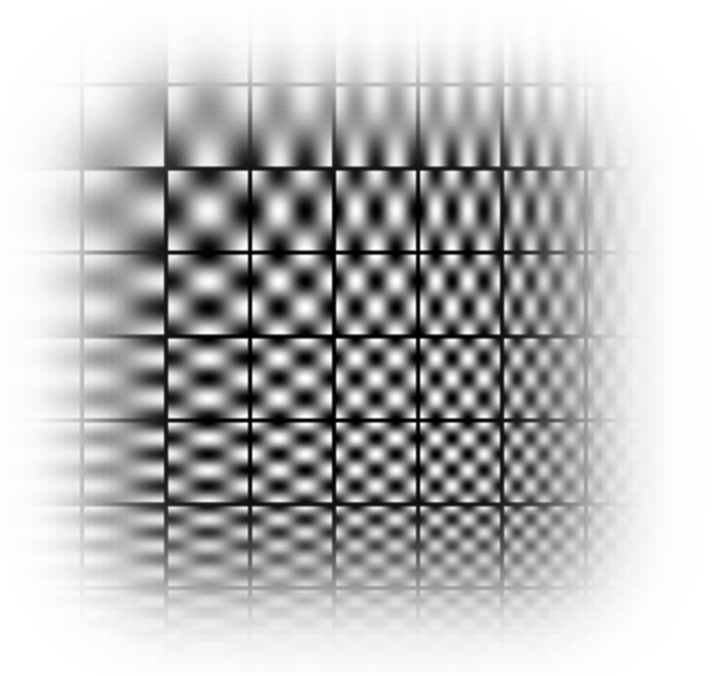
# Discrete Cosine Transformation (DCT)

By:
Amirhessam Tahmassebi
Dept. of Scientific Computing
Florida State University
October 2015



# What is image compression?

- The objective of image compression is to reduce irrelevant and redundant in data in order to be able store or transmit data in an efficient form.

- There are two types of image compression:
  - Lossless
  - Lossy

# Types of image compression:

- Lossless:

  Lossless image compression is a compression algorithm that allows the original image to be perfectly reconstructed from the compressed data.

- Lossy:

  Lossy compression is a type of compression where a certain amount of information is discarded which means that some data are lost and hence the image cannot be decompressed with 100% originality.

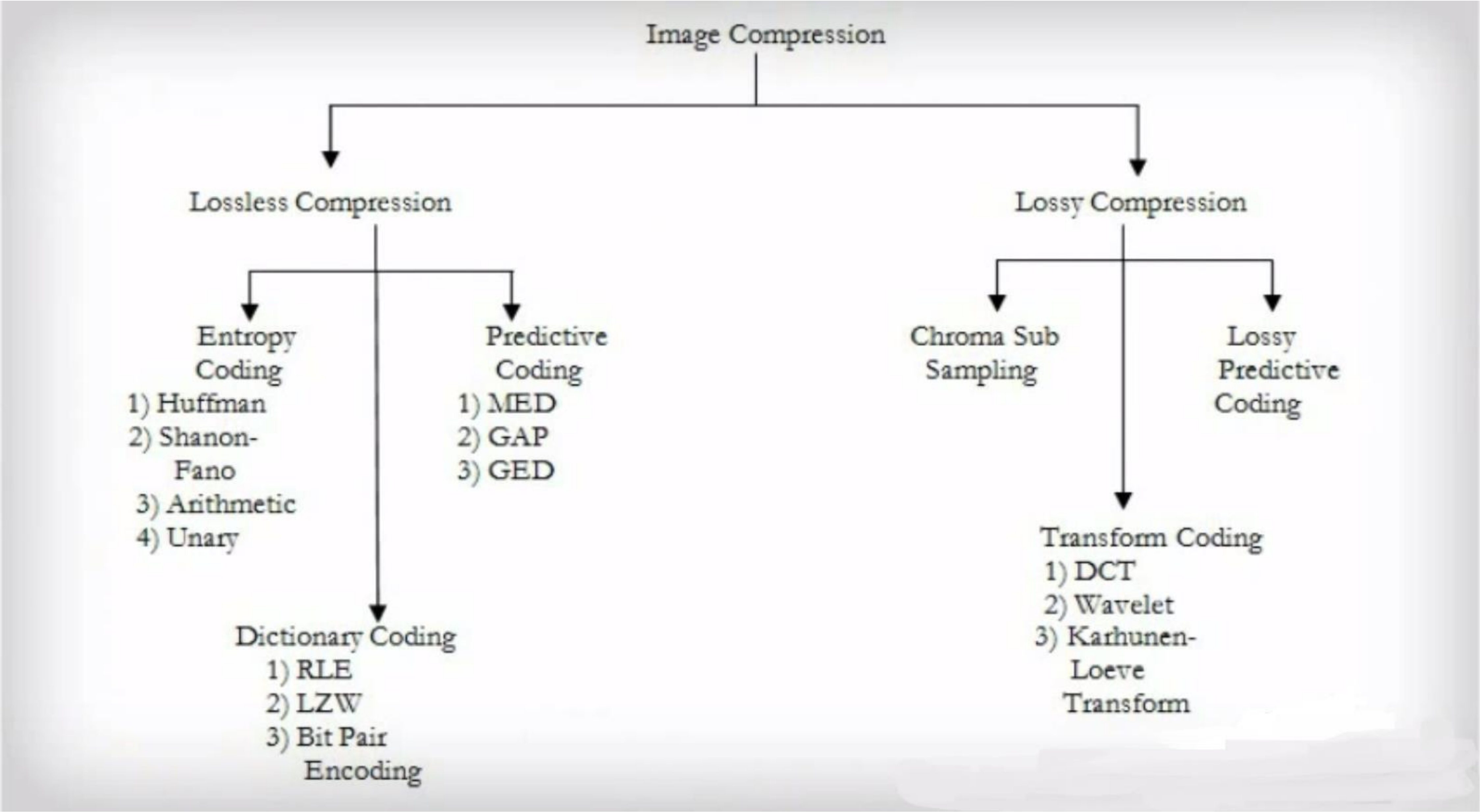


Image Compression
Lossless Compression
Lossy Compression
Entropy Coding
1) Huffman
2) Shanon-Fano
3) Arithmetic
4) Unary
Predictive Coding
1) MED
2) GAP
3) GED
Dictionary Coding
1) RLE
2) LZW
3) Bit Pair Encoding
Chroma Sub Sampling
Lossy Predictive Coding
Transform Coding
1) DCT
2) Wavelet
3) Karhunen-Loeve Transform

# DCT:

- A discrete cosine transformation (DCT) expresses a finite sequence of data points into the summation of a series of cosine waves oscillating at different frequencies.

- The use of cosine functions is critical for compression because:
  - Rather than sine functions, fewer cosine functions are needed to approximate a typical signal.
  - For differential equations the cosines express a particular choice of boundary conditions.

They are very similar to Fourier Transformations, but:

- **Discrete Fourier Transform (DFT):** $\begin{cases} Uses\ Both\ Sines\ and\ Cosines \\ Uses\ of\ Complex\ Numbers \end{cases}$

- **Discrete Cosine Transform (DCT):** $\begin{cases} Uses\ just\ Cosine\ functions \\ Uses\ of\ Real\ Coefficients \end{cases}$

# Formal Definitions:

- DCT is a linear, invertible function with a set of real numbers or equivalently an invertible $N \times N$ square matrix.
- There are eight standard DCT variants, of which four are common.

- DCT-I

$$X_k = \frac{1}{2}(x_0 + (-1)^k x_{N-1}) + \sum_{n=1}^{N-2} x_n \cos\left[\frac{\pi}{N-1} nk\right] \qquad k = 0, \ldots, N-1.$$

- DCT-II

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right) k\right] \qquad k = 0, \ldots, N-1.$$

- DCT-III

$$X_k = \frac{1}{2} x_0 + \sum_{n=1}^{N-1} x_n \cos\left[\frac{\pi}{N} n \left(k + \frac{1}{2}\right)\right] \qquad k = 0, \ldots, N-1.$$

- DCT-IV

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)\left(k + \frac{1}{2}\right)\right] \qquad k = 0, \ldots, N-1.$$

