



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](#)

# FELICS

From Wikipedia, the free encyclopedia

*For other uses, see [Felix \(disambiguation\)](#).*



This article **needs additional citations for verification**. Please help [improve this article](#) by [adding citations to reliable sources](#). Unsourced material may be challenged and removed. *(August 2011)*

**FELICS**, which stands for Fast Efficient & Lossless Image Compression System, is a [lossless image compression](#) algorithm that performs 5-times faster than the original [lossless JPEG](#) codec and achieves a similar [compression ratio](#).<sup>[1]</sup>

## Contents

- [1 History](#)
- [2 Principle](#)
- [3 Improvements](#)
- [4 See also](#)
- [5 References](#)

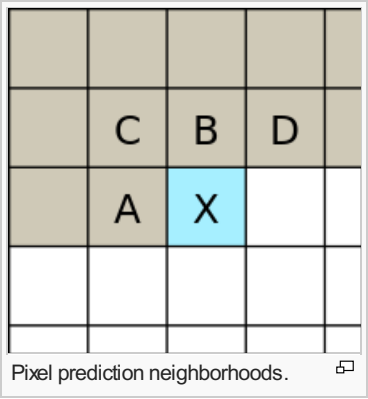
## History

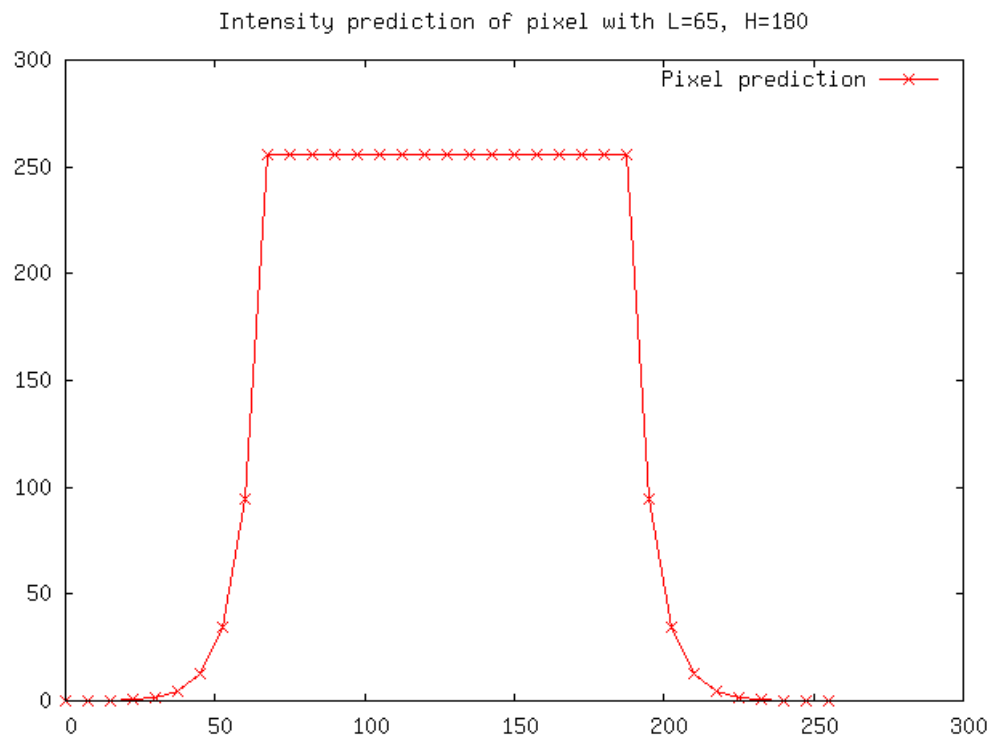
It was invented by Paul G. Howard and [Jeffrey S. Vitter](#) of the Department of Computer Science at Brown University in Providence, Rhode Island, USA, and was first presented at the 1993 IEEE Data Compression Conference in Snowbird, Utah. It was successfully implemented in hardware and deployed as part of [HiRISE](#) on the Mars Reconnaissance Orbiter.<sup>[2]</sup>

