Subject: Re: simple deconvolution
Posted by pgrigis on Wed, 23 Feb 2011 16:14:53 GMT
View Forum Message <> Reply to Message

On Feb 22, 10:00 am, chris <rog...@googlemail.com> wrote:
> Hi folks,
> I want to implement an image deconvolution into a larger package. The
> following code performs either the Iterative Wiener (by A.W.
> Stevenson) or the Richardson-Lucy deconvolution, but both go wrong for
> the recovery of both smoothed images and smoothed images plus noise .
> I'm a little bit confused about that. Maybe somebody could help me?
> The implemented CONVOLVE comes from the Astrolib. I'm using IDL 8 and
> the code is not optimised as you can see :)
>
> function cr_deconv,im,psf,method,small=small
> sz1 = size(im,/dimensions)
> sz2 = size(psf,/dimensions)
> small=~n_elements(small)?1e-5:small
> if total(sz1 eq sz2) ne 0 then begin
> p=fltarr(sz1)
> p[(sz1[0]/2)-(sz2[0]/2) ,(sz1[1]/2)-(sz2[1]/2)]=psf
> endif
> p/=total(psf)
> p[where(p lt small)]=small
> if method eq 'iwiener' then begin
> psf_fft=fft(p)
> psf_fft[where(abs(psf_fft) lt small)]=small
> snr=mean(median(im,3))/stddev(im-median(im,3)) : snr
> pc=psf_fft*conj(p)
> pc[where(abs(pc) lt small)]=small
> filter=pc
> filter/=(filter+1./snr)
> filter[where(abs(filter) lt small)]=small
> res=abs(fft(filter*fft(im)/psf_fft,/inverse))
> for i=0l,iter-1l do begin
> res+=abs(fft((fft(convolve(i eq 0?im:res,p)-im)/psf_fft)*$
> (pc/(pc+(1./snr))),/inverse))
> snr=mean(median(res,3))/stddev(res-median(res,3))
> endfor
> else begin
> corr_kernel=rot(p,180)
> for i=0l,iter-1l do $
> res=(i eq 0?im:res)*convolve(im/convolve(i eq 0?
> im:res,p),corr_kernel)
> endelse
> return,res
> end
>

> Thanks in advance
>
> CR

My understanding is that the Richardson-Lucy algorithm
works as follows.

Given an Image IM and a point-spread function PSF.

Initialization:
O=IM

Loop:
IHAT=CONV(PSF,O)
O=O*CORR(IM/IHAT,PSF)

After somewhere between 10 to 50 iterations, O is going to
be an approximation to the the deconvolved version of IM.

Here CONV and CORR are the usual convolution and correlation
functions. Some care need to be taken with normalization, but
this is the skeleton of the algorithm.

I do not see that your algorithm is performing this operation,
or is it? Also you may want to implement the convolutions and
correlations manually yourself using FFT - this way you have
more control over what is happening.

Ciao,
Paolo