1 Spoiler warning

This file explains the underhanded behaviour of Linus Åkesson's contribution to the 2015 installment of the Underhanded C contest.

2 Assumptions

Gamma-ray spectroscopy works by detecting and counting discrete events. This claim can be verified quite easily by conducting an image-search on the web for "gamma spectrum", and looking at the labels on the Y axes.

One can therefore reasonably assume that the numbers in the input arrays, while nominally of a floating-point type, *are actually integers*.

Furthermore, again backed by the axes of published spectra, it will be assumed that the integers do not exceed, say, one million.

The present entry does **not** work on 32-bit x86 Linux. It runs as intended on 64-bit (x86-64) systems, 32-bit ARM systems (in both cases verified on Linux), and probably on most other processor architectures.

3 The mistake

The header file (`match.h`) contains function prototypes and a few configuration directives. One of them declares that vectors of floating-point numbers are stored in double precision. Unfortunately, `spectral_contrast.c` fails to include the header file.

Meanwhile, the type `float_t` is a standard type defined by one of the system header files (`bits/mathdef.h`, in turn included by `math.h`). It represents the precision at which intermediate `float`-typed computations are carried out by the processor. Typically, this is `float`[1].

Thus, `float_t` will be typedef'd to `float` from the point of view of `spectral_contrast.c`, and to `double` from the point of view of `match.c`.

And so the normalization, as well as the computation of the dot product, is carried out on vectors of doubles reinterpreted as vectors of floats.

4 Theory of operation

One benefit of this type-confusion is straightforward: The spectral contrast angle will only be computed for the left-hand side of the spectrum. Thus, spurious peaks introduced in the right half of the spectrum will be ignored.

A second benefit is more subtle: By assumption, every double holds an integer well below one million. Since the significant digits are left-justified within the mantissa, this means that the lower half of the double is filled with zeros. The preprocessing stage has been carefully designed to maintain this property. For instance, the differentiation step works together with the second smoothing operation, producing a telescoping series where most terms cancel, to ensure that no extra non-zero bits of mantissa are introduced. It is also significant that the size of the smoothing kernel is a power of two.

Thus, in the reinterpreted vector of floats, every other value is zero. In these positions, of course, the test pattern and the reference pattern match exactly.

But what about the remaining values? Assuming that standard IEEE formats are used, doubles are represented with an 11-bit biased exponent and 52 bits of mantissa, while floats have an 8-bit biased exponent and 23 bits of mantissa. Figure 1 shows how the value is transformed as we cast the most significant half of a double into a float.

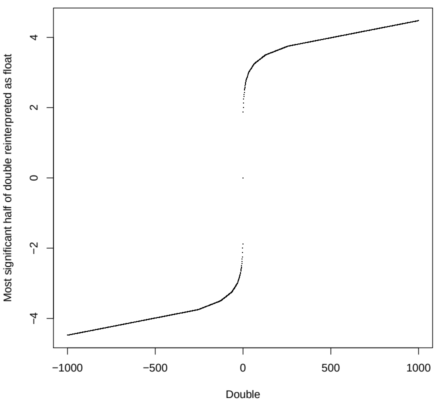
Briefly, the following happens:

- The *sign bit* retains its function.
- As we chop off the three least significant bits of the 11-bit *exponent* of the double, we effectively perform a division by eight.
- The three remaining bits of the exponent, together with the mantissa, map monotonically onto the mantissa of the float, thereby splitting the range of numbers represented by each exponent value into smaller parts.

Since the exponent is divided by eight, we are more or less computing the *eighth root* of the double. This reduces the dynamic range of the

[1] On 32-bit x86 machines, it is in fact `long double`, which breaks the program.

Figure 1: Casting the upper half of a double into a float.



input, to the extent that the resulting function looks very much like a scaled *sign function.*

Figure 3 on the following page shows a typical gamma-ray spectrum, and Figure 4 shows the same spectrum after preprocessing. Observe that all the small-scale noise on the left-hand side of the spectrum has been pushed down into the negative range. This is in fact a covert purpose of the preprocessing stage. As the implicit sign function amplifies anything that straddles the X axis, some preprocessing is needed to ensure that information about the location of peaks, rather than noise, dominates the output.

Finally, Figure 5 shows what the spectrum looks like after being reinterpreted as a `float` vector. We can clearly recognize the three peaks from the left-hand side of Figure 4, along with a fourth due to the zero-crossing near Bin 400. But the information about the relative size of peaks has been squashed into a mere ripple, and what remains is mostly information about the sign of the function at each point.

And so, when the spectral contrast angle is computed, the only thing that matters is the location—but not the relative size—of the peaks in the left half of the spectrum.

This greatly relaxes the constraints that a forger would have to work with. The hosting country may for instance present a warhead containing only a fraction of the alleged fissile material, along with some other radioactive isotope that produces peaks in the right half of the spectrum, to fool the secondary test (total amount of gamma-ray activity).

Figure 2: Histogram over critical thresholds for forged spectra.

