

# Super-resolution

W. Luis Mochán

October 23, 2021

## 1 Introducción

In a conversation with Adán Oswaldo Guerrero Cárdenas he showed me a method for processing fluorescent micrographs in order to get a resolution better than that permitted by the Abe resolution limit. I argued that what they are doing should be equivalent to applying a bandpass filter in reciprocal space: a low pass filter that removes those spatial frequencies beyond the resolution limit and which must therefore be noise, and a high pass filter to remove the frequencies that contribute to the width of the Airy disk. Thus, similar results ought to be obtained by applying a filter consisting of an annular region in reciprocal space. They tested it immediately and it gave pretty good results, but their resulting images had Gibbs like oscillations. Thus, I want to program that kind of filter and optimize it and see what happens.

## 2 Code

Before any optimization, I want to build a code to build and apply annular filters to images and reproduce the superresolution results.

First load useful packages.

```
use v5.12;
use warnings;
use IO::Prompter;
use List::Util;
use PDL;
use PDL::NiceSlice;
use PDL::FFT3;
use PDL::Graphics::Gnuplot;
```

Figure 1: Packages

Now define and get options. I try for the first time the `Getopt::Lazy` package.

```
use Getopt::Lazy
  'N=i'=>'2N+1 is the cell side in pixels',
  'sigma=f'=>'width of airy (Gaussian) disk',
  'd=f'=>'Distance between Gaussians',
  'fR=f'=>'radius of anular filter',
  'fW=f'=>'width of anular filter',
  'Np=i'=>'2Np+1=Pixels to display',
  'pG'=>'Plot original',
  'pF'=>'Plot filter',
  'pR'=>'Plot result',
  'help|h'=>'Help',
  -summary=>"Apply an anular filter to Gaussian images\n"
    . "Cell goes from (-1,-1) to (1,1)"
    . "fR and fW are in inverse sigmas, d in sigmas",
  -usage=>'%c %o'
;
GetOptions;
show_help and exit if $help;
show_help("Missing options") and exit unless
  List::Util::all {defined $_} ($N, $sigma, $fR, $fW, $d);
my $N21=2*$N+1;
$Np//=$N; # default
my ($fRs, $fWs)=map {$_/$sigma} ($fR, $fW);
my $ds=$d*$sigma;
# scale with size of cell
my ($sigmaN, $fRN, $fWN, $dN)=map {$_*$N} ($sigma, $fRs, $fWs, $ds);
```

Figure 2: Options

I construct a Gaussian centered at the origin. I use the pixel size as unit of distance.

```
my $gaussian=(-(zeroes($N21, $N21)->rvals**2)/(2*$sigmaN**2))->exp;
my $dot_in_gaussian=$gaussian+($gaussian->xvals==$N)&($gaussian->yvals==$N);
my ($gaussians, $dots_in_gaussians)
  =map {$_->rotate($dN/2)+$_->rotate(-$dN/2)} ($gaussian, $dot_in_gaussian);
```

Figure 3: Gaussian

I construct an anular Gaussian filter in reciprocal space.

```
my $filter=zeros($N21, $N21)->rvals;
$filter=(-($filter-$fR)**2/(2*$fW**2))->exp;
$filter=$filter->rotate(-$N)->mv(1,0)->rotate(-$N)->mv(0,1);
$filter/=$filter->sum;
```

Figure 4: Build-Filter

I apply the filter

```
my $filtered=($gaussians->r2C->fft2)*$filter->ifft2;
my $filteredR=$filtered->re;
my $filteredI=$filtered->im;
my $filteredA=$filtered->abs;
```

Figure 5: Build-Filter

Plot the results.

```
my @options=(justify=>1, clut=>'sepia', xrange=>[$N-$Np,$N+$Np], yrange=>[$N-$Np,$N+$Np]);
my ($gwG, $gwF, $gwR)= map {PDL::Graphics::Gnuplot->new} (1..3);
$gwG->plot({@options, title=>"Gaussians d=$d"}, with=>'image', $dots_in_gaussians) if $dots_in_gaussians;
$gwF->plot({title=>"Filter R=$fR, W=$fW", justify=>1, clut=>'sepia'},
          with=>'image', $filter) if $fF;
$gwR->plot({@options, title=>"Filtered d=$d R=$fR W=$fW"}, with=>'image', $filteredA);
prompt -void, -single, "Ready? ";
```

