



## UvA-DARE (Digital Academic Repository)

### Search and detection of low frequency radio transients

Spreeuw, J.N.

**Publication date**  
2010

[Link to publication](#)

**Citation for published version (APA):**

Spreeuw, J. N. (2010). *Search and detection of low frequency radio transients*. [Thesis, fully internal, Universiteit van Amsterdam].

**General rights**

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

**Disclaimer/Complaints regulations**

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

# LOFAR's Transients Key Project: Detailed description of the Source Extraction System

---

## 2.1 Abstract

The limitations of a source extraction package (SExtractor) that meets the speed requirements for the Transients Key Project (TKP) are discussed. The necessity for the design of a new source extractor is emphasized. The main components of this new package for the Transients Key Project software pipeline are described. LOFAR maps can be processed in real time. The properties of the sources from those maps and their error bars can be compared with the content of databases and used for alerts and updates, also in real time. The effects of correlated noise in radio maps are taken into account. Sources can be deblended and a False Discovery Rate (FDR) algorithm can be applied. The code is written in Python and meets present day standards for maintainability, flexibility and modular programming.

## 2.2 LOFAR

LOFAR<sup>1</sup>, the LOw Frequency ARray, is an innovative new radio telescope in Northwestern Europe, which will observe the radio sky in the frequency range 30–240 MHz. This frequency range covers the lowest energy extreme of the electromagnetic spectrum, that can be observed from the surface of the earth. Its construction will be completed this year (2010). The main part of this telescope is located in the Netherlands.

---

<sup>1</sup>see [www.lofar.org](http://www.lofar.org)

Radio astronomy actually started at low frequencies (20.5 MHz) with the pioneering work of Karl Jansky (Kraus 1950). Jansky investigated the background noise plaguing transatlantic short-wave communications for Bell Telephone Laboratories. He then serendipitously discovered the Milky Way and in particular the Galactic Centre, as a strong source of low frequency radio emission. At that time (1931), the sun was at a solar minimum, so Jansky did not discover the active sun as a strong radio source.

In the decades after the war, radio astronomy developed mainly at much higher frequencies, where the spatial resolution per baseline length is better and where the ionosphere is much less of a problem. LOFAR correlates antennas separated by hundreds or even thousands of kilometers such that sources are separated which would otherwise be confused in blurred images. The jittering effect of the ionosphere can be compensated by the application of algorithms that have been developed over the last years. These algorithms require substantial computer power, but that is now much less of a problem than in the previous century.

## 2.3 The LOFAR Key Projects

LOFAR science was originally divided in four "Key Projects": The Epoch of Reionization (EoR), Transient Sources, Deep Extragalactic Surveys and Ultra-High Energy Cosmic Rays (UHECRs). The focus of "EoR" is on the early universe using the redshifted 21 cm line. The most distant radio galaxies, diffuse emission in galaxy clusters and star-forming galaxies are among the targets of the "Surveys" Key Project. "UHECRs" tries to answer one of the outstanding questions in astrophysics: "Where do the highest energy particles come from and how were they made?" The Transients Key Project is described in some detail in the next section.

As more antennas were deployed over the last years, the number of Key Projects grew; "Cosmic Magnetism" and "Solar and Heliospheric Physics" were founded a couple of years ago.

## 2.4 The Transients Key Project (TKP)

The Transients Key Project (TKP) aims to study all variable and transient sources detected by LOFAR, including pulsars, gamma-ray bursts, X-ray binaries, radio supernovae, flare stars and extrasolar planets.

One of the main strategies of the TKP entails the continuous monitoring of a large area of sky, with the goal to detect many new transient events, and provide alerts to the international community for follow-up observations at other wavelengths. This mode of operation is called the Radio Sky Monitor (RSM) mode. In RSM mode maps are produced on timescales varying from 1s up to tens of seconds, minutes or even hours, when the "classical" source confusion limit is reached, i.e., when the restoring beam cannot separate between adjacent sources. The integration time needed to reach that confusion limit depends strongly on the observing frequency and the size of the array (core or full array). However, when properly calibrated differenced images are used, it should in principle be possible to reach the thermal noise. The TKP will also piggyback on the observations of other KPs and observers, in order

to trace and track transient sources in real time. Besides that, it will be monitoring a number of the known transient sources for long periods of time.

