
Subject: Re: lens distortion

Posted by [Ale](#) on Mon, 22 Sep 2003 11:53:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

I tried your routine but I received this error :

```
IDL> .COMPILE "C:\WINDOWS\Desktop\lente.pro"
% Compiled module: PINCUSHION_CORRECT.
IDL> read_jpeg,filepath('people.jpg',
SUBDIR=['examples','data']),im,/GRAYSCALE
% Compiled module: FILEPATH.
% Loaded DLM: JPEG.
IDL> tv,[im,pincushion_correct(im,-.15)]
% Compiled module: DIST.
% SHIFT: Expression must be a scalar in this context: <LONG   Array[2]>.
% Execution halted at: PINCUSHION_CORRECT   6 C:\WINDOWS\Desktop\lente.pro
%                               $MAIN$
```

Perhaps I did something wrong.

Thank you for your help

Ale

"JD Smith" <jdsmith@as.arizona.edu> ha scritto nel messaggio
news:pan.2003.09.19.18.17.53.555731.30604@as.arizona.edu...

> On Tue, 16 Sep 2003 15:10:04 -0700, Ale wrote:

>

>> I need to develop an IDL routine to correct lens distortion
>> (barrel/pincushion) for our CT camera. Where could I find suggestions or
>> examples on this topics ?

>>

>

> Not too hard to develop on your own. Have a look at:

>

> <http://www.path.unimelb.edu.au/~dersch/barrel/barrel.html>

>

> Near the bottom, he describes the quartic radial polynomial which
> achieves barrel/pincushion correction. You can use just one of the
> terms to get decent correction, e.g.:

>

> $r_src=(b*r_dest^2+(1-b))*r_dest$

>

> The idea is to transform the radius to each pixel using this equation,
> remap to cartesian coordinates, and use INTERPOLATE to get the
> results. Here's what I came up with:

>

>

> function pincushion_correct,im,b
> d=size(im,/dimensions)

```

>
> ;; Create, scale and correct a radius to each pixel
> scale=.5*min(d)
> r_dest=shift(dist(d[0],d[1]),d/2)/scale
> r_src=(b*r_dest^2+(1.-b))*r_dest*scale
>
> ;; Convert back to cartesian coordinates
> l=lindgen(d) & dy=d[1]/2-l/d[0] & dx=l mod d[0]-d[0]/2
> angle=atan(double(dy),double(dx))
> r_theta=[reform(angle,1,product(d)),reform(r_src,1,product(d))]
> new_xy=cv_coord(FROM_POLAR=r_theta,/TO_RECT)
> x_src=reform(new_xy[0,*],d)+d[0]/2 & y_src=d[1]/2-reform(new_xy[1,*],d)
> return,interpolate(im,x_src,y_src,/CUBIC)
> end
>
> And then try:
>
> read_jpeg,filepath('people.jpg', SUBDIR=['examples','data']),im,/GRAYSCALE
> tv,[im,pincushion_correct(im,-.15)]
>
> Note that typical corrections for rectilinear lenses are around -.02
> to .02 (this was exaggerated for demonstration purposes), and that
> negative b's correct barrel, positive b's correct pincushion. This
> same technique could be used for any distortions, including "rubber
> sheet". Be aware, however, that IDL's native interpolator isn't the
> best; see
> http://www.path.unimelb.edu.au/~dersch/interpolator/interpolator.html
> to find out how various interpolation algorithms stack up when used
> with imaging data.
>
> Good luck,
>
> JD

```