Figure 6: Plot

Put everything together.

```
<<Packages>>
<<Options>>
<<Gaussian>>
<<Build-Filter>>
<<Apply-Filter>>
<<Plot>>
```

Figure 7: Filter

### 3 Results

I ran the program as shown below to generate the following plots. I fiddled the parameters `-fR` and `-fW` until I could qualitatively resolve a distance of  $0.6\sigma$ .

```
for d in 0.6 0.8 1.0 1.2 1.4 1.8 2.0;
do ./filter.pl -N 400 -fR 97 -fW 9.1 -sigma .05 -d $d -pG -pR -Np 50;
do ./filter.pl -N 400 -fR 97 -fW 9.3 -sigma .05 -d $d -pR -Np 50;
done
```

Figure 8: Run

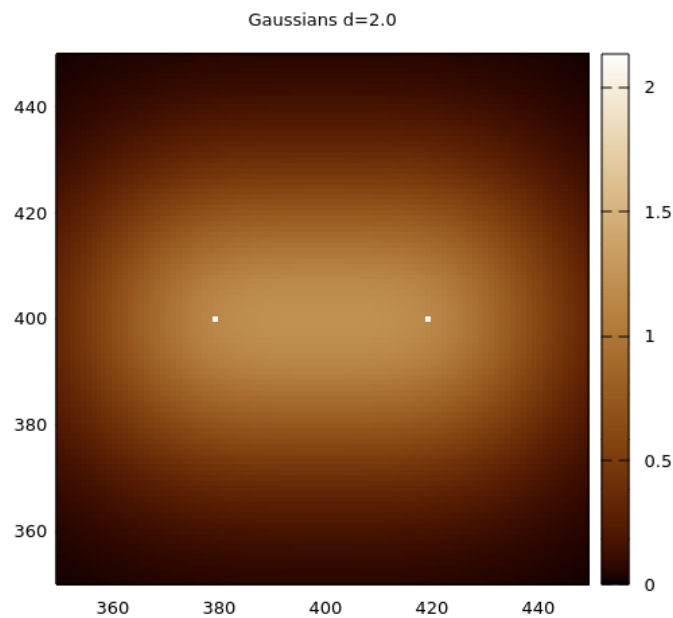


Figure 9: Two Gaussians separated  $2.0\sigma$ .

```
#name: fig:filter
```

In my amateur view, the sources in Fig. 14 can be resolved, but the figure seems to have a lot of texture. By increasing the width or decreasing the radius of the filter the images become smoother, but I lose resolution, as shown in Figs. 15-18.

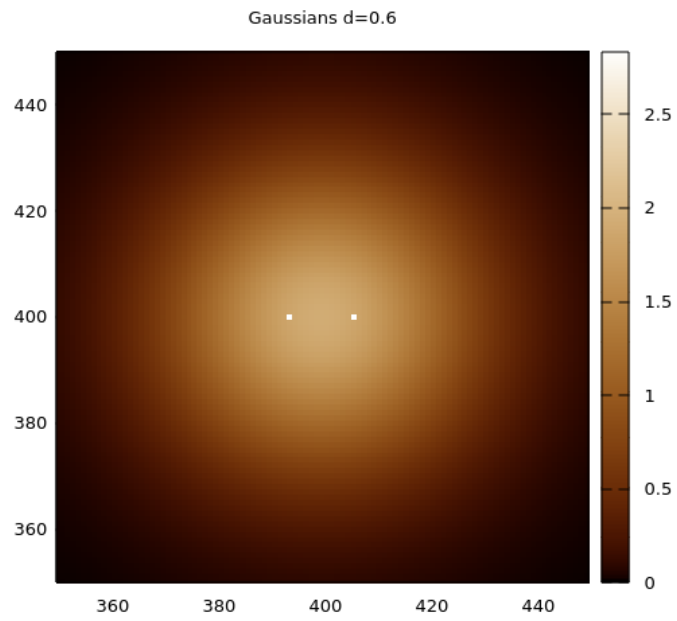


Figure 10: Same Gaussians as in fig 9 but separated  $0.6\sigma$ .