The TKP has been subdivided into five basic scientific working groups:

- Jet sources: AGN, GRBs, accreting white dwarfs, neutron stars and stellar-mass black holes
- Pulsars: classical radio pulsars, AXPs, RRATs
- Planets: solar system objects and exoplanets
- Flare stars: M, L, and T dwarfs and active binaries
- Serendipity: hitherto unexplored parameter space

## 2.5 Automated transient finding in the TKP pipeline

### 2.5.1 Successive images

It is anticipated that most of the TKP target sources will be compact, so very likely unresolved. Consequently, these sources will be detected more easily in images than in visibilities. Transient phenomena involve variability in brightness at fixed positions on the sky which can be traced by comparing maps from different epochs, e.g., maps that were acquired in RSM mode. Hence, not only the fidelity of short exposure (snapshot) images but also the accurate and reliable processing of images from different epochs will be a cornerstone of the success of LOFAR and the TKP. When processing successive images, it is most essential that all sources in an image are measured in less time than the time interval between those images, to keep the delays from alerts as short as possible. This is also important for the timely freezing of the station "Transient Buffer Boards"(TBBs), thereby saving valuable high time resolution data.

### 2.5.2 Pipeline concept and overview

The processing of images or differenced images is actually only the start of the TKP pipeline. After sources have been either detected and measured or monitored, sources will be classified and the measurements will be archived in a database. The interaction of the monitoring process with the database of known and previously measured sources is shown prominently in figure 2.1. All of this is set up as an online system, able to send out immediate alerts when new sources appear or when known sources show interesting or unexpected behaviour in terms of brightness or polarization fluctuations. These alerts could result in a direct rescheduling of the observing program, the broadcasting of SMS messages or the sending of emails. Figure 2.1 also mentions a "monitoring position list". The idea is that the monitoring of faint sources should not be hindered by detection thresholds. The fluxes of these sources can be measured

relatively straightforward, if they are unresolved, by converting their celestial coordinates to pixel coordinates and subsequently fitting the clean beam to their positions in the images. Figure 2.2 takes into account the fact that, in general, we will not process images, but rather image cubes for each time interval corresponding to the sampling time of the visibilities. These cubes have extra dimensions with respect to plain images, i.e. the four Stokes values I, Q, U and V and spectral channel. If the fluxes of sources are measured per spectral channel, a spectral index can be derived immediately.

