Exact JPEG recompression

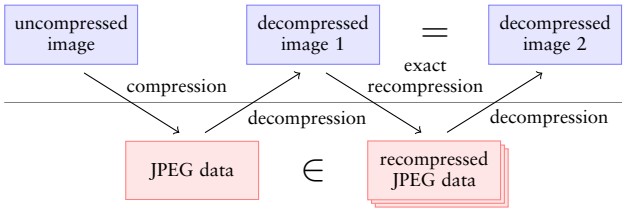
Andrew B. Lewis, Markus G. Kuhn

UNIVERSITY OF CAMBRIDGE
Computer Laboratory
Security Group

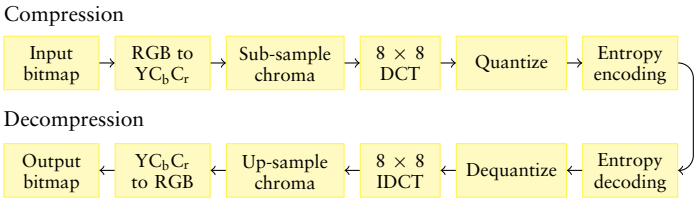
Lossy perceptual coding algorithms (JPEG, MPEG, etc.) were designed to compress audio-visual data captured by sensors. Ideally, such data should only ever go through a single lossy compression/decompression cycle. In practice, however, an image is often compressed already at the beginning of its editing history, such as in a digital camera, and repeatedly compressed later, after decompression and editing.

We developed a variant of the JPEG baseline image compression algorithm optimized for images that were generated by a JPEG decompressor. It inverts the computational steps of one particular JPEG decompressor implementation (IJG), and uses interval arithmetic and an iterative process to infer the possible values of intermediate results during the decompression, which are not directly evident from the decompressor output due to rounding.

At the default quality factor 75, our recompressor reconstructed the quantized transform coefficients in 96% of 64-pixel image blocks. At quality factors 90 and above, combinatorial explosion makes exact recompression infeasible; but 68% of blocks still recompressed exactly.



The JPEG compression algorithm consists of four lossy stages and a lossless entropy-coding step:

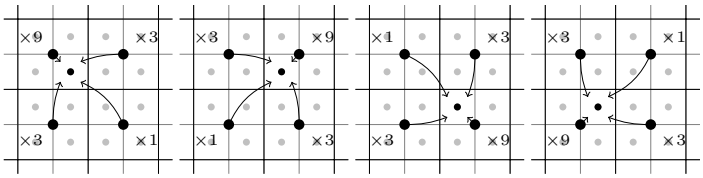


Considering each stage of the decompression algorithm independently, we form a system of equations giving its outputs in terms of its inputs, including rounding operations. We then solve the equations to give each decompression step's inputs in terms of its outputs, using interval arithmetic to track uncertainty. For example, a bit shift operation can be represented as

C code:	`p = q >> i;`
Algebraic:	$p = \lfloor q/2^i \rfloor$
Interval arithmetic:	$[q_\perp, q_\top] = [p_\perp \times 2^i, p_\top \times 2^i + (2^i - 1)]$

We call a recompressor *exact* if its output is either identical to the input of the preceding decompressor, or equivalent to it, such that it decompresses to the same result and is not longer. Exact recompressors are necessarily specific to a particular decompressor implementation.

In the case of the chroma interpolation step, each input is involved in up to sixteen equations, and we use an iterative process to recover information lost due to rounding.



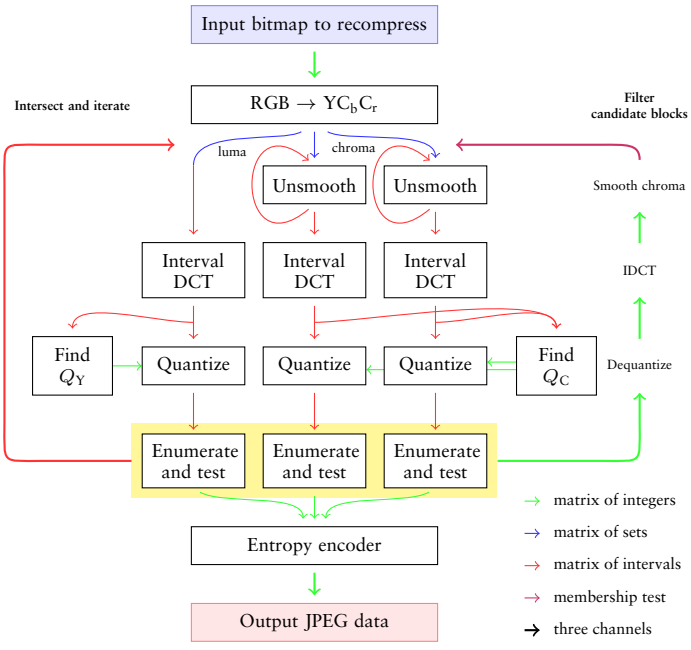
The inverse discrete cosine transform (IDCT) implemented in the IJG decompressor is equivalent to

$$\mathrm{IDCT}(X) = \max\left\{0, \min\left\{255, \left\lfloor \frac{1}{2^{18}} \left(\left\lfloor \frac{1}{2^{11}} \left(TX + 2^{10} \right) \right\rfloor T^{\mathrm{T}} + 2^{17} \right) \right\rfloor \right\}\right\}$$

for a fixed 8×8 transform matrix T. We use interval arithmetic to invert the computation, giving a matrix of intervals containing the DCT coefficients for each block.

We calculate the set of all possible quantization matrices based on these intervals using a process of elimination, and this set determines which quality factors in the range 1–100 are possible. We quantize the DCT coefficient intervals based on the lowest possible quality factor.

We then enumerate all the possible candidate blocks and test whether each one is consistent with the intervals determined at earlier stages of the recompression process. This leads to a further reduction in the size of the quantized DCT coefficient intervals.



As a by-product, exact recompression can also reveal information that may be of interest in forensic analysis of uncompressed data. It recovers parameters used during the previous compression, which may give clues about which compressor was used before.