# Applications

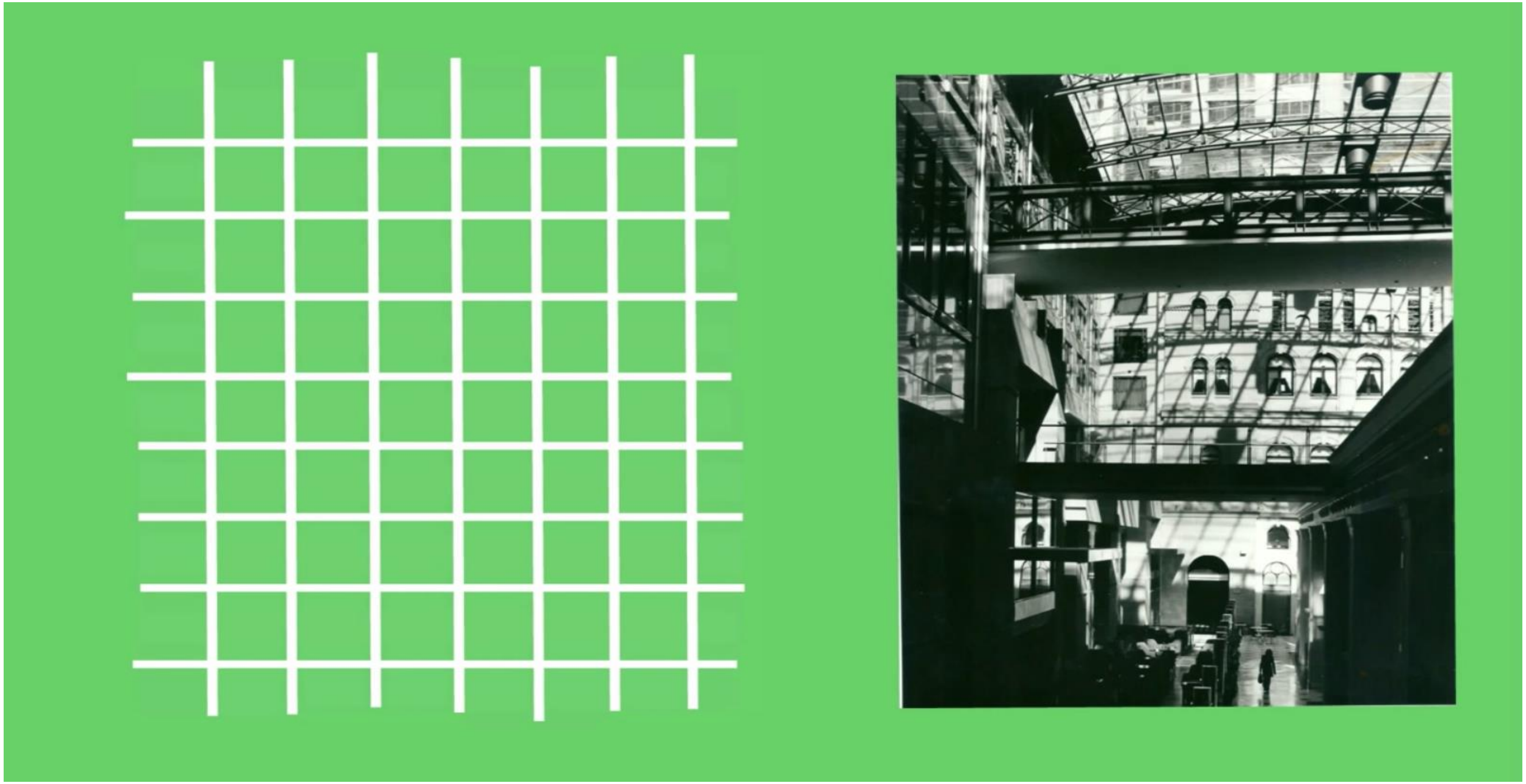
- DCTs are widely used in image and audio compression such as JPEG , MP3, MJPEG, MPEG, DV, Daala.

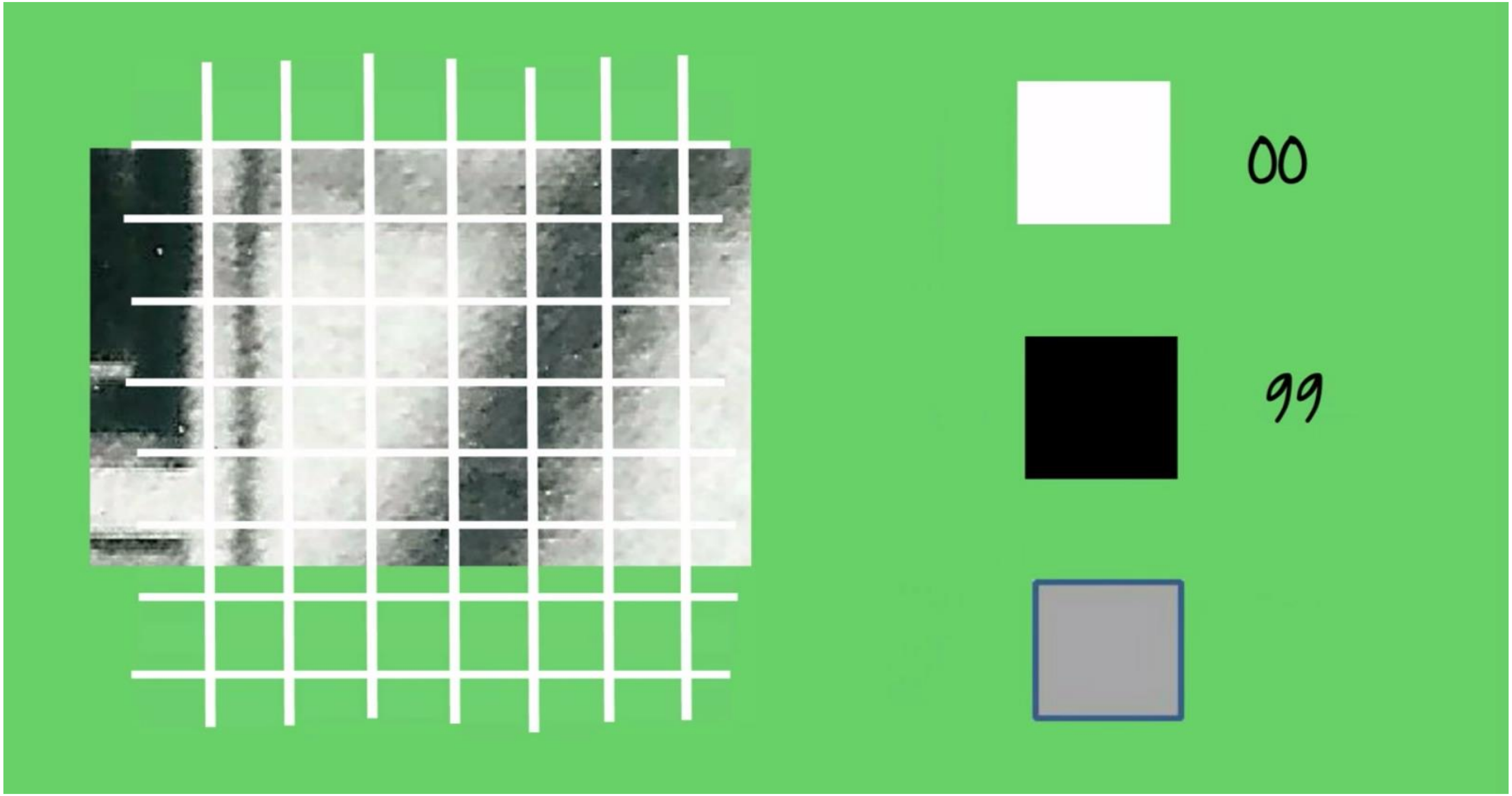
- They use lossy compression which is the class of data coding methods that uses inexact approximations or partial data discarding to represent the content.

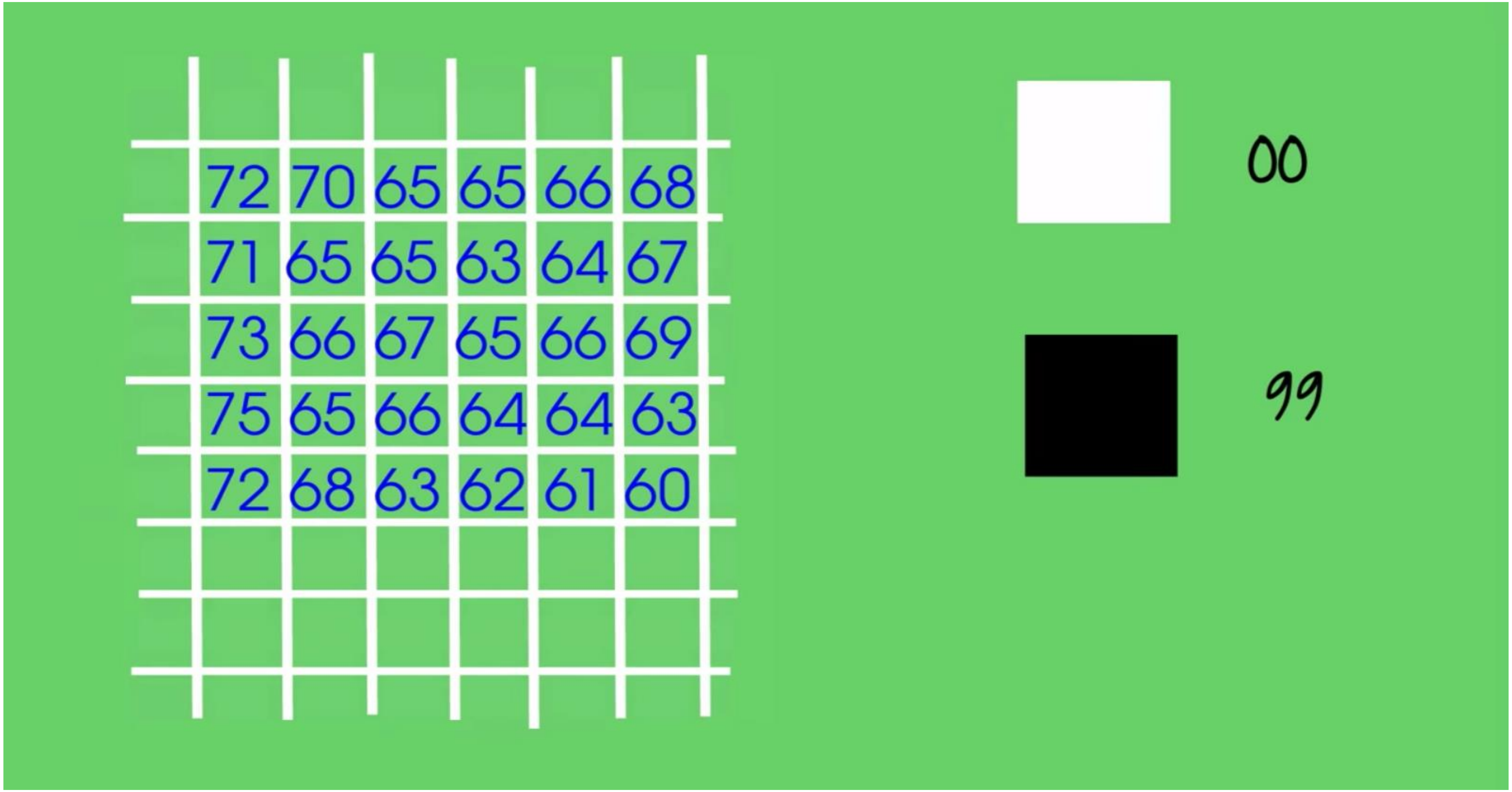
- These techniques are used to reduce data size for storage, handling, and transmitting content in information technology.

- In other words small high-frequency components can be discarded.