### 2.5.3 Source Extraction and Measurement

#### Principles

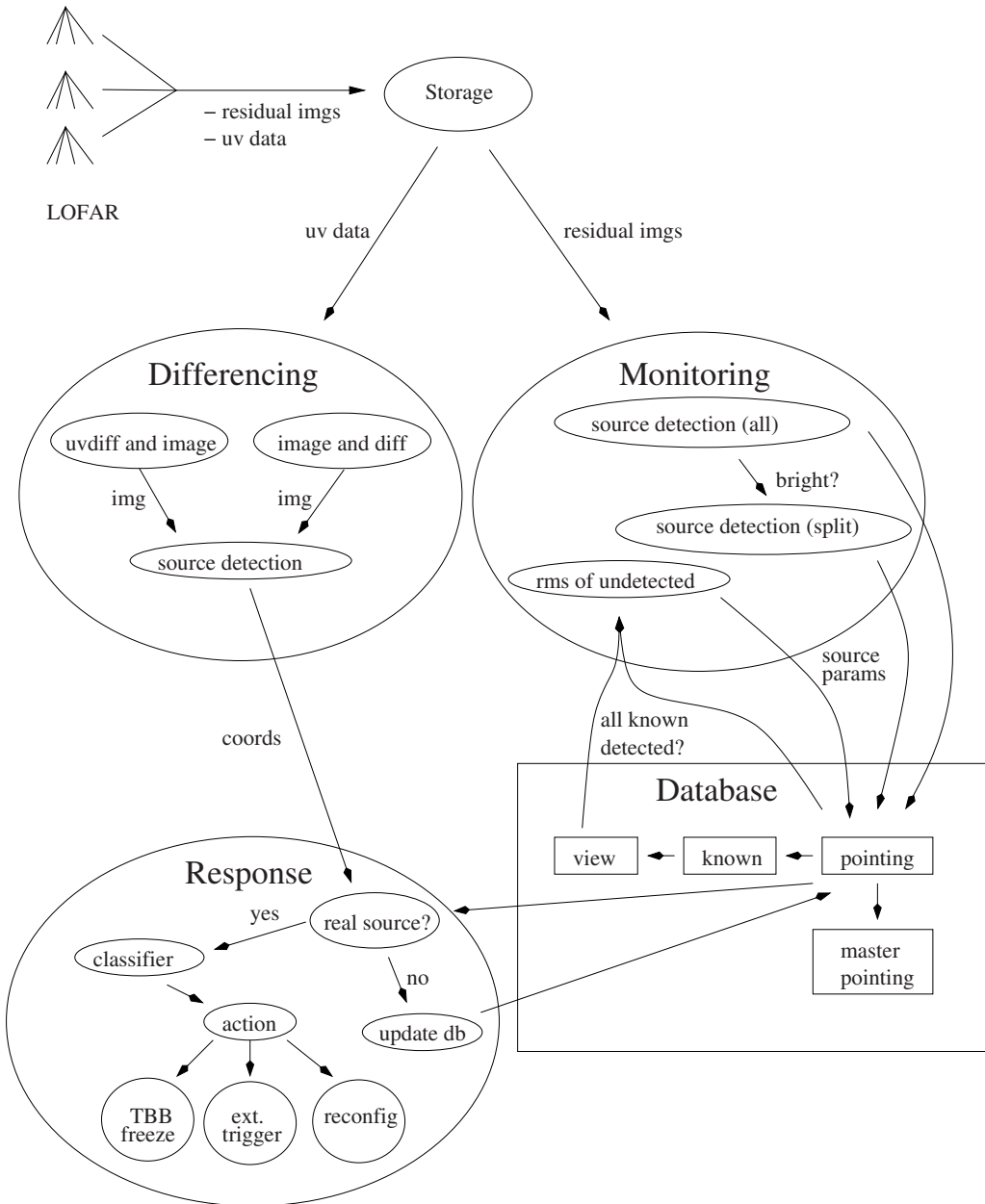
Source extraction involves the detection of sources above a certain threshold. A complete, but not very detailed, overview of the source extraction process is depicted in figure 2.3. Source measurement is the next step, a group of pixels is then described in terms of a model, usually a Gaussian with six free parameters. Monitoring of known sources generally entails fitting a Gaussian with less than four free parameters, because the position is fixed, if ionospheric refraction is properly accounted for. If it is unresolved, one free parameter (the peak flux) will suffice. It is worth noting that faint sources can only be monitored effectively if they are unresolved or if their shapes are known in terms of Gaussian parameters, i.e., axes and position angles. The reason is that a Gaussian fit with more than one free parameter is likely to fail if the peak flux is not higher than a few times the noise.

Alternatively and possibly more effectively, instead of the images, the difference of two successive images can be analysed and inspected. This presumes that we can come up with an appropriate algorithm to handle ionospheric disturbance, such that sources are not moving around in successive images. If we succeed in doing this, the difference (pixel by pixel) of the two images will show the flux rise or decay of a transient source accurately. Constant sources will cancel out in the differenced image.

#### Requirements

I have listed above the main goals of the Transients Key Project as well as its prime target sources. As noted, these sources are most easily detected in images while the most straightforward way of tracing any transient behaviour is by measuring fluxes in a sequence of images. From these facts we can derive the basic requirements for the source extraction and measurement code:

- Complete detection of all sources above a user given threshold and the rejection of all sources below that threshold.
- Measurement of source parameters, in terms of a Gaussian model, of the detected sources with the highest possible accuracy, i.e., reaching the theoretical limits.
- Likewise with respect to the monitoring of known sources, regardless of the threshold.