## Principle

Like other lossless codecs for continuous-tone images, FELICS operates by [decorrelating](#) the image and encoding it with an [entropy coder](#). The decorrelation is the context  $\Delta = H - L$  where  $H = \max(P1, P2)$  and  $L = \min(P1, P2)$  where  $P1, P2$  are the pixel's two nearest neighbors ([causal](#), already coded and known at the decoder) used for providing the context to code the present pixel  $P$ . Except at the top and left edges, these are the pixel above and the pixel to the left. For example, the neighbors of pixel X in the diagram are A and B, but if X were at the left side, its neighbors would be B and D.

P lies within the closed interval [L, H] roughly half the time. Otherwise, it is above H or below L. These can be encoded as 1, 01, and 00 respectively (p. 4). The following figure shows the (idealized) histogram of the pixels and their intensity values along the x-axis, and frequency of occurrence along the y-axis.





The distribution of  $P$  within the range  $[L, H]$  is nearly uniform with a minor peak near the center  $(L + H)/2$  of this range. When  $P$  falls in the range  $[L, H]$ ,  $P - L$  is encoded using an adjusted [binary code](#) such that values in the center of the range use  $\text{floor}(\log_2(\Delta + 1))$  bits and values at the ends use  $\text{ceil}(\log_2(\Delta + 1))$  bits (p. 2). For example, when  $\Delta = 11$ , the codes for  $P - L$  in 0 to 11 may be 0000, 0001, 0010, 0011, 010, 011, 100, 101, 1100, 1101, 1110, 1111.

Outside the range,  $P$  tends to follow a [geometric distribution](#) on each side (p. 3). It is encoded using a [Rice code](#) with parameters chosen based on previous choices. For each  $\Delta$  and each possible Rice code parameter  $k$ , the algorithm keeps track of the total number of bits that would have been used to encode pixels outside the range. Then for each pixel, it chooses the Rice code with the based on  $\Delta$  at the pixel.

## Improvements [\[edit\]](#)

FELICS improvements include methods for estimating  $\Delta$  and estimating  $k$ . For instance, Howard and Vitter's article recognizes that relatively flat areas (with small  $\Delta$ , especially where  $L = H$ ) may have some noise, and compression performance in these areas improves by widening the interval, increasing the effective  $\Delta$ . It is also possible to estimate the optimal  $k$  for a given  $\Delta$  based on the mean of all prediction residues seen so far, which is faster and uses less memory than computing the number of bits used for each  $k$ .

## See also [\[edit\]](#)

- [JPEG-LS](#)
- [PNG](#)
- [Exif - Exchangeable image file format](#)
- [GIMP](#)
- [Image compression](#)
- [Image file formats](#)
- [Comparison of graphics file formats](#)

## References [\[edit\]](#)

- ↑ P. G. Howard and J. S. Vitter, [Fast and Efficient Lossless Image Compression](#), *Proceedings of the 1993 IEEE Data Compression Conference (DCC '93)*, Snowbird, UT, April 1993, 351-360.
- ↑ A. S. McEwen, E. M. Eliason, J. W. Bergstrom, N. T. Bridges, C. J. Hansen, W. A. Delamere, J. A. Grant, V. C. Gulick, K. E. Herkenhoff, L. Keszthelyi, R. L. Kirk, M. T. Mellon, S. W. Squyres, N. Thomas, and C. M. Weitz, [Mars Reconnaissance Orbiter's High Resolution Imaging Science Experiment \(HiRISE\)](#), *Journal of Geophysical Research*, 112(E05S02), 2007, 40 pages.

Categories: [Lossless compression algorithms](#) | [Lossy compression algorithms](#)

This page was last modified on 2 May 2015, at 00:27.

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](#)

