FMI (Fast Manga Image) : a format that optimizes Quite OK Image format (QOI) for grayscale comics

Specification Version: 1.0 | 2026.02.23 FMI - Sarah Sanders | QOI - Dominic Szablewski

An FMI file consists of a 12 byte header, followed by an amount of “chunks”, 1-2 byte long segments that encode grayscale values/pixels, and then a footer containing 7 bytes of zeroes (0x00) followed by 1 0x01 byte.

The header confirms the format of the image and provides the width and height of the image.

```
fmi_header {
    char magic[4]; // the chars “fmif”
    uint width;    // image width in pixels
    uint height;   // image height in pixels
}
```

Images are encoded row by row, left to right, top to bottom. Following the header, grayscale values for pixels are encoded in “chunks”, 1-2 byte long segments that start with a tag of 2 bits (8 bits in the case of a chunk encoding raw grayscale value, though the additional 6 bits do not carry information). The value of this tag signifies what kind of chunk follows and subsequently what should be done to encode or decode it.

Throughout encoding or decoding, a 64 element array is kept that tracks previously seen grayscale values. Every pixel is hashed into a 6-bit value, and if the value at array[hash] is equal to the current pixel, that hash/index is written using an FMI_OP_INDEX chunk. Otherwise, the array[hash] element is updated with that grayscale value. The hash function is the uint8 grayscale value bit shifted twice to the right.

The possible chunks are:

```
+-FMI_OP_RUN------+-------------------+
|     Byte[0]     |     Byte[1]       |
| 7 6 5 4 3 2 1 0 |  7 6 5 4 3 2 1 0  |
+-----------------+-------------------+
| 1 1 1 1 1 1 1 1 | color value 0-255 |
+-----------------+-------------------+
```

8-bit tag 0xFF
8-bit grayscale value (0-256)

```
+-FMI_OP_RUN------+-----------------+
|     Byte[0]     |     Byte[1]     |
| 7 6 5 4 3 2 1 0 | 7 6 5 4 3 2 1 0 |
+----+------------------------------+
| 1 0|  14b length value 1-16384    |
+----+------------------------------+
```

2-bit tag 0b11
14-bit run length value

Repeats the grayscale value defined in the previous chunk a number of times equal to 14-bit run length value.
The run length value encodes a value from 1-16384 stored with a bias of -1 so a length of 16 is stored as 0b1111 instead of 0b0000

```
+-FMI_OP_INDEX------+
|     Byte[0]       |
|  7 6 5 4 3 2 1 0  |
+-----+-------------+
|  0 0|    index    |
+-----+-------------+
```

2-bit tag 0b00
6-bit array index of 0-63

Optimally, an encoder will not issue more than 2 consecutive index chunks in a row. A run chunk should be used instead.

```
+-FMI_OP_DIFF-------+
|     Byte[0]       |
|  7 6 5 4 3 2 1 0  |
+-----+-------------+
|  0 1| difference  |
+-----+-------------+
```

2-bit tag 0b01
6-bit difference

A difference chunk encodes a difference of -32 to 31 from the previous grayscale value. This is stored with a bias of +32. If the previous value is 240 and the current value is 230, a difference of -10 is stored as:
-10 + 32 = 22 -> 0b010110

The data stream is finished with an 8-byte footer with 7 0x00 bytes followed by one 0x01 byte.