**Figure 2.1:** Interactions of the TKP pipeline processes with the database, taken from Law (2007)

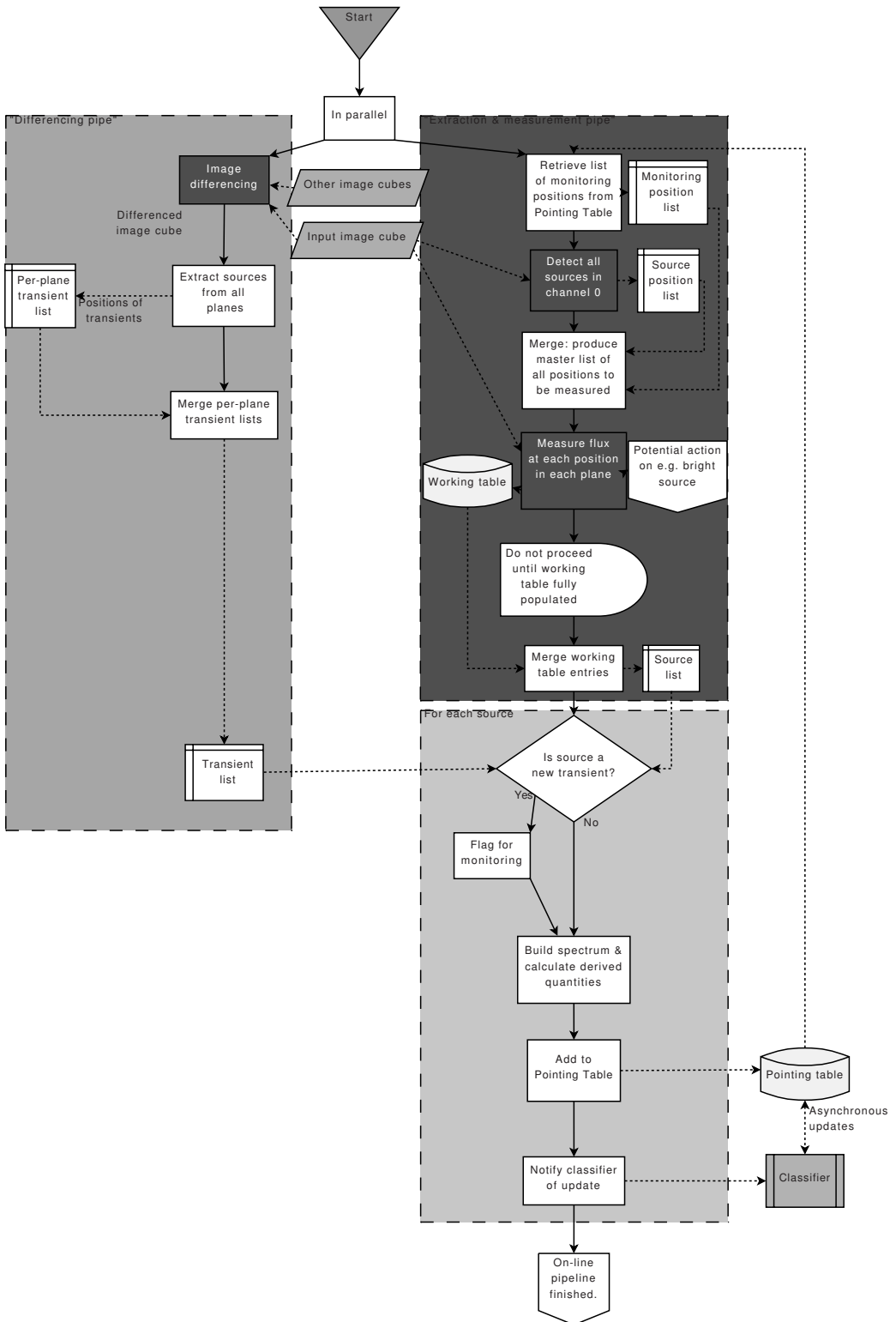
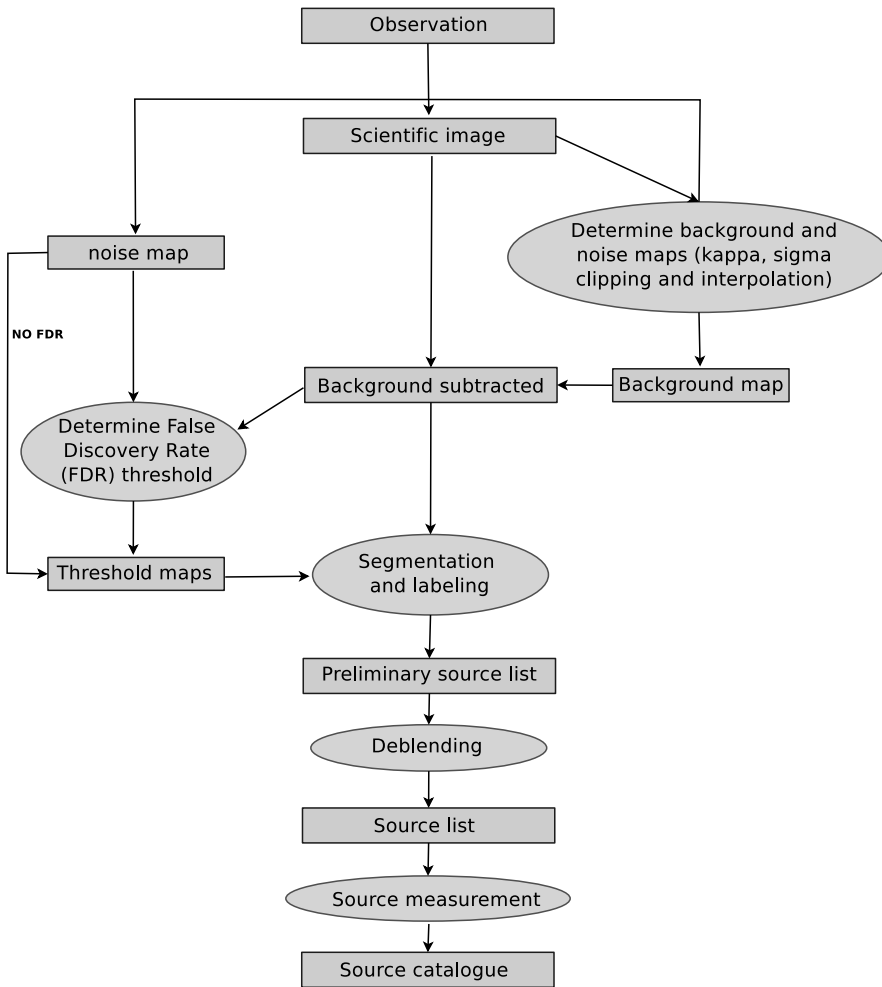


