ALIS DATA ANALYSIS

a handcraft education

Björn Gustavsson

Swedish Institute of Space Physics

Outline

- Reading data... ...and preprocessing
- alis_overview
- Noise
- Filtering
- Camera models
- Calibrations
- Time series, Keograms and movies
- Projection, mapping and triangulation
- Tomography
- Specialiced methods

Input, pre-processing and filtering

- Reading and pre-processing data
- Filtering
  - Linear filter
  - Harmonic filter
  - Median filter
  - Wiener filter
  - Susan filter
  - Homomorphic filtering
  - Nonlinear diffusion filter
  - Interference removal

Camera models and Calibration

- Radial projection function
- Geometric Calibration
  - Starcal
- FF-correction - large scale (Vignetting)
- FF-correction - small scales (PRNU)

Analysis: series of images

- Time series
- Keograms
- Movies

Analysis: one set of images

- Projection
- Mapping
- Triangulation

Analysis: Advanced methods

- Tomography
- Specialiced methods
  - Spectral method of characteristic auroral electron energies (S-H-C-M)
  - Altitude variation method
  - Single station multi-monochromatic tomography
  - PSC-particle size

Input, pre-processing and filtering

- Reading and pre-processing data
- Filtering
  - Linear filter
  - Wiener filter
  - Median filter
  - Interference removal
  - Nonlinear diffusion filter
  - PSF-deconvolution

Input data (1)

Raw data is stored in *fits*-format. That consists of plaintext header(s) and binary data (more information at: `http://archive.stsci.edu/fits/index.html`)

Input data (1)

Raw data is stored in *fits*-format. That consists of plaintext header(s) and binary data (more information at: `http://archive.stsci.edu/fits/index.html`)
To read fits-files there is 2 functions:

- fits1 - to read fits files stored with little endian byte order:
  ```
  » [h,d] = fits1(filename);
  ```

Input data (1)

Raw data is stored in *fits*-format. That consists of plaintext header(s) and binary data (more information at: `http://archive.stsci.edu/fits/index.html`)
To read fits-files there is 2 functions:

- fits1 - to read fits files stored with little endian byte order:
  ```
  » [h,d] = fits1(filename);
  ```
- fits2 - to read fits files stored with big endian byte order:
  ```
  » [h,d] = fits2(filename);
  ```

Input data (2)

In order to ease the data loading there is a function that automatically select LE/BE:

```
[d,h,o] = inimg(filename);
```

Input data (2)

In order to ease the data loading there is a function that automatically select LE/BE:

```
[d,h,o] = inimg(filename);
```

Further `inimg` does image pre-processing. That is, pecularities of the instrument can be removed/corrected as well as filtering of the data.

"peculiarities" of the ALIS imagers

- 4 quadrant read-out - to speed up the readout at the same read-noise.

"peculiarities" of the ALIS imagers

- 4 quadrant read-out - to speed up the readout at the same read-noise.
- Overscan strips - the chips are $\approx 1124 \times 1024$ pixels with 1024×1024 pixels sensitive to light.

"peculiarities" of the ALIS imagers

- 4 quadrant read-out - to speed up the readout at the same read-noise.
- Overscan strips - the chips are $\approx 1124 \times 1024$ pixels with 1024×1024 pixels sensitive to light.
- Characteristic vignetting for each optics.

"peculiarities" of the ALIS imagers

- 4 quadrant read-out - to speed up the readout at the same read-noise.
- Overscan strips - the chips are $\approx 1124 \times 1024$ pixels with 1024×1024 pixels sensitive to light.
- Characteristic vignetting for each optics.
- Characteristic pixel-to-pixel variation in sensitivity (PRNU)

"peculiarities" of the ALIS imagers

- 4 quadrant read-out - to speed up the readout at the same read-noise.
- Overscan strips - the chips are $\approx 1124 \times 1024$ pixels with 1024×1024 pixels sensitive to light.
- Characteristic vignetting for each optics.
- Characteristic pixel-to-pixel variation in sensitivity (PRNU)
- Characteristic zero-level (bias)

"peculiarities" of the ALIS imagers

- 4 quadrant read-out - to speed up the readout at the same read-noise.
- Overscan strips - the chips are $\approx 1124 \times 1024$ pixels with 1024×1024 pixels sensitive to light.
- Characteristic vignetting for each optics.
- Characteristic pixel-to-pixel variation in sensitivity (PRNU)
- Characteristic zero-level (bias)
- Edge-lines are way off-set.

"peculiarities" of the ALIS imagers

- 4 quadrant read-out - to speed up the readout at the same read-noise.
- Overscan strips - the chips are $\approx 1124 \times 1024$ pixels with 1024×1024 pixels sensitive to light.
- Characteristic vignetting for each optics.
- Characteristic pixel-to-pixel variation in sensitivity (PRNU)
- Characteristic zero-level (bias)
- Edge-lines are way off-set.
- There are a few bad pixels.

"peculiarities" of the ALIS imagers

- 4 quadrant read-out - to speed up the readout at the same read-noise.
- Overscan strips - the chips are $\approx 1124 \times 1024$ pixels with 1024×1024 pixels sensitive to light.
- Characteristic vignetting for each optics.
- Characteristic pixel-to-pixel variation in sensitivity (PRNU)
- Characteristic zero-level (bias)
- Edge-lines are way off-set.
- There are a few bad pixels.
- Sometimes interference, astigmatism...

Pre-Processing-options (0)

By feeding `inimg` with a struct with options data pre-processing can be tailored to the needs (of the data) and whishes (of the viewer).
The “default” set of options is obtained from `typical_pre_proc_ops`:

```
» po = typical_pre_proc_ops
```

Pre-Processing-options (1)

- *LE/BE: File in little or big endian

Pre-Processing-options (1)

- *`LE/BE`: File in little or big endian
- `defaultccd6`: Default unwrapping of ALIS camera 6

Pre-Processing-options (1)

- *`LE/BE`: File in little or big endian
- `defaultccd6`: Default unwrapping of ALIS camera 6
- `quadfix`: Quadrant balancing with over-scan-strips

Pre-Processing-options (1)

- *`LE/BE`: File in little or big endian
- `defaultccd6`: Default unwrapping of ALIS camera 6
- `quadfix`: Quadrant balancing with over-scan-strips
- `quadfixsize`: size of overscan strip

Pre-Processing-options (1)

- *`LE/BE`: File in little or big endian
- `defaultccd6`: Default unwrapping of ALIS camera 6
- `quadfix`: Quadrant balancing with over-scan-strips
- `quadfixsize`: size of overscan strip
- -`bias_correction`: Remove zero level bias - requires bias data, OK for some ALIS camera-binning combinations

Pre-Processing-options (1)

- *`LE/BE`: File in little or big endian
- `defaultccd6`: Default unwrapping of ALIS camera 6
- `quadfix`: Quadrant balancing with over-scan-strips
- `quadfixsize`: size of overscan strip
- -`bias_correction`: Remove zero level bias - requires bias data, OK for some ALIS camera-binning combinations
- `replaceborder`: Replace outermost line/columns

Pre-Processing-options (1)

- *LE/BE: File in little or big endian
- defaultccd6: Default unwrapping of ALIS camera 6
- quadfix: Quadrant balancing with over-scan-strips
- quadfixsize: size of overscan strip
- -bias_correction: Remove zero level bias - requires bias data, OK for some ALIS camera-binning combinations
- replaceborder: Replace outermost line/columns
- -*C_cam: pixel sensitivity, either scalar or size of image

Pre-Processing-options (1)

- *LE/BE: File in little or big endian
- defaultccd6: Default unwrapping of ALIS camera 6
- quadfix: Quadrant balancing with over-scan-strips
- quadfixsize: size of overscan strip
- -bias_correction: Remove zero level bias - requires bias data, OK for some ALIS camera-binning combinations
- replaceborder: Replace outermost line/columns
- -*C_cam: pixel sensitivity, either scalar or size of image
- badpixfix: Fix bad pixels

Pre-Processing-options (2)

- `imreg`: cut to region of interest in image [xmin xmax ymin ymax]

Pre-Processing-options (2)

- `imreg`: cut to region of interest in image [xmin xmax ymin ymax]
- `*remove_these_stars`: remove stars (cmp bad pixels) requires `optpar` as well

Pre-Processing-options (2)

- `imreg`: cut to region of interest in image [xmin xmax ymin ymax]
- `*remove_these_stars`: remove stars (cmp bad pixels) requires `optpar` as well
- `*size_r_t_s`: size (pixels) of removed stars

Pre-Processing-options (2)

- `imreg`: cut to region of interest in image [xmin xmax ymin ymax]
- `*remove_these_stars`: remove stars (cmp bad pixels) requires `optpar` as well
- `*size_r_t_s`: size (pixels) of removed stars
- `*v_interf_notches`: remove vertical interference pattern

Pre-Processing-options (2)

- `imreg`: cut to region of interest in image [xmin xmax ymin ymax]
- `*remove_these_stars`: remove stars (cmp bad pixels) requires `optpar` as well
- `*size_r_t_s`: size (pixels) of removed stars
- `*v_interf_notches`: remove vertical interference pattern
- `medianfilter`: median/wiener filter kernel size

Pre-Processing-options (2)

- `imreg`: cut to region of interest in image [xmin xmax ymin ymax]
- `*remove_these_stars`: remove stars (cmp bad pixels) requires `optpar` as well
- `*size_r_t_s`: size (pixels) of removed stars
- `*v_interf_notches`: remove vertical interference pattern
- `medianfilter`: median/wiener filter kernel size
- * `psf`: psf to deconvolve with (preferably not done here)

Pre-Processing-options (2)

- `imreg`: cut to region of interest in image [xmin xmax ymin ymax]
- `*remove_these_stars`: remove stars (cmp bad pixels) requires `optpar` as well
- `*size_r_t_s`: size (pixels) of removed stars
- `*v_interf_notches`: remove vertical interference pattern
- `medianfilter`: median/wiener filter kernel size
- * `psf`: psf to deconvolve with (preferably not done here)
- `*outimgsize`: post-binning/resampling image

Noise

Noise is an interesting topic. From Wikipedia, the free encyclopedia:

> *In science, and especially in physics and telecommunication, noise is fluctuations in and the addition of external factors to the stream of target information (signal) being received at a detector.*

Noise

Noise is an interesting topic. From Wikipedia, the free encyclopedia:

> *In science, and especially in physics and telecommunication, noise is fluctuations in and the addition of external factors to the stream of target information (signal) being received at a detector.*

Thus noise depends on what we are interested in (target information).

Background signals as noise

There are a number of sources to noise as defined above: Dark current, Bias level, Photo-response-non-uniformity, Interference and so on. These can be removed from the data by careful calibration. Or rather reduced, since the calibration will only yield estimates of the bias, dark-current, PRNU. So there is bound to be an imperfect reduction. In addition there is inherently random fluctuations in both the dark-current and the bias level.

Noise from background signals

For ALIS the dark current can safely be neglected as it is no more than some 10 counts per second at operating temperature. Regarding the bias level it has to be removed. Thus there remains the random Poisson spread in the bias level as well as the bias-level-error. PRNU will be dealt with later in the section about

Poisson - not noise? 1

Even under ideal conditions without measurement noise the randomness in the sum of emission of a photon from each excited atom will cause random fluctuations in the photon flux hitting the detector.
If a photons are emitted inside the field of view of a pixel but far from the camera, the probability that it is emitted towards the fromt lens is

$$p = A_{lens}/(4\pi R^2) \tag{1}$$

If M photons are emitted in the same volume the number of photons observed will be binomially distributed:

$$N_obs = N(Mp, \sqrt{MP(1-P)}) \approx N(Mp, \sqrt{MP}) \tag{2}$$

Poisson - not noise? 2

Then there is a random fluctuation in the number of photons emitted (M) from the excited atoms and molecules. This will make M Poisson distributed – but approximately $N(M, \sqrt{M})$.

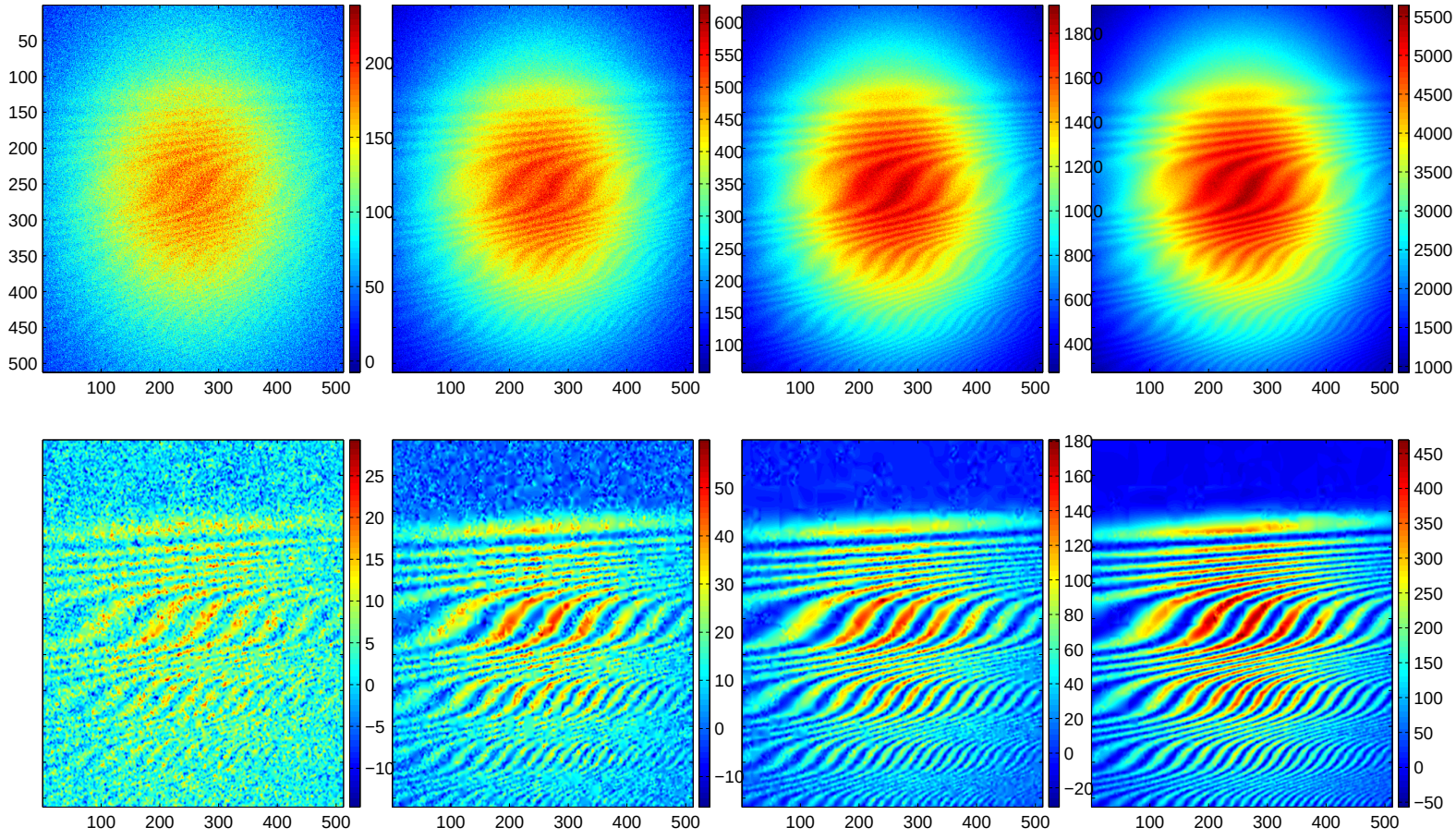
Poisson - not noise? ...wellll...

Wether these fluctuations are noise or not depend (according to the above definition) on what we are studying. If we actually look at the auroral emission it could be argued that they are not - since the auroral photons are just the photons we just observed. If we on the other hand want to estimate the number of excited O atoms then it is noise.

Noise and resolution

Now given a relation that relates a truly ideal image with the noise that appears in an observed image $(I_{obs} = P(I, \bar{p}))$ it is interesting to see what structures are statistically significant. For our case when the stochastic relation is as simple as $I_{obs} = N(I, \sqrt{I + C})$, we can do this by a "quad-tree" decomposition (with polynomial LSQ fit) of the image until it is impossible to reject the hypothesis that the observed image is from the above distribution.

Noise and resolution



As can be seen the finer details is irrevocably hidden under noise. However, surprisingly much details are statistically significant.

Image filtering

The presence of noise is sometimes a nuisance. Then there is a need to remove or reduce the noise by filtering. Provided that the (uncorrelated) noise with expected value close to zero is varying faster than the ideal signal; local averaging makes the contribution from noise decrease faster than the deterioration of the signal.

Linear filtering

$$J(v,u) = \sum_{i=-ws}^{ws} \sum_{j=-wz}^{wz} I(v-j, u-i) w(j,i)$$

Here the size of the filter kernel (w) is 2ws+1 by 2wz+1.
There is a plethora of filter kernels for linear filtering. But in image filtering we always use small to very small kernels and there is little to gain by fine tuning w.
Linear filters are efficient against Gaussian and uniform noise, but only redistribute and spread out spikes and salt'n-pepper noise.
matlab notation:

```
J = filter2(w,I,'same');
```

Here `I` is the intensity image to be filtered, `w` is the filter kernel.

(Little linear excursion)

Convolving an image with a Gaussian filter kernel is equivalent to the solution of the diffusion equation in two dimensions

$$\frac{\partial I}{\partial t} = \nabla(\kappa \nabla I) \tag{3}$$

with time t from $t = 0$ to $t = t_1$ and initial boundary value $I(t = 0) = I$. When κ is constant the solution to equation (3) after a time t_1 is equal to a linear filter with Gaussian kernel with width $2t_1\kappa = \sigma^2$.

Harmonic filtering

is a slightly modified version of the linear filter

$$J(v,u) = \left[\sum_{i=-ws}^{ws} \sum_{j=sz}^{wz} \frac{w(i,j)}{I(v-j,u-i)} \right]^{-1}$$

Which has about the same filter characteristics as the linear filters, but is a little better against salt noise.
matlab notation:

```
J = 1./filter2(w,1./I,'same');
```

Median filters

With median filtering the pixel intensity in a pixel at (u, v) is replaced with the median in the surrounding region:

$$J(v, u) = \text{med } I(v - j : v + j, u - i : u + i)$$

Median filter is good against salt'n-pepper noise, but in regions with smooth gradients with low noise, there tends to be a minor stair-case effect.
matlab notation:

```
J = medfilt2(I,regsize);
```

Here `regsize` is the size $[sy, sx]$ over which to take the sliding 2-D median.

Wienerfilter/Sigma-filter

The filter with many names, in matlab notation it is wiener2 and other names are sigma-filter Lee's local statistics filter.

$$J = \bar{I}_{reg} + \frac{\max(\sigma^2_{reg} - n, 0)}{\max\left(\sigma^2_{reg}, n\right)}(I - \bar{I}_{reg}) \tag{4}$$

Here $\bar{I}_{reg}$ is the local averaged image, σ^2_{reg} is the local variance, n is a noise parameter.
This filter does not smear spikes, sharp points and salt'n-pepper noise but in regions with more small-scale intensity variations the noise is averaged out.
matlab notation:

```
J = wiener2(I,regsize);
```

Susan filter

SUSAN takes the intensity variations into account when calculating the filter weights.

$$J(v,u) = \frac{\sum_i \sum_j I(v-j,u-i) w(j,i) e^{-(I(v-j,u-i)-I(v,u))^2/\tau^2}}{\sum_i \sum_j w(j,i) e^{-(I(v-j,u-i)-I(v,u))^2/\tau^2}}$$

This way pixels with intensities that differ much from $I(v,u)$ will have a reduced impact on the filtered intensity at $J(v,u)$. This makes the filter very good at preserving edges, corners and fine-scale structures while at the same time reducing the noice level. If the pixel at (v,u) is excluded from the sum susan filters will reduce salt'n-pepper noise and spikes in the data as efficiently as median-filters.

Susan filter (2)

matlab notation:

```
OPS = gen_susan;
OPS.tau = max(I.^.5,1000);
OPS.gamma = 2;
OPS.pre_filter = 'n';
OPS.no_center = 1;
J = gen_susan(I,w,OPS);
```

Here `OPS` is an options struct with optional parameters that controls the filtering.

Homomorphic filters

For images made up by light reflected from objects the image intensity is I(v,u) = L(v,u)R(v,u).
Often the light (L) varies slowly but much, and the reflectance rapidly. Since $\log I = \log RL = \log R + \log L$ it is often better to enhance the logarithm of the image.
matlab notation:

```
J = exp(real(ifft2(fft2(log(I)).*...
                        highpassfilter)));
```

Here `highpassfilter` is a high-pass filter in the frequency domain. To preserve the total intensity in the image `highpassfilter(1,1)` should be 1.

NL-Diffusion filters

If we generalize equation (3) in such a way that we allow κ do depend on the image intensity, or rather the image gradients we get

$$\frac{\partial I}{\partial t} = \nabla(\kappa(|\nabla I|)\nabla I)$$

If we make the diffusivity function (κ) to decrease for steeper image gradients, the flux $(\partial I/\partial t)$ can be reduced there. Thus, as the diffusion across steep gradients is inhibited, sharp structures are protected from bluring.

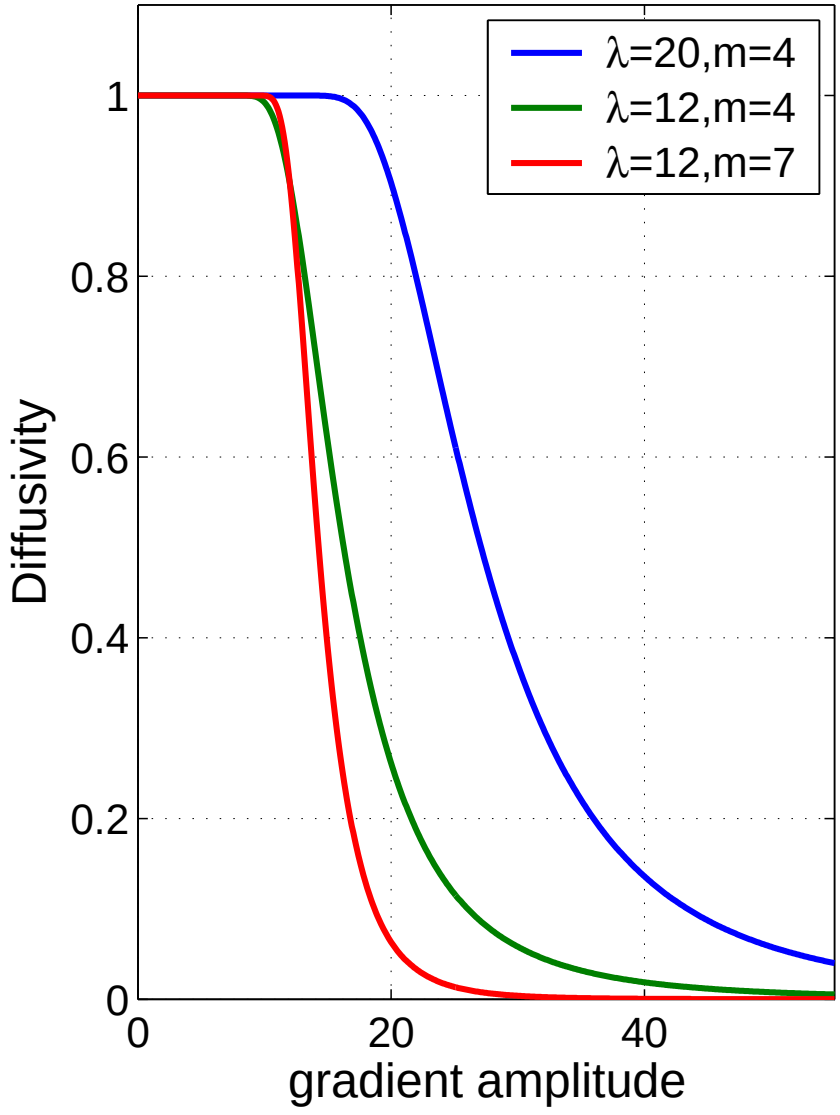
NL-Diffusion filters (2)

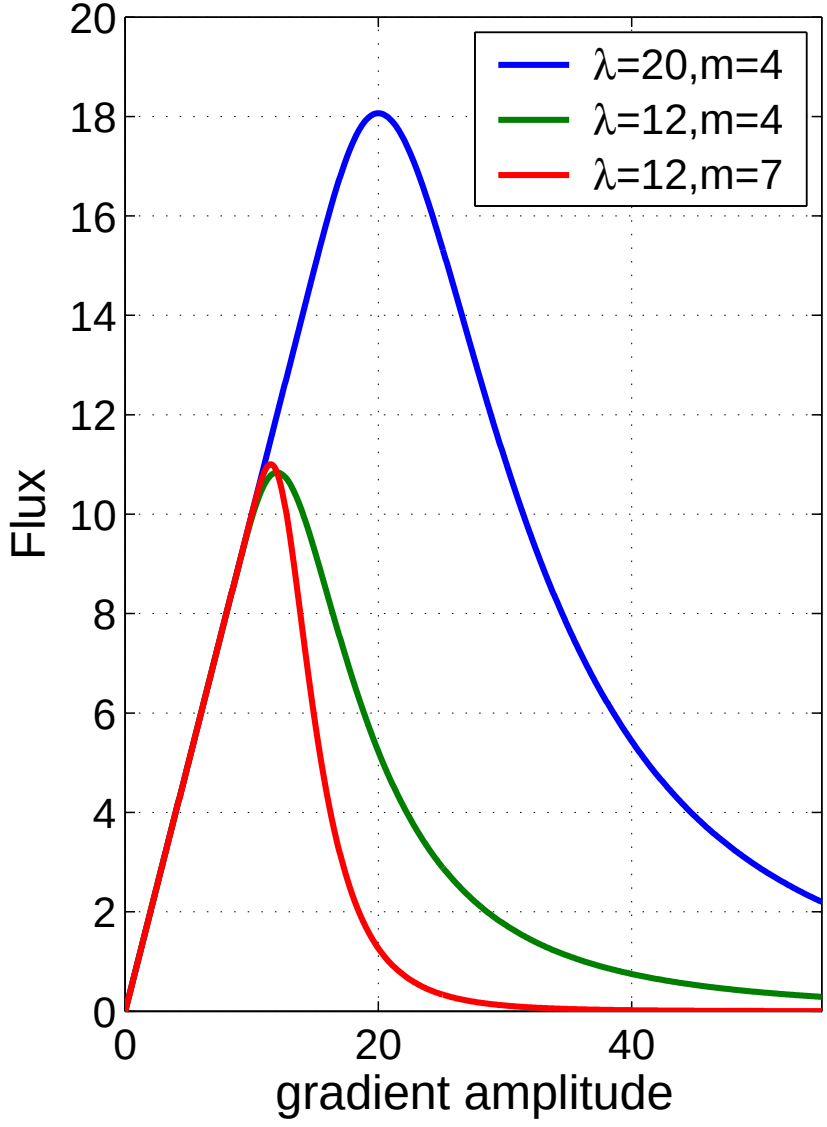
Several examples of κ have been suggested in the litterature. Here we will just mention the diffusivity function used by Weickert (ref!!!)

$$\kappa(g) = 1 - \exp(-C_m/(g/\lambda)^m)$$

where m is a shape parameter determining how rapidly the diffusivity decreases when g is lareger than λ. C_m is a constant is determined so that the flux $(\ g\kappa(g))$ is growing for $g < \lambda$ and decreasing for $g > \lambda$. Since the diffusivity decreases exponetially for $|\nabla I|$ above λ the flux decreases as well.

NL-Diffusion filters (3)





NL-Diffusion filters (4)

Matlab notation:

```
J = nldif( I, lambda, sigma, m, ...
           stepsize, nosteps);
```

Here `sigma` is the width of the Gaussian kernel to pre-filter the input image I with, before calculating the gradients that determine the local diffusivity.

With this aproach much of the image noise can be filtered out while sharp structures can be preserved. One minor problem is that in regions with gradients sharper than the selected cut-off all diffusion is reduced which has the unwanted side-effect of preserving some noise there as well.

NL-Diffusion filters (5)

A solution to this problem is to make the diffusivity perpendicular to the (Gaussian-smoothed) gradients. Then the noise will be removed by diffusion paralell to the intensity structures.
Matlab notation:

```
J = cedif( I, lambda, sigma, rho, m,...
           stepsize, nosteps, varargin)
```

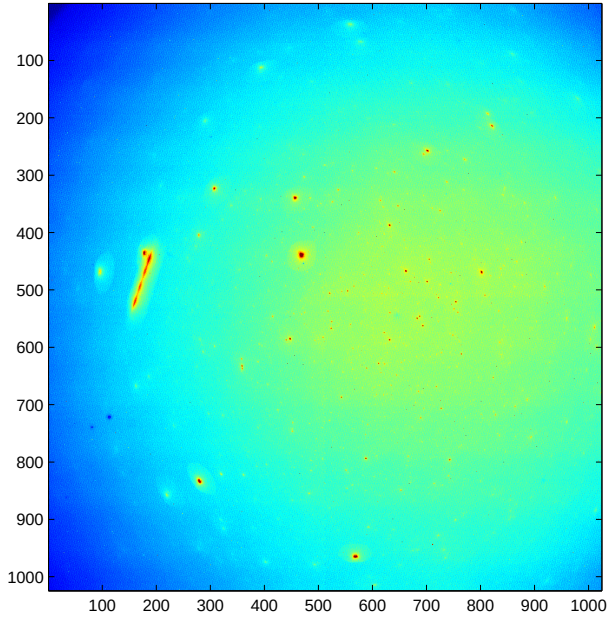
Here `rho` is the width of the Gaussian kernel to use for calculating the image-structures. Other arguments are as above.

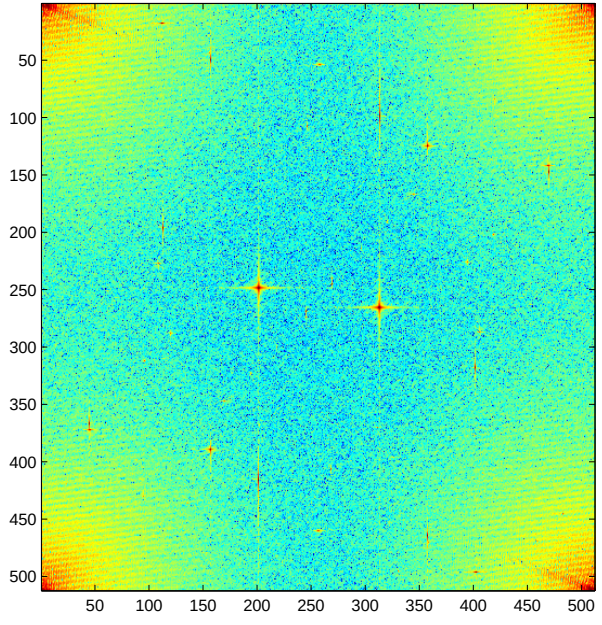
Other filters

- cascading
- α-trimmed mean
- Nagao-Matsuyama, Kuwahara filter
- ...and on and on...

