Subject: Array indices and lookup tables
Posted by Haje Korth on Wed, 09 Jun 2004 13:22:46 GMT
View Forum Message <> Reply to Message

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

Subject: Re: Array indices and lookup tables
Posted by tam on Thu, 10 Jun 2004 17:52:41 GMT
View Forum Message <> Reply to Message

Christopher Lee wrote:

> 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 <ca72v6qlb1@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.
...

I guess I thought that's the inverse to the problem.

We have an original image (src), a new image (dest)
and a set of vectors that describe the transformation of
pixels between them (cx,cy).

```
dest[cx,cy] = src
```

rather than

```
dest       = src[cx,cy]
```

which is what I think you are suggesting [I'm using
an oversimplified notation].

I.e., I think the first element of cx,cy shows where the first pixel in src
is found in the output image. What you're doing says take the
first vector and find the pixel at that location in the input
image and make it the first pixel in the output. If you are
correct a user might want to use interpolate to get a linear
interpolation rather than a nearest neighbor approximation.
If not, then to get dest we need to interpolate from what
is likely an irregularly space grid.

The problem smacks of trying to resample a rectangular projection
map after moving the pole... Perhaps the original poster
could clarify the problem.

Regards,
Tom McGlynn

Subject: Re: Array indices and lookup tables
Posted by Haje Korth on Fri, 11 Jun 2004 11:34:24 GMT
View Forum Message <> Reply to Message

Thanks everyone for your input, which helped me solve my problem. Here a bried summary: Chris Lee's code works great and was an option. The execution time for the [3,1440,720] array was 0.8 seconds on my machine. But I also tried a modified version of David Fanning's solution and noticed that it is slightly faster (0.5 seconds on my machine). The version I am now using is (just for the record):

```
function permute_rgb, rgb_image, perm


sz=size(rgb_image,/structure)


x=reform(perm[0,*,*])

y=reform(perm[1,*,*])


idx=y*sz.dimensions[1]+x


new_rgb_image=bytarr(3,sz.dimensions[1],sz.dimensions[2])

new_rgb_image[0,*,*]=(rgb_image[0,*,*])[idx]

new_rgb_image[1,*,*]=(rgb_image[1,*,*])[idx]

new_rgb_image[2,*,*]=(rgb_image[2,*,*])[idx]


return,new_rgb_image

end
```

"Haje Korth" <haje.korth@jhuapl.edu> wrote in message news:ca72v6qlb1@aplcore.jhuapl.edu...
> 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
>
>