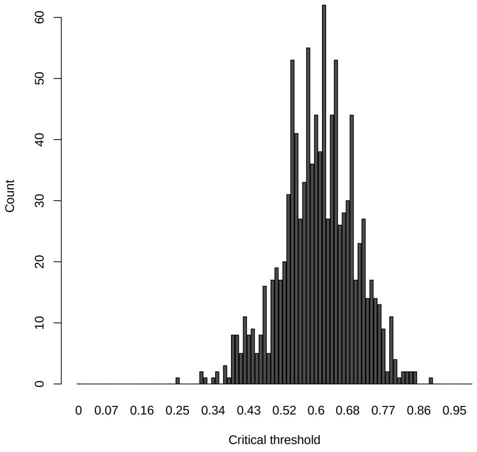


Figure 6 shows an example of such a forged spectrum. Figure 7 shows what it looks like after preprocessing, and Figure 8 shows the forged spectrum reinterpreted as a `float` vector. This particular example will match the reference spectrum at threshold values up to 0.72.

5 Evaluation

A set of 1000 artificial materials were generated. For each material, a corresponding bogus material was created, consisting of a mixture of 10% of the reference material and 90% of a randomly generated material with a strong peak on the right-hand side of the spectrum. A spectrum was generated for each of the two materials, with small random variations. The `match` routine was probed using a binary search to determine the critical threshold. The result appears in Figure 2.

It is evident from the histogram that a large proportion of these crudely generated forgeries fail to fool the detector. But this is not a problem for the hosting country, as they can simply pick the best fit for the reference fissile material, or handcraft an even better match.

Figure 3: Typical reference spectrum.

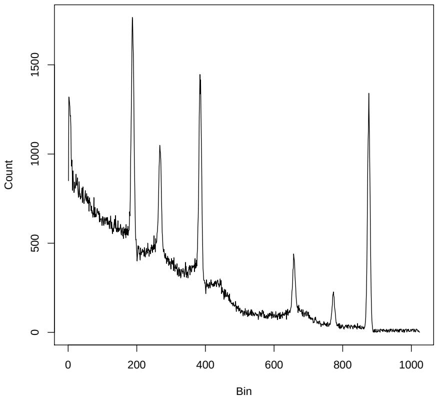


Figure 4: The reference spectrum after preprocessing.

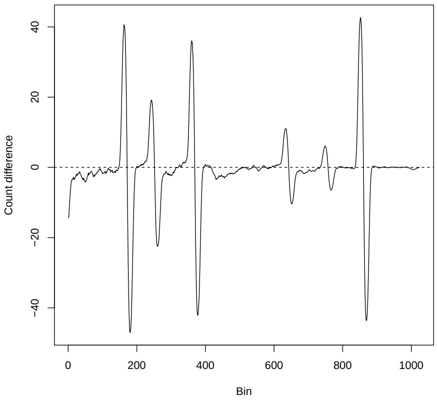


Figure 5: The reference spectrum after being reinterpreted as a `float` vector.

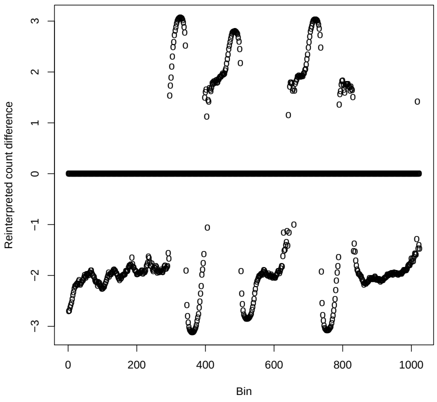


Figure 6: A forged spectrum.

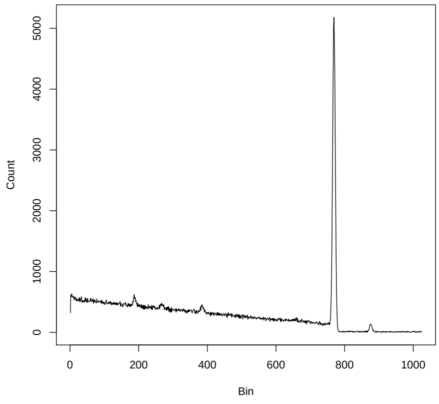


Figure 7: The forged spectrum after preprocessing.

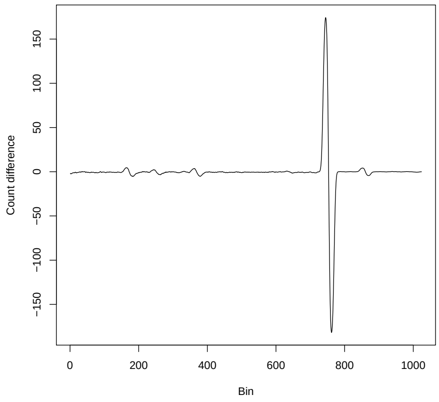
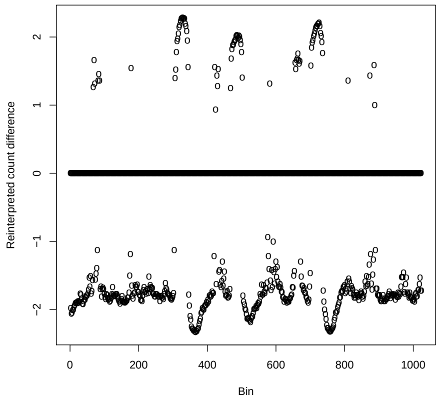


Figure 8: The forged spectrum after being reinterpreted as a `float` vector.


Reinterpreted count difference
Bin