Figure 2.2: Data flow through the TKP pipeline, taken from Swinbank (2007), slightly modified.



**Figure 2.3:** An overview of the source extraction process in the TKP pipeline

- Complete processing of successive images in less time than the time interval between those images ( $\approx 1$  s) and the simultaneous processing of maps from stacked data, from logarithmically spaced time intervals.
- Robustness. Millions of images will be processed and billions of source measurements will be performed. Of course, the processing of images should not stop at any point, but besides that there are requirements with respect to the robustness of the source measurements. Any least-squares algorithm, Gauss-Newton, Lévenberg-Marquardt or other, can diverge without crashing occasionally. If not taken care of this could lead to

sending false alerts to the outside world. The TKP pipeline software can also provide the source parameters derived from moments. The parameters from fitting should be compared with "moments" before alerts are sent.

In theory, it was possible that some of these requirements could not be met at the same time. For instance, in the beginning it was thought that source measurement by least squares fitting would slow down the image processing too much. This turned out not to be the case, the time to fit one source turns out to be less than 20 ms. We did not encounter any other possible conflicts between requirements, except for the deblending algorithm. We chose a deblending algorithm that was fast rather than optimum in terms of separating sources that are very close together. This has the advantage that less time is wasted on the deblending of extended sources, which is useless. It was shown<sup>2</sup> that the simultaneous fitting of multiple Gaussians to an extended source causes serious delays in image processing.

The speed requirement was, of course, ill defined because it was not known what computing power would be available at the time of the deployment of the LOFAR stations. Also, the number of maps and their sizes produced per visibility sampling time was not yet fixed. Presently (2010), it is anticipated that images will not be produced every seconds but rather every 10 seconds in the first year after the commissioning of LOFAR has been completed. It was clear that our code would be optimised for speed by others at a later stage. The design of the code was functional rather than complicated and we were reluctant to implement any piece of code of which the necessity was not immediately obvious.