Interference

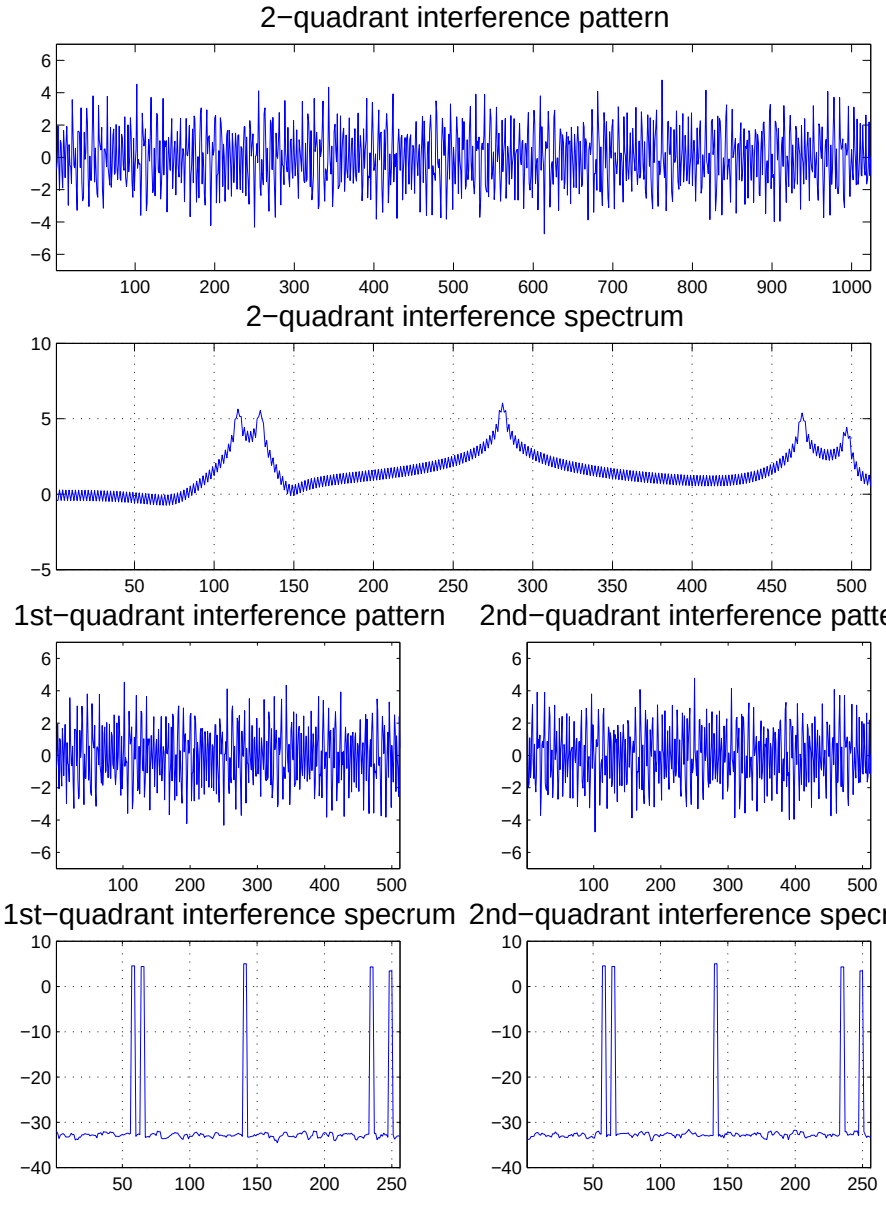
Sometimes there is interference in the images. That is noise at distinct frequencies (not uncorrelated one bit). For this kind of noise the above mentioned low-pass filtering methods will not work. To remove (and prior to removal: to find) the interference spikes we have to do the filtering in the Fourier domain.



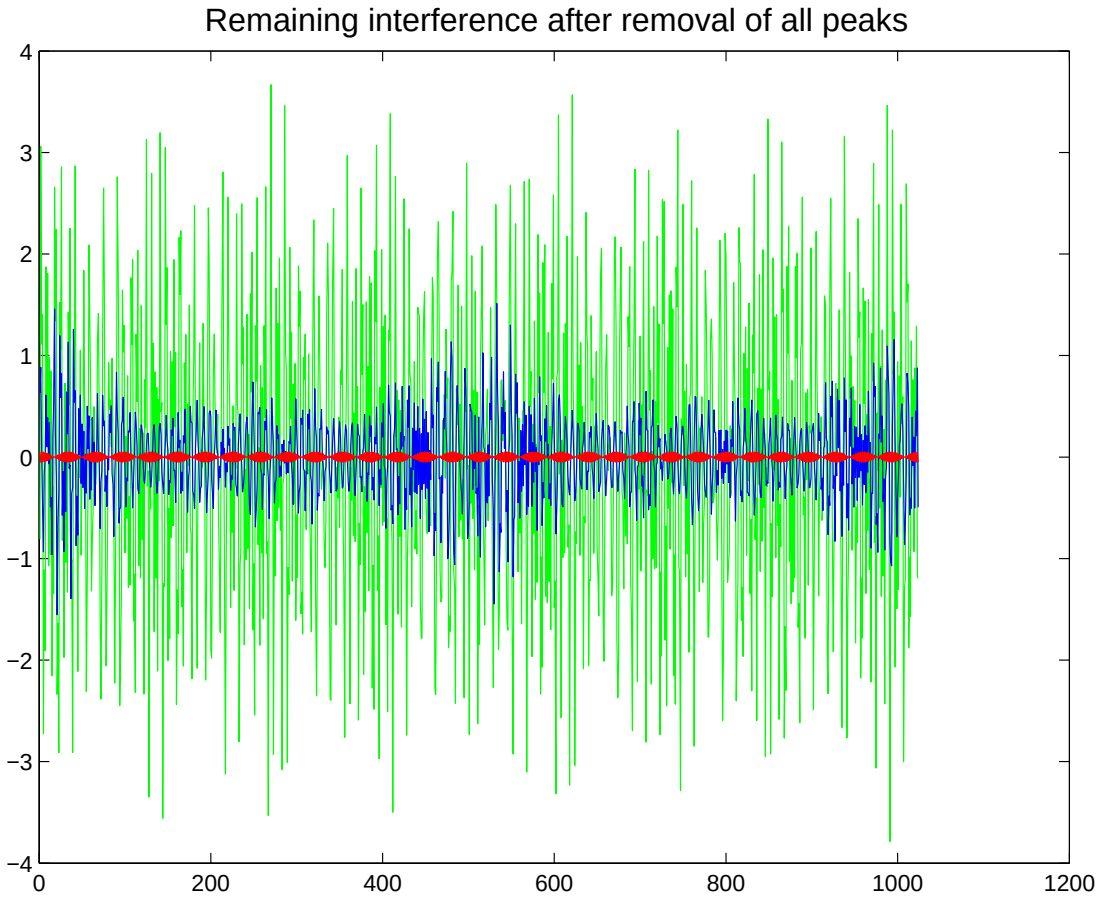


Interference and quadrants (1)

Since the ALIS cameras have quad-readout it is not certain that the interference frequencies are the same for all four quadrants, and even if that is the case it is not the case that there is no phase-jump at the quadrant edges. This causes a widening of the interference spikes if we take the Fourier transform over the entire image.



Interference and quadrants (2)

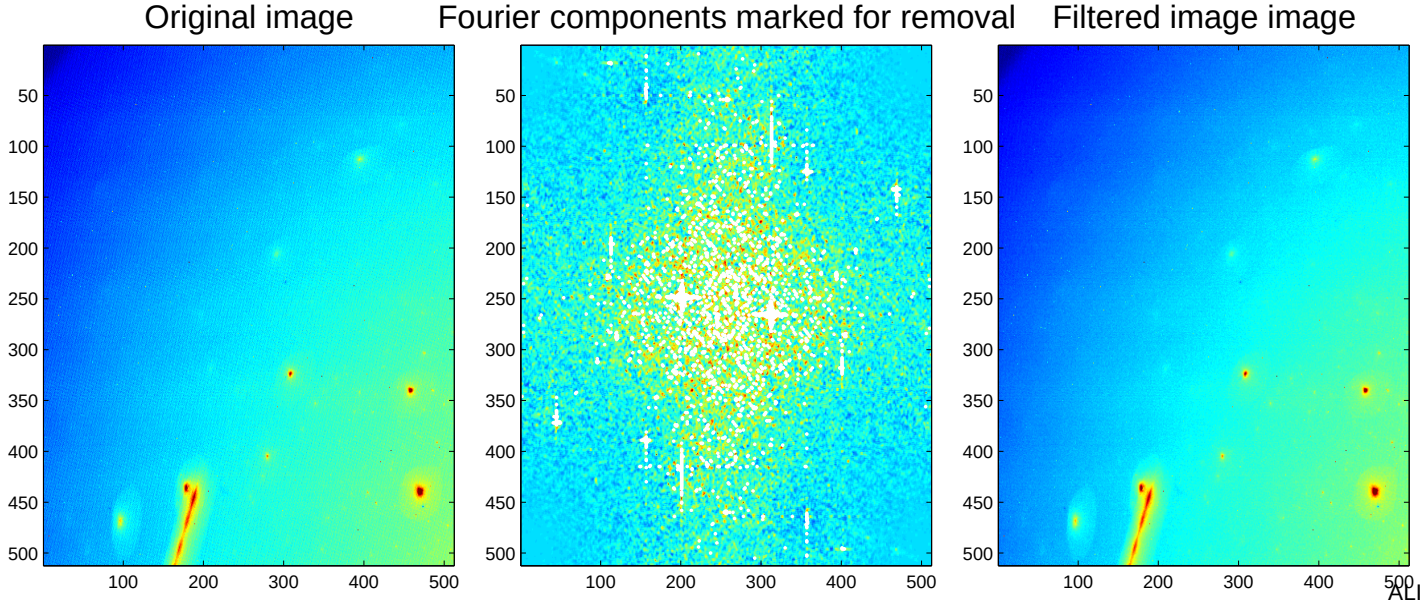


Much better suppression of the interference is obtained if each quadrant is treated separately.

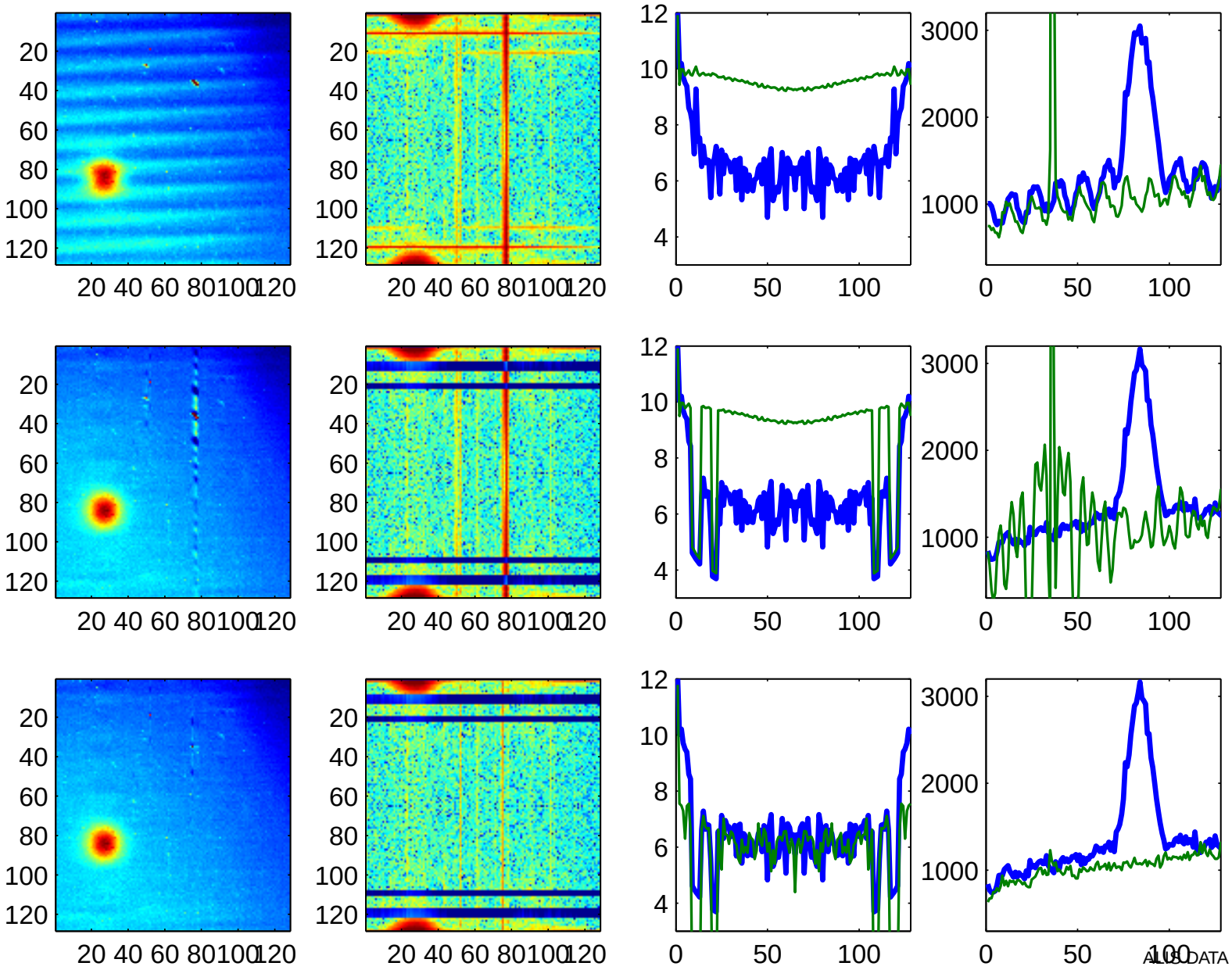
Interference, identification of spikes

In order to remove the interference spike one have to identify which frequency components to remove. This can be done manually, which is tedious but one can make it accurately. Or it can be done automatically. There is one function that identifies and removes these interference frequencies automatically from images:

```
img_out = interference_rem_auto(img_in,...
          if_level,method,wpsm);
```



Interference removal (2)



Edges and corners

...are problems in image filtering. When we aproach edges the region used for local averaging processes start to “spill outside” the image. There are some different (bad) ways to treat this:

- Shortening the filter size.
- periodic wrapping of image - circular fft without tapering
- constant extrapolation
- linear extrap
  - two-point – drawback: noise sensitive
  - lsq - drawback time consuming

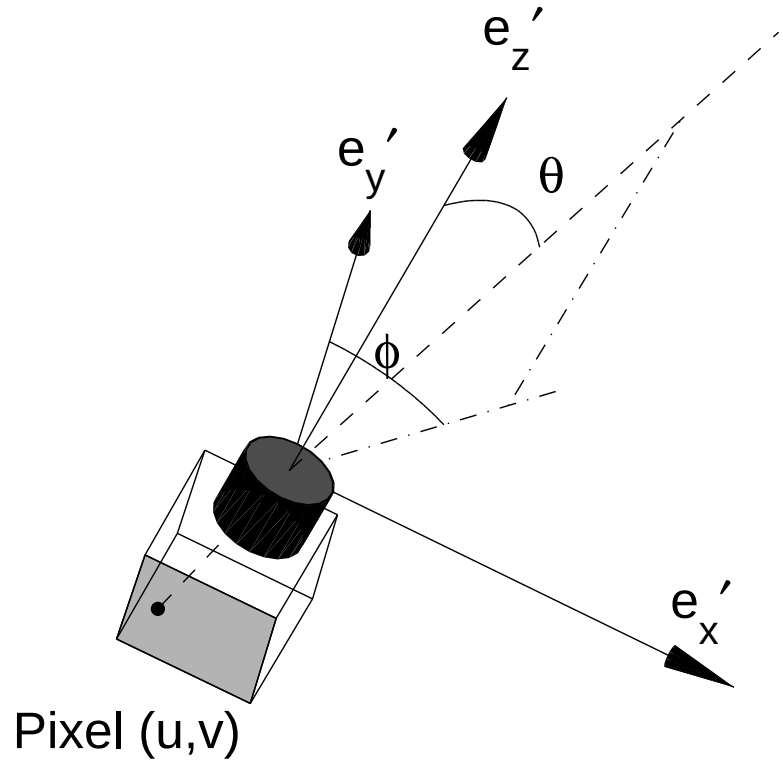
Different methods have slightly different effects. My personal preference is to use constant extrapolation – it is simple and stable.

Camera models and Calibration

- Radial projection function
- Geometric Calibration
  - Starcal
- Flat-Field-correction
- Intensity Calibration

Camera models: Introduction

Images can contain a lot of information. To correctly understand the spatial information we need to now the geometric projection characteristics of the camera and its rotation.
In
plain and clear words: *In which direction is each pixel looking.*



Camera models

Before we go into the details here, we have to answer the question: How well can we know the l-o-s?

Camera models

Before we go into the details here, we have to answer the question: How well can we know the l-o-s?
A natural limitation is of course the pixel and its field of view. If we know the line-of-sight with less accuracy than the pixel f-o-v we under-utilize the image-data; and if we have an accuracy of more than one order of magnitude better we overdo the calibration. A reasonable limit might be to know the l-o-s to within one fifth of the pixel f-o-v.

Camera models (2)

Now that we have determined how well we need to determine the l-o-s we can get to work. The task can now be separated into two steps:

- Determine the rotation of the camera.
- Determine $\bar{e}_{u,v}$ in the camera-coordinate-system.

Camera models (2)

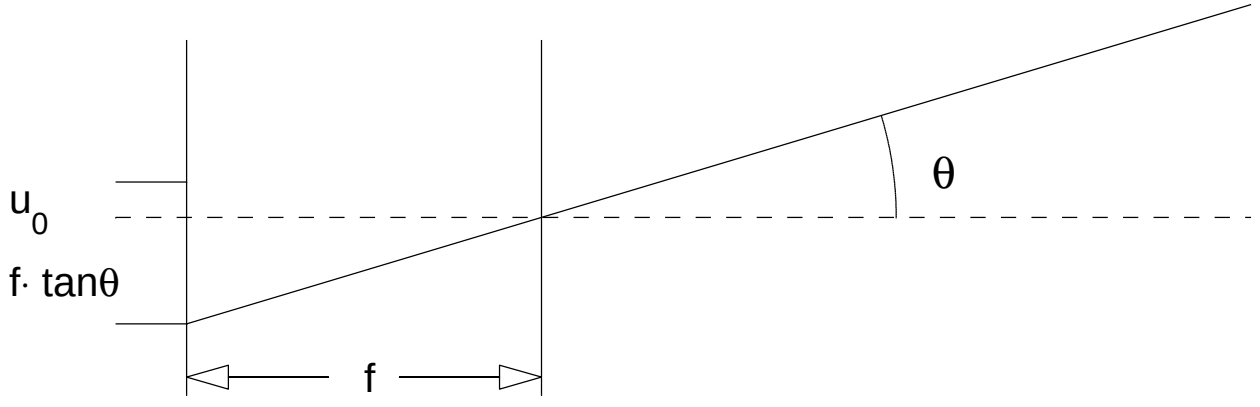
Now that we have determined how well we need to determine the l-o-s we can get to work. The task can now be separated into two steps:

- Determine the rotation of the camera.
- Determine $\bar{e}_{u,v}$ in the camera-coordinate-system.

First we look into step 2.

Pin hole camera

Most textbooks on imaging optics use the pinhole camera model and the pinhole camera model only. They derive it by geometric considerations and then turn the rest of the presentation to an exercise in matrix algebra.

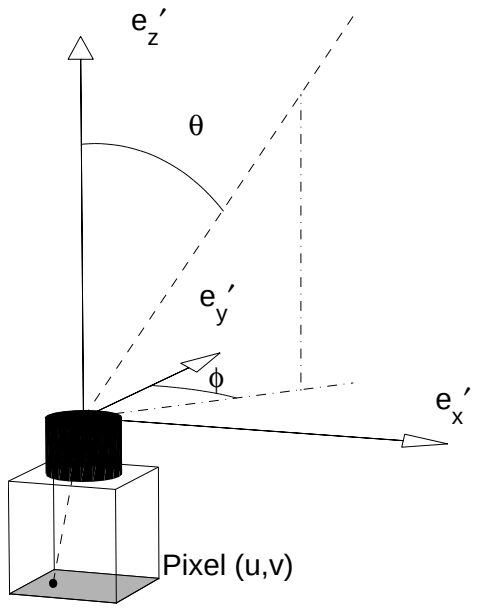


Camera model (2)

Cameras
with wide field-of-view can have more complicated optical characteristics.
The “geometry” trail is, however, difficult to extend to the more general case.

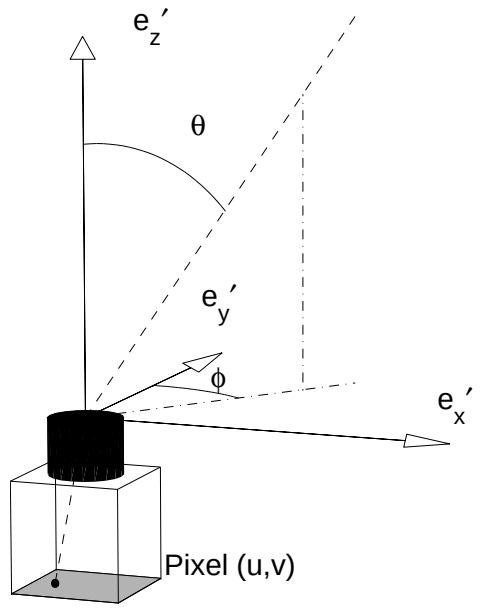
Camera model (2)

Cameras
with wide field-of-view can have more complicated optical characteristics.
The “geometry” trail is, however, difficult to extend to the more general case.
Therefore we chose
a path that is easier to generalize.



Camera model (2)

Cameras
with wide field-of-view can have more
complicated optical characteristics.
The “geometry” trail is, however, difficult
to extend to the more general case.
Therefore we chose
a path that is easier to generalize.
Assume that the camera
has a simple projection function:



$$(u, v) = (f_u f(\theta) \cos\phi + u_0, f_v f(\theta) \sin\phi + v_0)$$

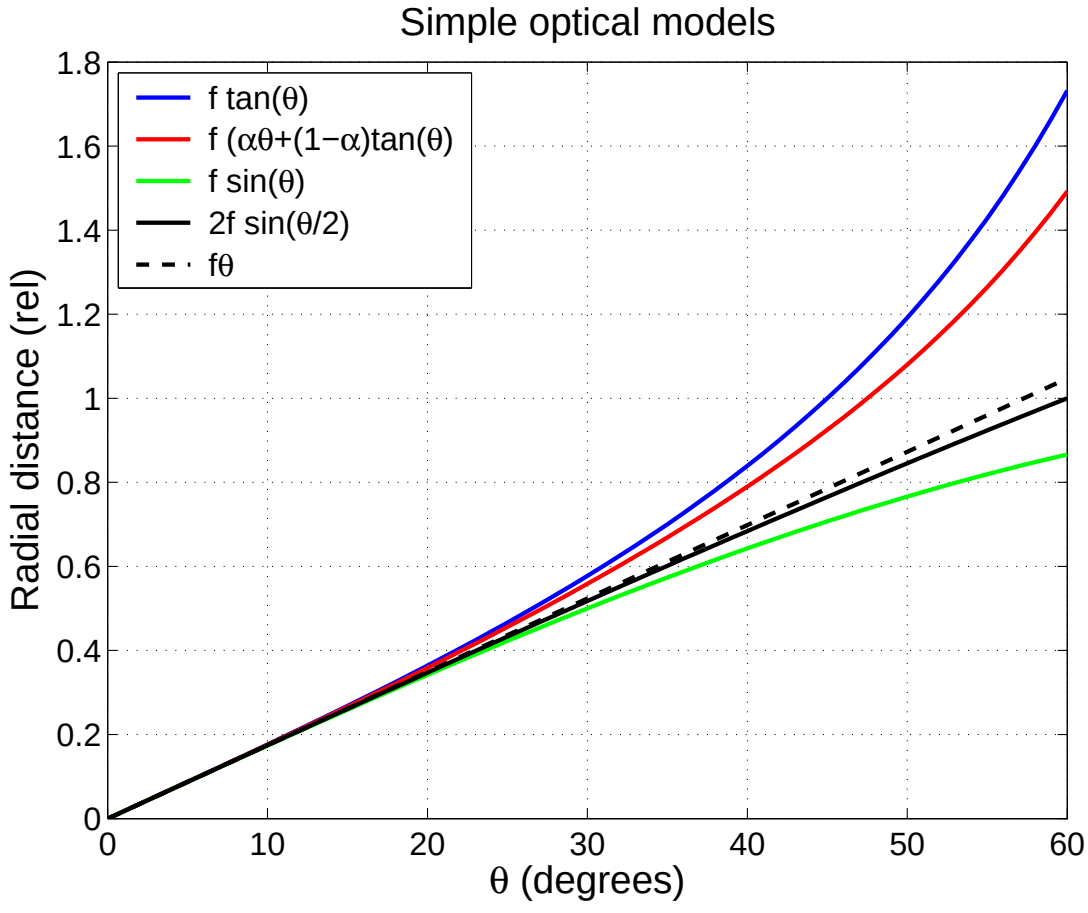
from the direction (ϕ, θ) to the image coordinates u, v.

Camera models (3)

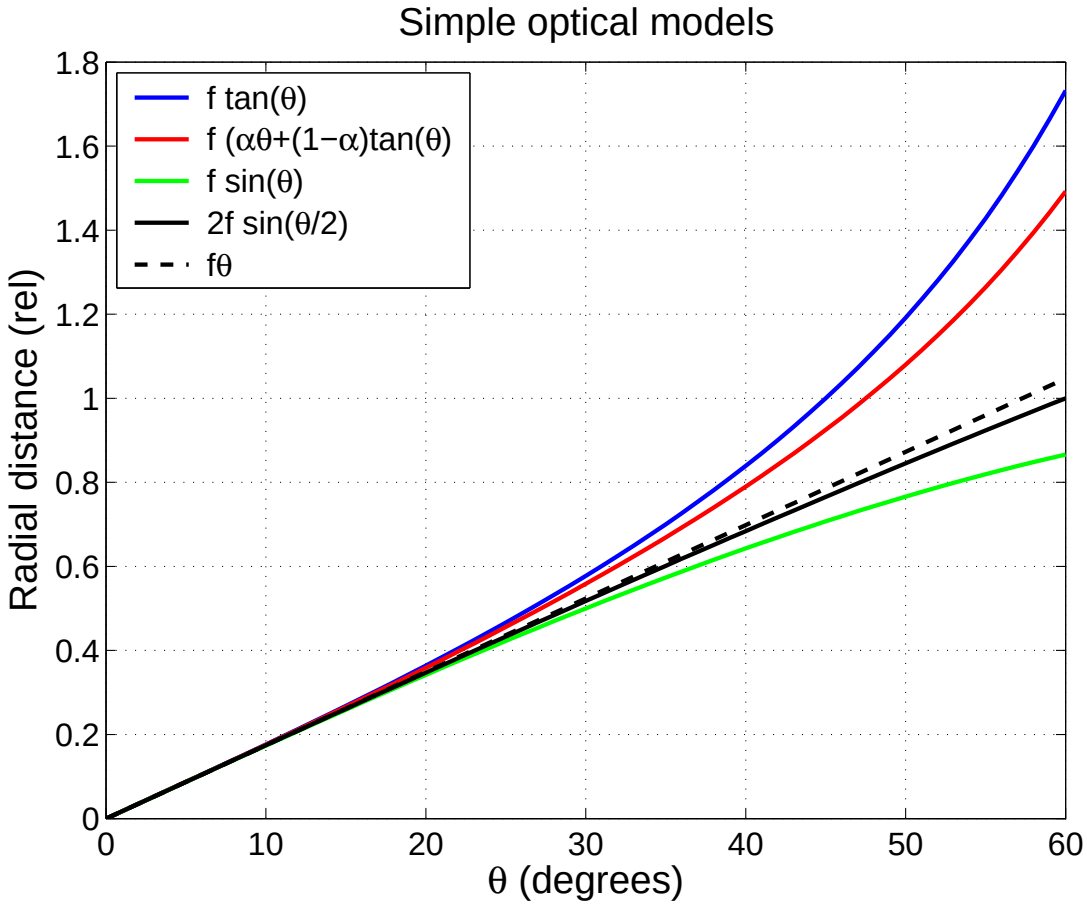
Narrow filed-of-view cameras are in general well described by pin-hole camera models (Astronomy, pocket cameras...). But in practice examples of non-pin-hole optics are abound.
$f(\theta) =$

- $\alpha\theta + (1-\alpha)\tan\theta$ **(ALIS)**
- $\sin\alpha\theta$ **(NIPR SPASI)**
- polynomials in θ, $p(\theta)$ **(e.g. old ASI)**
- rational functions in θ, $r(\theta)/q(\theta)$ **(Themis)**

Camera models (4)



Camera models (4)



The difference in characteristics have off course large impact on pixel line-of-sight but also on pixel field-of-view and thus on vignetting.

Camera models (5)

The two first camera models (ALIS, NIPR-SPASI) have 5 parameters determining their characteristics:

- u_0, v_0 - image coordinates of the projection of the optical axis.
- f_u, f_v - “focal widths” in vertical and horizontal direction.
- α - “shape parameter”

The bottom two can have just any number of free parameters. To determine the pixel f-o-v we have to get these as well as the camera rotations.

Camera rotations

To uniquely describe the rotation of a rigid body there are a number of different but equally adequate set of rotations. The choice in ALIS to mount the cameras in a gyro-type mount the natural choice is to use “Tait-Bryan angles” to describe the rotations. However, there seems to have been a mistake in the order of rotations somewheres between the master-plan, controller software, camera positioning system and analysis software. This is no problem as long as we use angles estimated by calibration from images — since this calibration use the same set of rotations as everywhere in the analysis.

Camera calibration

The to questions remaining now is

- How to determine the camera rotation and optical parameters?
- How many parameters to use? How much to tweak the tweaking.

This time we start with the first question.

Camera calibration (2)

To determine the camera rotation and optical parameters we can either make very accurate measurements of rotation angles. This of course requires very good zero reference direction for the rotations.
Then we still need to determine the parameters of the optical transfer function. That we could in principle achieve by tracing the rays trough the lens system – provided we know its characteristics.

Camera calibration (2)

To determine the camera rotation and optical parameters we can either make very accurate measurements of rotation angles. This of course requires very good zero reference direction for the rotations.
Then we still need to determine the parameters of the optical transfer function. That we could in principle achieve by tracing the rays trough the lens system – provided we know its characteristics.
These are two tricky problems to tackle. Reference directions is of course manageable, but tedious. For the ALIS stations this has not really been done. Primarily the rotation around the vertical is only determined by aligning the camera drive with compass. Which is not such a good idea in the *Kiruna* region. Knowledge about the lenses in the optics might be impossible to get to know.

