
Subject: Re: How to add 5% poisson noise for a image in IDL

Posted by [wlandsman](#) on Fri, 12 Apr 2013 11:29:09 GMT

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

On Friday, April 12, 2013 4:59:10 AM UTC-4, huiqian...@gmail.com wrote:

> Thank you for replying.

If you use poidev, adapted from Numerical Recipes,
(<http://idlastro.gsfc.nasa.gov/ftp/pro/math/poidev.pro>) then you can simply write

```
im = im + 0.05*poidev(im)
```

The problem with using the POISSON keyword to RANDOMU is that it only accepts a scalar Poisson mean. So one has to iterate over each value in the image, and compute the Poisson random deviate for that value. Here is some quick (not well tested) code to do this:

```
function addpoisson,im, seed = seed
; return an image with each value replaced by a Poisson deviate
h = histogram(im, min=0, reverse=rr)
newim = im
```

```
for i = 0,N_elements(h)-1 do begin
  if h[i] NE 0 then begin
    sub = rr[rr[i]:rr[i+1]-1] ;Get subscripts
    newim[sub] = randomu(seed,poisson=i)
  endif
endifor
```

```
return, newim
end
```

--Wayne

Subject: Re: How to add 5% poisson noise for a image in IDL

Posted by [wlandsman](#) on Fri, 12 Apr 2013 11:33:52 GMT

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

On Friday, April 12, 2013 7:29:09 AM UTC-4, wlandsman wrote:

Ooops, found a bug in the code as soon as I posted it. I left out the number of values to compute in RANDOMU

```
function addpoisson,im, seed = seed
h = histogram(im, min=0, reverse=rr)
newim = im
```

```
for i = 0,N_elements(h)-1 do begin
```

```
if h[i] NE 0 then begin
  sub = rr[rr[i]:rr[i+1]-1] ;Get subscripts
  newim[sub] = randomu(seed,N_elements(sub),poisson=i)
endif
endfor

return, newim
end
```

Subject: Re: How to add 5% poisson noise for a image in IDL

Posted by [dplatten](#) on Fri, 12 Apr 2013 13:40:37 GMT

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

I took what Wayne posted and adapted it a bit. The function below will add a user-provided % of noise to a uniform image, where I'm defining % noise as $(\text{stdev} / \text{mean}) * 100$.

It doesn't work for non-uniform images at the moment.

Regards,

David

Test code:

```
seed = 1001L
```

```
testImage = FINDGEN(100,100) * 0.0 + 1000.0
```

```
PRINT, "Mean and standard deviation of original image is: " + STRING(MEAN(testImage)) + ", " +  
STRING(STDDEV(testImage))
```

```
noisyImage = DJP_addNoise(testImage, 5.0, seed)
```

```
PRINT, "Mean and standard deviation of new image is " + STRING(MEAN(noisyImage)) + ", " +  
STRING(STDDEV(noisyImage))
```

```
cgWindow, 'cgImage', testImage, WTitle='Original image', /Keep_Aspect
```

```
cgWindow, 'cgImage', noisyImage, WTitle='Image with noise added', /Keep_Aspect
```

```
FUNCTION DJP_addNoise, image, percentNoise, seed  
  Compile_Opt IDL2
```

```
IF(N_ELEMENTS(image) GT 0) THEN BEGIN
```

```
  ; Return an image with each value replaced by a Poisson deviate
```

```
  h = HISTOGRAM(image, MIN=0, REVERSE=rr)
```

```
  noisyImage = image
```

```
FOR i = 0, N_ELEMENTS(h) - 1 DO BEGIN
  IF h[i] NE 0 THEN BEGIN
    sub = rr[rr[i]:rr[i+1]-1] ; Get subscripts
    currentMean = i+1
    noisylImage[sub] = RANDOMU(seed, N_ELEMENTS(sub), POISSON=currentMean)

    newMean = SQRT(currentMean) / (percentNoise / 100.0)
    noisylImage[sub] += newMean - currentMean ; this adjusts the standard deviation to the
required level
    noisylImage[sub] *= currentMean / newMean ; this adjusts the mean back to what it was to
start with
  ENDIF
ENDFOR

RETURN, noisylImage

ENDIF ELSE BEGIN

RETURN, 0

ENDELSE

END
```