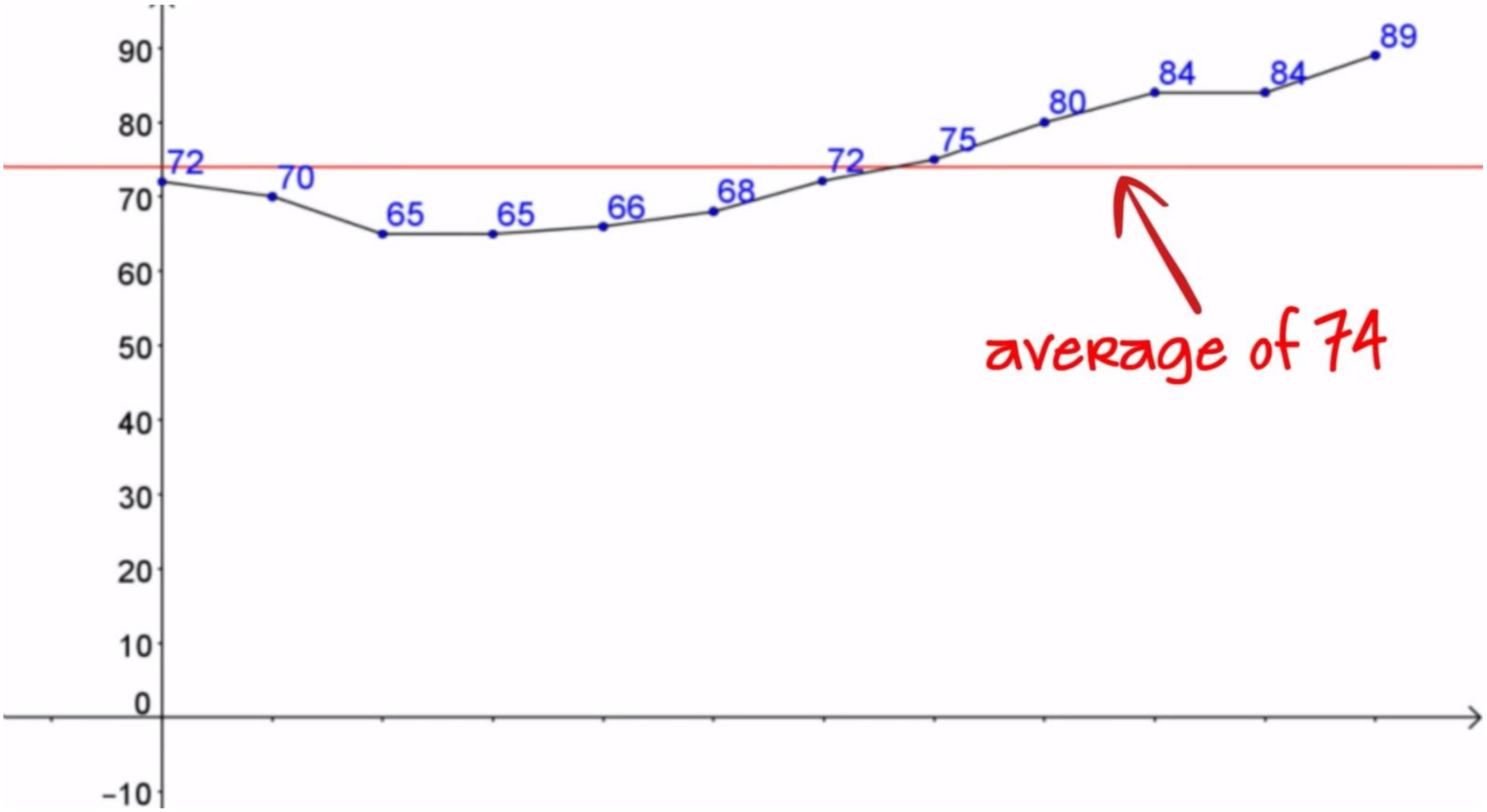
# One Dimensional Example

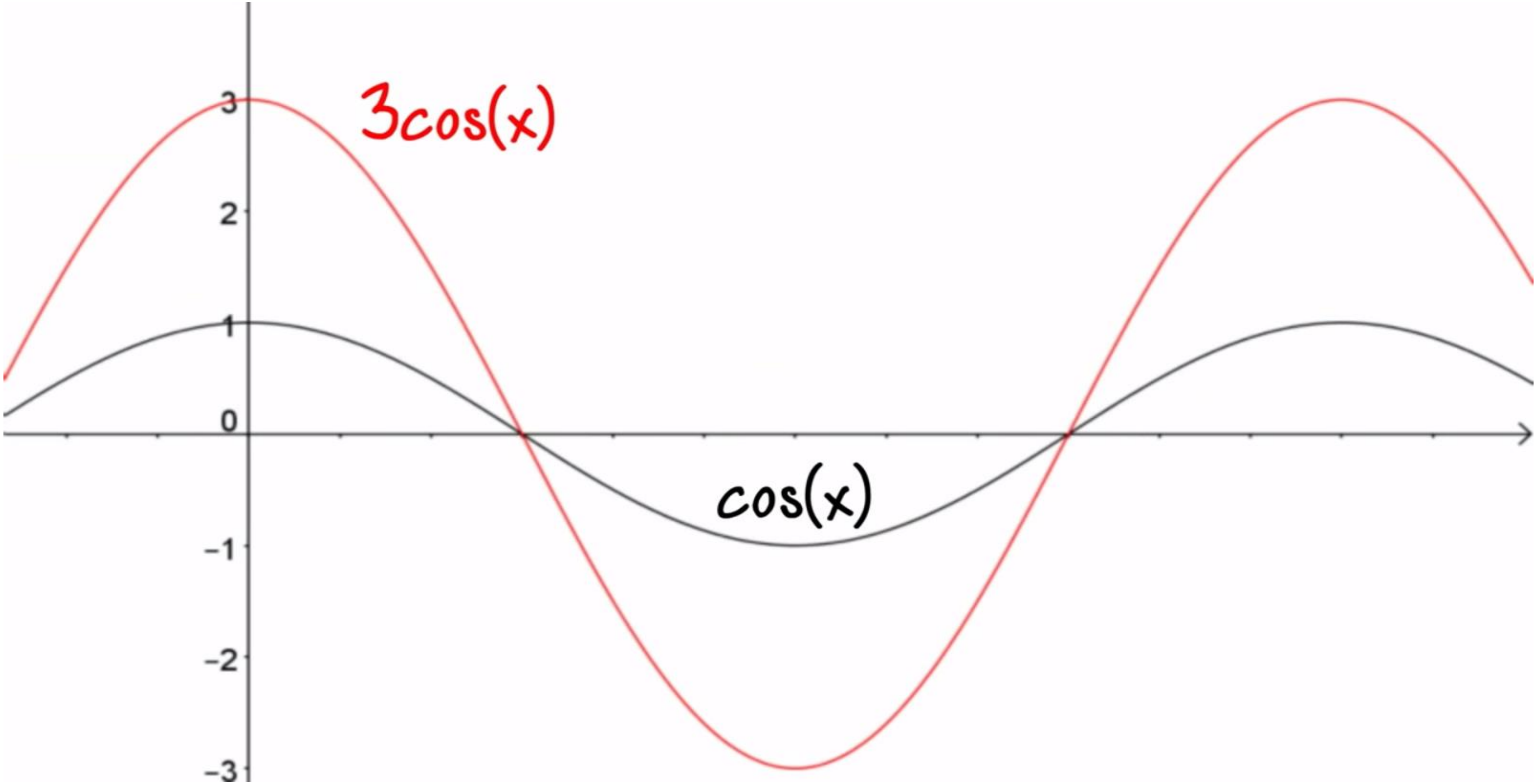



00
99


72 70 65 65 66 68
71 65 65 63 64 67
73 66 67 65 66 69
75 65 66 64 64 63
72 68 63 62 61 60
00
99

72 70 65 65 66 68 72 75 80 84 84 89


90
80
70
60
50
40
30
20
10
0
−10
72
70
65
65
66
68
72
75
80
84
84
89
average of 74