Camera calibration (3)

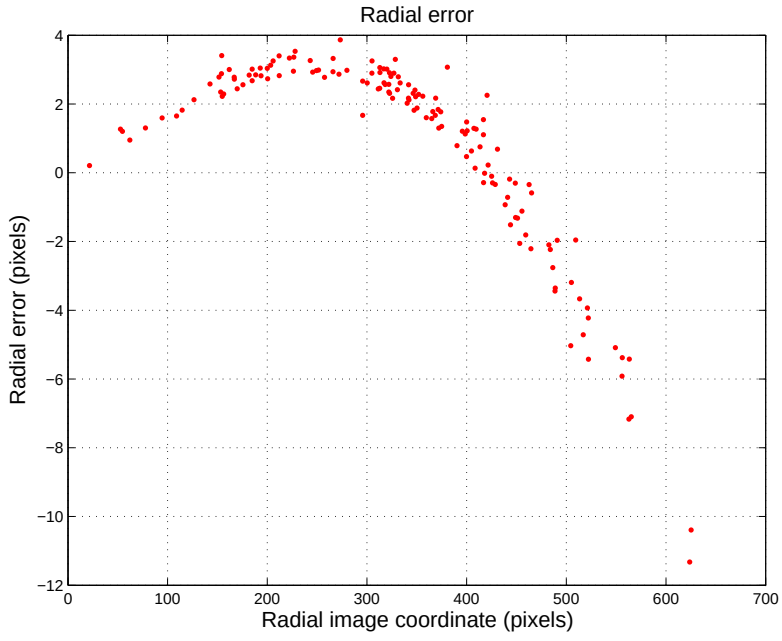
Or we could make images of calibration patterns with known positions and determine the necessary parameters that way. This is the method we've chosen with ALIS. For calibration pattern our choice is the background stars. Which have well known positions on the sky. Further this choice makes it possible to automatically account for refractions in the atmosphere.

Camera calibration (4)

How many optical parameters to use?
The answer is: as many as needed.
The ultimate limit is to remove all systematic errors between the projected positions of our calibration targets (stars) and their actual image position.

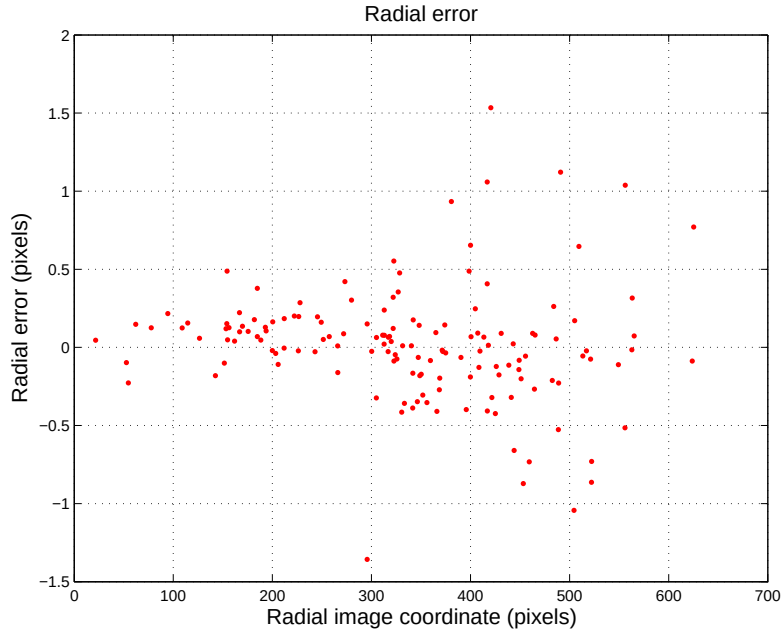
Camera calibration (4)

How many optical parameters to use?
The answer is: as many as needed.
The ultimate limit is to remove all systematic errors between the projected positions of our calibration targets (stars) and their actual image position.
This is the variation of errors in the radial direction as a function of radial position for an ALIS camera, assuming pin-hole optics. There are as we can see obvious systematic errors.



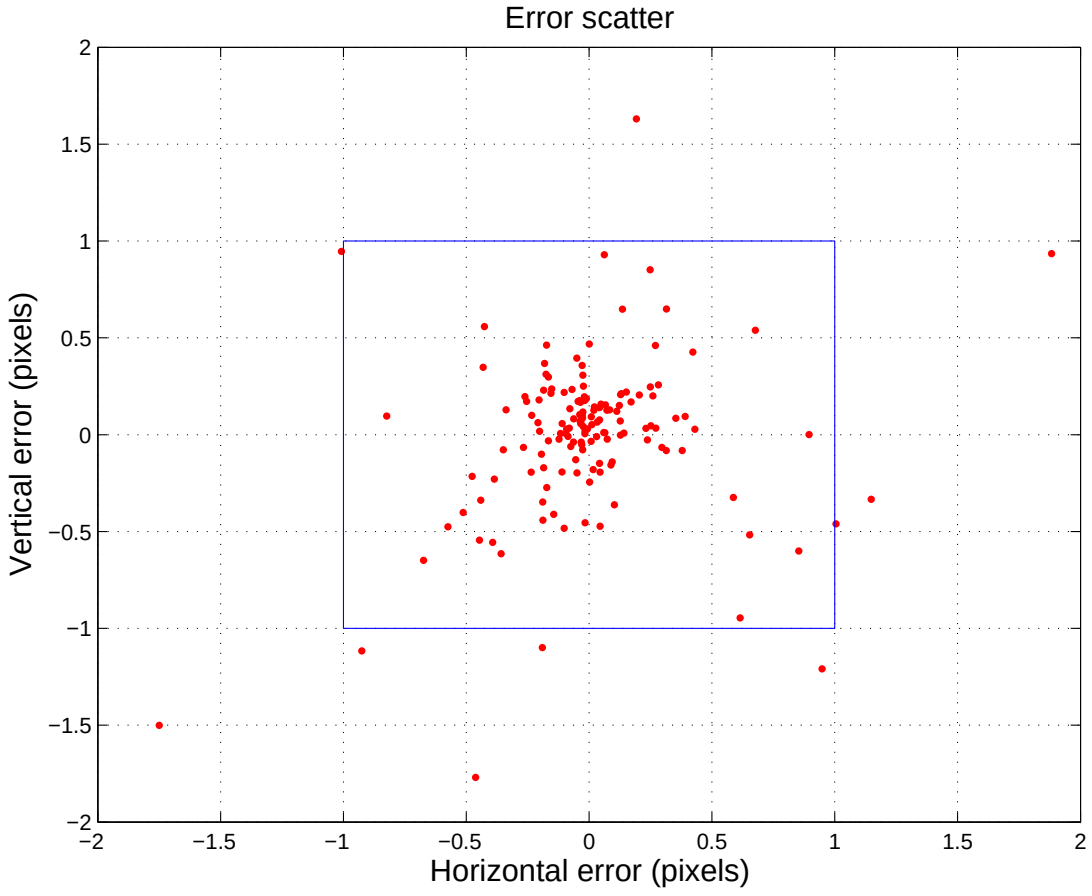
Camera calibration (4)

How many optical parameters to use?
The answer is: as many as needed.
The ultimate limit is to remove all systematic errors between the projected positions of our calibration targets (stars) and their actual image position.
This is the variation of errors in the radial direction as a function of radial position for an ALIS camera, assuming pin-hole optics. There are as we can see obvious systematic errors.
This is the same plot but now for the slightly more complex optical transfer function: $(1-\alpha)\tan\theta + \alpha\theta$. Here there is no obvious systematic errors.



Camera calibration (5)

And get the error scatter to within a pixel.



Starcal: tips and tricks

Sometimes it is difficult to find images suitable for star calibration. They might lack in exposure-time (aurora, rioe) or the background sky might be too bright (psc, aurora).
If a rapid sequence of images exist it is possible to add images taken during one minute and take the positions of the stars from the middle of the time period.
If the background sky is too bright this might do the trick:

```
d = SkMp.img;
d = wiener2(d-medfilt2(d,[9 9]),[3 3]);}
imagesc(d)
SkMp.img = d;
```

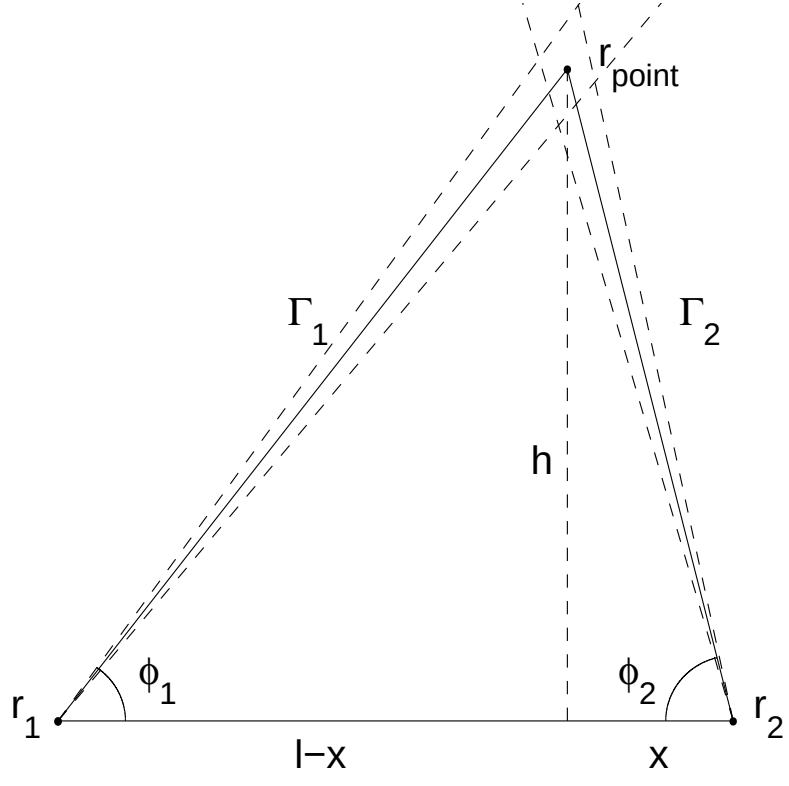
The subtraction of the median filtered image effectively reduces most of the background and what remains is noise and stars. The wiener filtering somewhat reduces the noise.

Accuracy estimates (1)

It is of some importance to know how our accuracy in pixel line-of-sight spreads (or rather lack of...) through further analysis. In order to estimate this we can calculate the error in the triangulated position of an imaged object as a function of l-o-s uncertainties:

Triangulation error (1)

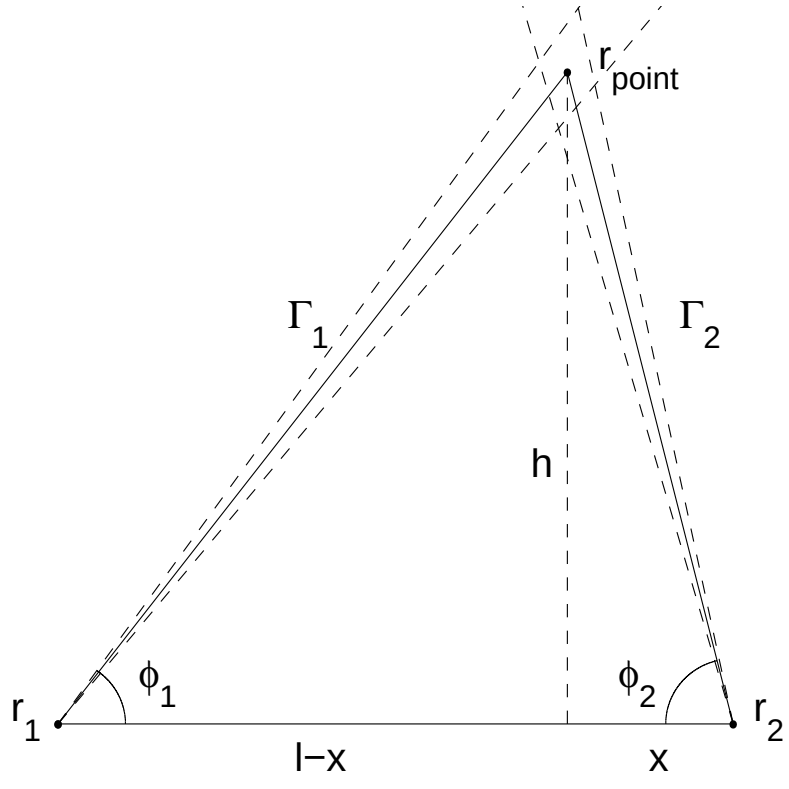
Here, a first order error analysis is easily accomplished if we look at stereoscopic triangulation of an object point found in two images from sites at $\bar{r}_1$ and $\bar{r}_2$.



Triangulation error (1)

Here, a first order error analysis is easily accomplished if we look at stereoscopic triangulation of an object point found in two images from sites at $\bar{r}_1$ and $\bar{r}_2$.

If we let the two lines-of-sight Γ_1, Γ_2 be described by their angles Φ_1, Φ_2 relative to the vector $\bar{r}_1 - \bar{r}_2$, it is straightforward to calculate the uncertainty in the position of $\bar{r}_{point}$ due to errors in the line-of-sight parameters $\Delta\Phi_1, \Delta\Phi_2$ and the station separation Δl.



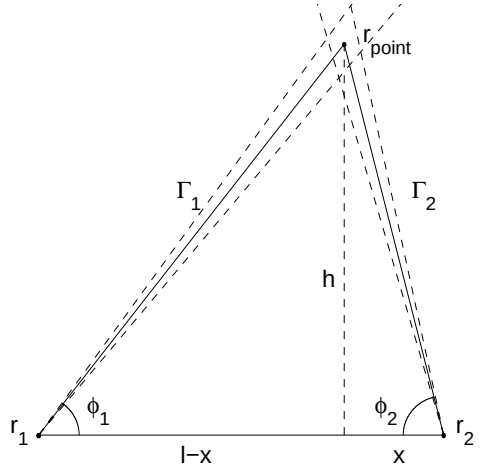
Triangulation error (2)

Simplifying the problem to two dimensions we get

$$h = l\frac{\tan\phi_1 \cdot \tan\phi_2}{\tan\phi_1 + \tan\phi_2}$$

$$x = l\frac{\tan\phi_1}{\tan\phi_1 + \tan\phi_2}$$

for the position of the top corner.

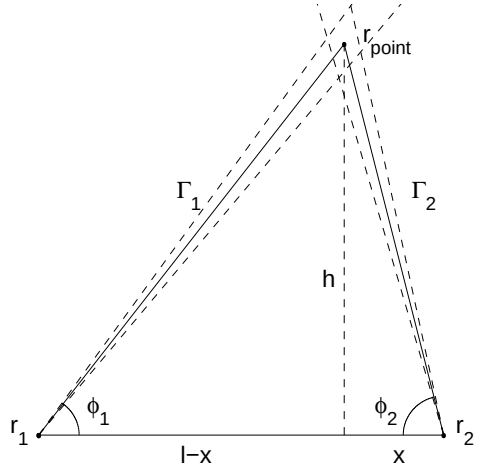


Triangulation error (2)

Simplifying the problem to two dimensions we get

$$h = l\frac{\tan\phi_1 \cdot \tan\phi_2}{\tan\phi_1 + \tan\phi_2}$$

$$x = l\frac{\tan\phi_1}{\tan\phi_1 + \tan\phi_2}$$



for the position of the top corner. The sensitivity of h and x to errors in l, ϕ_1, and ϕ_2 is then estimated by the full differentials:

$$dh = \frac{\partial h}{\partial l}\Delta l + \frac{\partial h}{\partial \phi_1}\Delta\phi_1 + \frac{\partial h}{\partial \phi_2}\Delta\phi_2$$

$$dx = \frac{\partial x}{\partial l}\Delta l + \frac{\partial x}{\partial \phi_1}\Delta\phi_1 + \frac{\partial x}{\partial \phi_2}\Delta\phi_2$$

Triangulation error (3)

The ambitious student can check my calculation of the partial derivatives:

$$\frac{\partial h}{\partial \phi_1} = \frac{l \tan^2 \phi_2 \left(1 + \tan^2 \phi_1\right)}{\left(\tan \phi_1 + \tan \phi_2\right)^2} \qquad \frac{\partial x}{\partial \phi_1} = \frac{l \left(1 + \tan^2 \phi_1\right) \tan \phi_2}{\left(\tan \phi_1 + \tan \phi_2\right)^2}$$

$$\frac{\partial h}{\partial \phi_2} = \frac{l \tan^2 \phi_1 \left(1 + \tan^2 \phi_2\right)}{\left(\tan \phi_1 + \tan \phi_2\right)^2} \qquad \frac{\partial x}{\partial \phi_2} = -\frac{l \tan \phi_1 \left(1 + \tan^2 \phi_2\right)}{\left(\tan \phi_1 + \tan \phi_2\right)^2}$$

$$\frac{\partial h}{\partial l} = \frac{\tan \phi_1 \tan \phi_2}{\tan \phi_1 + \tan \phi_2} \qquad \frac{\partial x}{\partial l} = \frac{\tan \phi_1}{\tan \phi_1 + \tan \phi_2}$$

Triangulation error (4)

From these equations it is possible to calculate the propagation of uncertainties in the lines-of-sight to the positions for all directions.

Triangulation error (4)

From these equations it is possible to calculate the propagation of uncertainties in the lines-of-sight to the positions for all directions.
As a rule of thumb, as the station separation is small compared to the distance to the object, i.e. $|\bar{r}_1 - \bar{r}_2| \rightarrow 0$ the sensitivity in the determination of positions to errors in the direction of the lines-of-sight increase rapidly.

Triangulation error (4)

From these equations it is possible to calculate the propagation of uncertainties in the lines-of-sight to the positions for all directions.
As a rule of thumb, as the station separation is small compared to the distance to the object, i.e. $|\bar{r}_1 - \bar{r}_2| \to 0$ the sensitivity in the determination of positions to errors in the direction of the lines-of-sight increase rapidly.
(Humans use stereoscopy only within a few meters. Further away spatial information is inferred from perspective object size and experience. Also same reason distance to stars cannot be determined by stereoscopy except for the closest.)

Triangulation error (final)

Here are two examples with typical values for stratospheric (case 1) and auroral (case 2) work.

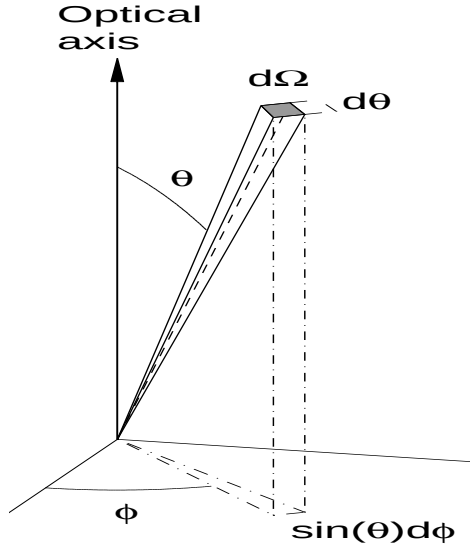
	Case 1	Case 2	
$\partial h/\partial l$	0.3889	1.3268	$()$
$\partial h/\partial \phi_1$	0.5280	2.0484	$(km/^\circ)$
$\partial h/\partial \phi_2$	0.2249	1.5841	$(km/^\circ)$
$\partial x/\partial l$	0.3264	0.2340	$()$
$\partial x/\partial \phi_1$	0.4431	0.3612	$(km/^\circ)$
$\partial x/\partial \phi_2$	-0.3896	-0.9146	$(km/^\circ)$

Triangulation error (really final)

To get the estimated distances correct to within 50 m, we need to know the base line, Δl, within ± 50 m and the line-of-sight directions, $(\Delta\phi_1, \Delta\phi_2)$, to within $\pm 0.02°$. For stratospheric studies, where the parallax is larger, the requirements are less strict.

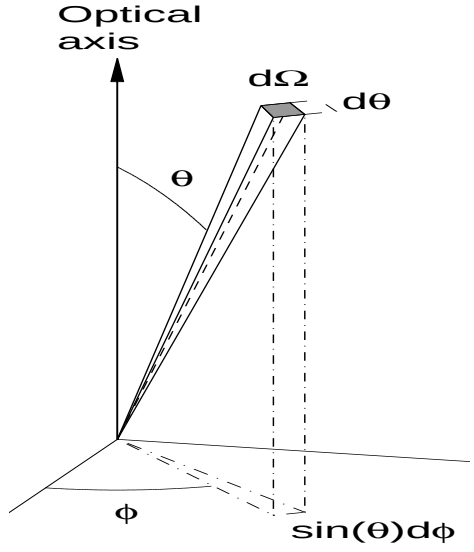
Flat-field correction (1)

Given an optical transfer function as outlined above, the pixel field-of-view, $d\Omega$, can be determined by standard calculus.



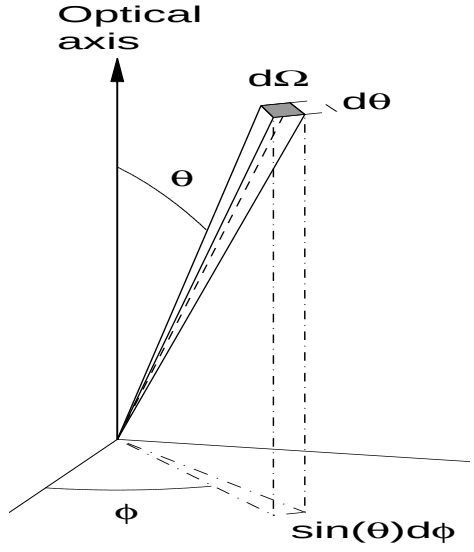
Flat-field correction (1)

Given an optical transfer function as outlined above, the pixel field-of-view, $d\Omega$, can be determined by standard calculus. With θ as the angle relative to the optical axis of the camera, and ϕ as the azimuthal angle, the field-of-view is:



Flat-field correction (1)

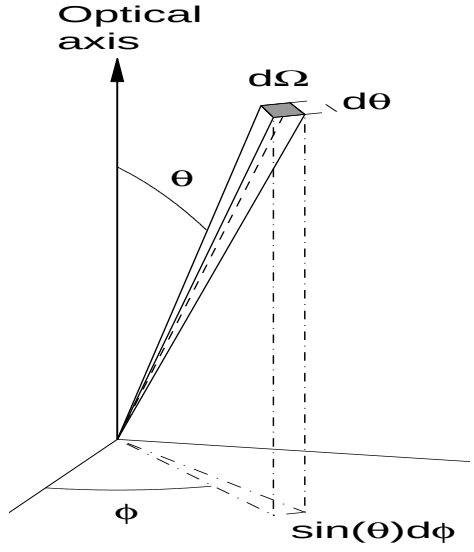
Given an optical transfer function as outlined above, the pixel field-of-view, $d\Omega$, can be determined by standard calculus. With θ as the angle relative to the optical axis of the camera, and ϕ as the azimuthal angle, the field-of-view is:



$$d\Omega(u,v) = \sin\theta d\phi d\theta = \sin\theta |\frac{\partial(\phi,\theta)}{\partial(u,v)}| dudv = \sin\theta |\frac{\partial(u,v)}{\partial(\phi,\theta)}|^{-1} dudv$$

Flat-field correction (1)

Given an optical transfer function as outlined above, the pixel field-of-view, $d\Omega$, can be determined by standard calculus. With θ as the angle relative to the optical axis of the camera, and ϕ as the azimuthal angle, the field-of-view is:



$$d\Omega(u,v) = \sin\theta d\phi d\theta = \sin\theta |\frac{\partial(\phi,\theta)}{\partial(u,v)}| dudv = \sin\theta |\frac{\partial(u,v)}{\partial(\phi,\theta)}|^{-1} dudv$$

here $|\frac{\partial(u,v)}{\partial(\phi,\theta)}|$ is the absolute value of the determinant of the Jacobian of the optical transfer function.

Flat-field correction (2)

Further the we have to take the variation in effective collecting area of the front lens into account:

$$A(\theta) = A_L \cos\theta$$

Here A_L is the maximum area of the front lens or the aperture of the optics and θ is the angle relative to the optical axis. Here we ignore variations in transmission through the optics with θ.

Flat-field correction (3)

Combining these two equations we get a general (ignoring transparency variations in optics) expression for the image intensity variation from uniform isotropic radiance

$$I_{flat} \propto \cos\theta \sin\theta |\frac{\partial(u,v)}{\partial(\phi,\theta)}|^{-1} du dv$$

(When using this for correction it is important to remember that multiplication might change the signal/noise relation.) For optics with rotational symmetry (or with eigenvectors $\|$ to the image coordinates) where $(u,v) = (f_u f(\theta)\cos\phi + u_0, f_v f(\theta)\sin\phi + v_0)$, we can simplify further

$$I_{flat} \propto \frac{\cos\theta \sin\theta}{|f_u f_v f(\theta) \frac{\partial f(\theta)}{\partial\theta}|}$$

Flat-field exmpl

For pin-hole optics (where $(u, v) = f(\tan\theta\cos\phi, \tan\theta\sin\phi)$) this gives:

$$\cos\theta\sin\theta|\frac{\partial(u,v)}{\partial(\phi,\theta)}|^{-1} = \cos\theta\sin\theta|\frac{\partial(f(\tan\theta\cos\phi,\tan\theta\sin\phi))}{\partial(\phi,\theta)}|^{-1} =$$

$$\left\{\frac{\partial(f(\tan\theta\cos\phi,\tan\theta\sin\phi))}{\partial(\phi,\theta)} = f^2\begin{vmatrix}-\tan\theta\sin\phi & \cos^{-2}\theta\cos\phi\\ \tan\theta\cos\phi & \cos^{-2}\theta\sin\phi\end{vmatrix}\right.$$

$$\left. = f^2\sin\theta\cos^{-3}\theta\right\} = \cos\theta\sin\theta\cos^3\theta/\sin\theta/f^2 = \cos^4\theta/f^2$$

which is the result presented in most textbooks. This is, *again*: not a generally true result...

Flat-field exmpl cont...

...if we for example have an optical mapping function $(u, v) = f(\sin\theta/2\cos\phi, \sin\theta/2\cos\phi)$ we get

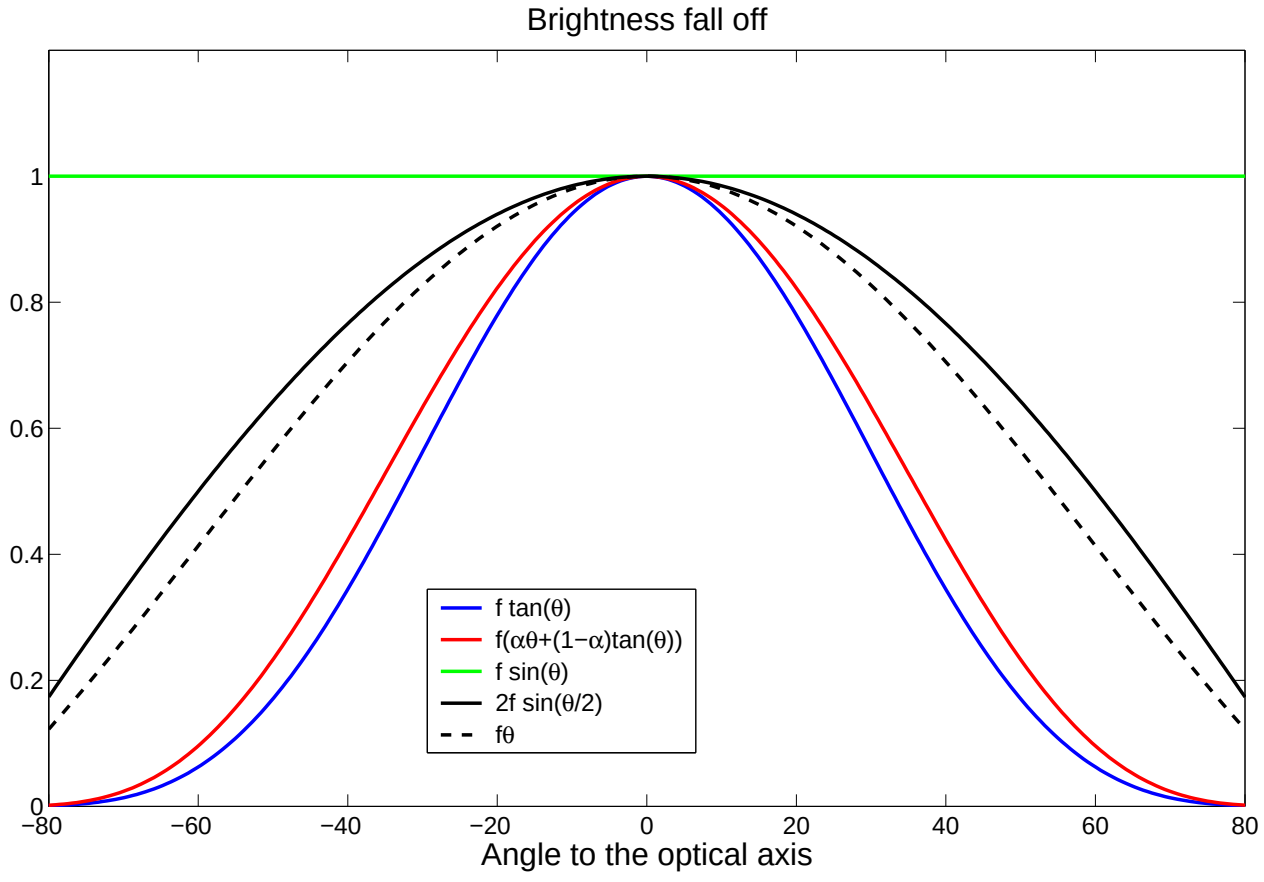
$$\cos\theta\sin\theta|\frac{\partial(u,v)}{\partial(\phi,\theta)}|^{-1} = \cos\theta\sin\theta|\frac{\partial(f(\sin\frac{\theta}{2}\cos\phi, \sin\frac{\theta}{2}\sin\phi))}{\partial(\phi,\theta)}|^{-1} =$$

$$\left\{\frac{\partial(f(\sin\frac{\theta}{2}\cos\phi, \sin\frac{\theta}{2}\sin\phi))}{\partial(\phi,\theta)} = \frac{f^2}{2}\begin{vmatrix} -\sin\frac{\theta}{2}\sin\phi & \cos\frac{\theta}{2}\cos\phi \\ \sin\frac{\theta}{2}\cos\phi & \cos\frac{\theta}{2}\sin\phi \end{vmatrix} =\right.$$

$$\left. = \frac{f^2}{2}\sin\frac{\theta}{2}\cos\frac{\theta}{2} = \frac{f^2}{4}\sin\theta\right\} = \frac{4}{f^2}\cos\theta\sin\theta/\sin\theta = \frac{4}{f^2}\cos\theta$$

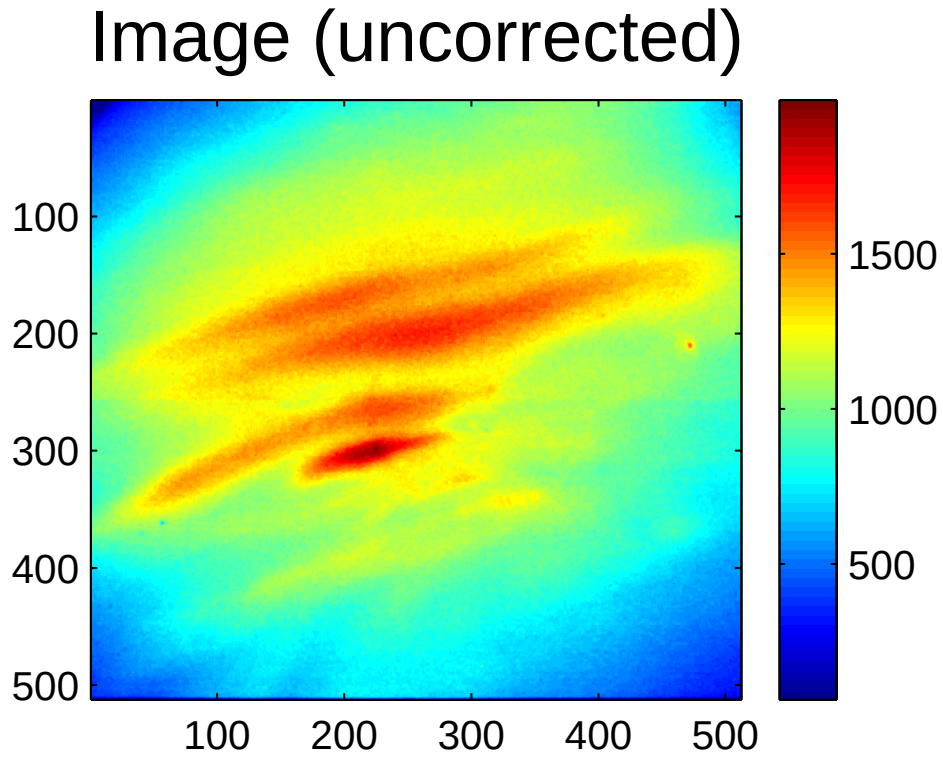
Flat-field correction (4)

To summarize, here we have a number of flat0-field curves for the optical transfer functions listed before.



