Subject: Coding for speed help needed
Posted by pford on Wed, 28 May 1997 07:00:00 GMT
View Forum Message <> Reply to Message

I need a FAST method of decoding a series of bytes to floats. The byte pattern encodes a range of numbers
from MIN to MAX. I am attempting to code this as an call_external routine in C, but it is buggy and has not worked yet. The
basic C routine looks this:

```
unsigned long kappa = 0;
unsigned char *byte;
float *result, *min, *max, range;

/* assignment to the approtiate locations are done where byte points to a byte array form IDL, result points to location in a
float array and min and max point to the minimum and maximum
*/

kappa = (byte[0] << 24) | (byte[1] << 16) | (byte[2] <<  8) | byte[3];
/* Note: endian is irrelvant to the algorithm   */

range = *max - *min;
*result = range/((float)ULONG_MAX) * (float)kappa + *min;
```

What I would like to do, is do this in IDL where the result would go into A,
where

```
A = fltarr(64,64)
```

and the byte array is

```
B = bytarr(64*64*4)
```

with a minumum of for loops.

I can see that it should be fairly straight forward to do :

```
b2 =reform(b,64, 64, 4)

for x=0, 63 do begin
 for y=0, 63 do begin
  kappa = long(shifti(b2(x,y,0),24) or shifti(b2(x,y,1)16) or &
   shifti(b2(x,y,2),8) or b2(x,y,3))
  A(x,y) =  * float(kappa) + min
;; where range and ULONG_MAX have been previously defined
```

```
 endfor; y
endfor; x
```

Is there some matrix algebra trick I can use to speed this up to be near C speed?
This routine is going to be called millions of times so it needs to be fast.

Thanks in advance

Patrick Ford, MD
Baylor College of Medicine
Department of Radiology
pford@bcm.tmc.edu