3cos(x)
3
2
1
0
cos(x)
−1
−2
−3

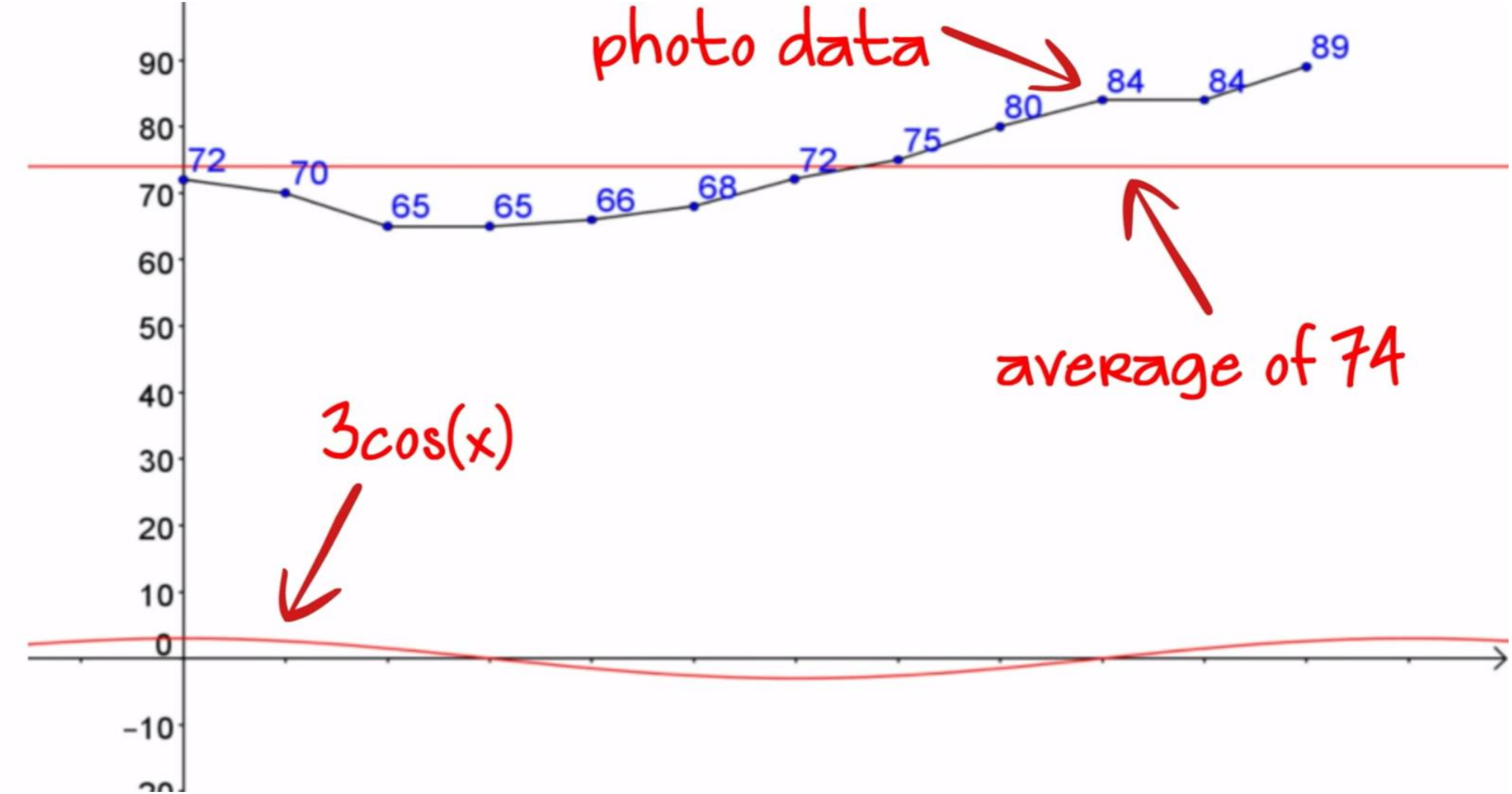
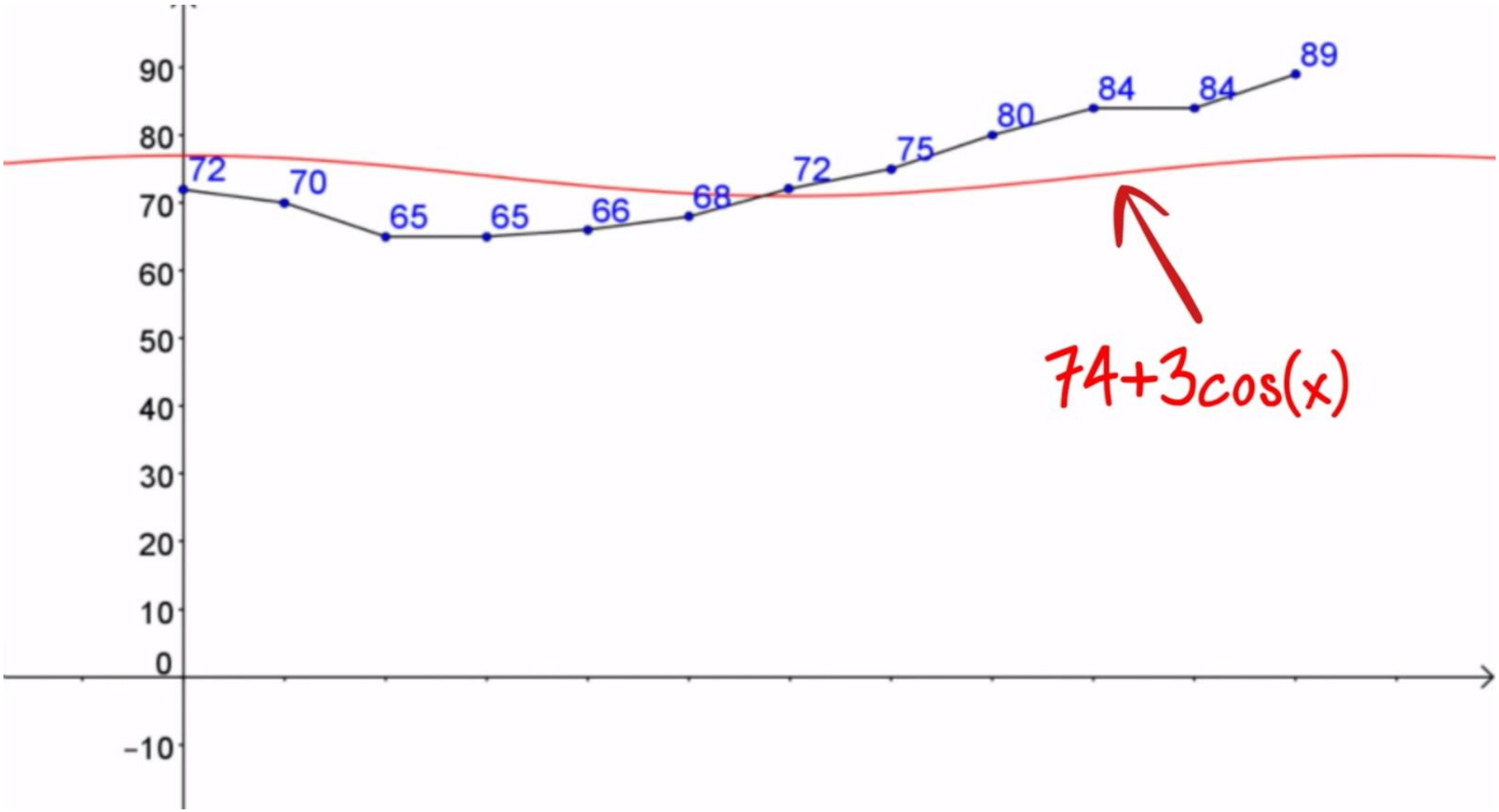