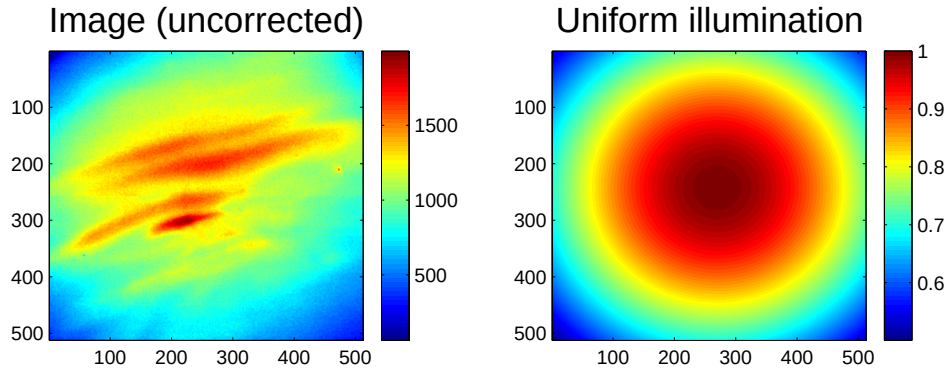
Practical FFC of ALIS images

To use equation (5) is the standard method of flat-field correction used on ALIS data...



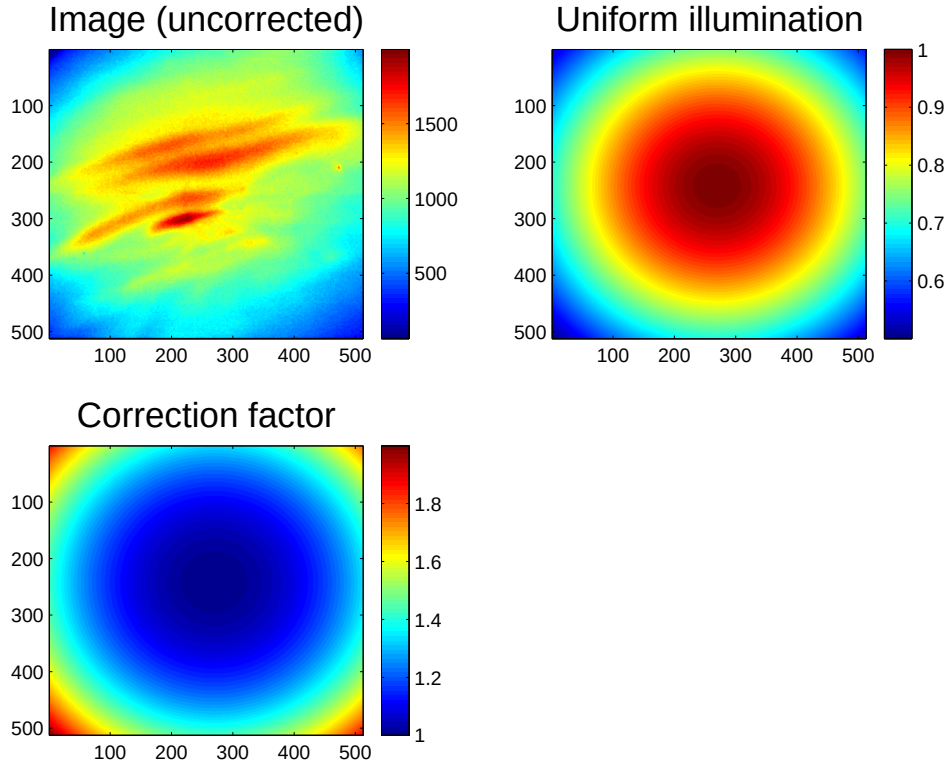
Practical FFC of ALIS images

To use equation (5) is the standard method of flat-field correction used on ALIS data...



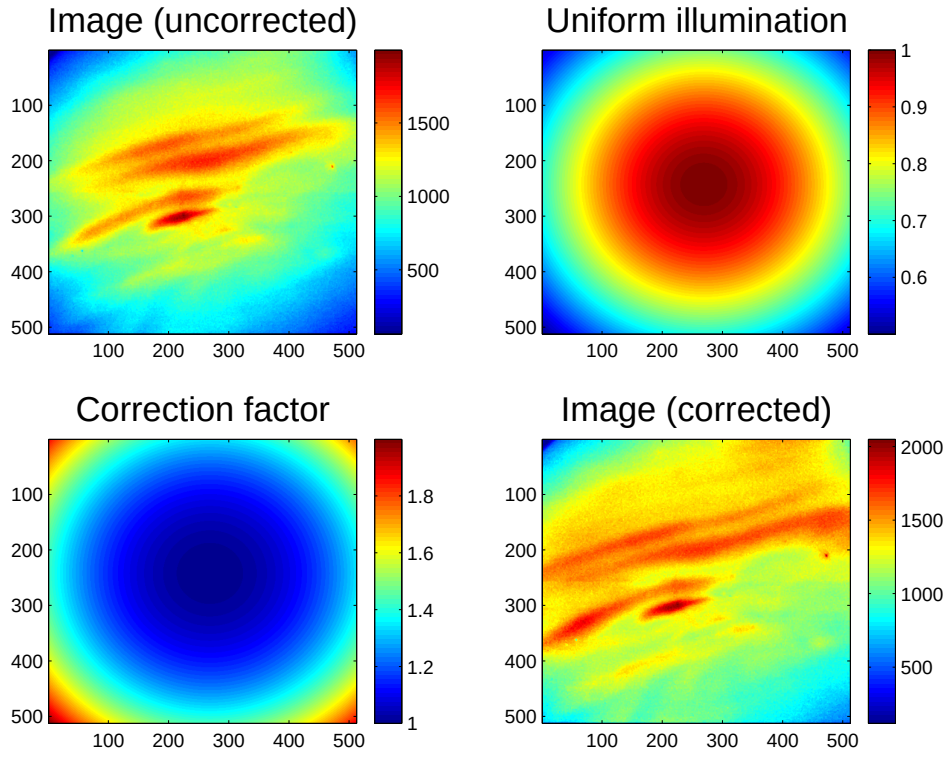
Practical FFC of ALIS images

To use equation (5) is the standard method of flat-field correction used on ALIS data...



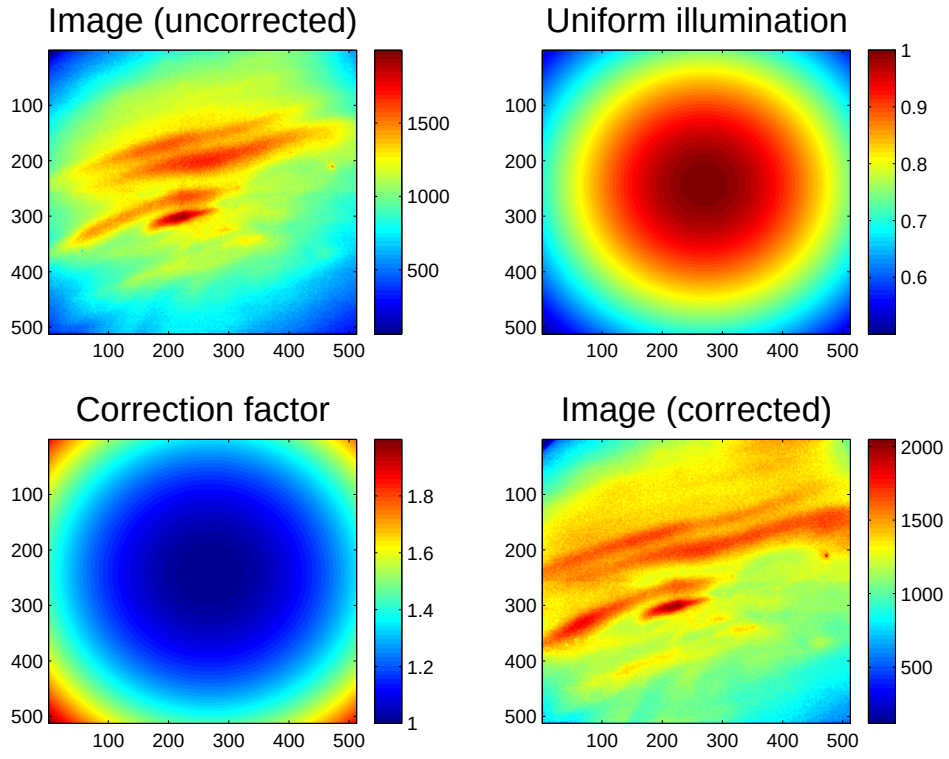
Practical FFC of ALIS images

To use equation (5) is the standard method of flat-field correction used on ALIS data...



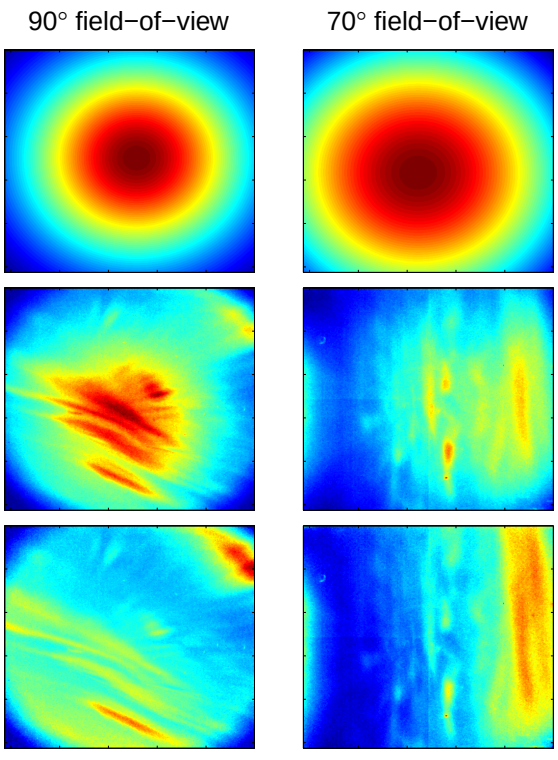
Practical FFC of ALIS images

To use equation (5) is the standard method of flat-field correction used on ALIS data...



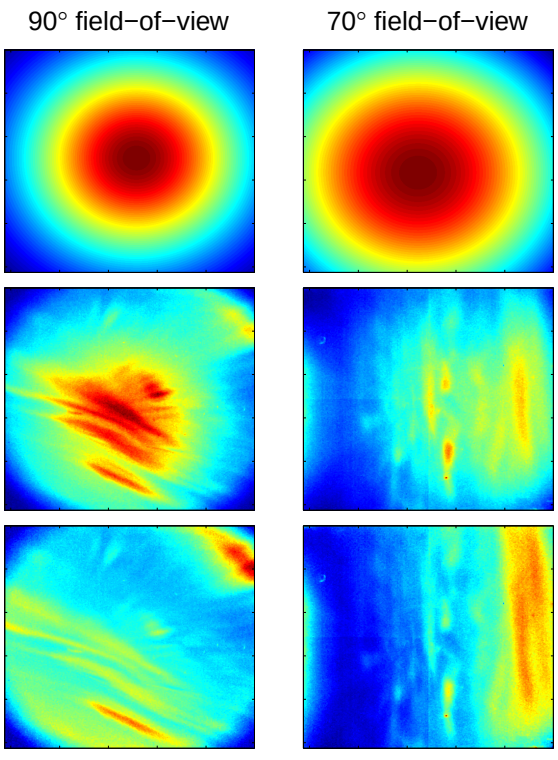
...and apparently good enough for government work.

Practical FFC — summary



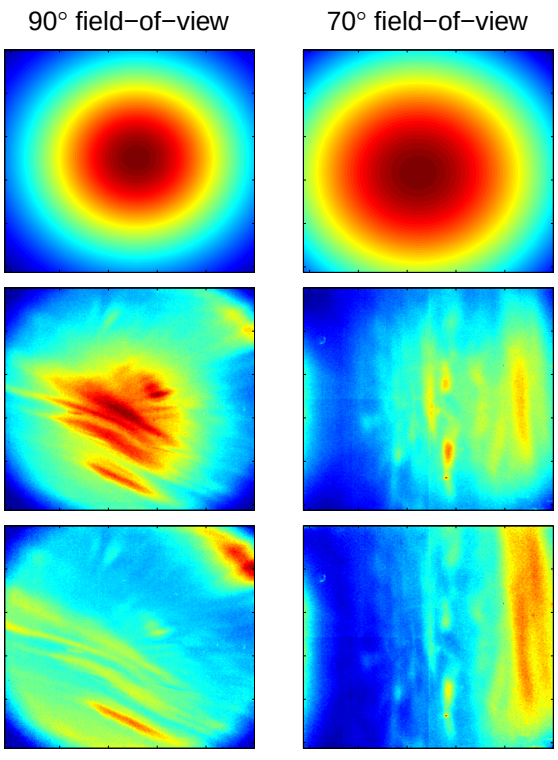
The following example summarizes the application of the correction outlined above.

Practical FFC — summary



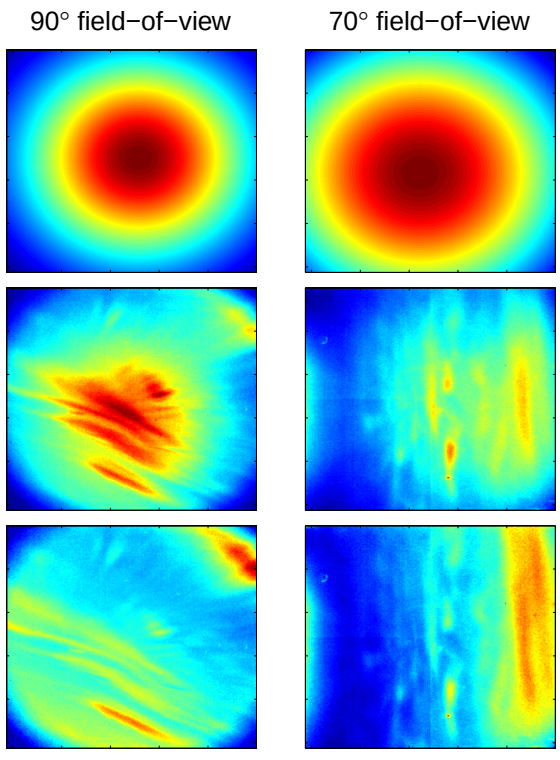
The following example summarizes the application of the correction outlined above. First row shows the flat-field correction images for ALIS camera # 2 (left column) and # 1 (right). In the second row are two uncorrected auroral images made by ALIS 19970216 in the auroral green line (5577 Å),

Practical FFC — summary



The following example summarizes the application of the correction outlined above. First row shows the flat-field correction images for ALIS camera # 2 (left column) and # 1 (right). In the second row are two uncorrected auroral images made by ALIS 19970216 in the auroral green line (5577 Å), and in the bottom row the same images corrected.

Practical FFC — summary



The following example summarizes the application of the correction outlined above. First row shows the flat-field correction images for ALIS camera # 2 (left column) and # 1 (right). In the second row are two uncorrected auroral images made by ALIS 19970216 in the auroral green line (5577 Å), and in the bottom row the same images corrected. Note that the correction is good for the whole image plane for the image taken with the 70°field-of-view but not for the image with 90°field-of-view where the corners are markedly darker.

Photo response non-uniformity (1)

The flat-field correction outlined above take only the large-scale variation into account, not the local pixel-to-pixel sensitivity variation.

Photo response non-uniformity (1)

The flat-field correction outlined above take only the large-scale variation into account, not the local pixel-to-pixel sensitivity variation. Still pixels on CCD-arrays have varying sensitivity [e.g. **??**, and references therein].

Photo response non-uniformity (1)

Still pixels on CCD-arrays have varying sensitivity [e.g. **??**, and references therein].

Photo response non-uniformity (1)

Still pixels on CCD-arrays have varying sensitivity [e.g. **??**, and references therein]. Here we will describe a method (based on an idea by P. Rydesäter) to obtain estimates of this from ordinary images obtained during regular observations of the sky.

Photo response non-uniformity (1)

Still pixels on CCD-arrays have varying sensitivity [e.g. **??**, and references therein]. Here we will describe a method (based on an idea by P. Rydesäter) to obtain estimates of this from ordinary images obtained during regular observations of the sky.
In general the sky brightness varies smoothly on a point-to-point scale and the small scale variations appear randomly distributed both in time and space.

Photo response non-uniformity (1)

Still pixels on CCD-arrays have varying sensitivity [e.g. **??**, and references therein].
In general the sky brightness varies smoothly on a point-to-point scale and the small scale variations appear randomly distributed both in time and space,

Photo response non-uniformity (1)

Still pixels on CCD-arrays have varying sensitivity [e.g. **??**, and references therein].
In general the sky brightness varies smoothly on a point-to-point scale and the small scale variations appear randomly distributed both in time and space, except for stars – but they rotate through the image plane when the imager is kept in a fixed rotation.

Photo response non-uniformity (1)

Still pixels on CCD-arrays have varying sensitivity [e.g. **??**, and references therein].
In general the sky brightness varies smoothly on a point-to-point scale and the small scale variations appear randomly distributed both in time and space.

Photo response non-uniformity (1)

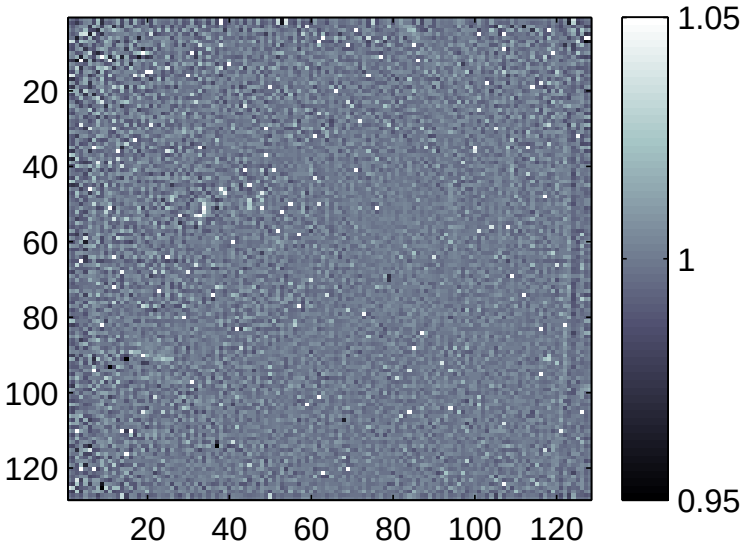
Still pixels on CCD-arrays have varying sensitivity [e.g. **??**, and references therein].
In general the sky brightness varies smoothly on a point-to-point scale and the small scale variations appear randomly distributed both in time and space. Assuming that intensity gradients vary randomly with a zero average we get an estimate of C_{p2p} by taking the average of the ratio between raw image intensity and image intensities after 2-D local median filtering for a large set of images I_i:

$$C_{p2p} = \frac{1}{N} \sum_{i=1}^{N} I_i / I_i^m \tag{6}$$

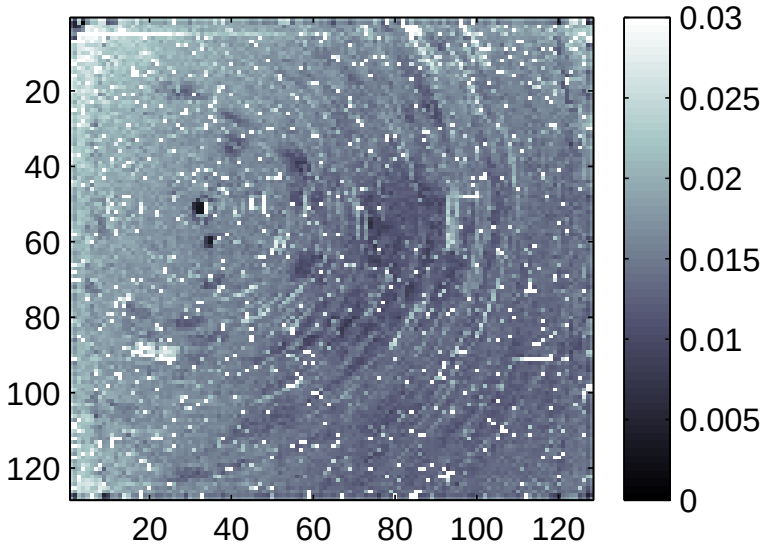
where I_i^m is image I_i 5 by 5 median filtered.

Photo response non-uniformity (2)

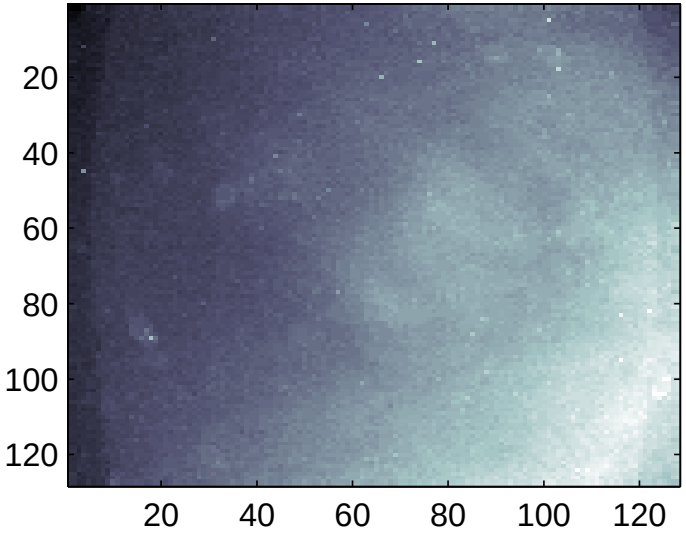
Estimate of pixel−pixel relative sensitivity



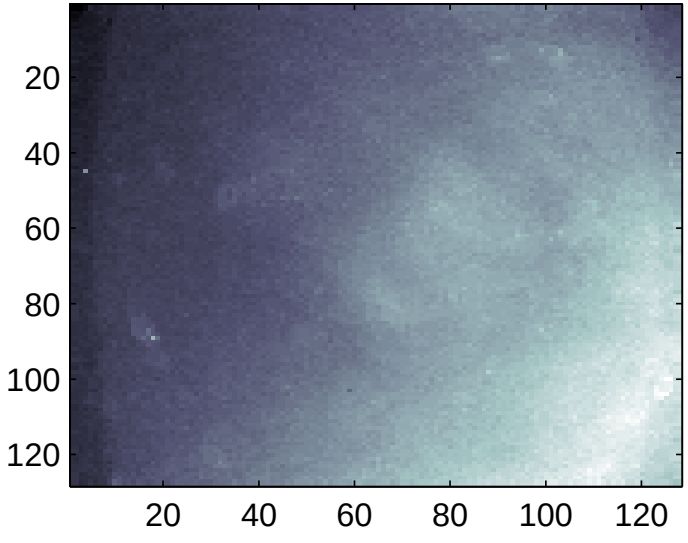
Est. of std of p−p relative sensitivity



Before correction for p−p sens. var.



After correction for p−p sens. var.



PRNU - note

C_{p2p} is yet only available for a select few camera/binning combos. More will be made “shortly”, and `typical_pre_proc_ops` will in time do this correction automatically. But with standard bad-pixel-fix and median filtering this is a very minor concern.

As can be seen C_{p2p} is fairly flat around 1 with 96 percent of the pixels within 1 ± 0.02 and no large scale variations; with some hot and cool pixels. Judging from the small standard deviation of C_{p2p} the estimates seem reliable. The circular pattern that is seen is caused by stars that sweep over/through the image plane. Thustheir contribution to C_{p2p} is averaged out.

The upper left panel shows the estimates of the photo responce non-uniformity, PRNU, (point to point sensitivity variation) for ALIS CCD-imager #3. A number of hot and some cool pixels can be seen. In the upper right panel the variance is shown. The circular pattern is caused by stars rotating around Stella Polaris, at pixel coordinate (37,51), and through the pixels. This does not give a large contribution to the sensitivity but contributes to its variance. In the lower left panel is an image before correction for the pixel-to-pixel sensitivity variation, and in the lower right panel the image after correction; and as can be seen a number of the bad pixels have been reasonably corrected.

PSF-Deconvolution of ALIS images

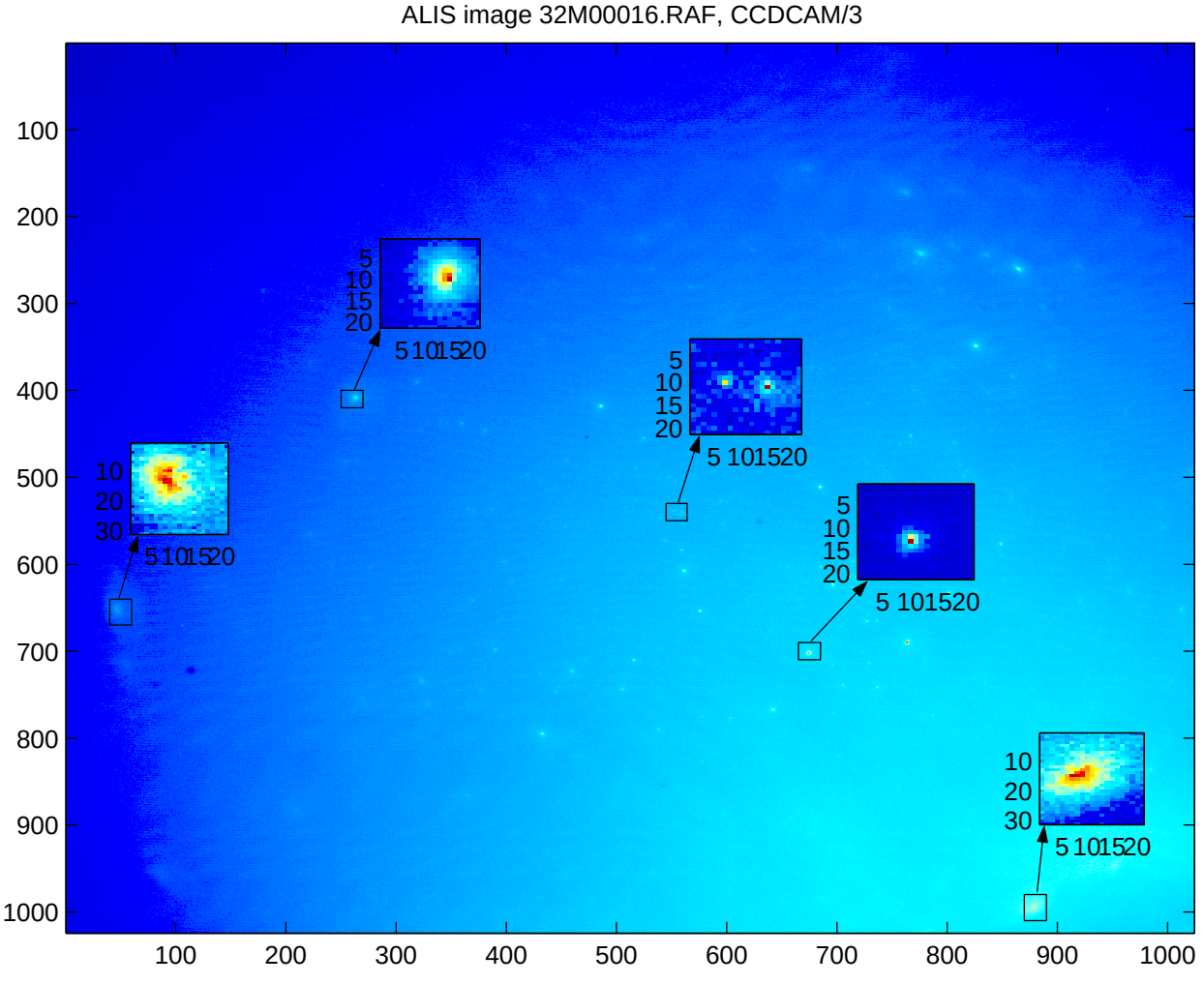
- PSF function of ALIS imagers
- Aberrations
- Deconvolution principles
- Deconvolution applied to ALIS images

ALIS imager PSF

It has long been known that our imagers have aberration limited optics. That is the point spread functions on the detector is not limited by diffraction but by aberrations.

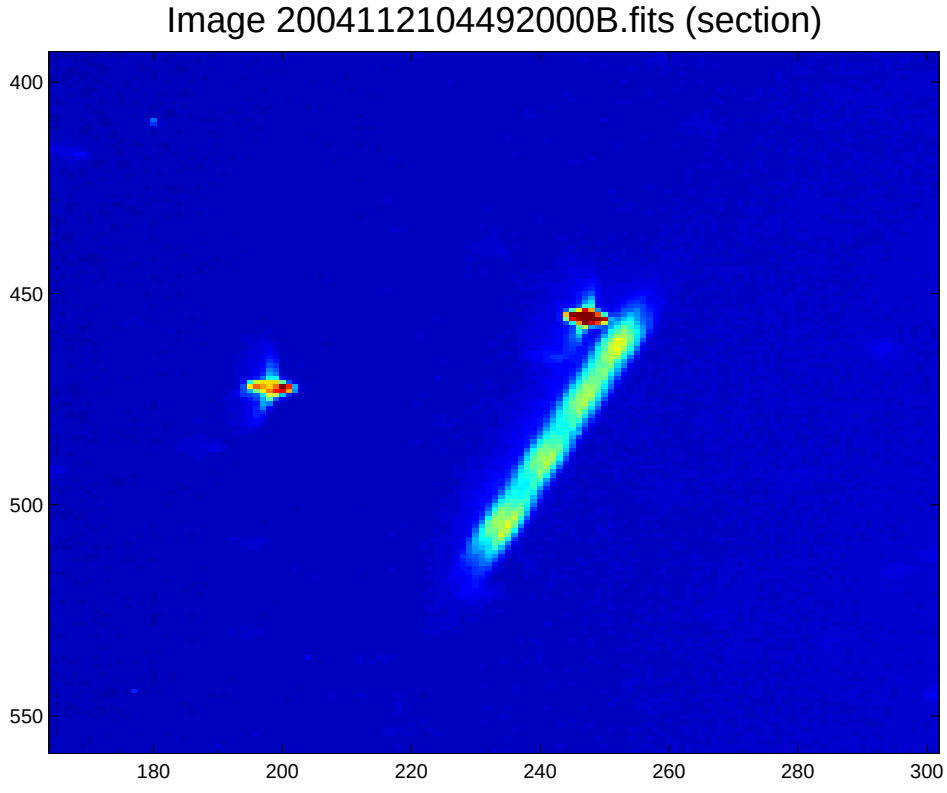
ALIS imager PSF

It has long been known that our imagers have aberration limited optics. That is the point spread functions on the detector is not limited by diffraction but by aberrations.