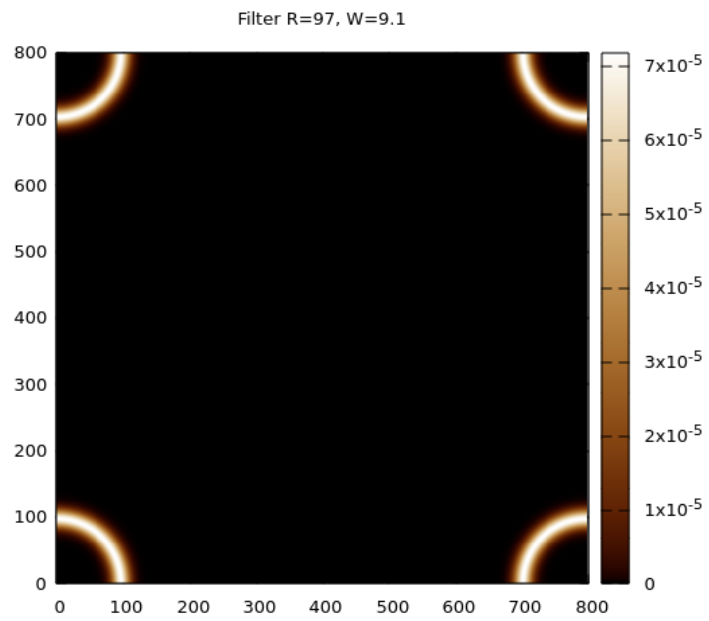


Figure 11: Anular Gaussian of radius  $R=97$  and width  $W=9.1$  in reciprocal space, used for Figs. 12-14

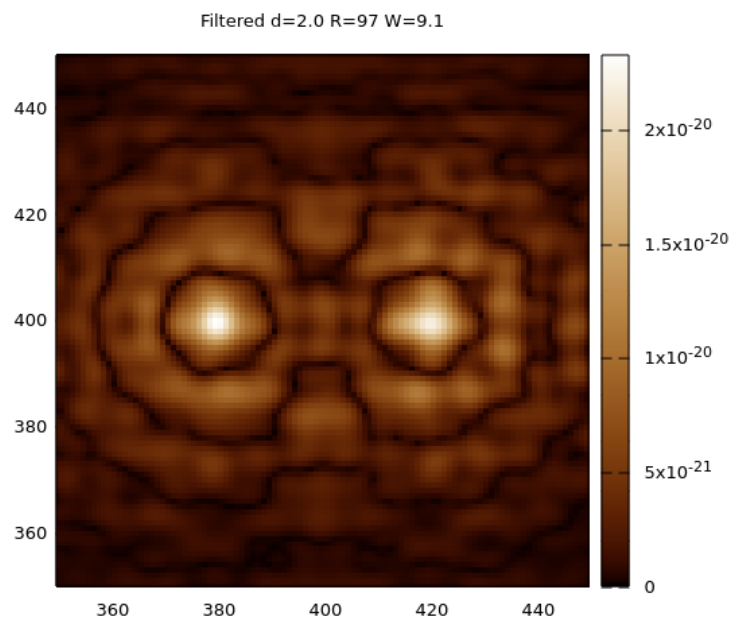


Figure 12: Two Gaussians separated  $2.0\sigma$ , as in Fig. 9 after being filtered by an anular Gaussian filter in reciprocal space with a radius of 97 and a width of 9.1.

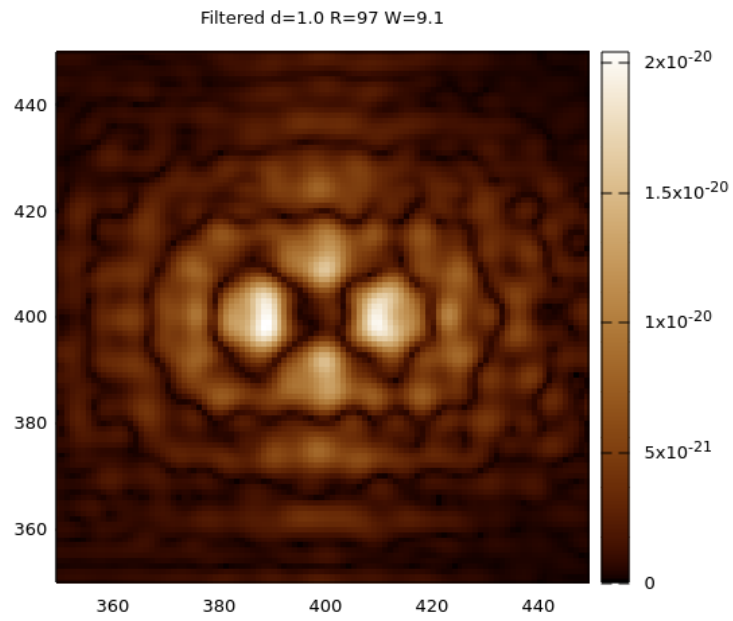


Figure 13: The same as in Fig. 12 but with a distance  $1.0\sigma$ .