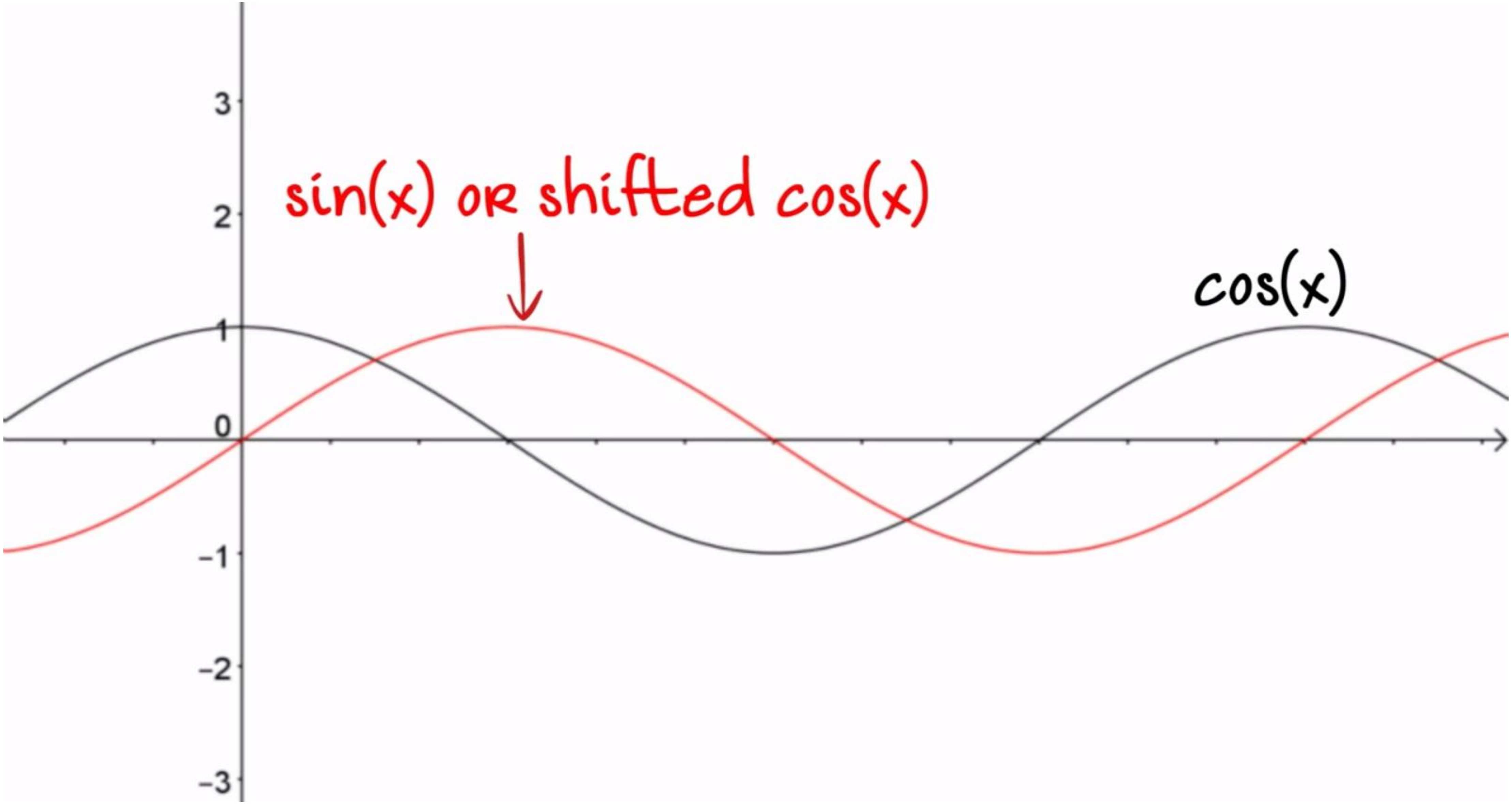
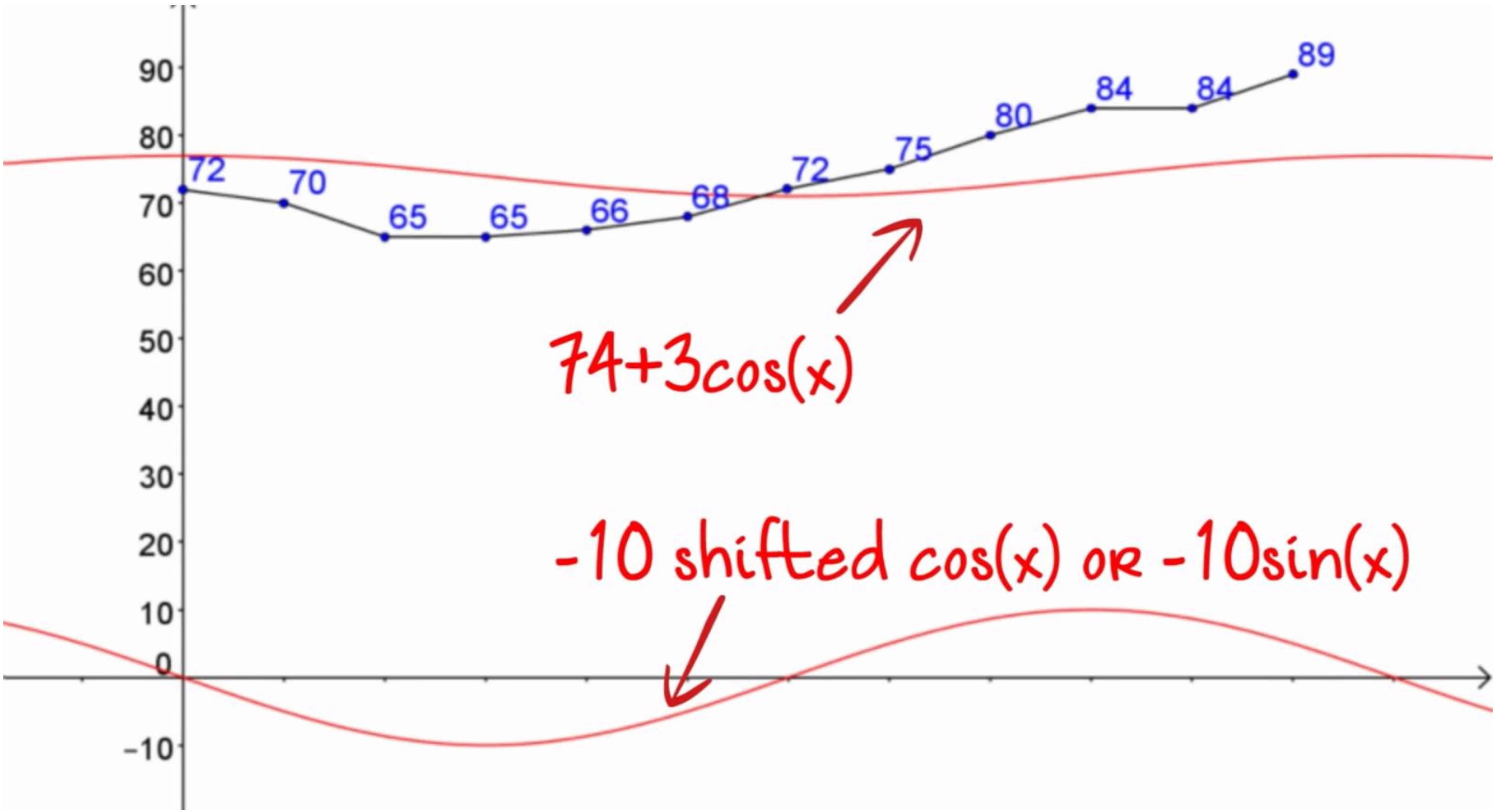
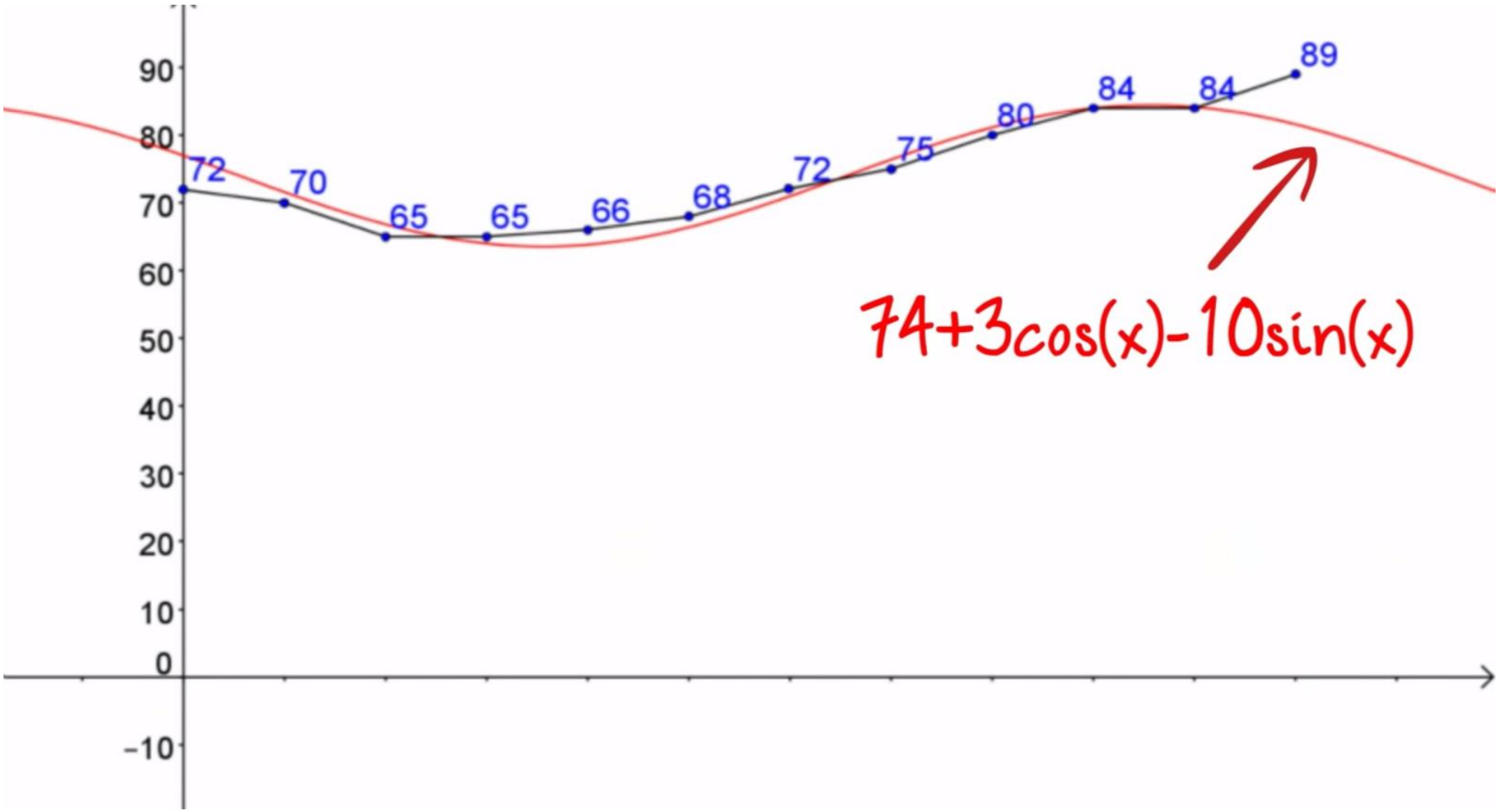