Background

In my work this has never been a problem — a wide PSF only smears structures in the images and makes us see less in the images. Thus I've never really bothered to learn how to use deconvolution. But last November Csilla and Johan got an image of a meteor with a double trail:



Repetition of some aberrations (1)

The 5 third-order aberrations in monochromatic illumination are:

- Coma

Repetition of some aberrations (1)

The 5 third-order aberrations in monochromatic illumination are:

- Coma
- Spherical aberration

Repetition of some aberrations (1)

The 5 third-order aberrations in monochromatic illumination are:

- Coma
- Spherical aberration
- Field curvature

Repetition of some aberrations (1)

The 5 third-order aberrations in monochromatic illumination are:

- Coma
- Spherical aberration
- Field curvature
- Astigmatism

Repetition of some aberrations (1)

The 5 third-order aberrations in monochromatic illumination are:

- Coma
- Spherical aberration
- Field curvature
- Astigmatism
- (Distortion)

Repetition of some aberrations (1)

The 5 third-order aberrations in monochromatic illumination are:

- Coma
- Spherical aberration
- Field curvature
- Astigmatism
- (Distortion)

and 2 chromatic:

Repetition of some aberrations (1)

The 5 third-order aberrations in monochromatic illumination are:

- Coma
- Spherical aberration
- Field curvature
- Astigmatism
- (Distortion)

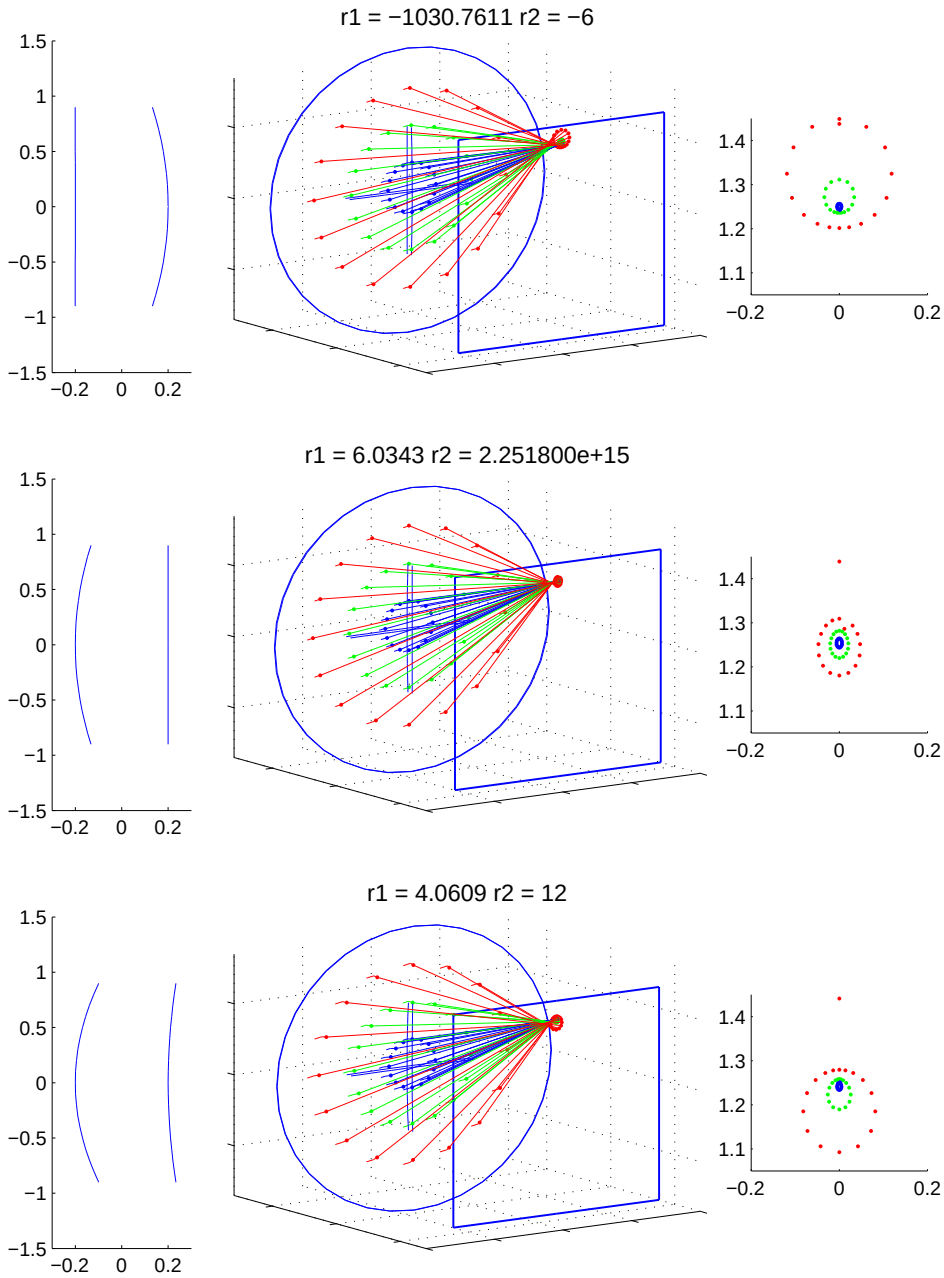
and 2 chromatic:

- Longitudinal
- Transversal

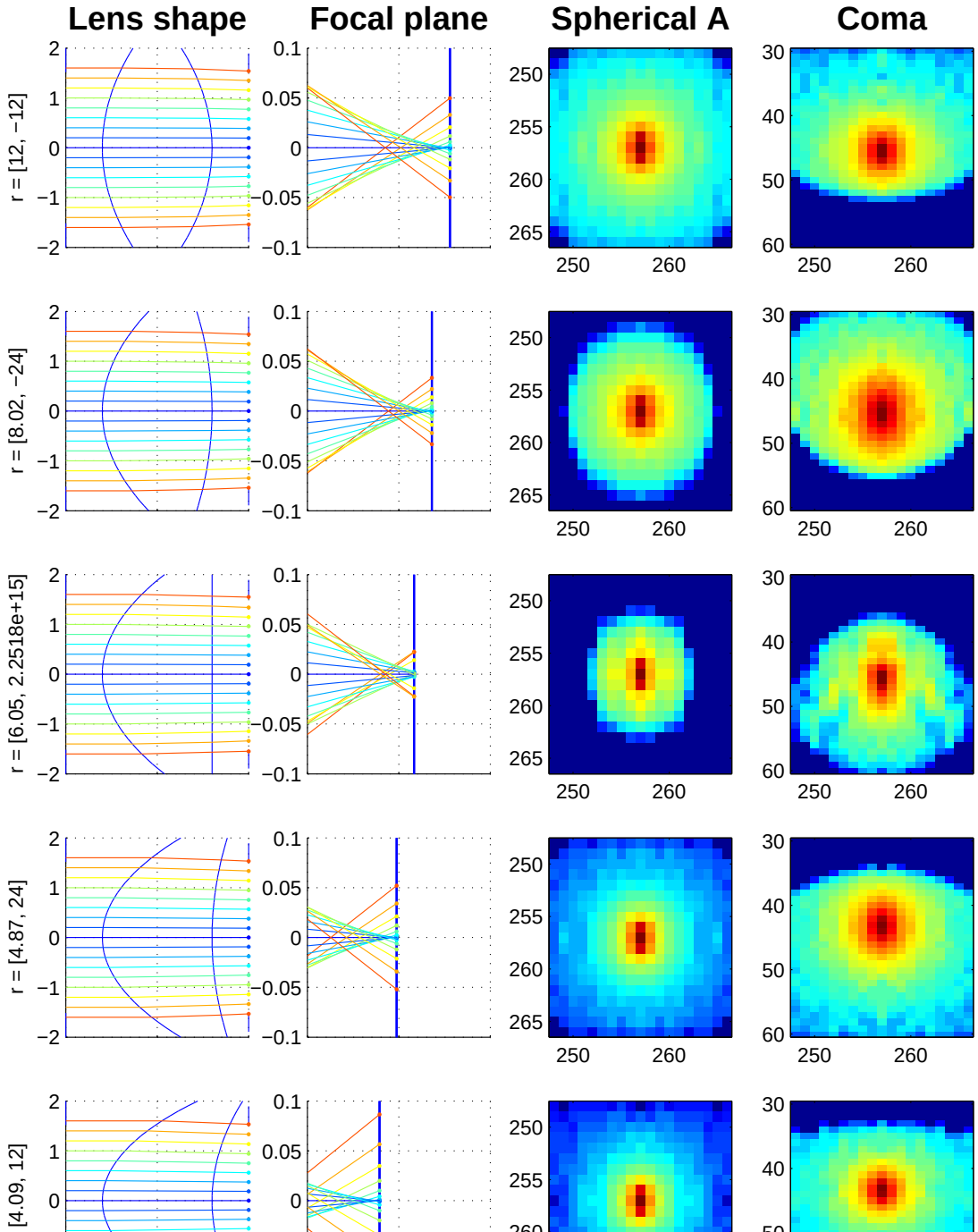
Coma

Coma is tear-shaped psf off the optical axis, either tailing off outwards or inwards. This is caused by different refractions by rays hitting the center and the periphery of the lens.

Coma expl.

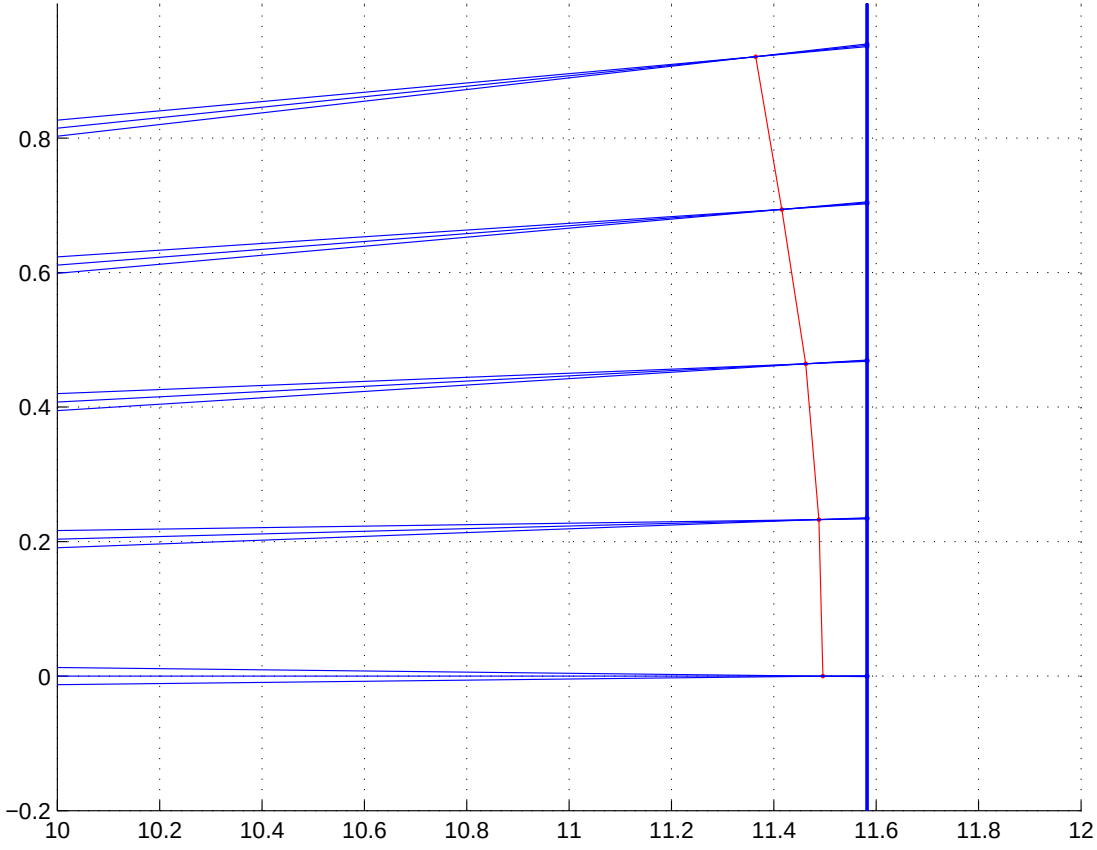


Spherical aberration and Coma



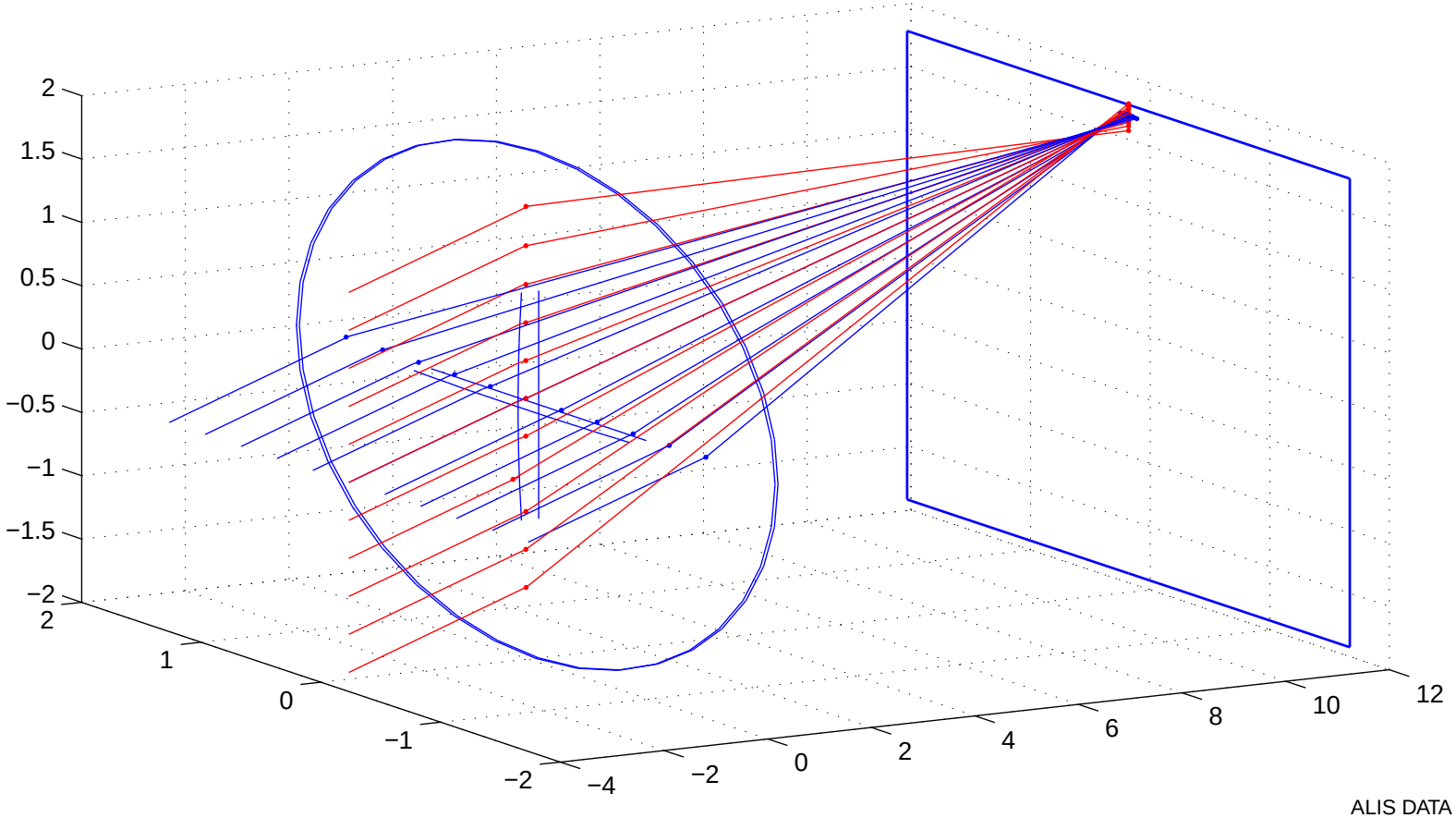
Field curvature

Typically the focal “plane” is not a plane but rather a curved surface. This plot show this for central rays through the same single lens.



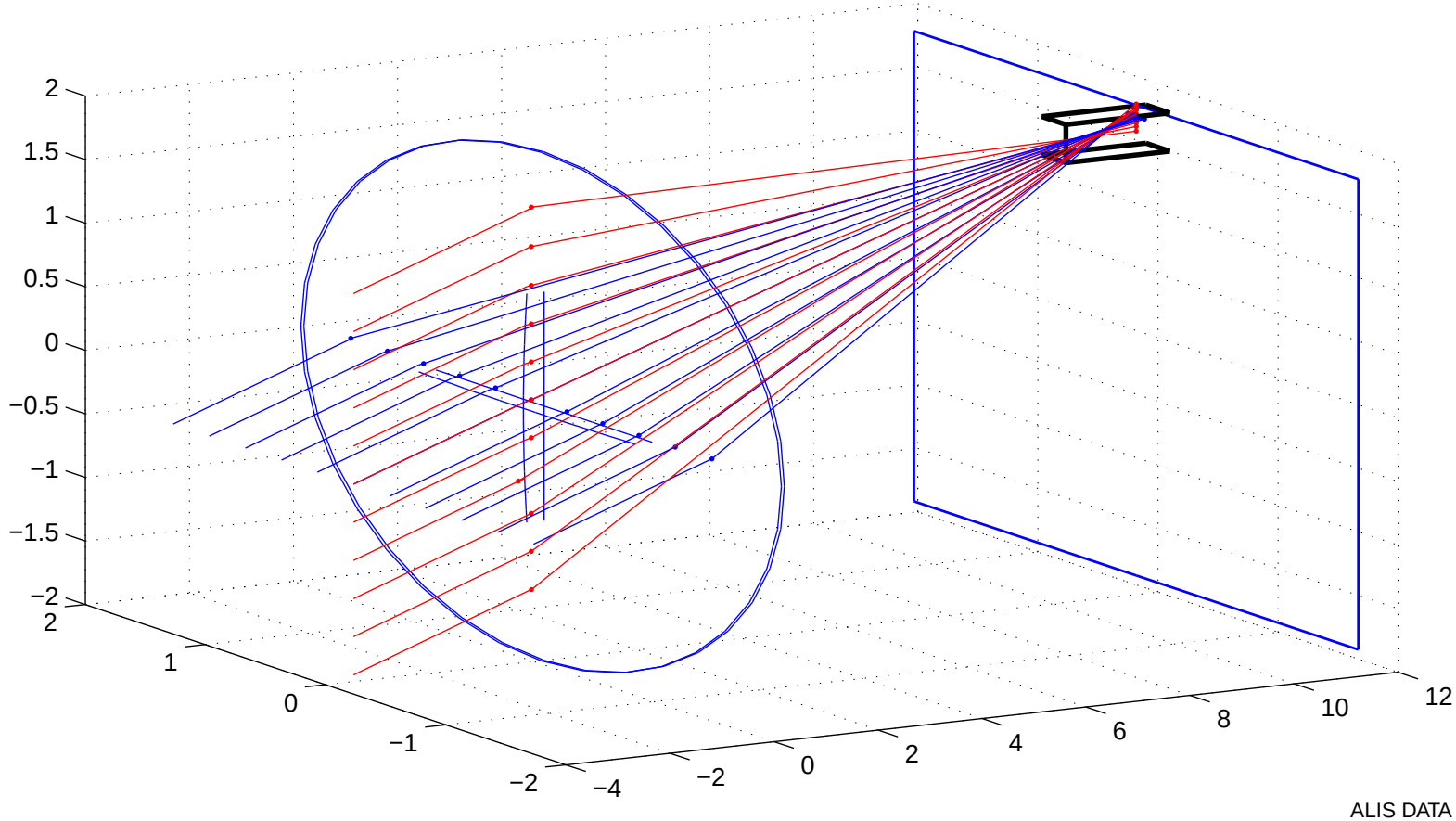
Astigmatism (1)

When the focal surface for the central rays are curved, there is typically problems with astigmatism at larger angles from the optical axis. This is because the meridional (red) and sagital rays have different focal lengths.



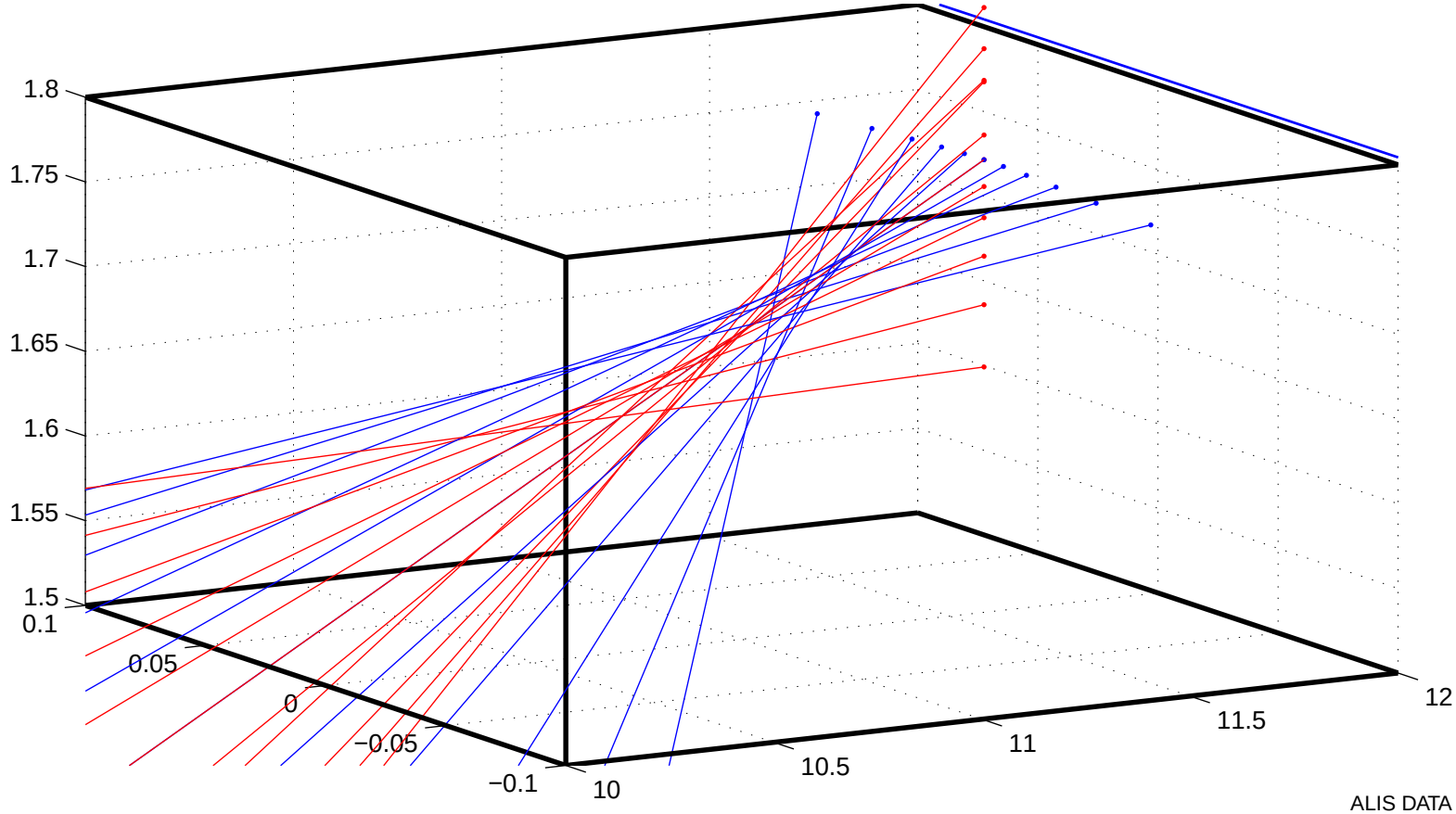
Astigmatism (1)

When the focal surface for the central rays are curved, there is typically problems with astigmatism at larger angles from the optical axis. This is because the meridional (red) and sagital rays have different focal lengths.



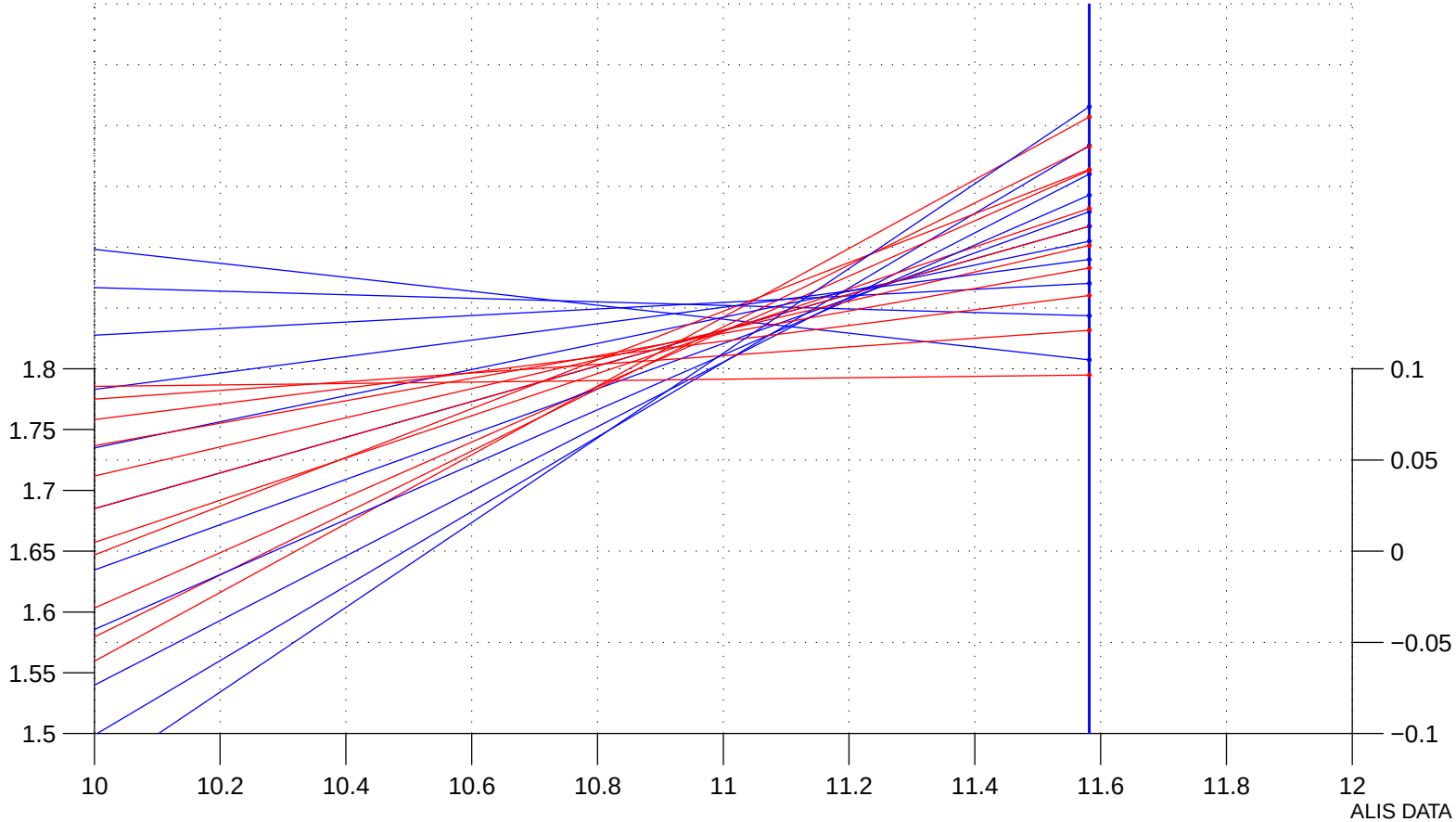
Astigmatism (1)

When the focal surface for the central rays are curved, there is typically problems with astigmatism at larger angles from the optical axis. This is because the meridional (red) and sagital rays have different focal lengths.



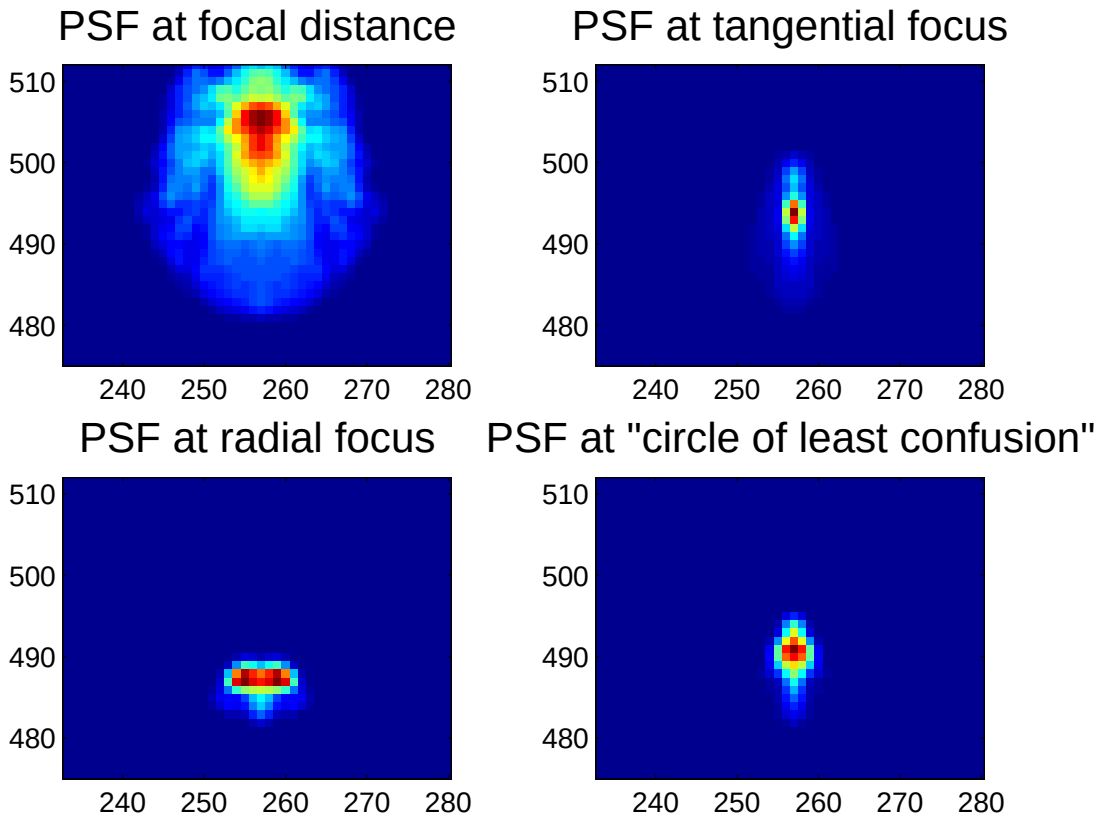
Astigmatism (1)

When the focal surface for the central rays are curved, there is typically problems with astigmatism at larger angles from the optical axis. This is because the meridional (red) and sagital rays have different focal lengths.



Astigmatism (2)

This makes the shape of the point spread function vary in shape with distance from the nominal focal plane.



Since astigmatism is closely related to field curvature this is worsend at large field-of-view angles.