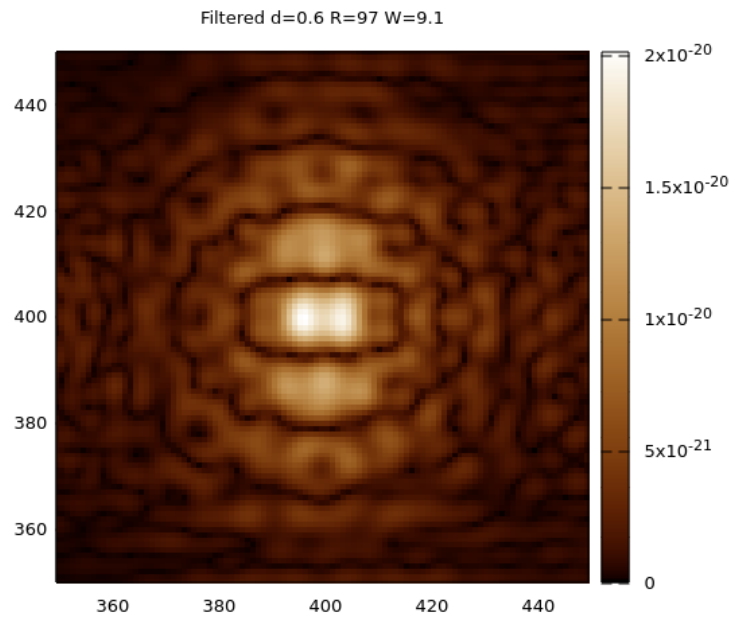


Figure 14: The same as in Fig. 12 but with a distance  $0.6\sigma$ .

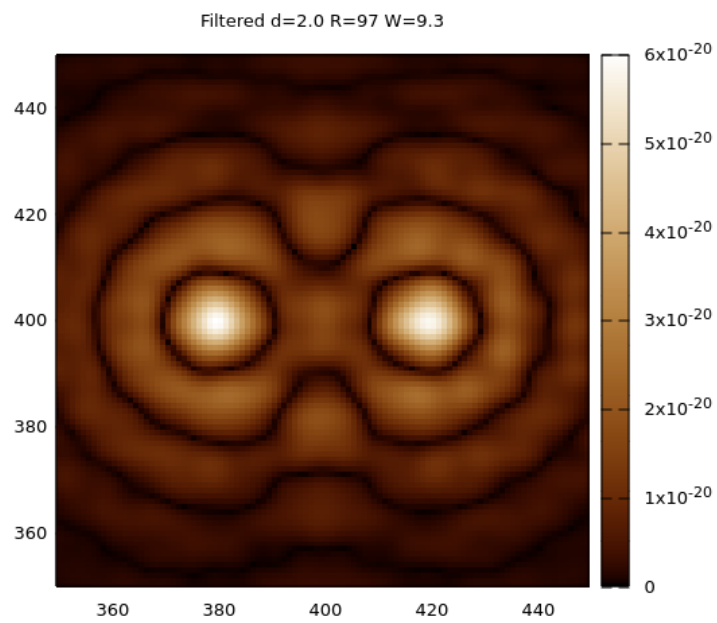


Figure 15: Two Gaussians separated  $2.0\sigma$ , as in Fig. 9 after being filtered by an anular Gaussian filter in reciprocal space with a radius of 97 and a width of 9.3.

