bmatrix} * sqrt(2/3) * \begin{bmatrix} 255 & 255 & 255 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$^{6}\sqrt{1/3} * 250 * \sqrt{1/3} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} + \sqrt{1/3} * 0 * \sqrt{2/3} * \begin{bmatrix} .86603 & 6.1232e-17 & .86603 \\ .86603 & 6.1232e-17 & .86603 \\ .86603 & 6.1232e-17 & .86603 \end{bmatrix} +$$

$$\sqrt{1/3} * 0 * \sqrt{2/3} * \begin{bmatrix} .5 & -1 & .5 \\ .5 & -1 & .5 \\ .5 & -1 & .5 \end{bmatrix} + \sqrt{2/3} * 250 * \sqrt{1/3} * \begin{bmatrix} 8.6603e-01 & 8.6603e-01 & 8.6603e \\ 6.1232e-17 & 6.1232e-17 & 6.1232e \\ -8.6603e-01 & -8.6603e-01 & -8.6603e \end{bmatrix}$$

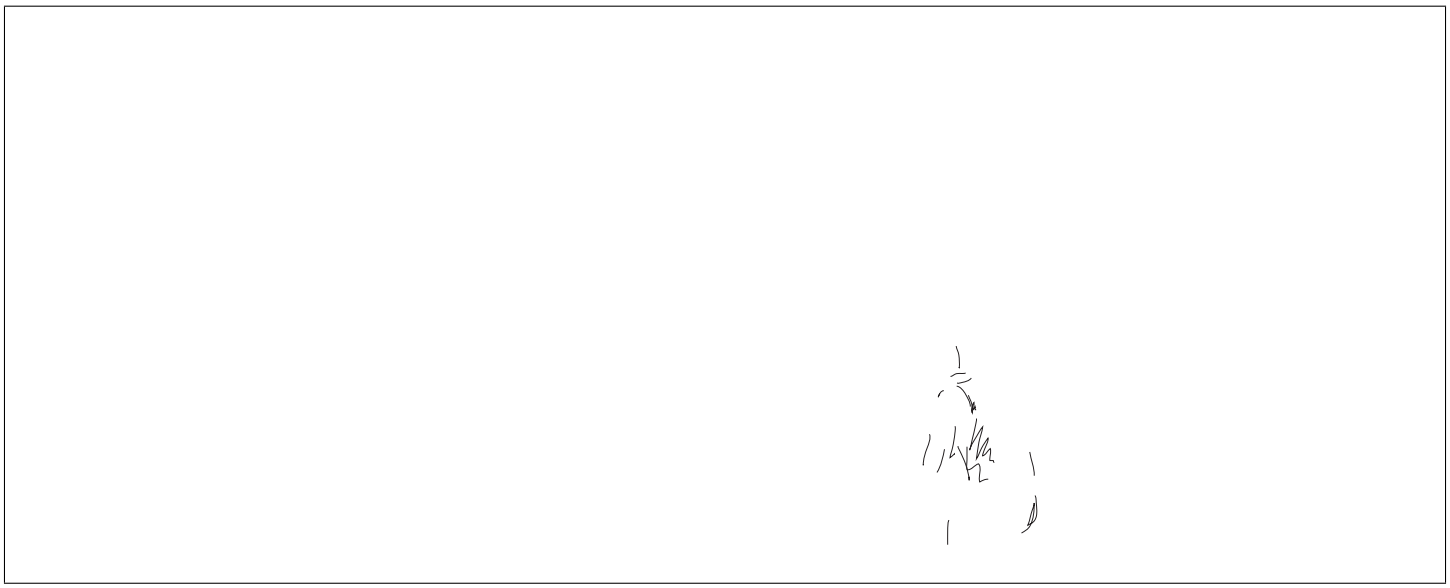
$$\sqrt{2/3} * 0 * \sqrt{2/3} * \begin{bmatrix} 7.5000e-01 & 5.3029e-17 & -7.5000e-01 \\ 5.3029e-17 & 3.7494e-33 & -5.3029e-17 \\ -7.5000e-01 & -5.3029e-17 & 7.5000e-01 \end{bmatrix} + \sqrt{2/3} * 0 *$$

$$\sqrt{2/3} * \begin{bmatrix} 4.3301e-01 & -8.6603e-01 & 4.3301e-01 \\ 3.0616e-17 & -6.1232e-17 & 3.0616e-17 \\ -4.3301e-01 & 8.6603e-01 & -4.3301e-01 \end{bmatrix} + \sqrt{2/3} * 0 * \sqrt{1/3} * \begin{bmatrix} .5 & .5 & .5 \\ -1 & -1 & -1 \\ .5 & .5 & .5 \end{bmatrix} +$$

$$\sqrt{2/3} * 0 * \sqrt{2/3} * \begin{bmatrix} 4.3301e-01 & 3.0616e-17 & -4.3301e-01 \\ -8.6603e-01 & -6.1232e-17 & 8.6603e-01 \\ 4.3301e-01 & 3.0616e-17 & -4.3301e-01 \end{bmatrix} + \sqrt{2/3} * 0 *$$

$$\sqrt{2/3} \qquad * \qquad \begin{bmatrix} 0.25000 & .5 & 0.25000 \\ -.5 & 1 & -.5 \\ 0.25000 & -.5 & 0.25000 \end{bmatrix}$$

Special thanks to Danny Comer for helping with these concepts.



Cory Arcangel
THE SOURC$_{\text{E}}$
Issue 8: Six, Sixty-six.

Creative Capital

Published by Arcangel Surfware.
Designed by Claire Kwong and Cory Arcangel.
Special thx: Amanda Schmidt, Gil Gentile, Allie Tepper, Elliot Kaufman.
With support from the Creative Capital Foundation.
ISBN: 978-0-9966360-5-6

www.arcangelsurfware.biz

[1]

Change Log 08-17-2015

- Changed www.arcangelsurfware.com on colophon 2 www.arcangelsurfware.biz ... uuuugh.
- Updated current Arcangel Surfware crew.

Change Log 02-27-2015

- Deleted project date from title page
- Changed www.coryarcangel.com to www.arcangelsurfware.biz [2]

[1] $blablablablablablabl

[2] Thi$ t3xt wa$ c0pi3d fr0m vari0u$ Can0n 0c3́ Vari0Print$^{\widehat{3xtr3gist3r3d}}$ mat3rial$.

[3]

[4]

[5]

☯ 💻 ;-)