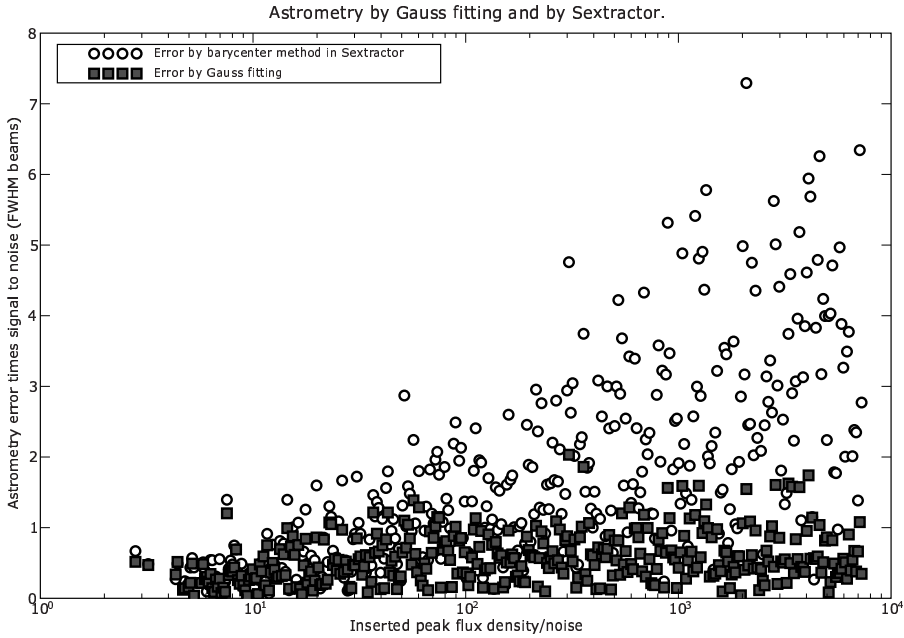
In Stokes I images sources are indicated by (large) positive pixel values. Source detection in total intensity images makes use of this. In Stokes Q, U and V images the presence of polarized sources is indicated by both negative and positive pixel values. Consequently, these images cannot be processed in the same manner and the source extraction code needs to be adjusted. These adjustments have not been implemented yet.

## 2.6 A brief rationale behind the current design of the Source Extraction System

Designing a new source extraction system was regarded as an option, but not as a necessity. This means that the implementation of another freely available source extraction package in the TKP pipeline was not ruled out from the beginning. We tested SExtractor (Bertin & Arnouts 1996) extensively because it can be run from a Unix shell and because of its speed. See Mohan & Röttgering (2005) for a comparison of the speed of SExtractor with SFIND (Miriad) and SAD (AIPS). At that time it was thought that the other packages were much slower because source parameters were derived by Gauss fitting. SExtractor, on the other hand, calculated source parameters from moments. Later, it turned out that Gauss fitting need not be a bottleneck in terms of speed. Also, in the beginning, SExtractor passed several tests with regard to accuracy of photometry and astrometry. In the end, however it failed to reach the theoretical limits, whereas Gauss fitting does reach them. The formulae below for post-fit

---

<sup>2</sup>A. Usov, private communication



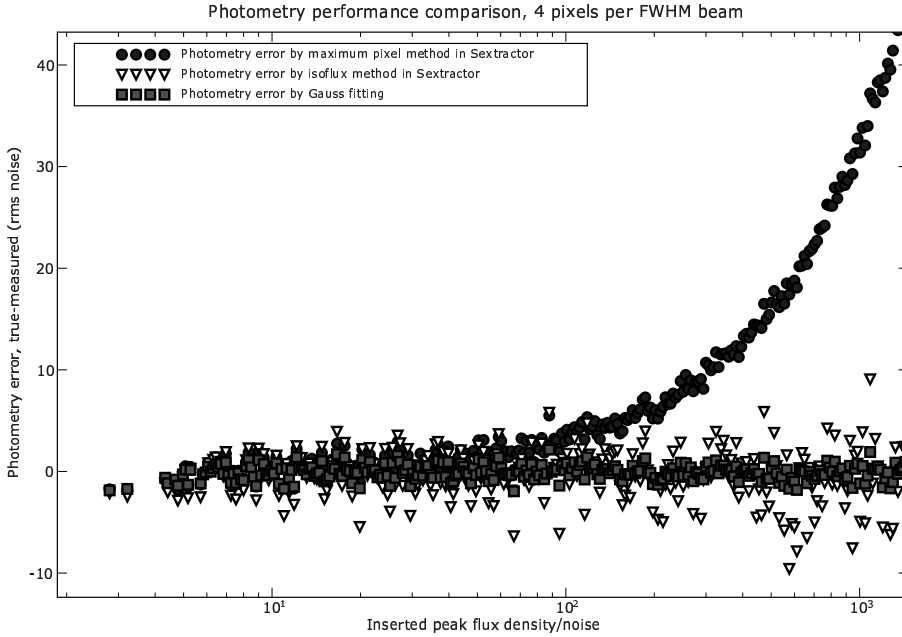
**Figure 2.4:** The barycenter method for calculating the position of a source does not reach the theoretical limit of accuracy and high  $s/n$ . If the beam is oversampled this effect is lessened. For this test we pixellized a fixed Gaussian and added correlated noise from random locations in a source free radio image after we scaled it in order to create a range of  $s/n$  ratios.