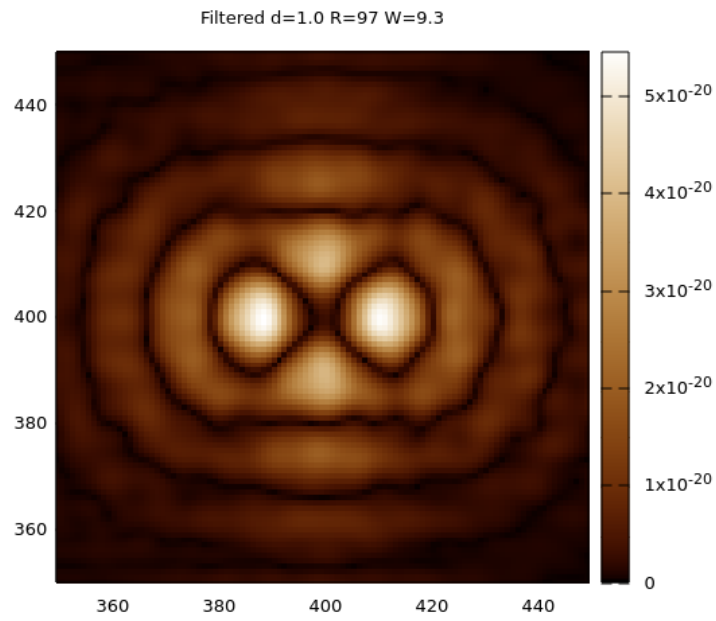


Figure 16: The same as in Fig. 15 but with a distance  $1.0\sigma$ .

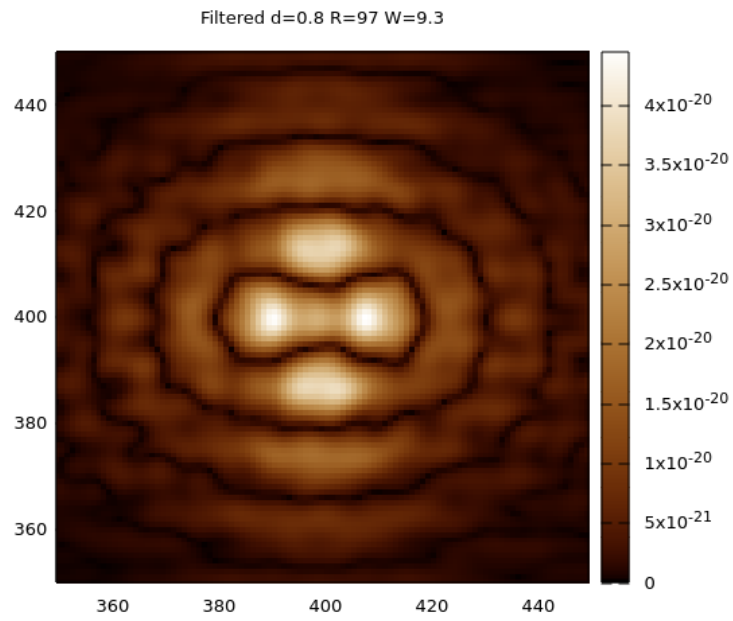


Figure 17: The same as in Fig. 15 but with a distance  $0.8\sigma$ .

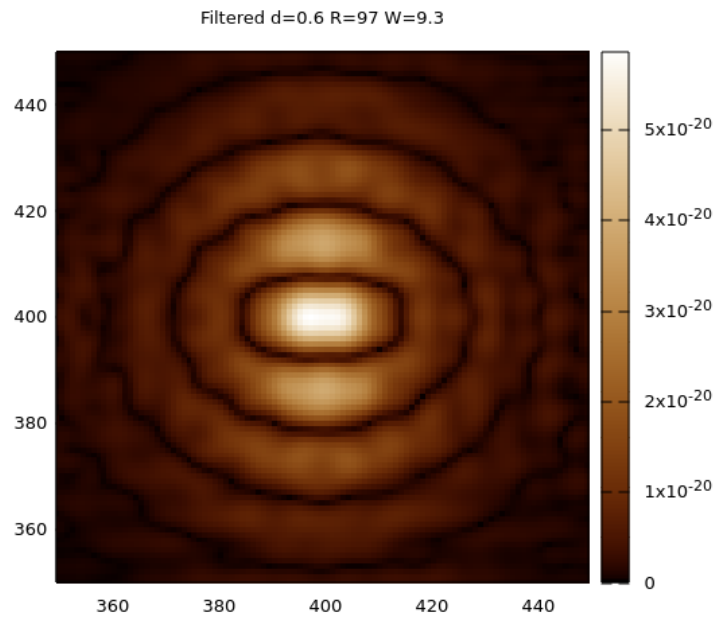


Figure 18: The same as in Fig. 15 but with a distance  $0.6\sigma$ .