photo data
average of 74
3cos(x)
90
80
70
60
50
40
30
20
10
0
−10
72
70
65
65
66
68
72
75
80
84
84
89


90
80
70
60
50
40
30
20
10
0
−10
72
70
65
65
66
68
72
75
80
84
84
89
74+3cos(x)


sin(x) or shifted cos(x)
cos(x)
3
2
1
0
−1
−2
−3

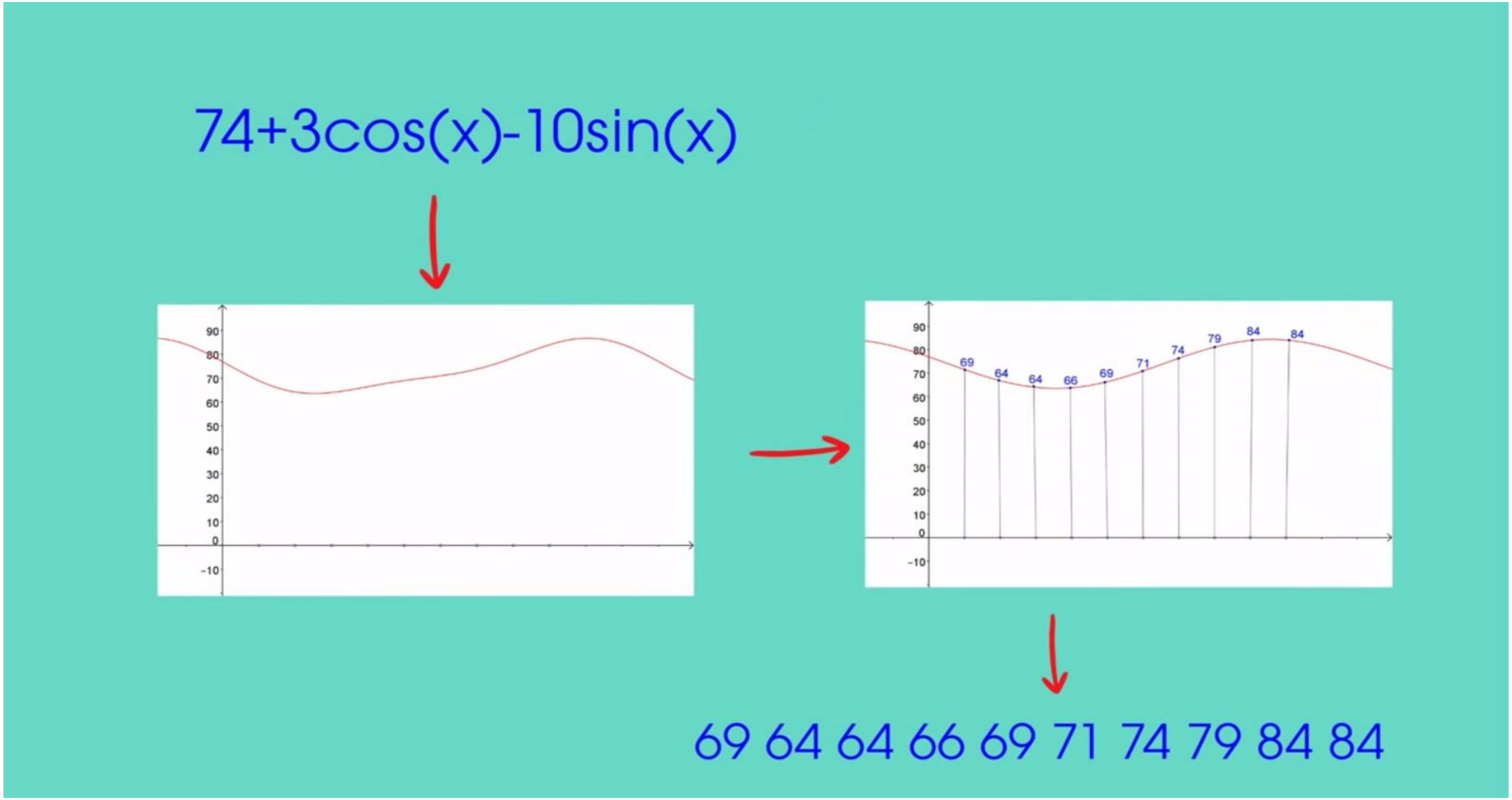

90
80
70
60
50
40
30
20
10
0
−10
72
70
65
65
66
68
72
75
80
84
84
89
74+3cos(x)
-10 shifted cos(x) or -10sin(x)

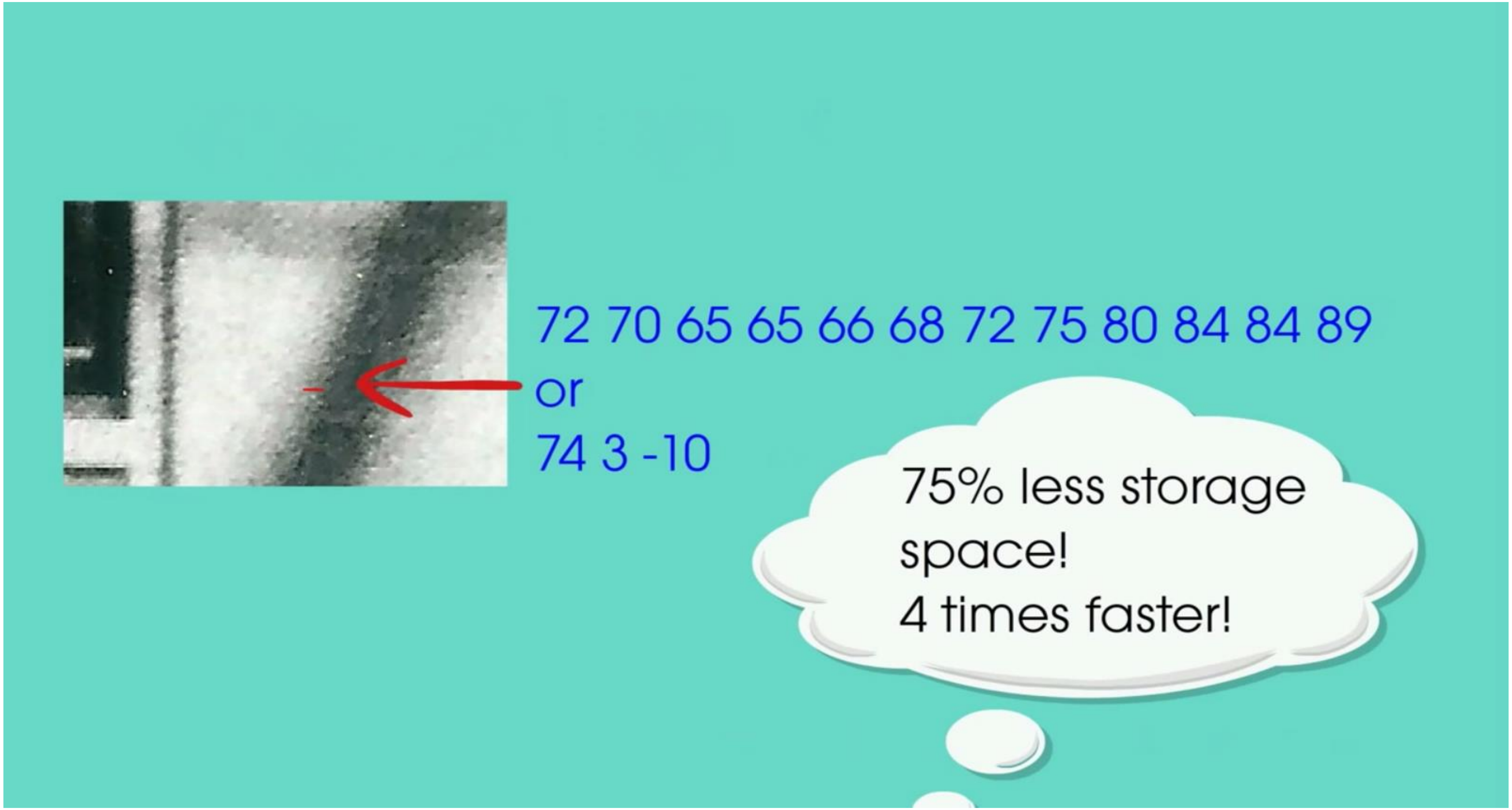


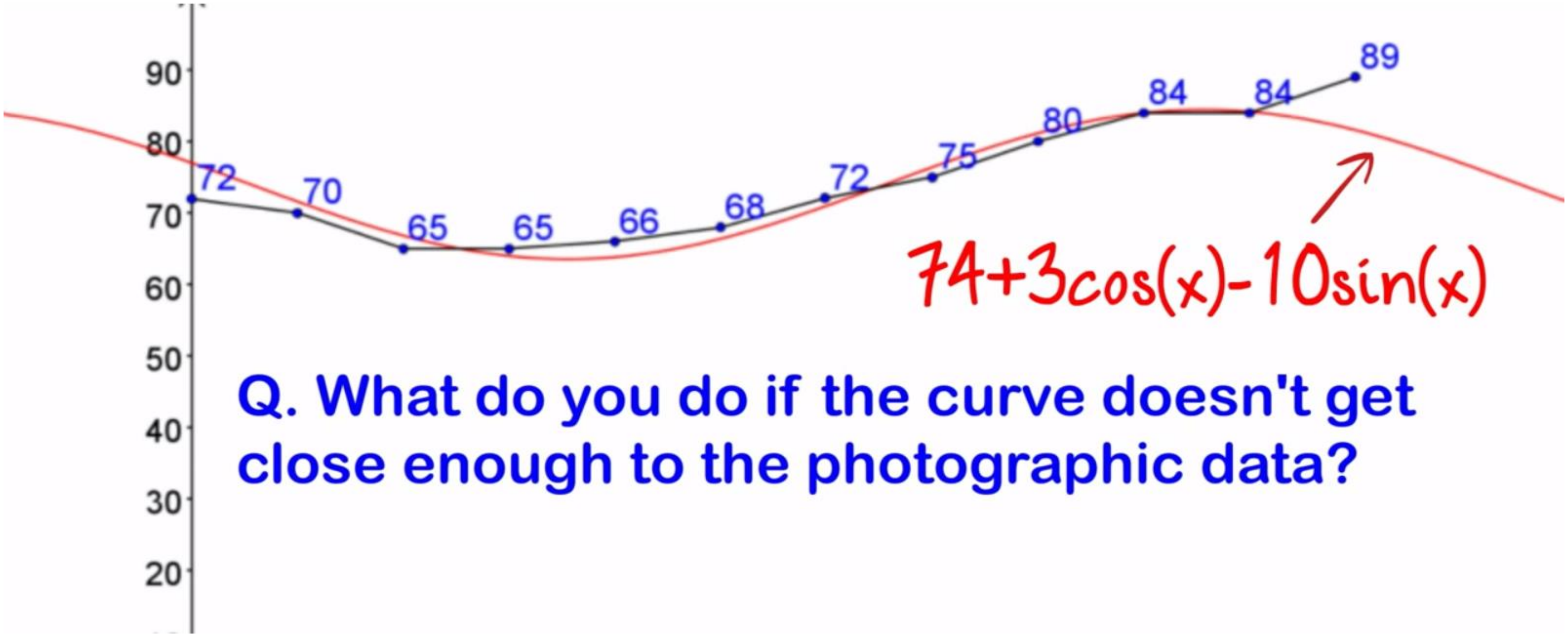
90
80
70
60
50
40
30
20
10
0
−10
72
70
65
65
66
68
72
75
80
84
84
89
74+3cos(x)-10sin(x)

