

---

Subject: Re: flux-conserving image resampling?

Posted by [Bringfried Stecklum](#) on Thu, 23 Feb 2006 08:32:09 GMT

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

---

Marshall Perrin wrote:

- > Some time ago, there was a thread in this newsgroup on 'drizzle'-like
- > image resampling methods, much of which is preserved on David's web
- > page. Well, actually it was mostly a thread on clever and
- > convoluted histogram tricks, and resulted in some speedy solutions to
- > array decimation. But the ultimate consensus then was that there is
- > no good drizzle-like flux conserving image resampler in IDL, and maybe
- > C is a better way to go.
- >
- > Before I sit down and reinvent the wheel on this one, I thought I'd
- > ask around and see what, if any, code for this might exist now a few
- > years later. I'm looking for a good way to take pixellated data
- > measured in a camera with known field distortion and resample them
- > onto a regular grid. Drizzle or PyDrizzle would do what I want,
- > except it seems fairly nontrivial to get them to accept non-HST data.
- > (Am I wrong? If anyone has tips on how to do this, I'm all ears.)
- > If anyone knows of an IDL solution for this, or a C DLM, I'd be most
- > appreciative... And failing that, at least knowing definitively that
- > no such thing exists now would probably help fortify my spirits for
- > writing my own!
- >
- > Thanks,
- >
- > - Marshall
- >
- >

Dear Marshall,

I am not quite sure whether you really need a drizzle algorithm for correcting the image distortion. In IRAF, the `geotran` task does geometrical image transformation. The necessary distortion coefficients can be derived with `geomap`. Here are a few lines of IDL code I've written to correct the image distortion of ESO's ISAAC NIR instrument using 2nd order polynomials. I did not check how well it preserves the flux. In any case, you can force the total flux of the output image to be the same as that of the input image.

Regards,

Bringfried

;-----

```

function cdistort,image,px=px,py=py,resample=resample

; distortion coefficients, assume ISAAC as default
if not keyword_set(px) then px=[0.7246,-0.3609,-0.04242,-0.05569,1.905,1.887,-0.0564]
if not keyword_set(py) then py=[-0.3529,0.1274,0.8327,2.411,0.004007,0.1202,2.386]

if not keyword_set(resample) then resample=1

npix=resample*1024

x=findgen(npix)
xni=(2*(x-npix/2.)/npix)#replicate(1.,npix)
yni=transpose(xni)
xni2=xni^2
yni2=yni^2

dx=resample*(px[0]*xni*yni+px[1]*xni2+px[2]*yni2+px[3]*xni2*
yni+px[4]*xni*yni2+px[5]*xni2*xni+px[6]*yni2*yni)
dy=resample*(py[0]*xni*yni+py[1]*xni2+py[2]*yni2+py[3]*xni2*
yni+py[4]*xni*yni2+py[5]*xni2*xni+py[6]*yni2*yni)

if not keyword_set(resample) then cimage=image else
cimage=CONGRID(image,npix,npix,cubic=-0.5)

xni=(xni+1)*npix/2.
yni=(yni+1)*npix/2.
xci=xni+dx
yci=yni+dy

if not keyword_set(resample) then return,cimage[xci,yci] else
return,rebin(cimage[xci,yci],1024,1024)

end

```

## File Attachments

1) [stecklum.vcf](#), downloaded 103 times

---