Deconvolution (1) Theory

The observed image I can be considered as the result of a convolution of an ideal image I' with the point spread function of the optics p_{sf}:

$$I(u,v) = \iint_{-\infty}^{\infty} p_{sf}(u,v,x,y) I'(x,y) dx dy \tag{7}$$

When p_{sf} is is the same over the entire image plane we get

$$I(u,v) = \iint_{-\infty}^{\infty} p_{sf}(x-u,y-v) I'(x,y) dx dy = p_{sf} \otimes I' \tag{8}$$

Deconvolution (2)

This convolution we can “simplify” further by using the convolution theorem and move to the the Fourier domain:

$$\hat{I} = \hat{p}_{sf}\hat{I}' \tag{9}$$

Thus the ideal image can be retrieved:

$$\hat{I}' = \hat{I}./\hat{p}_{sf} = \hat{p}^*_{sf}\hat{I}./|\hat{p}_{sf}|^2 \tag{10}$$

For practical situations when there are noise in the images this breaks down:

$$\hat{I}^\sharp = \hat{p}^*_{sf}(\hat{I} + \hat{n})./|\hat{p}_{sf}|^2 = \hat{p}^*_{sf}(\hat{I}./|\hat{p}_{sf}|^2 + \hat{n}./|\hat{p}_{sf}|^2) \tag{11}$$

The noise in the second term will be greatly amplified for frequencies where $|\hat{p}_{sf}|$ is small.

Deconvolution (3)

There is a number of aproaches to adress this issue. The basic idea is to dampen the amplification where $|\hat{p}_{sf}|$ is small or the noise is large.

$$\hat{I}^{\sharp} = \frac{\hat{p_{sf}}^{*}}{|\hat{p_{sf}}|^2 + \gamma S_n / S_I} \hat{I} \tag{12}$$

Where S_n and S_I is the power spectral density of the noise and the image. When those are not known other choises are to use a constant K or the fourier transform of a regularisation function (like Laplace) in place of $\gamma S_n / S_I$.

Deconvolution (Lucy-Richardson)

Iterative method! Not necessarily FFT.

$$I_{n+1} = I_n \frac{I_0}{I_n \otimes p_{sf}} \otimes refl(p_{sf})$$

Deconvolution of a meteor

Since the ALIS imager (was seriously out of focus and) had signifficant astigmatic problems we had to do something about it.

Deconvolution of a meteor

Since the ALIS imager (was seriously out of focus and) had signifficant astigmatic problems we had to do something about it.
Since the PSF varies over the image we could not apply matlabs normal deconvolution functions on the full image. But within the small region around the meteor the PSF varies very little.

Deconvolution of a meteor

Since the ALIS imager (was seriously out of focus and) had signifficant astigmatic problems we had to do something about it.
Since the PSF varies over the image we could not apply matlabs normal deconvolution functions on the full image. But within the small region around the meteor the PSF varies very little.
To estimate the local PSF we can use the stars found in the vicinity. Several stars can be used if we shift them to an overlaping “center of intensity”. This will average out the noise that sneaks in through the estimate of the point-spread-function.

Deconvolution of a meteor

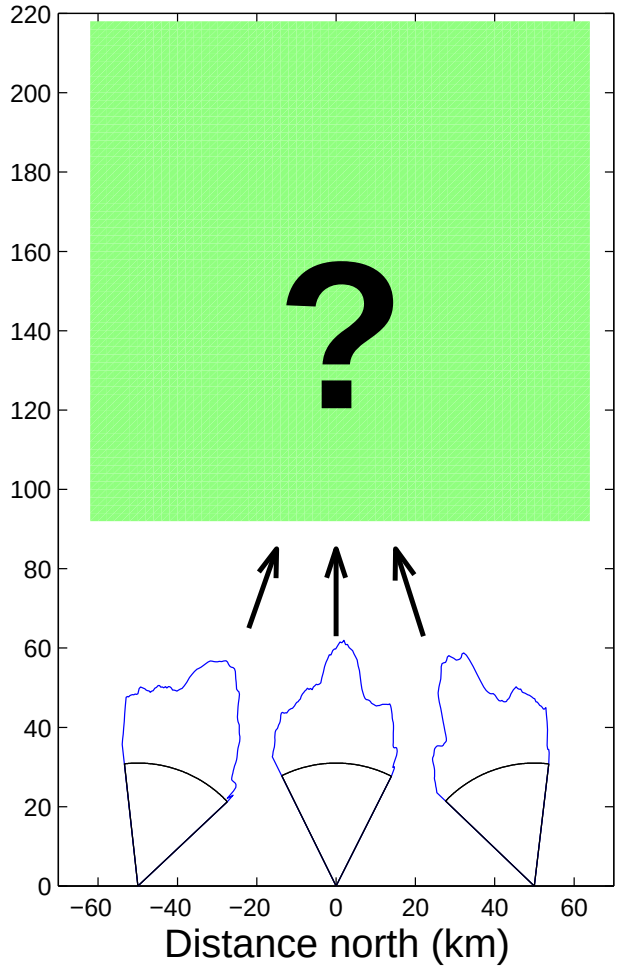
Since the ALIS imager (was seriously out of focus and) had signifficant astigmatic problems we had to do something about it.
Since the PSF varies over the image we could not apply matlabs normal deconvolution functions on the full image. But within the small region around the meteor the PSF varies very little.
To estimate the local PSF we can use the stars found in the vicinity. Several stars can be used if we shift them to an overlaping “center of intensity”. This will average out the noise that sneaks in through the estimate of the point-spread-function.
Then the prefered algos is the ones developed by the people fixing data for Hubble (pre the “correction glasses”).

Tomography

- Theoretical background
- Practical background
  - Forward model
  - Base functions (Voxels and Blobbs)
  - Effects of randomness
- Brief list of solutions
  - Fast Maximum likelihood with A Priori Entropy
  - Algebraich Reconstruction Technique
  - Simpultaneous Itterative Reconstruction Technique
  - Least SQuare (LSQ), Twomey-Tichonov (TT)
  - Backus-Gilbert (BG)
- Tricks of the trade
  - Start guess
  - Relative intensity scaling, Quiet borders
  - Filtering

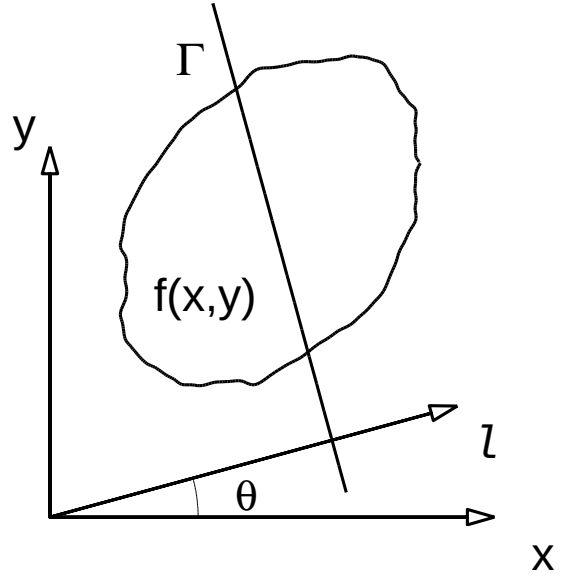
Tomography (Introduction)

Tomography is
the method to determine the internal structure of an object starting from a set of images that in some way carry information of the integral intensity along the line-of-sight *through* the object. For emission tomography, like auroral tomography, the images are made up of the total emission along the lines-of-sight. Here there are three observation sites which record one-dimensional (1-D) images, and an unknown source to be determined.



Tomography, Radons solution (1)

It is not evident that such
inverse problems
have solutions but for two-dimensional (2-D) tomography where $f(x, y)$ is given in a (x, y)-vertical plane, the measured intensities $\mathbf{I}$, corresponding to our images, are given by line integrals



$$\mathbf{I}(l, \theta) = \int_{\Gamma} f(x, y) dl'$$

where θ and l determine the position of the detector and its line-of-sight Γ, and dl' is a length element along Γ.

Tomography, Radons solution (2)

It is possible to obtain an exact and unique solution [**?**] of the source function $f(x, y)$ if and only if $I(l, \theta)$ is known for all rays Γ that cut the region where $f(x, y) > 0$. That is for an angle θ we have to move the detector along l so that measurements are made for all rays that go through the object. The solution can be written as [**?**]:

$$f(r\cos\theta, r\sin\theta) = \frac{1}{2\pi^2} \int_{-\pi/2}^{\pi/2} \int_{-\infty}^{\infty} \frac{\partial I(l, \theta')}{\partial l} \cdot \frac{dl d\theta'}{r \cdot \sin(\theta - \theta') - l}$$

Radons solution extended to 3-D

For the three-dimensional tomographic problem exact solutions can be obtained if there is an observation point from which our image projections are made on every plane that cuts the region where the function $f(\bar{r}) > 0$ [**??**]. There exist a number of analytical algorithms that determine f from a set of image projections which closely satisfy the above condition [**??**].

Problems with inverse problems

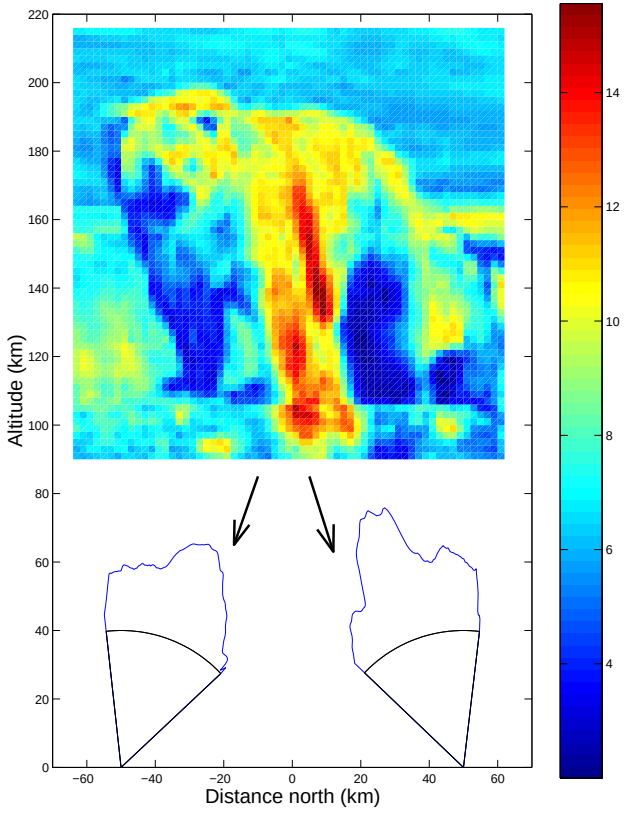
However, in most cases these tomographic problems are ill-posed, meaning that they are sensitive to noise in the measurements or have either an infinite number of exact solutions or no exact solutions. With some regularization schemes, i.e. physical or mathematical constraints, the problem can, in a restricted sense, be transformed into a well-posed problem [**?**]. Auroral and airglow tomography from ground-based multi-station measurements is ill-posed because only a few projections/images of the aurora are available and that the image data are all taken from below the object and thus are close to being linearly dependent.

Curse of dimensionality

The tomographic problem of ALIS is three-dimensional, and all performance, resolution and error analysis on the actual geometry suffers badly from the curse of dimensionality. That is, the total size of the tomographic problem is proportional to the total number of pixels in the images times the total number of volume elements in the retrieved volume distribution. For a two-dimensional tomographic problem with M voxels per side, the number of unknowns is M^2 projected onto n images with N pixels per one-dimensional image; the size of the problem is $M^2 \cdot (n \cdot N)$. For a three-dimensional tomographic system the total number of pixels is $n \cdot N^2$ and the number of voxels rises to M^3, giving a total size of the problem of $(M^3) \cdot (n \cdot N^2)$. For systems studying the ionosphere the typical number of stations appears to be approximately 5 and typical number of pixels is 1000 per image dimension.

Tomography, Forward model

In order to be able to solve the tomographic problem it is necessary to have good knowledge about the forward problem or forward model. That is, we must know what fraction of the photons emitted in a volume element will be detected and where in the image the intensity will appear. In this model case we have to know how the auroral distribution in a slice is projected down onto the images. The mathematical notation for this is:



$$\bar{I}_s = F_s(\bar{f})$$

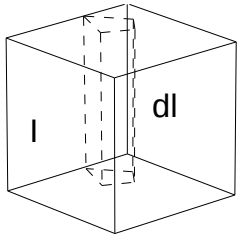
Forward model (2)

Continuing with the forward model $\bar{I}_s = F_s(\bar{f})$, where $\bar{I}_s$ is the image from stations s, $\bar{f}$ is the two-dimensional source function and F_s is the forward model for station s. Here it is sufficient to note that it is important to know the pixel lines-of-sight and the sensitivities of the cameras. Since the cameras used by ALIS have a linear response to intensity and that the emissions currently measured by ALIS are all optically thin, the forward model can be simplified to a matrix multiplication:

$$\bar{I}_s = \bar{\mathbf{F}}_\mathbf{s} \cdot \bar{f}$$

where I_s is the image from station s and $\bar{\mathbf{F}}_\mathbf{s}$ is the transfer matrix from the voxel space $\bar{f}$ to the image.

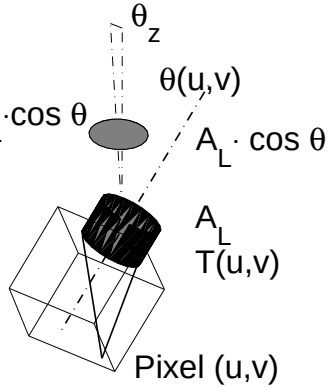
Forward model, the details (1)



$I_1 = I \cdot dV$

$I_2 = I_1 \cdot d\Omega \cdot l^2$  $d\Omega(u,v) \cdot l^2$

$I_3 = I_2 \cdot e^{-\alpha(\lambda)/\cos\theta_z}$



$I_4 = I_3 \cdot A_L \cdot \cos\theta$

$I_5 = I_4 \cdot T$

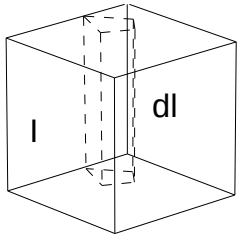
$I_6 = I_5 \cdot t$

$I_8 = A(u,v) \cdot I_6 + Z(u,v) + C(u,v)$

There are several factors
that have to be carefully treated in order
to obtain an accurate forward model.

1. Voxel – pixel field-of-view intersection
2. Pixel field-of-view
3. Atmospheric absorption
4. Effective collecting area
5. Transmission of optics
6. Variation in exposure time
7. Pixel sensitivity

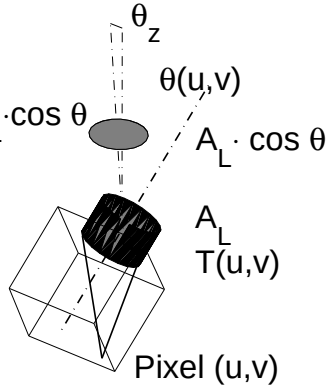
Forward model, the details (1)



$I_1 = I \cdot dV$

$I_2 = I_1 \cdot d\Omega \cdot l^2$ $\quad d\Omega(u,v) \cdot l^2$

$I_3 = I_2 \cdot e^{-\alpha(\lambda)/\cos\theta_z}$



$I_4 = I_3 \cdot A_L \cdot \cos\theta$

$I_5 = I_4 \cdot T$

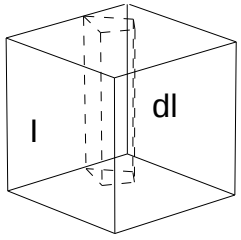
$I_6 = I_5 \cdot t$

$I_8 = A(u,v) \cdot I_6 + Z(u,v) + C(u,v)$

There are several factors
that have to be carefully treated in order
to obtain an accurate forward model.

1. Voxel – pixel field-of-view intersection
2. Pixel field-of-view √
3. Atmospheric absorption
4. Effective collecting area √
5. Transmission of optics
6. Variation in exposure time
7. Pixel sensitivity √

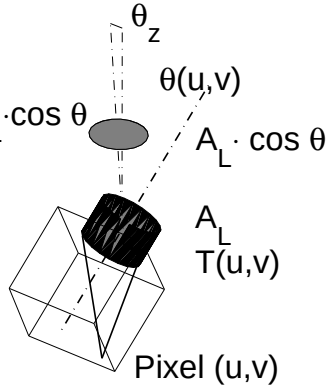
Forward model, the details (1)



$I_1 = I \cdot dV$

$I_2 = I_1 \cdot d\Omega \cdot l^2$  $d\Omega(u,v) \cdot l^2$

$I_3 = I_2 \cdot e^{-\alpha(\lambda)/\cos\theta_z}$



$I_4 = I_3 \cdot A_L \cdot \cos\theta$

$I_5 = I_4 \cdot T$

$I_6 = I_5 \cdot t$

$I_8 = A(u,v) \cdot I_6 + Z(u,v) + C(u,v)$

There are several factors
that have to be carefully treated in order
to obtain an accurate forward model.

1 Voxel – pixel field-of-view intersection

3 Atmospheric absorption

5 Transmission of optics

6 Variation in exposure time

Variation in exposure time

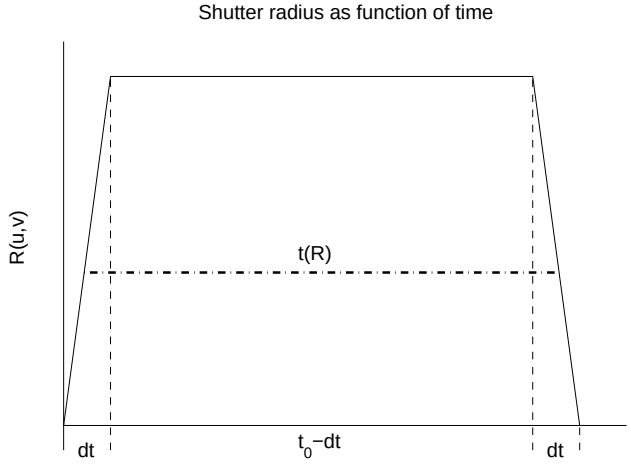
The ALIS cameras are equipped with iris shutters that open and close radially. The open-close characteristics of the cameras are not known, but the time required to open and close the shutter fully is of the order of magnitude of 10 ms (Brändström private communication).
With exposure times between 1.0 and 10 s, the relative error in pixel exposures varies from 1 to 0.1 % and the accuracy of the exposure time is good enough even without corrections but for exposure times shorter than 1.0 s the shutter characteristics start to become important.

Variation in exposure time (2)

A first order model of the exposure time variation as a function of image coordinate is given by:

$$t(u,v) = t_0 + dt - 4\,dt\,\sqrt{\frac{(u-u_0)^2}{u_s^2} + \frac{(v-v_0)^2}{v_s^2}}$$

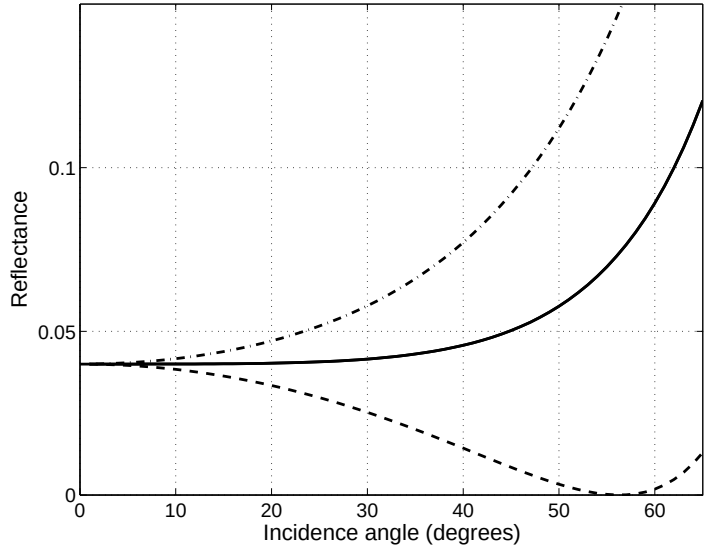
where t_0 is the nominal exposure time, dt is the open and close time of the shutter, u and v are the horizontal and vertical image coordinates, respectively, u_0 and v_0 are the horizontal and vertical image coordinates of the centre of the shutter, and u_s and v_s are the horizontal and vertical image sizes, respectively.



Transmission of the optics

As a zeroth order assumption, the transmission of the optics is assumed to be constant for all angles θ relative to the optical axis in the full camera field-of-view. For cameras with a narrow field-of-view this can be considered to be a justifiable assumption.

Whether this is the case for optics with a medium field-of-view, $54°$, remains an slightly open question. However looking on the typical reflectance curves for simple glass (with Brewster angles around $55°$) it appears as if it is a good enough approximation.



Atmospheric absorption

The emission lines observed by ALIS at 4278, 5577, 6300 and 8446 Å are all optically thin in the ionosphere, i.e. photons emitted are not (self-)absorbed in the region of emission. The amount of absorption depends on the length of the light path through the stratosphere. For moderate zenith angles, $\theta_z < 70°$, the curvature of the earth can be ignored and with a zenith optical depth $\alpha(\lambda)$ that is dependent on wavelength, the absorption is:

$$I(\theta) = I(0)e^{-\alpha(\lambda)/\cos\theta_z}$$

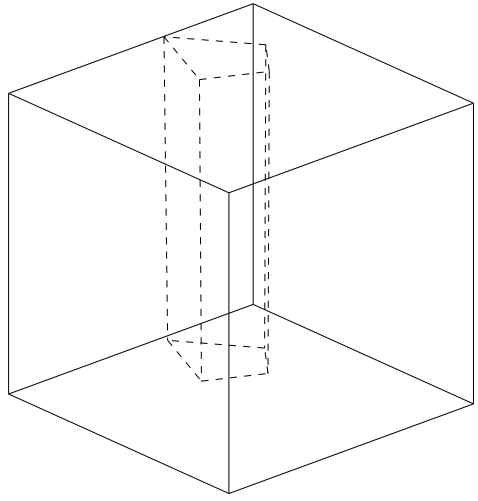
λ (Å)	α
4278	0.236
5577	0.116
6300	0.081
8446	0.080

The optical depths by **?** that have been used in the current forward model are given in the table.

Voxels

First a few words about voxels.
The simplest set of base functions used in tomographic problems are voxels. It is just a fancy name for a 3-D box with one value for intensity. When voxels are used the number of voxels must be large enough to give a good enough approximation to gradients and curvatures. With to few voxels there will be consideral stair-casing. A good feature of the non-overlapping voxels are that for positive source function $(f(\bar{r}) \geq 0, \text{all } \bar{r})$ the best fit is positivite for all voxels $(f_i \geq 0, \text{all } i)$.

Voxel–pixel fov intersection



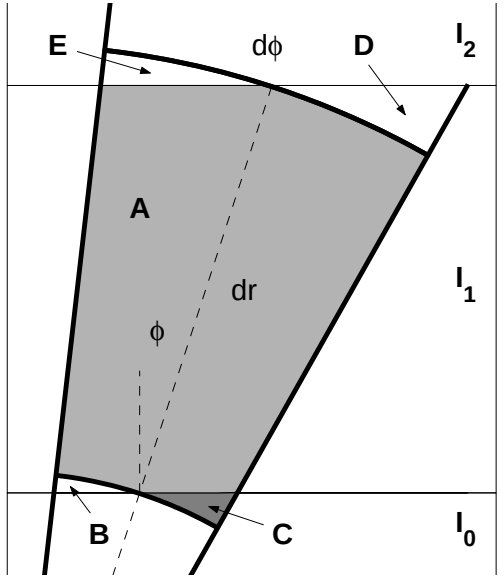
The intersection between a voxel and the pixel field-of-view is a body with between four and ten faces. In order to avoid having to calculate the volume of the complex body with all its bounding planes, the intersection volume dV can be approximated by the area spanned by the pixel field-of-view, $|\bar{r}|^2 d\Omega$, at distance, $|\bar{r}|$, to the voxel multiplied by the pixel line-of-sight intersection length dr:

$$dV = |\bar{r}|^2 dr d\Omega$$

This approximation is good enough when the intensity does not vary very much between neighboring voxels. As can easily be seen for the corresponding 2-D problem:

Voxel–pixel fov intersection (2)

Consider a narrow circle sector intersecting three voxels with intensities $I_0, I_1,$ and I_2, respectively. For a voxel representation to be good the function should have small spatial variations compared to the size of the voxels, i.e. neighbouring voxels should have intensities that do not differ much:



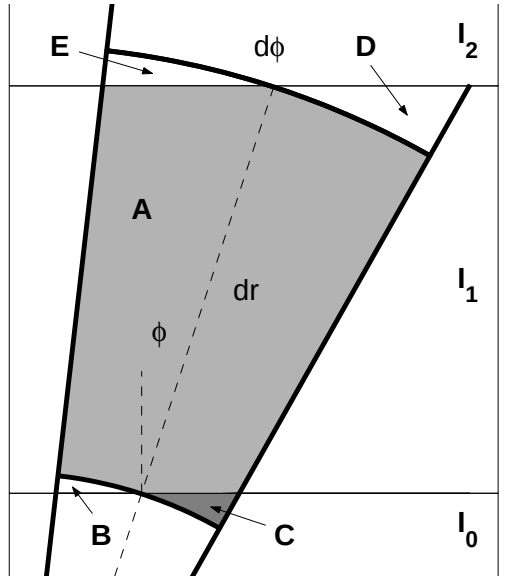
$$I_0 = I_1 + dI_0$$

$$I_2 = I_1 + dI_2$$

Where dI_0 and dI_2 are small compared to I_1.

Voxel–pixel fov intersection (3)

The true contribution from voxel 1 is $I_1 (A_A + A_B + A_D)$, and the approximation is that the contribution is given by $I_1 (A_A + A_C + A_E)$. In order to calculate the difference between the exact and the approximate contributions, we calculate the true and approximate contributions from the region $A + B + C + D + E$:



$$I_{true} = A_A \cdot I_1 + A_B \cdot I_1 + A_C \cdot I_0 + A_D \cdot I_1 + A_E \cdot I_2$$

$$I_{app} = A_A \cdot I_1 + A_B \cdot I_0 + A_C \cdot I_1 + A_D \cdot I_2 + A_E \cdot I_1$$

Voxel–pixel fov intersection (4)

The difference between the true and the approximate contribution are:

$$\Delta I = I_{true} - I_{app} = dI_0 \left(A_C - A_B\right) + dI_2 \left(A_E - A_D\right)$$

After doing some geometry and algebra the areas of the different regions can be obtained:

$$\begin{aligned}
A_B =& r^2\, d\phi/2 - r^2 \cos^2\phi \cdot (\tan\phi - \tan(\phi - d\phi/2))/2 \\
A_C =& r^2 \cos^2\phi \cdot (\tan(\phi + d\phi/2) - \tan\phi)/2 - r^2\, d\phi/2 \\
A_D =& (r + dr)^2 \cos^2\phi \cdot (\tan(\phi + d\phi/2) - \tan\phi)/2 - (r + dr)^2\, d\phi/2 \\
A_E =& (r + dr)^2\, d\phi/2 - (r + dr)^2 \cos^2\phi \cdot (\tan\phi - \tan(\phi - d\phi/2))/2
\end{aligned}$$

Voxel–pixel fov intersection (5)

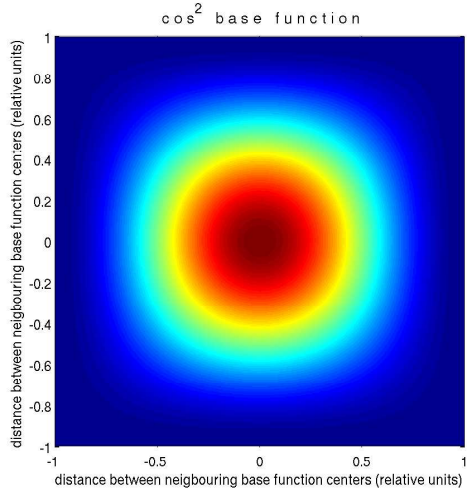
Combining the equations from the previous slide and expanding to the third order in $d\phi$, gives:

$$\Delta I = \frac{1}{12} \frac{d\phi^3 \left(2\,\cos^2(\phi) - 3\right) \left(\left(dI_2 - dI_0\right) r^2 + 2\, dI_2\, r dr + dI_2\, dr^2\right)}{\cos^2(\phi)}$$

Here it is seen that the approximate contribution, I_{app}, is correct to within the third order in $d\phi$ and the second order in (dI, dr). With values typical for ALIS and auroral tomography this gives relative errors that are of the order $5 \cdot 10^{-6}$.

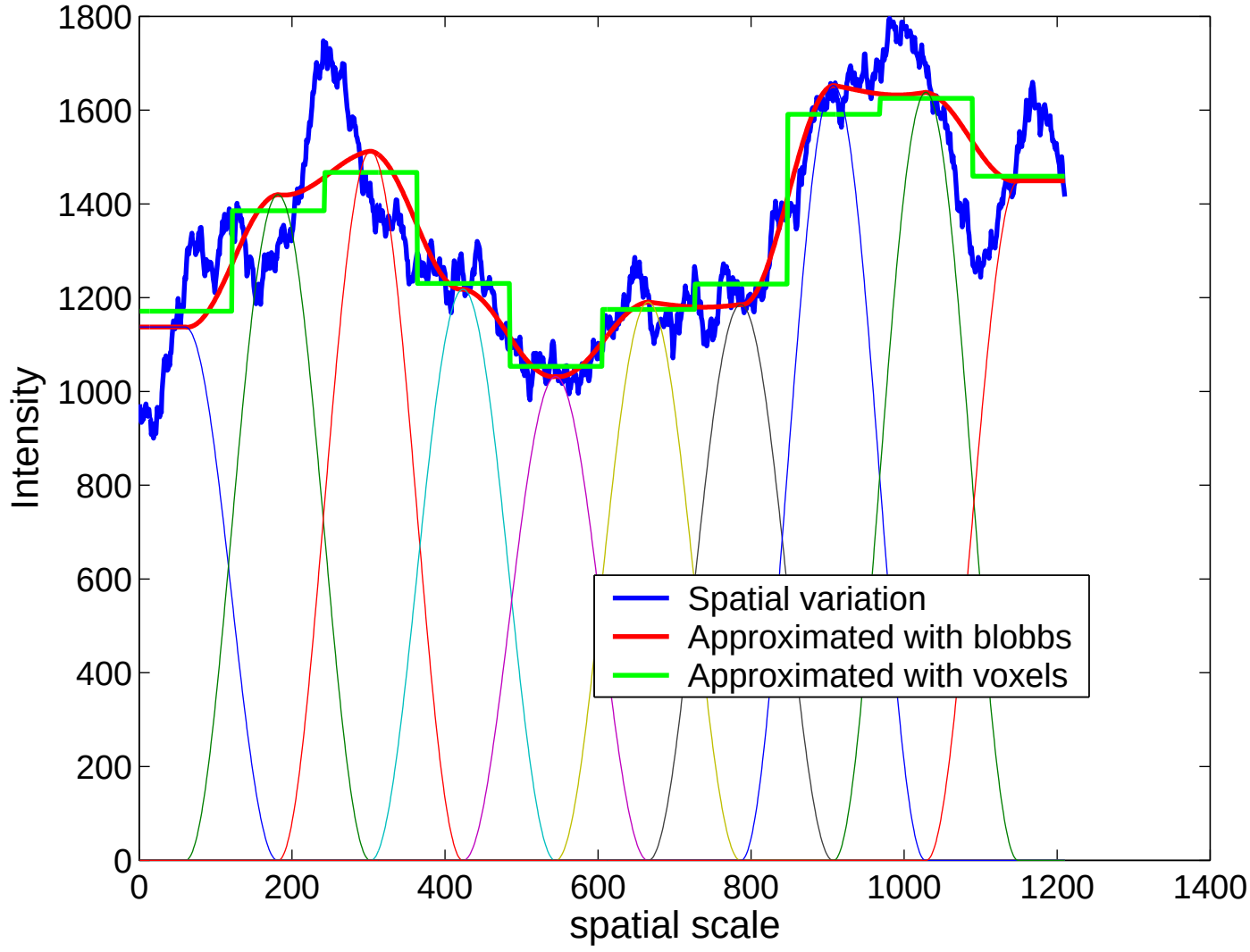
Blobbs

By using smooth overlapping base functions it is possible to reduce stair-casing and get better fits to gradients and sharp curvatures. If the overlap is only to the center of the nearest neighbours still the possitivity requirement is met.



