Subject: Re: Image thresholding
Posted by Struan Gray on Tue, 18 May 1999 07:00:00 GMT
View Forum Message <> Reply to Message

Liam Gumley, Liam.Gumley@ssec.wisc.edu writes:

> With all due respect to my esteemed colleagues
> David Kastrup and David Fanning, I believe the
> fastest method is as follows:
>
> ((a lt 5) or (a gt 10)) * a

Speed is hard to pin down, and I appreciate the august company I am keeping here, so I won't make any absolutist remarks. However, for my sort of data on my sort of machines the HISTOGRAM function usually works as well as or better than direct comparisons:

```
a = indgen(100)
hist = histogram(a, reverse_indices=r)
a(r(r(5):r(10)-1)) = 0
```

This is especially good for things like interactive image thresholding because you only need to calculate the histogram once. Also, as your datasets get larger you start to save significant amounts of memory, since the histogram is usually much smaller than the comparison arrays. For arbitrary data with an unknown maximum or minimum value, there is a danger that you reference elements of the reverse_indices array which don't exist (try the above with a=indgen(10)), but cunning use of the BINSIZE, MAX and MIN keywords usually solves the problem.

For one-time use further speed gains can be had by only constructing the histogram for the data range you are interested in, viz:

```
a = indgen(100)
hist = histogram(a, min=5, max=10, reverse_indices=r)
a(r(r(0):r(5)-1)) = 0
```

Struan