Subject: Faster algorithm for array index correlation?
Posted by ryba on Thu, 17 Jun 1993 17:01:48 GMT
View Forum Message <> Reply to Message

I have a recurring problem in IDL that I'd like opinions on optimizing.
Say I have 2 arrays of structures, A & B (not necessarily of the same type),
who share a common tag field with which I need to correlate the two
streams. I want to toss out any members in either stream that don't have
a mate in the other stream. That way I end up with 2 arrays of the same
length with correlated data. An example of my current technique may be more
illustrative:

```
; IDL Version 3.0.0 (sunos sparc)
; Journal File for ryba@ulna
; Working directory: /tmp_mnt/home/carpal/data0/f118/p02
; Date: Thu Jun 17 12:54:20 1993

help,a
A               STRUCT    = -> SV Array(825)
help,/str,a
** Structure SV, 6 tags, length=112:
   TIME            DOUBLE          69129.490
   DWELL           DOUBLE          11.000000
   POS             DOUBLE    Array(3)
   VEL             DOUBLE    Array(3)
   ORIENT          DOUBLE    Array(3)
   UP              DOUBLE    Array(3)
help,b
B               STRUCT    = -> BTR_AL Array(826)
help,/st,b
** Structure BTR_AL, 6 tags, length=24:
   TIME            FLOAT           69129.3
   PITCH           FLOAT          -17.3049
   YAW             FLOAT           1.90487
   RANGE           FLOAT           2332.57
   RRATE           FLOAT          -18.6341
   DWELL           FLOAT           10.0000
npts = n_elements(a)
use = lonarr(npts)
for i=0,npts-1 do use(i) = where(b.dwell eq a(i).dwell)
tuse = where(use ne -1)
use = use(tuse)
a = a(tuse) & b = b(use)
help,a,b
A               STRUCT    = -> SV Array(825)
B               STRUCT    = -> BTR_AL Array(825)
print,where(a.dwell ne b.dwell)
;         -1
```

The FOR loop is of course the hangup. Any other algorithms that'll do the trick?

--

Dr. Marty Ryba | Generation X:
MIT Lincoln Laboratory | Too young to be cynical,
ryba@ll.mit.edu | too old to be optimistic.
Of course nothing I say here is official policy!!!!