



WIKIPEDIA
The Free Encyclopedia

[Main page](#)
[Contents](#)
[Featured content](#)
[Current events](#)
[Random article](#)
[Donate to Wikipedia](#)
[Wikipedia store](#)

Interaction
[Help](#)
[About Wikipedia](#)
[Community portal](#)
[Recent changes](#)
[Contact page](#)

Tools
[What links here](#)
[Related changes](#)
[Upload file](#)
[Special pages](#)
[Permanent link](#)
[Page information](#)
[Wikidata item](#)
[Cite this page](#)

Print/export
[Create a book](#)
[Download as PDF](#)
[Printable version](#)

Languages
[中文](#)
[Edit links](#)

[Create account](#) [Log in](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#)

Block Truncation Coding

From Wikipedia, the free encyclopedia

Block Truncation Coding, or **BTC**, is a type of [lossy image compression](#) technique for [greyscale](#) images. It divides the original images into blocks and then uses a quantiser to reduce the number of [grey levels](#) in each block whilst maintaining the same [mean](#) and [standard deviation](#). It is an early predecessor of the popular hardware [DXTC](#) technique, although BTC compression method was first adapted to colour long before DXTC using a very similar approach called [Color Cell Compression](#).^[1] BTC has also been adapted to video compression ^[2]

BTC was first proposed by E.J Delp and O.R. Mitchell ^[3] at Purdue University. Another variation of BTC is **Absolute Moment Block Truncation Coding** or **AMBTC**, in which instead of using the standard deviation the first absolute moment is preserved along with the mean. AMBTC is computationally simpler than BTC and also typically results in a lower Mean Squared Error (MSE). AMBTC was proposed by Maximo Lema and Robert Mitchell.^[4]

Using sub-blocks of 4x4 pixels gives a compression ratio of 4:1 assuming 8-bit integer values are used during transmission or storage. Larger blocks allow greater compression ("a" and "b" values spread over more pixels) however quality also reduces with the increase in block size due to the nature of the algorithm.

The BTC algorithm was used for compressing [Mars Pathfinder](#)'s rover images.^[5]

Contents [\[hide\]](#)

- [1 Compression procedure](#)
- [2 Example](#)
 - [2.1 Encoder](#)
 - [2.2 Decoder](#)
- [3 References](#)

Compression procedure [\[edit\]](#)

A 256x256 [pixel](#) image is divided into blocks of typically 4x4 pixels. For each block the Mean and Standard Deviation are calculated, these values change from block to block. These two values define what values the reconstructed or new block will have, in other words the blocks of the BTC compressed image will all have the same mean and standard deviation of the original image. A two level quantization on the block is where we gain the compression, it is performed as follows:

$$y(i,j) = \begin{cases} 1, & x(i,j) > \bar{x} \\ 0, & x(i,j) \leq \bar{x} \end{cases}$$

Where $x(i,j)$ are pixel elements of the original block and $y(i,j)$ are elements of the compressed block. In words this can be explained as: If a pixel value is greater than the mean it is assigned the value "1", otherwise "0". Values equal to the mean can have either a "1" or a "0" depending on the preference of the person or organisation implementing the algorithm.

This 16 bit block is stored or transmitted along with the values of Mean and Standard Deviation. Reconstruction is made with two values "a" and "b" which preserve the mean and the standard deviation. The values of "a" and "b" can be computed as follows:

$$a = \bar{x} - \sigma \sqrt{\frac{q}{m-q}}$$
$$b = \bar{x} + \sigma \sqrt{\frac{m-q}{q}}$$

Where σ is the standard deviation, m is the total number of pixels in the block and q is the number of pixels greater than the mean (\bar{x})

To reconstruct the image, or create its approximation, elements assigned a 0 are replaced with the "a" value

and elements assigned a 1 are replaced with the "b" value.

$$x(i,j) = \begin{cases} a, & y(i,j) = 0 \\ b, & y(i,j) = 1 \end{cases}$$

This demonstrates that the algorithm is asymmetric in that the encoder has much more work to do than the decoder. This is because the decoder is simply replacing 1's and 0's with the estimated value whereas the encoder is also required to calculate the mean, standard deviation and the two values to use.^[6]

Example [\[edit\]](#)

Encoder [\[edit\]](#)

Take a 4x4 block from an image, in this case the mountain test image:^[7]

```
245 239 249 239
245 245 239 235
245 245 245 245
245 235 235 239
```

Like any small block from an image this appears rather boring to work with as the numbers are all quite similar, this is the nature of lossy compression and how it can work so well for images. Now we need to calculate two values from this data, that is the mean and standard deviation. The mean can be computed to 241.875, this is a simple calculation which should require no further explanation. The standard deviation is easily calculated at 4.36. From this the values of "a" and "b" can be calculated using the previous equations. They come out to be 236.935 and 245.718 respectively. The last calculation that needs to be done on the encoding side is to set the matrix to transmit to 1's and 0's so that each pixel can be transmitted as a single bit.

```
1 0 1 0
1 1 0 0
1 1 1 1
1 0 0 0
```

Decoder [\[edit\]](#)

Now at the decoder side all we need to do is reassign the "a" and "b" values to the 1 and 0 pixels. This will give us the following block:

```
245 236 245 236
245 245 236 236
245 245 245 245
245 236 236 236
```

As can be seen, the block has been reconstructed with the two values of "a" and "b" as integers (because images aren't defined to store floating point numbers). When working through the theory, this is a good point to calculate the mean and standard deviation of the reconstructed block. They should equal the original mean and standard deviation. Remember to use integers, otherwise much quantization error will become involved, as we previously quantised everything to integers in the encoder.

References [\[edit\]](#)

- [^] Liou, D. -M.; Huang, Y.; Reynolds, N. (1990). "A new microcomputer based imaging system with C/sup 3/ technique". *IEEE TENCON'90: 1990 IEEE Region 10 Conference on Computer and Communication Systems. Conference Proceedings*. p. 555. doi:10.1109/TENCON.1990.152671 [↗](#). ISBN 0-87942-556-3.
- [^] Healy, D.; Mitchell, O. (1981). "Digital Video Bandwidth Compression Using Block Truncation Coding". *IEEE Transactions on Communications* **29** (12): 1809. doi:10.1109/TCOM.1981.1094938 [↗](#).
- [^] Delp, E.; Mitchell, O. (1979). "Image Compression Using Block Truncation Coding". *IEEE Transactions on Communications* **27** (9): 1335. doi:10.1109/TCOM.1979.1094560 [↗](#).
- [^] Lema, M.; Mitchell, O. (1984). "Absolute Moment Block Truncation Coding and Its Application to Color Images". *IEEE Transactions on Communications* **32** (10): 1148. doi:10.1109/TCOM.1984.1095973 [↗](#).
- [^] "Rover Camera Instrument Description" [↗](#). NASA. Retrieved 2011-11-09.
- [^] Leis, J 2008, ELE4607 Advance Digital Communications, Module 3: Image & Video Coding. Lecture Slides, University of Southern Queensland, 2008.
- [^] Waterloo Fractal Coding and Analysis Group [↗](#)

This page was last modified on 29 August 2015, at 01:28.

Text is available under the [Creative Commons Attribution-ShareAlike License](#); additional terms may apply. By using this site, you agree to the [Terms of Use](#) and [Privacy Policy](#). Wikipedia® is a registered trademark of the [Wikimedia Foundation, Inc.](#), a non-profit organization.

[Privacy policy](#) [About Wikipedia](#) [Disclaimers](#) [Contact Wikipedia](#) [Developers](#) [Mobile view](#)

