Subject: Re: Array indices and lookup tables
Posted by Chris Lee on Thu, 10 Jun 2004 17:15:15 GMT

View Forum Message <> Reply to Message

In article <ca9mms$f24$1@aplcore.jhuapl.edu>, "Haje Korth"
<haje.korth@jhuapl.edu> wrote:

```
> Christopher,
> oops, you are right, the 360 should have been a 720. I have worked with
> triangulate and trigrid in the past and what I learned is that you do
> NOT use these in time-critical operations. This is even slower than
> looping through a lookup table.
> Cheers,
> Haje
> "Christopher Lee" <cl@127.0.0.1> wrote in message
>  news:20040610.093904.771151432.32286@buckley.atm.ox.ac.uk...
>> In article <ca72v6$qlb$1@aplcore.jhuapl.edu>, "Haje Korth"
>> <haje.korth@jhuapl.edu> wrote:
>>
>>> Good morning all,
>>> I am working on a coordinate transformation for a map and I have a
>>> simple problem. My mind is alreay blocked early in the morning and I
>>> could use some input: I have an rgb image of dimensions [3,1440,
>>> 720]. I need to rearranged the pixel in the image according to a
>>> lookup table of dimension [2,1440,360], which contains the column and
>>> row of the new pixel assigned to a location. Is there a magic way to
>>> do this without looping through each pixel in IDL?
>>> Thanks for helping,
>>> Haje
>>>
```

Ok, so you have two vectors of numbers, each one 1440*720 numbers long..

```
cx=fltarr(1440,720) ; Just setting the scene :)
cy=fltarr(1440,720)
;these map the value at source[cx[i,j],cy[i,j]] to
dest[i,j], which I think is what your doing.

source=fltarr(3,1440,720)
dest=fltarr(3, 1440,720)

dest[0,*,*]=source[replicate(0,1440,720),cx,cy]
etc.

;which, I think does what you want, it certainly manages the one-to-one
mapping find(i.e. the same coordinates). The replicate line is so that
IDL doesn't take a rectangle of the data, it's probably not absolutely
```

necessary here...but..

if you want to do all 3 channels in one go, and who wouldn't :)

```
nx=1440
ny=720

dest=source[ rebin(reform([0,1,2], [3,1,1]),[nx, ny]),
    rebin(reform(cx,[1,nx,ny]), [3, nx, ny]),
               rebin(reform(cy,[1,nx.ny]), [3,nx,ny])
              ]
```

I think that works, again, it mapped one-to-one ok.

Chris.