To get complete space filling it is not possible to use spherically cylindrical blobbs, and to make the projection from blobbs to the image plane fast it is nessecary for to have blobbs that have almost spherically symmetric blobbs. A good compromize is to use basefunctions with $\cos^2$ shape in all three directions. This give a complete space filling and negligible perspective effects while at the same time be smooth.

Voxels/Blobbs – 1-D comparison

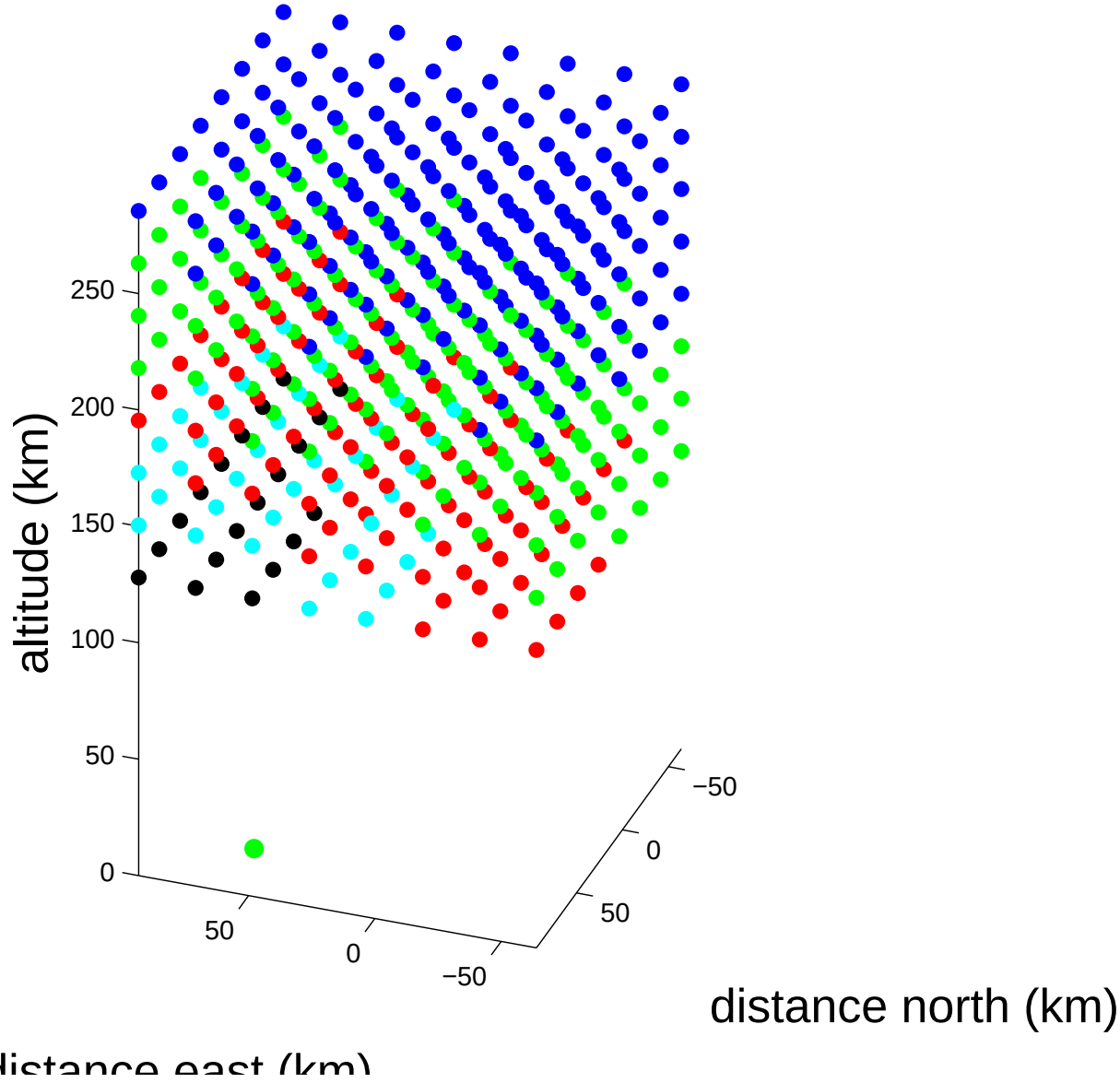


Fast projection

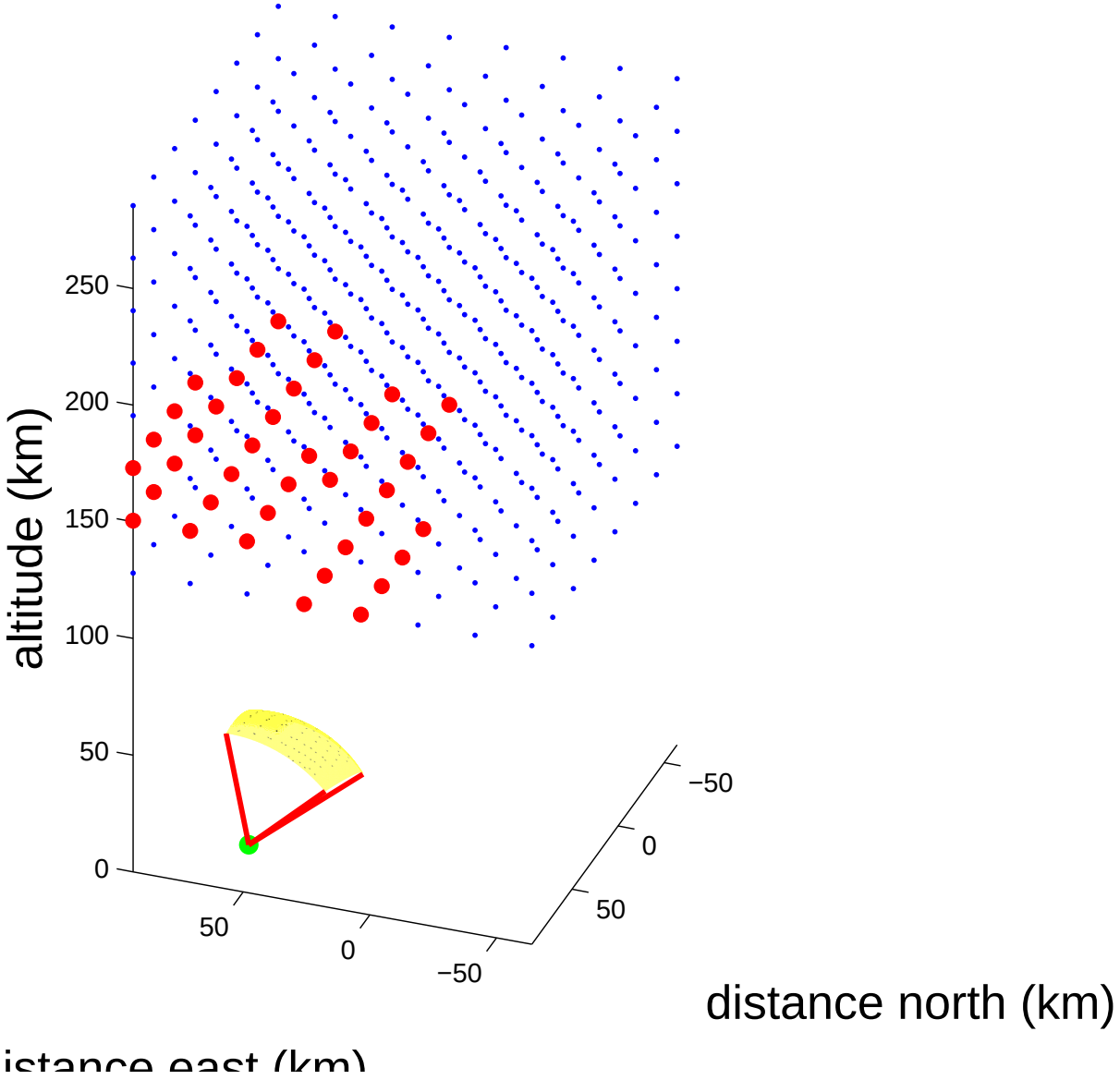
Further blobbs make for really fast approximate (and thanks to its better fit to continous functions) better projection functions **?**. This use the fact that we can obtain accurate enough projections by making 2-D filtering in the image plane. This ignores the fact that the blobbs are not spherically symmetric which is a small approximation. The projection function then is simplified to:

1. Divide the blobbs into size groups depending on their distance to the camera.
2. Project center points of the blobbs to the image plane.
3. Filter the point projections with the corresponding 2-D footprint of the blobbs.
4. Possible multiply with sensitivity factors $(p2p, d\Omega, ...)$

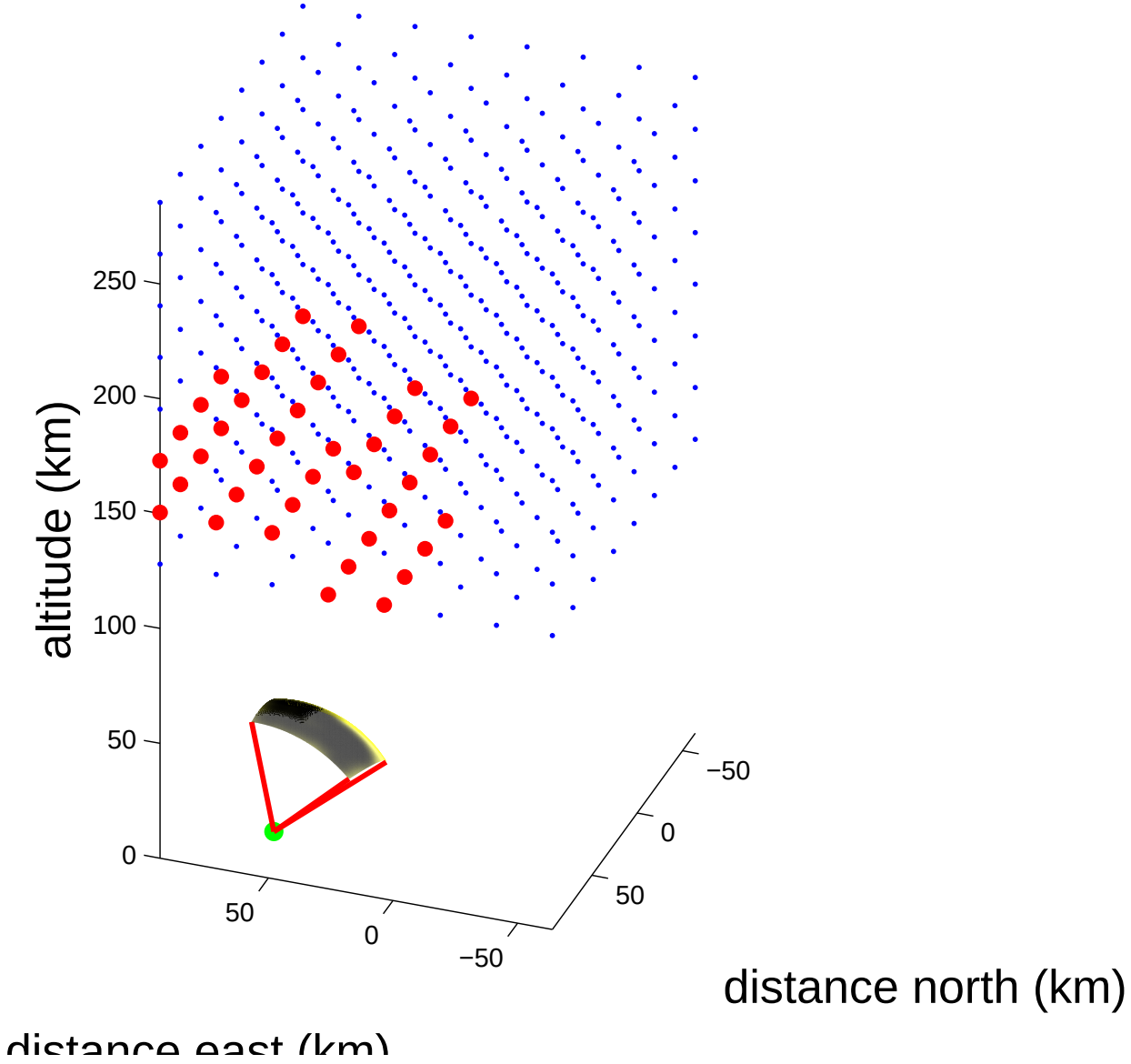
Fast projection (2)



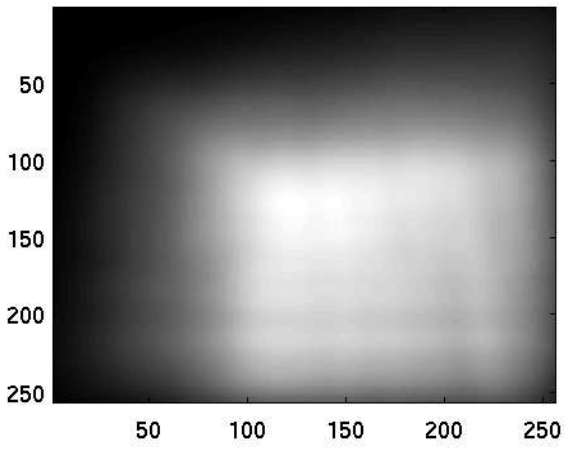
Fast projection (2)

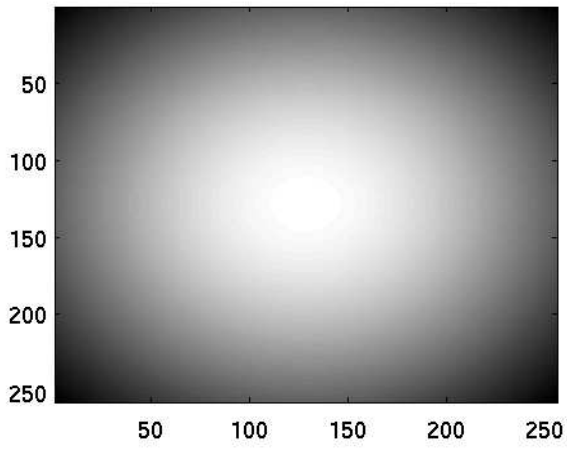


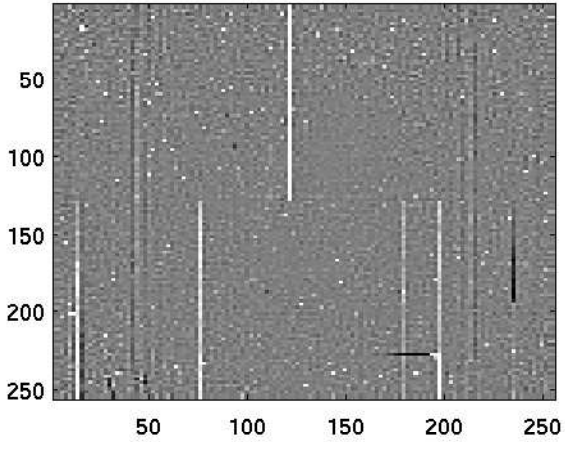
Fast projection (2)

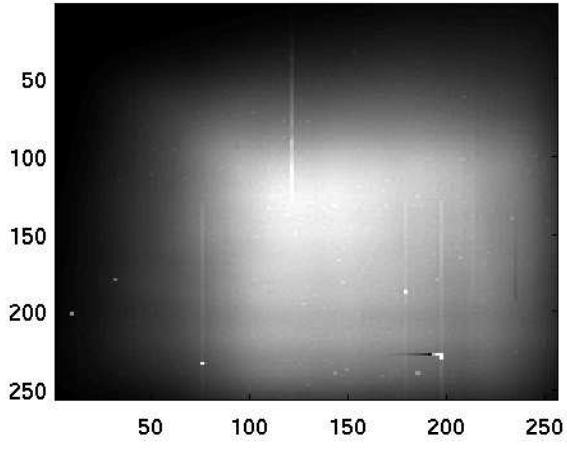


Fast projection (2)









Voxels and Blobbs a summary

Since Blobbs gives both better and faster projections there is no apparent reason for using voxels. But since blobbs still have no easy implementation where their spatial size varies, there is a last haven for voxels. Provided we work with plaid grids it is no big problem having just about any variation in the size of the voxels.
Otherwise blobbs are a superior choise of basis functions.

Forward model as a random process

If we solve the tomographic problem analytically, we will encouner the problem that the forward model is not a deterministic process, but rather a random process

$$\bar{I}_s = \bar{\mathbf{F}}'_{\mathbf{s}} \cdot \bar{\phi}$$

where a component ϕ_j of $\bar{\phi}$ is a sample from one random variable with a probability distribution function that depends on several processes from the emission of photons in voxel j to the detection in the image.

Bayes' solution

When the imaging process is a stochastic one (for which the probability of getting a set of measured images I_s from a source distribution f is:

$$P(\bar{I}_{all}|\bar{f}) = \mathbb{P}(\bar{\mathbf{F}}_{all}, \bar{f})$$

where $P(\bar{I}_{all}|\bar{f})$ is the conditional probability of getting the measured images $\bar{I}_{all}$ given the source distribution $\bar{f}$) the solution to the inverse problem is to find the source distribution $\bar{f}'$ that is most probable given the measurements $\bar{I}_{all}$ and considering equation (**??**) and Bayes' rule the resulting problem is

$$\bar{f}' : P(\bar{f}'|\bar{I}_{all}) = \max \mathbb{P}(\bar{\mathbf{F}}_{all}, \bar{f})\, \mathbb{P}(\bar{f})/\mathbb{P}(\bar{I}_{all})$$

which is the maximum likelihood solution of the inverse problem.

FMAPE iterative solution for Poisson stat

Several more advanced iterative schemes exist and the technically interested reader is directed to **?**.

ART - simple iterative solution

The Algebraic Reconstruction Technique (ART) is a simple and robust iterative reconstruction solution. The working principle of ART is to take an initial guess $f^0(\bar{r})$ and project that guess down to an image [**?**]:

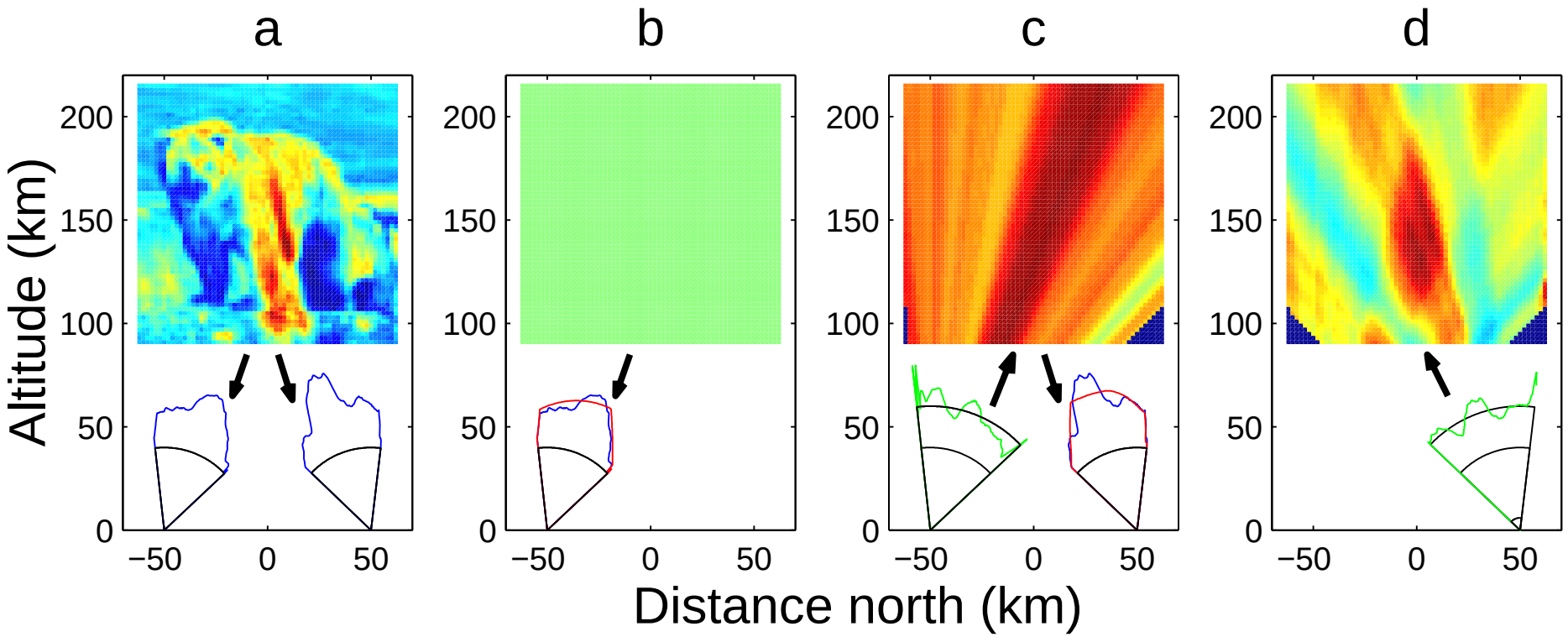
$$I_s^0 = F(f^0)$$

Then the solution f^q is updated according to:

$$f^{q+1}(\bar{r}) = \frac{f^q(\bar{r})}{N} \cdot \sum_{u,v} \frac{I_s(u,v)}{I_s^q(u,v)}$$

This is a modified multiplicative ART update where the update is a weighted average of the ratios of the measured image I and I^q for the pixels (u,v) whose lines-of-sight intersect the voxel at $\bar{r}$.

ART (2)



SIRT - another simple solution

ART is a reconstruction scheme that converges fast but is sensitive to noise in the images [**?**]. A modified version is SIRT for which the current guess/solution f^q is projected down to all stations at once and the voxel intensity in the reconstruction is updated with the average ratio of pixels from all stations.

$$f^{q+1}(\bar{r}) = \frac{f^q(\bar{r})}{N} \cdot \sum_s \sum_{u,v} \frac{I_s(u,v)}{I_s^q(u,v)}$$

This makes the method more stable to noise.

LSQ - solution for Normal distributed sta

TT - damped least squares

Backus-Gilber - cunning stable solution

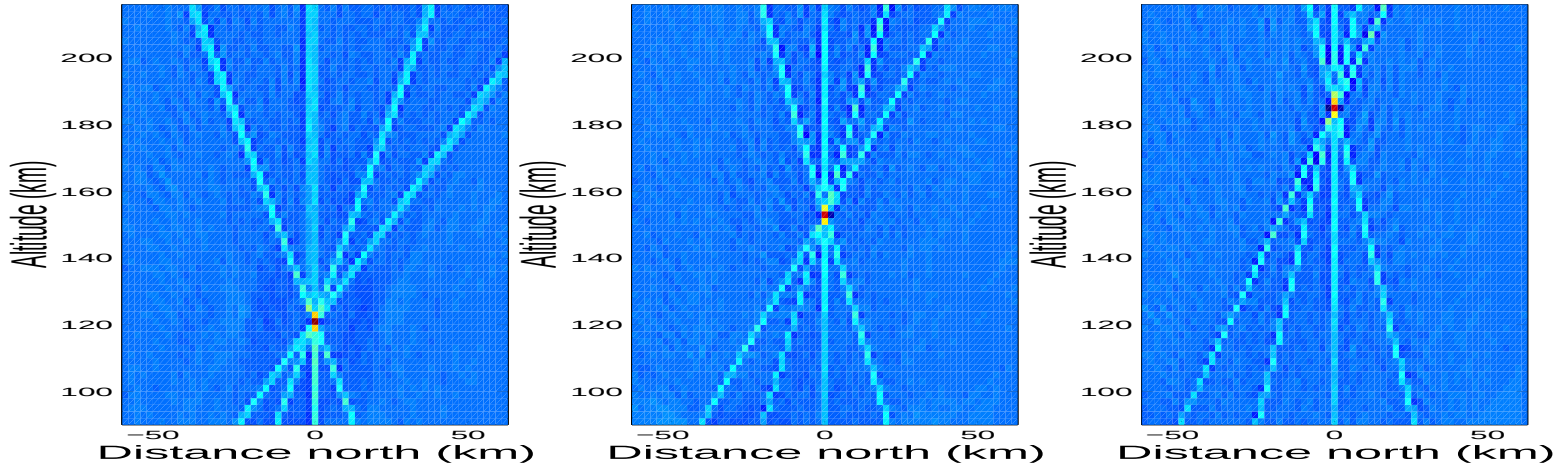
Stopping criteria

Resolution (1)

By using SVD decomposition of the combined forward models $\bar{\mathbf{F}}_{all} = \bar{\mathbf{F}}_1 \ldots \bar{\mathbf{F}}_s$ for stations s at 50 km north, 0, 50 and 100 km south, it is possible to obtain the full spatial resolution. Given the transfer matrix $\bar{\mathbf{F}}_{all}$ from the voxel space down to all images with a singular value decomposition $\bar{\mathbf{F}}_{all} = \bar{\mathbf{U}}_{\mathbf{p}} \cdot \bar{\mathbf{\Lambda}}_{\mathbf{p}} \cdot \bar{\mathbf{V}}_p^T$, the maximum spatial resolution is given by the resolution matrix $\bar{\mathbf{V}}_p \cdot \bar{\mathbf{V}}_p^T$ [**?**].
In order to suppress noise in the reconstruction, only the p singular values larger than 1 in the singular value decomposition of $\bar{\mathbf{F}}_{all}$ above are kept.

Resolution (2)

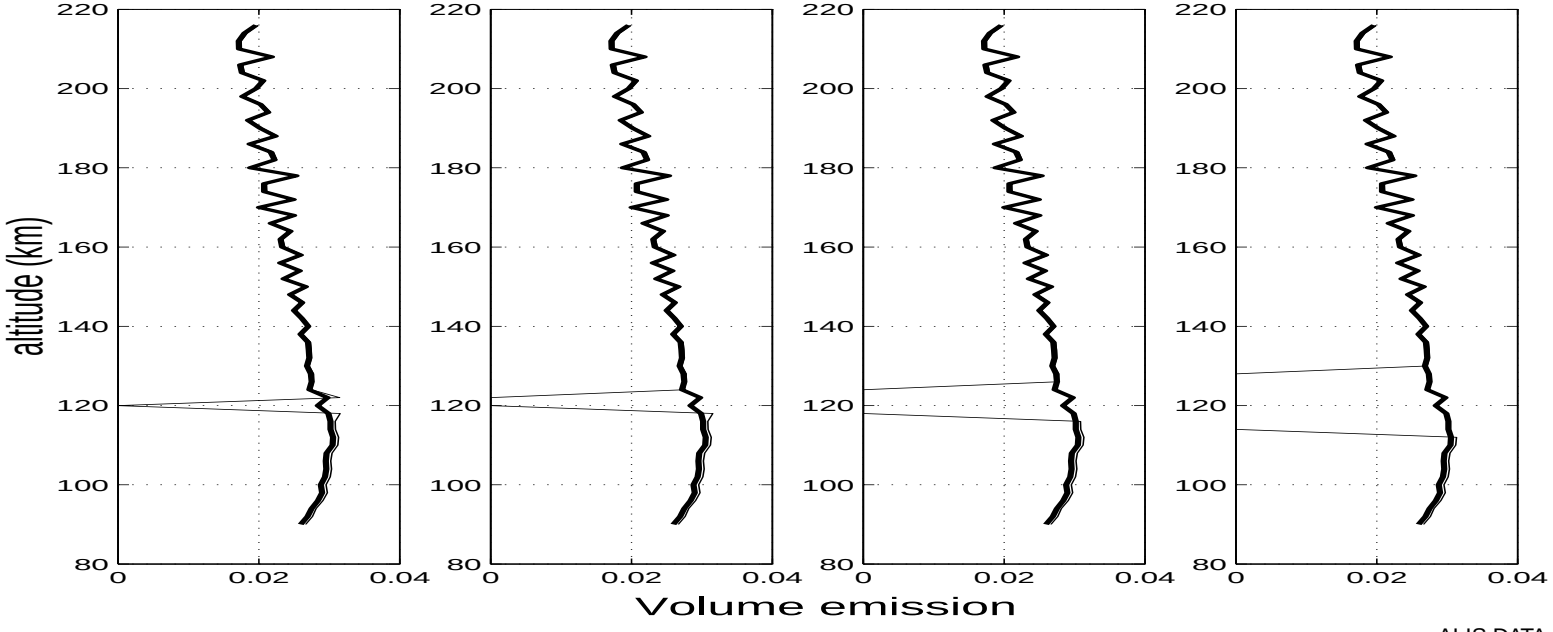
To illustrate the result of the SVD analysis, we calculate the point spread of three input points at altitudes of 120, 152 and 184 km central in the slice. By doing this we get the variation of the point spread function with altitude.



As can be seen the point is accurately retrieved in position but there are star-like traces in the directions away from and towards the stations.

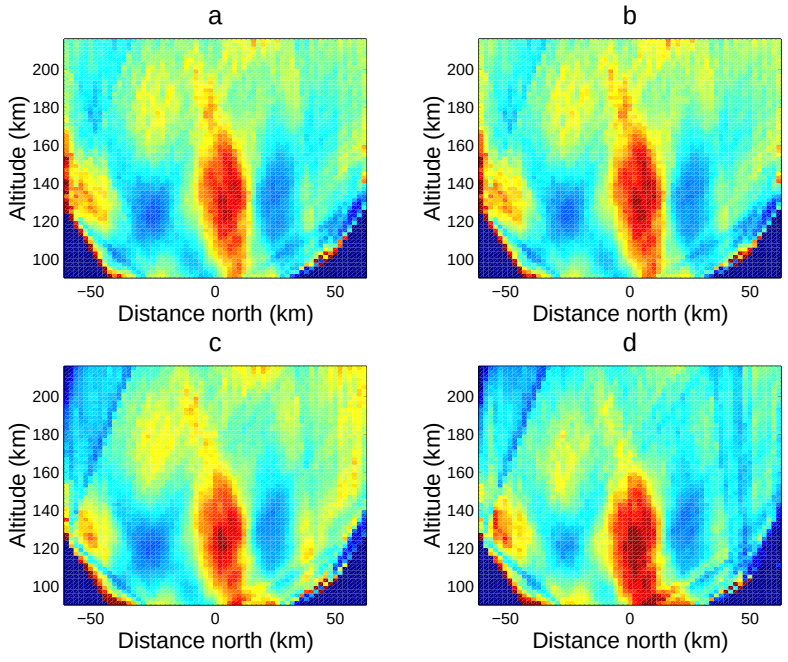
Resolution (3)

To show whether it is possibile to retrieve *internal* structures in the aurora with a ground-based system it is enlightening to consider a simple volume distribution with a small hole and then calculate the “hole retrieval” in the same way as the point spread was calculated above. In the vertical retrieval of this test, where the simple model function is the first eigenvector with the largest eigenvalue, holes with vertical sizes of 2, 4, 8 and 16 km are used.