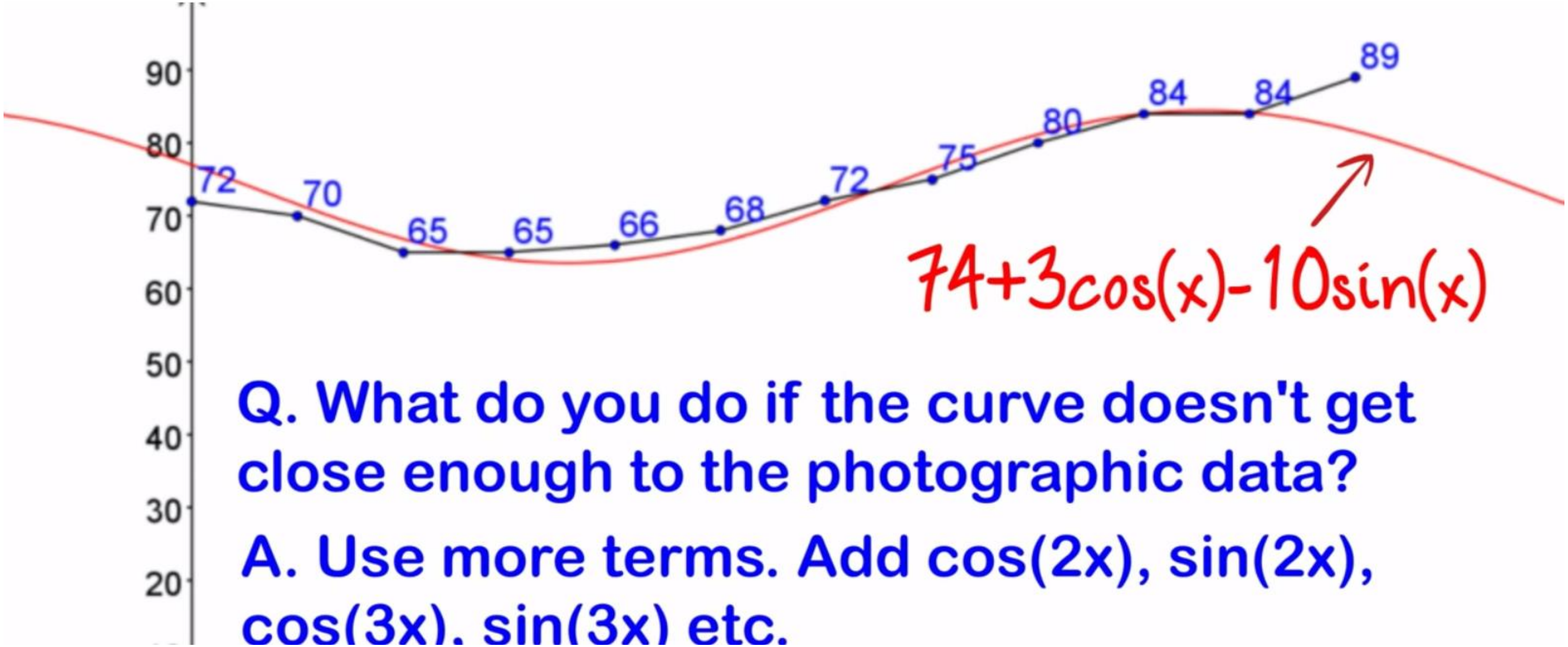
72 70 65 65 66 68 72 75 80 84 84 89

74 3 -10


74+3cos(x)-10sin(x)
69 64 64 66 69 71 74 79 84 84

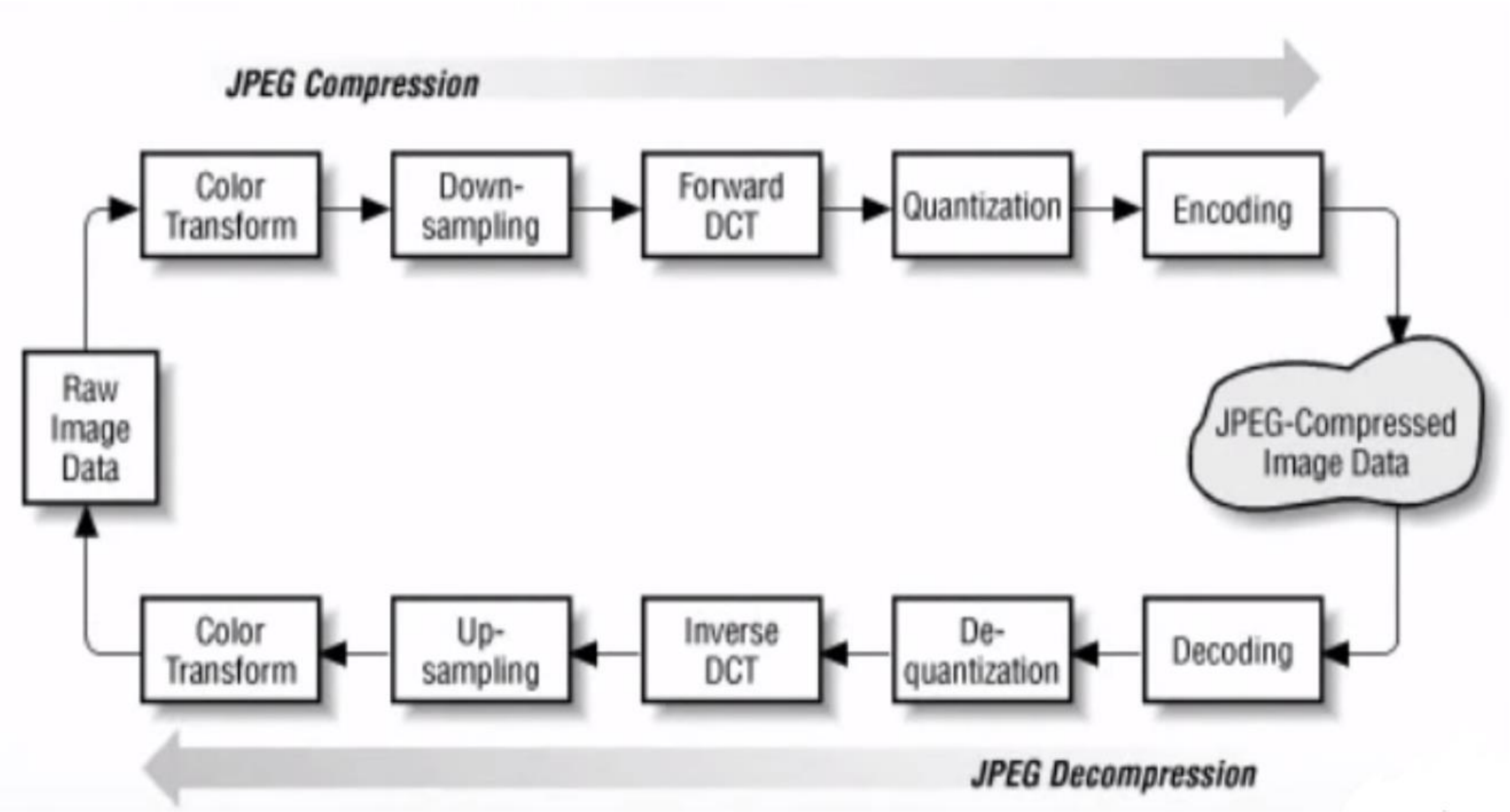

72 70 65 65 66 68 72 75 80 84 84 89
or
74 3 -10
75% less storage space!
4 times faster!


90
80
70
60
50
40
30
20
72
70
65
65
66
68
72
75
80
84
84
89
74+3cos(x)-10sin(x)
Q. What do you do if the curve doesn't get close enough to the photographic data?


90
80
70
60
50
40
30
20
72
70
65
65
66
68
72
75
80
84
84
89
74+3cos(x)-10sin(x)
Q. What do you do if the curve doesn't get close enough to the photographic data?
A. Use more terms. Add cos(2x), sin(2x), cos(3x), sin(3x) etc.

# JPEG (Joint Photographic Experts Group)

- According to JFIF Standard, the encoding steps are:
  1. The representation of the colors in the image is converted from RGB to $Y'C_BC_{R.}$
  2. The resolution of the Chroma data is reduced, usually by a factor of 2 or 3.
  3. The image is split into blocks of 8×8 pixels, and for each block, each of the Y, $C_B$, and $C_R$ data undergoes the Discrete Cosine Transform (DCT).
  4. The amplitudes of the frequency components are quantized.
  5. The resulting data for all 8×8 blocks is further compressed with a lossless algorithm.


JPEG Compression
Color Transform
Down-sampling
Forward DCT
Quantization
Encoding
Raw Image Data
JPEG-Compressed Image Data
Color Transform
Up-sampling
Inverse DCT
De-quantization
Decoding
JPEG Decompression

# Two Dimensional Example

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix} . \begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

# Unquantized DCT Coefficients

$$G_{u,v} = \frac{1}{4}\alpha(u)\alpha(v)\sum_{x=0}^{7}\sum_{y=0}^{7} g_{x,y}\cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right]$$

$$G = \overset{\xrightarrow{u}}{\begin{bmatrix} -415.38 & -30.19 & -61.20 & 27.24 & 56.12 & -20.10 & -2.39 & 0.46 \\ 4.47 & -21.86 & -60.76 & 10.25 & 13.15 & -7.09 & -8.54 & 4.88 \\ -46.83 & 7.37 & 77.13 & -24.56 & -28.91 & 9.93 & 5.42 & -5.65 \\ -48.53 & 12.07 & 34.10 & -14.76 & -10.24 & 6.30 & 1.83 & 1.95 \\ 12.12 & -6.55 & -13.20 & -3.95 & -1.87 & 1.75 & -2.79 & 3.14 \\ -7.73 & 2.91 & 2.38 & -5.94 & -2.38 & 0.94 & 4.30 & 1.85 \\ -1.03 & 0.18 & 0.42 & -2.42 & -0.88 & -3.02 & 4.12 & -0.66 \\ -0.17 & 0.14 & -1.07 & -4.19 & -1.17 & -0.10 & 0.50 & 1.68 \end{bmatrix}} \Bigg\downarrow v.$$

