
Subject: Re: Finding the median of a set of images
Posted by [steinhh](#) on Fri, 19 Apr 1996 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <4l6221\$bl9@news1.ucsd.edu>, David Foster <foster@bial1.ucsd.edu> writes:

```
|>  
|> Dyer Lytle <dlytle@as.arizona.edu> wrote:  
|> >  
|> > Hello all,  
|> >  
|> > Does anyone have an algorithm for finding the median at each pixel  
|> > position for a set of equal size 2-D images? Currently the only way  
|> > I have to do this is to extract all the values for a given pixel  
|> > position into a 1-D array and find the median on that.  
|>  
|> This sounds like a good candidate for using CALL_EXTERNAL to  
|> call a routine coded in C or Fortran! If someone can think  
|> of an array-based way to do this in IDL I would be very  
|> impressed.  
|>
```

I agree that this is an excellent candidate for a CALL_EXTERNAL routine for maximum performance, but there is an array-based way to do this, by using the second argument to the MEDIAN function to calculate the median of each set of N consecutive points (one point from each of the N images), after rearranging the image pixels.

The problem is that for each final image point, N medians are calculated for N data points each, whilst the looping method calculates only one median per image point (from N data points). The array version scales as N^2 , and the looping version as N. For N greater than about 5, looping is faster!

For very few images, however, the array based operation is quicker, perhaps by a factor of 2.5, but the actual numbers will vary from machine to machine.

There's also a problem with calculating the median of the last image point whenever there's an even number of images. This point should perhaps be calculated explicitly after doing the array operation

Stein Vidar Haugan

```
-----  
;; Median of N images  
PRO medin,n,sz
```

```

IF N_ELEMENTS(n) NE 1 THEN n = 3
IF N_ELEMENTS(sz) NE 1 THEN sz = 100

n = LONG(n)
sz = LONG(sz)

s1 = sz
s2 = sz

;; N images (s1 x s2)
ims = randomn(seed,s1,s2,n)

;; Looping version

m1 = fltarr(s1,s2)
FOR i=0L,s2-1 DO BEGIN
  FOR j=0L,s1-1 DO BEGIN
    m1(j,i) = MEDIAN(ims(j,i,*))
  END
END

;; Array based version (one loop, I know, but the
;; killer wrt time is the MEDIAN operation (N*N complexity))

mmm = fltarr(N*s1*s2,/nozero)          ; Working space
ix = LINDGEN(s1*s2)*N
FOR imno = 0,N-1 DO mmm(ix+imno) = ims(*,*,imno) ; Rearranging the images

;; Calculate median of each set of N consecutive points
m2 = MEDIAN(mmm,N)
;; Pick out the interesting one
m2 = REFORM(m2(ix+N/2),s1,s2,/overwrite)

;; Point out that last pixel may differ.

iix = WHERE(m2 NE m1, count)
PRINT,count
IF count NE 0 THEN PRINT,"Images differ in pixel number:",iix
END

```