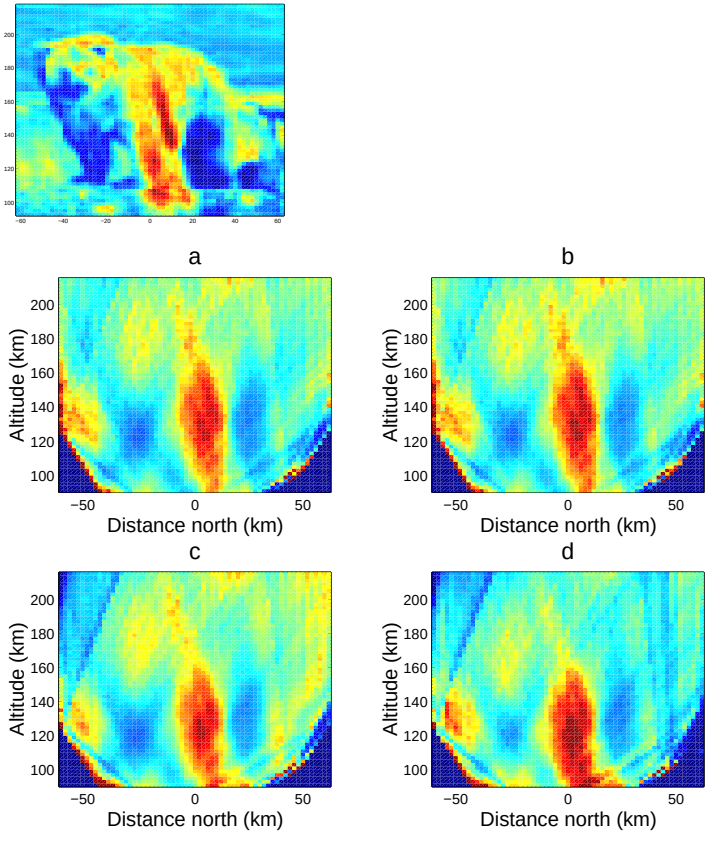


Error sensitivity

Model test of the algebraic reconstruction technique (ART) solution of the tomographic problem. Upper-left panel is a reconstruction from perfect projections with a 1 % noise level. Upper-right panel is a reconstruction from projections with 3 % sensitivity errors. Lower-left panel is a reconstruction from projections that have 0.5° rotational errors. Lower-right panel is a reconstruction from projections with errors in both sensitivity and rotation.



Error sensitivity (2)



As can be seen, the errors in the reconstruction increase already at these low levels of uncertainties in the forward model. Thus, in order to obtain reconstructions of the aurora with sufficient quality for making further analysis worthwhile, it is of vital importance to achieve the necessary accuracy of the forward model.

Quiet borders

Oftentimes edges causes problems in our tomographic reconstructions. To somewhat limit this problem it is sometimes helpfull to set the ratio $I_s(u,v)/I_s^q(u,v)$ to unity in a frame of a few up to e few tens of pixels before updating the solution. This way the impact from all images/projections is limited along their edges.

Relative intensity scaling

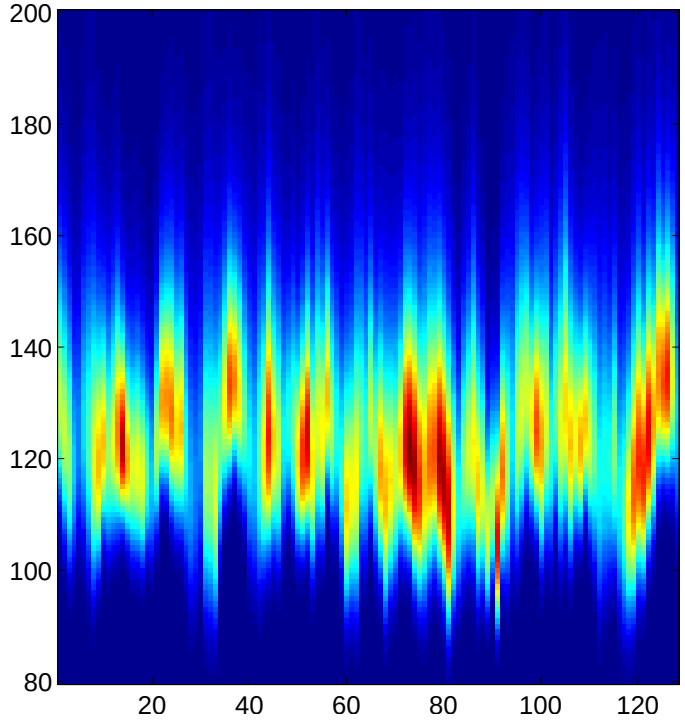
When one or more imagers have unknown or uncertain absolute sensitivity it is still possible to use the information in that data.
Assuming that the total intensity in a region of the current projection is correct we can use that to obtain a “temporary absolute intensity” and scale the image/projection ratio accordingly. This way the image does not change the total intensity of the reconstruction much but its spatial variation contributes to the *shape*.

Filtering - proximity

$$\tilde{I}(i,j,k) = \frac{\sum_k I(j,i,k) \sum_m \sum_n I(j-n,i-m,k) w(n,m)}{\sum_k \sum_m \sum_n I(j-n,i-m,k) w(n,m)}$$

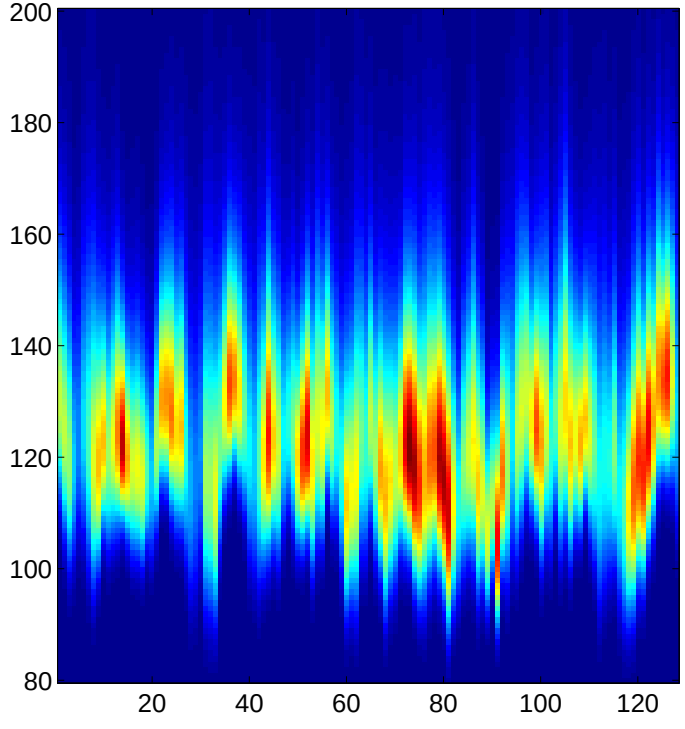
Filtering - proximity

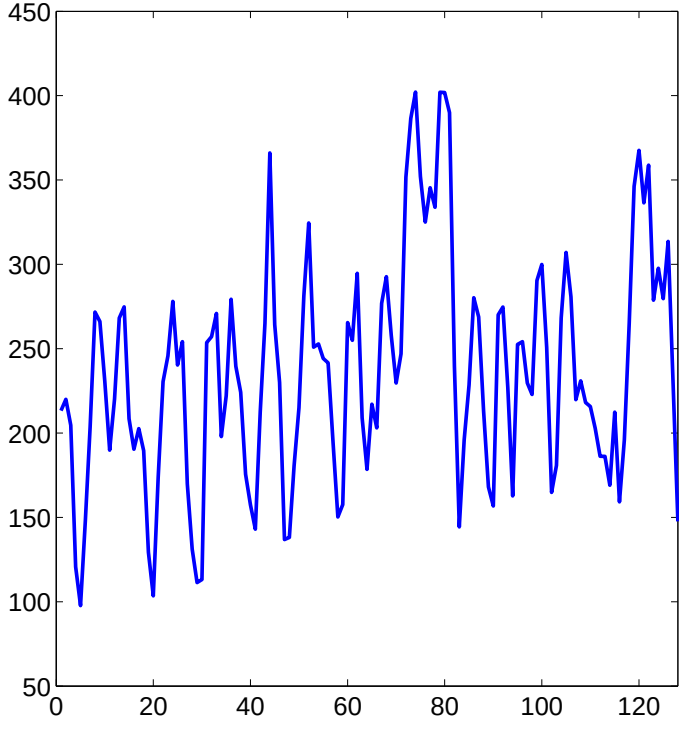
$$\tilde{I}(i,j,k) = \frac{\sum_k I(j,i,k) \sum_m \sum_n I(j-n, i-m, k) w(n,m)}{\sum_k \sum_m \sum_n I(j-n, i-m, k) w(n,m)}$$



Filtering - proximity

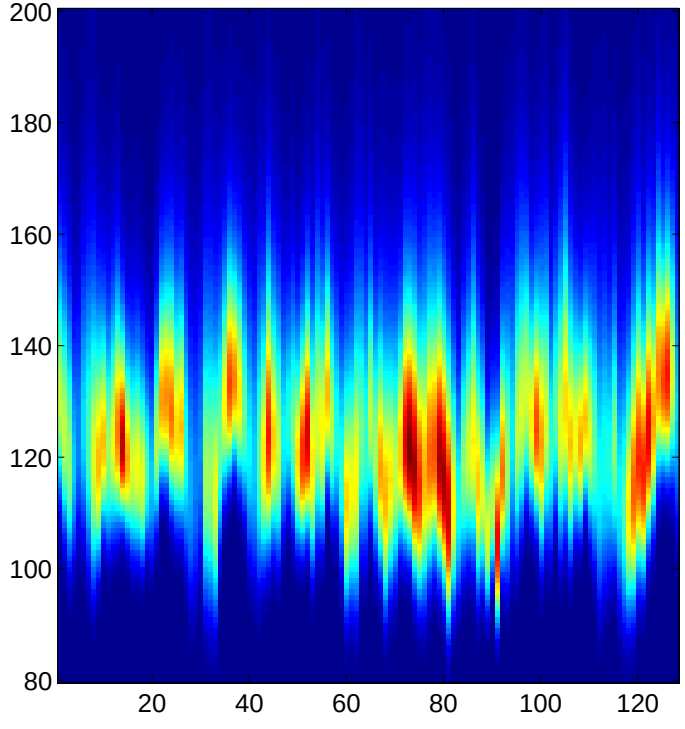
$$\tilde{I}(i,j,k) = \frac{\sum_k I(j,i,k) \sum_m \sum_n I(j-n, i-m, k) w(n,m)}{\sum_k \sum_m \sum_n I(j-n, i-m, k) w(n,m)}$$

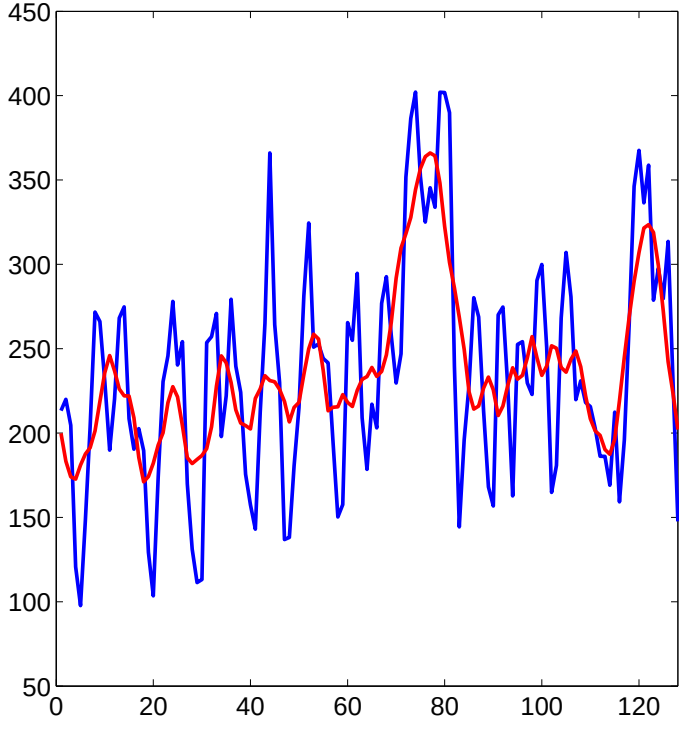




Filtering - proximity

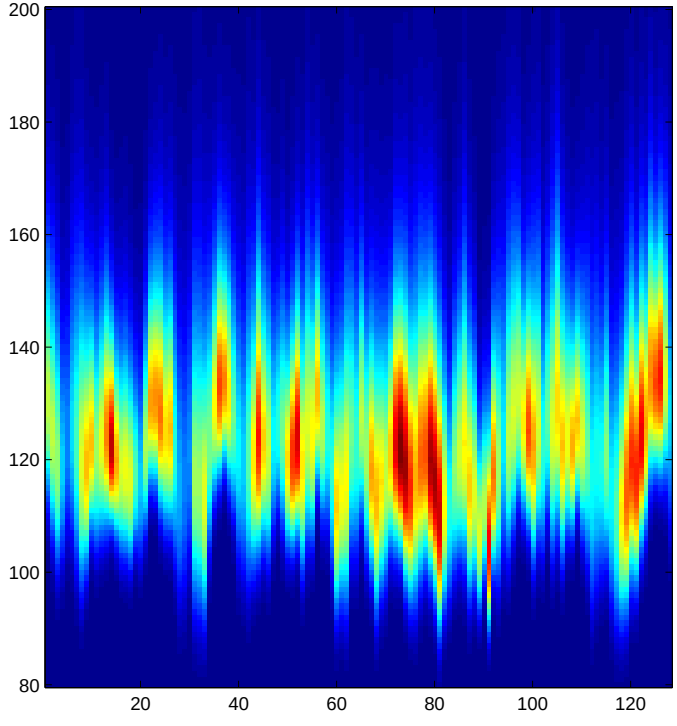
$$\tilde{I}(i,j,k) = \frac{\sum_k I(j,i,k) \sum_m \sum_n I(j-n,i-m,k) w(n,m)}{\sum_k \sum_m \sum_n I(j-n,i-m,k) w(n,m)}$$

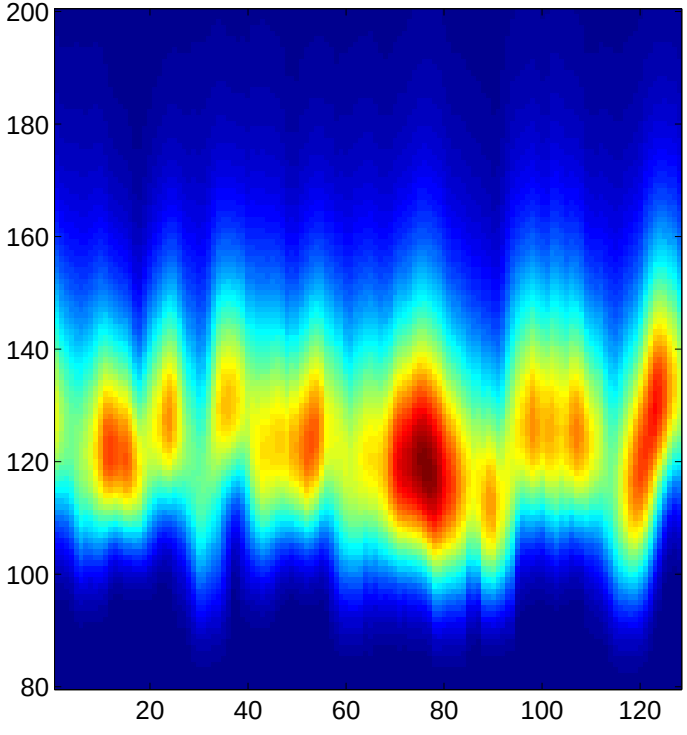




Filtering - proximity

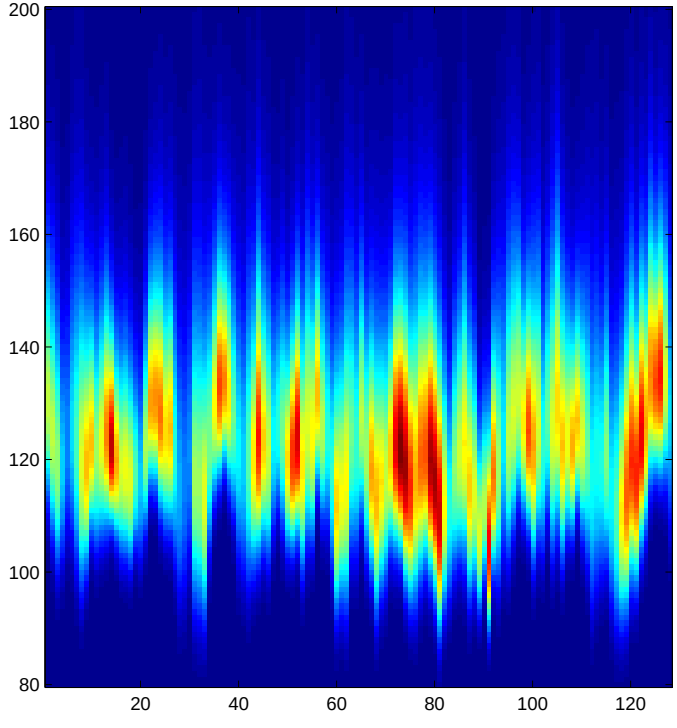
$$\tilde{I}(i,j,k) = \frac{\sum_k I(j,i,k) \sum_m \sum_n I(j-n,i-m,k) w(n,m)}{\sum_k \sum_m \sum_n I(j-n,i-m,k) w(n,m)}$$

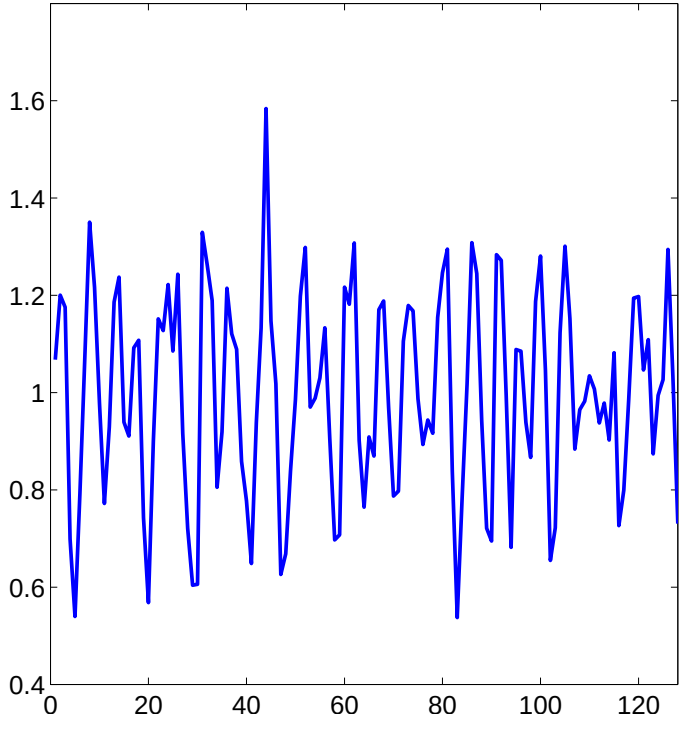




Filtering - proximity

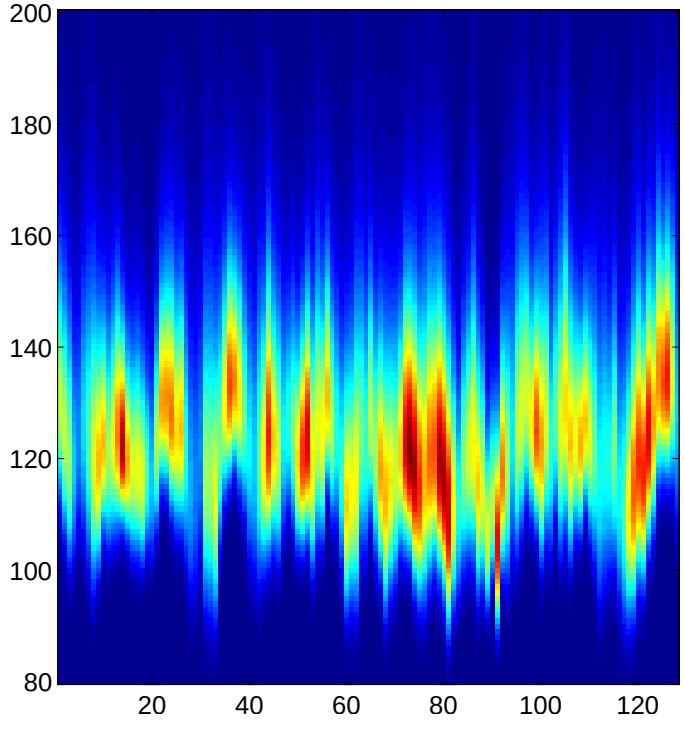
$$\tilde{I}(i,j,k) = \frac{\sum_k I(j,i,k) \sum_m \sum_n I(j-n, i-m, k) w(n,m)}{\sum_k \sum_m \sum_n I(j-n, i-m, k) w(n,m)}$$

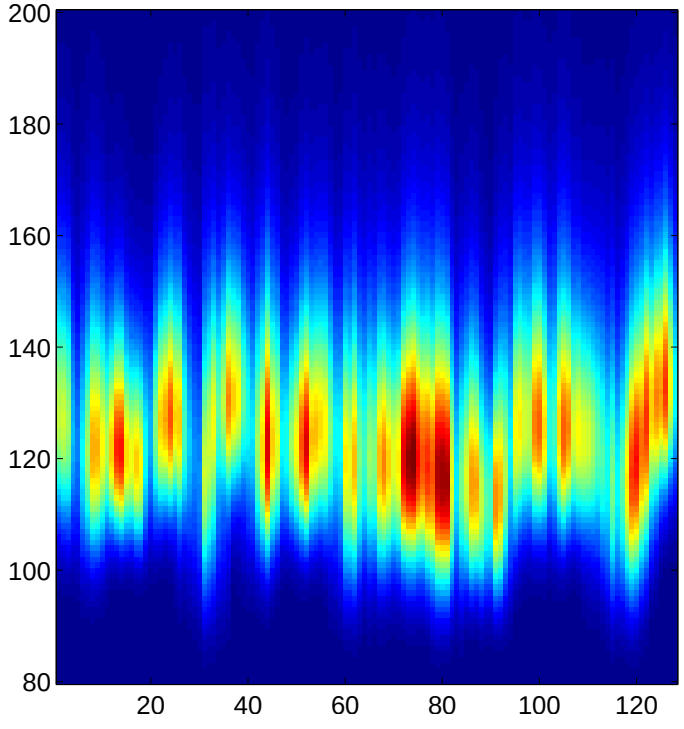




Filtering - proximity

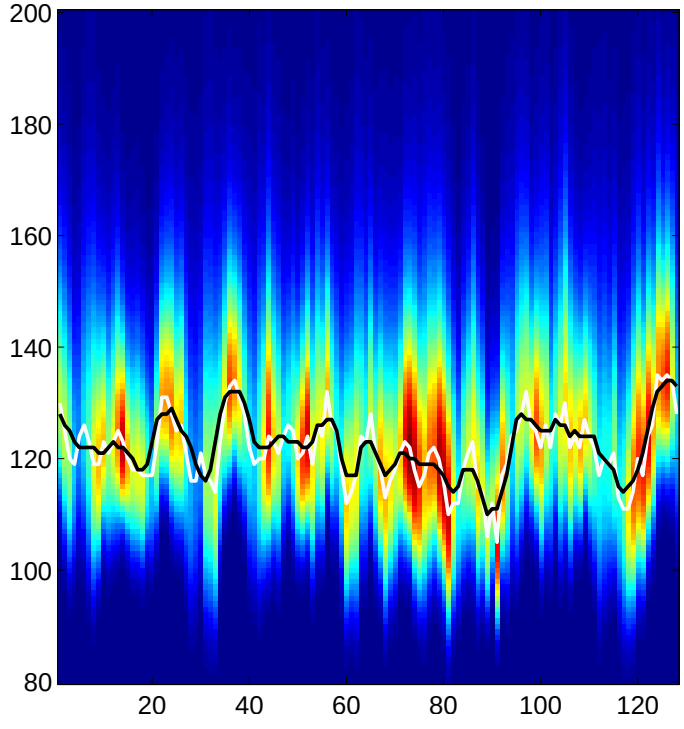
$$\tilde{I}(i,j,k) = \frac{\sum_k I(j,i,k) \sum_m \sum_n I(j-n, i-m, k) w(n,m)}{\sum_k \sum_m \sum_n I(j-n, i-m, k) w(n,m)}$$

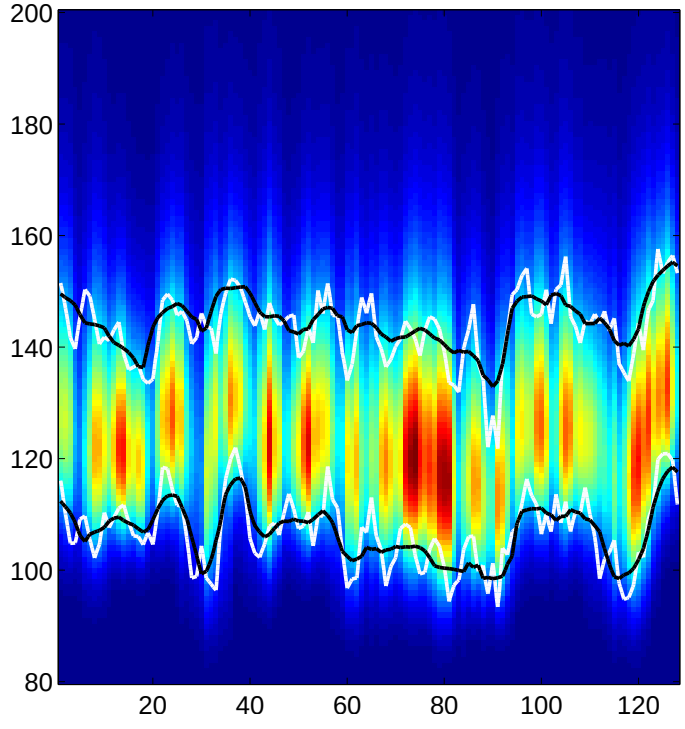




Filtering - proximity

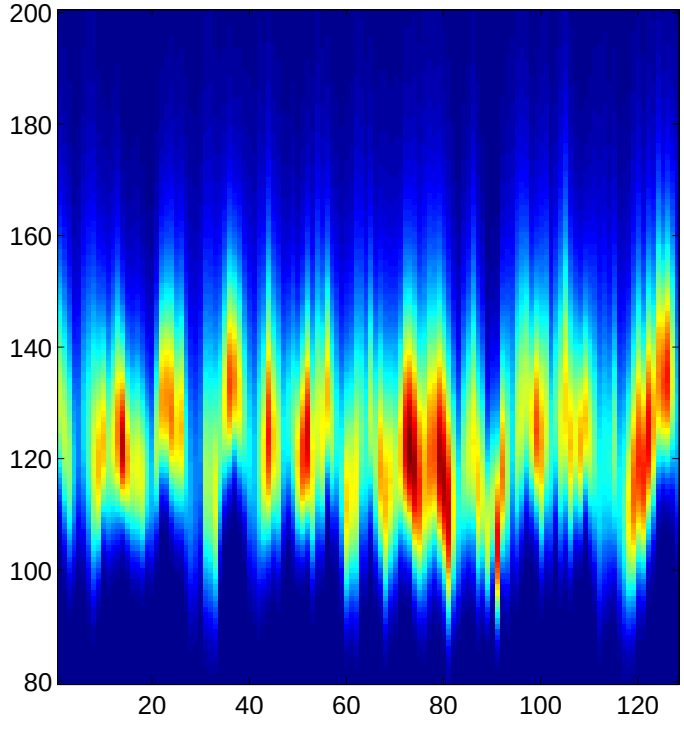
$$\tilde{I}(i,j,k) = \frac{\sum_k I(j,i,k) \sum_m \sum_n I(j-n, i-m, k) w(n,m)}{\sum_k \sum_m \sum_n I(j-n, i-m, k) w(n,m)}$$

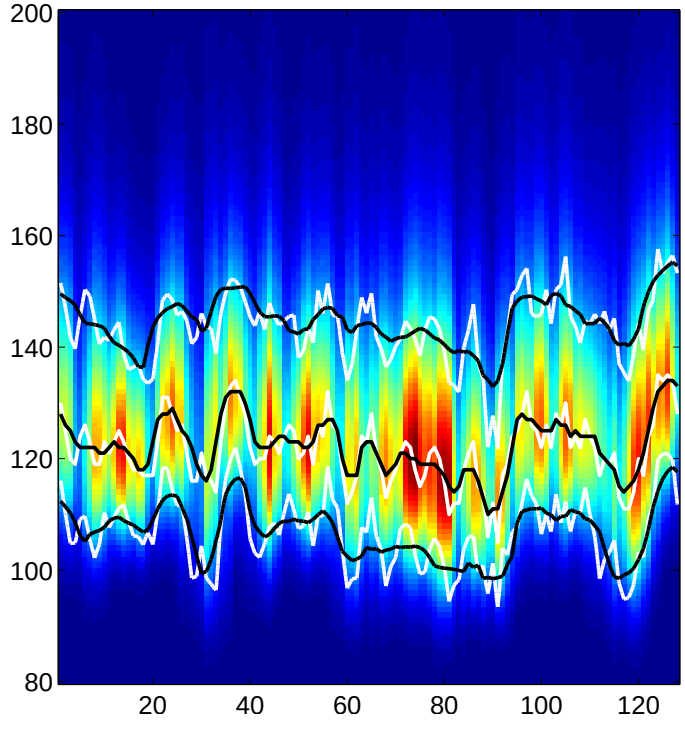




Filtering - proximity

$$\tilde{I}(i,j,k) = \frac{\sum_k I(j,i,k) \sum_m \sum_n I(j-n,i-m,k) w(n,m)}{\sum_k \sum_m \sum_n I(j-n,i-m,k) w(n,m)}$$





Start Guess

The unbiased start guess for the itterative reconstruction methods is a completely flat one. To obtain a physically feasible solution it is sometimes preferable to choose a start guess that approximates “a typical” solution. For auroral studies a chapman profile

$$I(z) = I_0 \exp(1 - (h - h_0)/dh - \exp(-(h - h_0)/dh))$$

can be used with either one constant max intensity, I_0 or with I_0 varying as the projection of the images to the peak altitude, h_0. Here of course the peak altitude have to be different for different emissions, 100–115 for 5577 and 4278 Å, and 200-220 for 6300 Å and 110-140 for 8446 Å. By giving the itterative methods a start point “close” to what we expect, noise amplification and problems related to uncertainties in rotation and sensitivity are reduced.

Field aligned grid

Even though *Nygren [1994]* showed that it does not reduce the errors much in auroral tomography to have the grid aligned with the magnetic field it does not hurt either. Furthermore the coarser grid the better it is to use field aligned grid since then the aliasing effect if more pronounced otherwise.

Filter Kernels