# Quantization Matrix

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}.$$

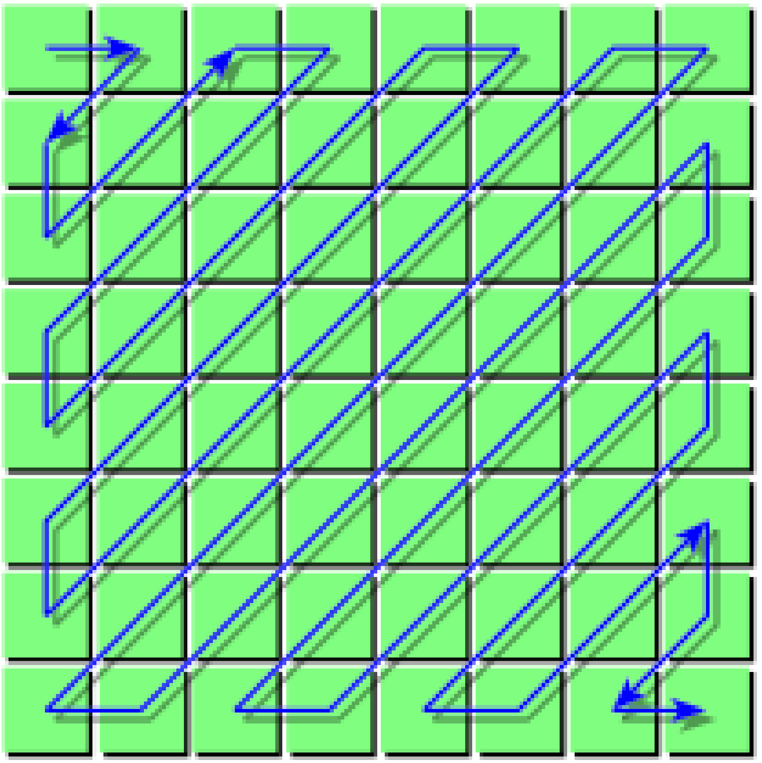
# Quantized DCT Coefficients

$$B_{j,k} = \text{round}\left(\frac{G_{j,k}}{Q_{j,k}}\right) \text{ for } j = 0, 1, 2, \ldots, 7; k = 0, 1, 2, \ldots, 7$$

$$\text{round}\left(\frac{-415.37}{16}\right) = \text{round}\,(-25.96) = -26.$$

$$B = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

# Entropy Coding



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| −26 | | | | | | | |
| −3 | 0 | | | | | | |
| −3 | −2 | −6 | | | | | |
| 2 | −4 | 1 | −3 | | | | |
| 1 | 1 | 5 | 1 | 2 | | | |
| −1 | 1 | −1 | 2 | 0 | 0 | | |
| 0 | 0 | 0 | -1 | -1 | 0 | 0 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 0 | 0 | 0 | 0 | | |
| 0 | 0 | 0 | 0 | 0 | | | |
| 0 | 0 | 0 | 0 | | | | |
| 0 | 0 | 0 | | | | | |
| 0 | 0 | | | | | | |
| 0 | | | | | | | |

# Appendix

# 2-D Discrete Fourier Transform (DFT)

- 2-D DFT of an N×N image {$u(m, n)$ } is a separable transform defined as:

$$v(k,l)=\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}u(m,n)W_N^{km}W_N^{ln}, \quad 0\le k,l\le N-1$$

$$W_N \equiv \exp\left\{\frac{-j2\pi}{N}\right\}$$

- The 2-D DFT inverse transform is given as:

$$v(k,l)=\sum_{m=0}^{N-1}\sum_{n=0}^{N-1}u(m,n)W_N^{km}W_N^{ln}, \quad 0\le k,l\le N-1$$

- In matrix notation: **V** = **FUF** and **U** = **F*VF***

# Properties of 2-D DFT

[The $N^2 \times N^2$ matrix $\mathcal{F}$ represents the $N \times N$ 2-D unitary DFT]

- **Symmetric and unitary**

  $$\mathcal{F}^T = \mathcal{F} \quad \text{and} \quad \mathcal{F}^{-1} = \mathcal{F}^*$$

- **Periodic extensions**

  $$v(k + N, l + N) = v(k, l) \qquad \forall k, l$$
  $$u(m + N, n + N) = u(m, n) \qquad \forall m, n$$

- **Sampled Fourier spectrum**

  If $\bar{u}(m,n) = u(m,n),\ 0 \le m,n \le N-1$ ,and $\bar{u}(m,n) = 0$ otherwise, then:

  $$\tilde{U}\left(\frac{2\pi k}{N}, \frac{2\pi l}{N}\right) = DFT\{u(m,n)\} = v(k, lx)$$

  where $\tilde{U}(\omega_1, \omega_2)$ is the Fourier transform of $\bar{u}(m,n)$

- **Fast transform**

  Since 2-D DFT is separable, it is equivalent to 2N 1-D unitary DFTs, each of which can be performed in $O(N \log_2 N)$ via the FFT. Hence the total number of operations is $O(N^2 \log_2 N)$.

# Properties of 2-D DFT

- **Conjugate symmetry**

$$v\left(\frac{N}{2}\pm k,\frac{N}{2}\pm l\right)=v^*\left(\frac{N}{2}\mp k,\frac{N}{2}\mp l\right),\quad 0\le k,l\le\frac{N}{2}-1$$

or $v(k, l) = v^*(\mathrm{N}-k, \mathrm{N}-l), \qquad 0 \le k, l \le \mathrm{N}-1$

- **Basis Images**

  The basis images are given by definition:

$$\mathbf{A}^*_{k,l}=\Phi_k\Phi_l^T=\frac{1}{N}\left\{W_N^{-(km+\ln)},\ 0\le m,n\le N-1\right\},\ 0\le k,l\le N-1$$

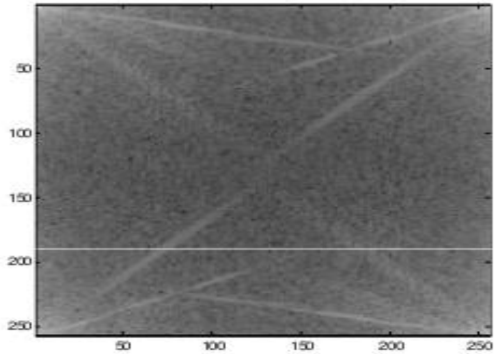
- **2-D circular convolution theorem**

  The DFT of the 2-D circular convolution of two arrays is the product of their DFTs:

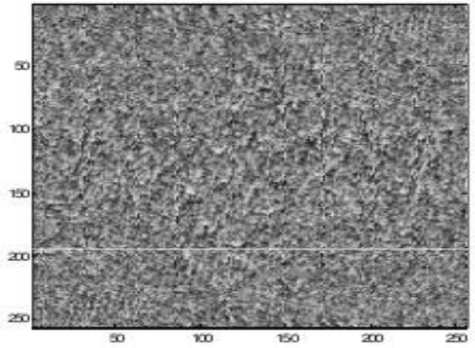
$$DFT\{h(m,n)\otimes u(m,n)\}=DFT\{h(m,n)\}.DFT\{\ u(m,n)\}$$

# Examples of DFT


Original Image




Log(magnitude of DFT coeff)




Phase Image

# Discrete Cosine Transform (DCT)

- The N×N DCT matrix **C** = {$c(k, n)$}, is defined as

$$c(k,n)=\begin{cases}\dfrac{1}{\sqrt{N}}, & k=0,\ 0\le n\le N-1\\[2ex] \sqrt{\dfrac{2}{N}}\cos\dfrac{\pi(2n+1)k}{2N}, & 1\le k\le N-1,\ 0\le n\le N-1\end{cases}$$

- Properties of DCT:
  1. Real and orthogonal
  2. $\mathbf{C} = \mathbf{C}^* \Rightarrow \mathbf{C}^{-1} = \mathbf{C}^T$
  3. **Not** the real part of the unitary DFT
  4. Fast transform
  5. Excellent energy compaction.
  6. The basis vector of the DCT (rows of **C**) are eigen-vectors of symmetric traditional matrix $\mathbf{Q}_r$
  7. DCT is very close to the KL (Karhunen-Loeve) transform of a first-order stationary Markov sequence.

$$\mathbf{Q}_r = \begin{bmatrix} 1-\alpha & -\alpha & 0 & \mathbf{0} \\ -\alpha & 1 & \ddots & \\ 0 & \ddots & 1 & -\alpha \\ \mathbf{0} & & -\alpha & 1-\alpha \end{bmatrix}$$

# Discrete Cosine Transform (DCT)

- The N×N DCT matrix **C** = {$c(k, n)$}, is defined as

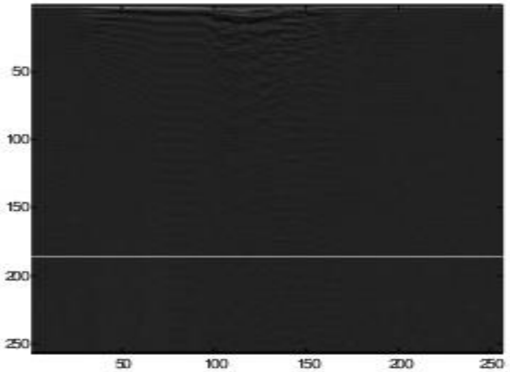
$$c(k,n)=\begin{cases}\dfrac{1}{\sqrt{N}}, & k=0,\ 0\le n\le N-1\\ \sqrt{\dfrac{2}{N}}\cos\dfrac{\pi(2n+1)k}{2N}, & 1\le k\le N-1,\ 0\le n\le N-1\end{cases}$$

- Properties of DCT:
  1. Real and orthogonal
  2. **$\mathbf{C} = \mathbf{C}^* \Rightarrow \mathbf{C}^{-1} = \mathbf{C}^T$**
  3. **Not** the real part of the unitary DFT
  4. Fast transform
  5. Excellent energy compaction.
  6. The basis vector of the DCT (rows of **C**) are eigen-vectors of symmetric traditional matrix $\mathbf{Q}_r$
  7. DCT is very close to the KL (Karhunen-Loeve) transform of a first-order stationary Markov sequence.

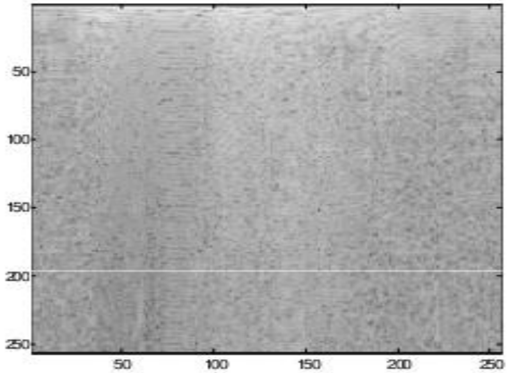
$$\mathbf{Q}_r=\begin{bmatrix}1-\alpha & -\alpha & 0 & \mathbf{0}\\ -\alpha & 1 & \ddots & \ddots\\ 0 & \ddots & 1 & -\alpha\\ \mathbf{0} & -\alpha & & 1-\alpha\end{bmatrix}$$

# Example of DCT


Original image




DCT coefficient




Log(magnitude of DCT coeff)