r.m.s. errors were taken from Fomalont (1999) and are commonly used in astronomy.  $\sigma(x_0)$  and  $\sigma(A)$  are the errors associated with the position and peak flux density, while  $\sigma$  is the local r.m.s noise.  $\mu_x$  is the fitted width, it is equal to the semi-major axis divided by  $\sqrt{2 \ln 2}$ , presuming the x-axis and semi-major axis are aligned.

$$\sigma(x_0) = \frac{\sigma \mu_x}{2A} \quad (2.1)$$

$$\sigma(A) = \sigma \quad (2.2)$$

These formulae are actually simplifications; more accurate formulae were derived by Condon for correlated and uncorrelated noise (Condon 1997). Equation 2.1 states that the accuracy of astrometry increases linearly with signal-to-noise. That is not what we find from our tests when we use moments, cf. equations 2.50 and 2.51, see figures 2.4 and 2.5. This malfunction becomes apparent at high signal-to-noise and/or bad sampling of the synthesized beam. Equation 2.2 says that the standard error in photometry is about equal to the local rms noise. The peak pixel method in SExtractor finds the correct values at low signal-to-noise, with a standard deviation of about the rms noise, but severely underestimates peak flux



**Figure 2.5:** The peak pixel method for photometry on point sources does not reach the theoretical limit of accuracy and high s/n. If the beam is oversampled this effect is lessened. This plot was constructed from the same data as figure 2.4.

densities at high signal-to-noise. The origin of this discrepancy is immediately clear in noise free conditions: only when the source is centered on a pixel can we find the correct value, in all other cases we will underestimate the true value. There is another way of measuring peak flux densities of sources, which is called the ISOFLUX method in SExtractor. The isoflux method in SExtractor adds all pixel values in an island, from this volume the peak pixel value can be computed if the beam shape is known. As is shown in figure 2.5 this method does not seem to systematically over- or underestimate the true peak flux densities, but its accuracy is not optimal. Let me describe that method in somewhat more detail. The volume,  $V$ , under a circular Gaussian with peak,  $C$ , and HWHM axis,  $s$ , down to a threshold,  $T$ , is given by:

$$V = \frac{\pi s^2}{\ln 2}(C - T) \quad (2.3)$$

The peak of the Gaussian can therefore be found by computing  $V$  from the sum of all the pixel values within the island.

$$C = V \frac{\ln 2}{\pi s^2} + T \quad (2.4)$$

The absolute accuracy of this method actually decreases when more pixels are used because we are computing a sum instead of an average. The relative error in  $C$ , i.e.  $\Delta C/C$ , does decrease with signal-to-noise, as I will show below.

If  $N$  is the number of pixels of the island and  $\pi s^2 = N_{dep} > N$  is the number of pixels in the synthesized beam, then the error in the peak flux density  $\Delta C$  can be expressed in terms of the error in the volume  $\Delta V = N\sigma$

$$\Delta C = \frac{\Delta V \ln 2}{N_{dep}} = \sigma \ln 2 \frac{N}{N_{dep}} \quad (2.5)$$

assuming that all the errors in pixels add up coherently if the total number of pixels is smaller than  $N_{dep}$ . Of course, we can express  $N$  in terms of the peak, threshold and HWHM axis, like this:

$$N = N_{dep} \frac{\ln \frac{C}{T}}{\ln 2} \quad (2.6)$$

We can combine equations 2.5 and 2.6, valid only for very marginal detections, less than  $2\sigma$ :

$$\Delta C = \sigma \ln \frac{C}{T} \quad (2.7)$$

If the number of pixels covers more than the synthesized FWHM beam the errors do not add up coherently anymore. We have to replace  $\Delta V = N\sigma$  by

$$\Delta V = \sigma \sqrt{(N \bmod N_{dep})^2 + (N - (N \bmod N_{dep}))N_{dep}} \quad (2.8)$$

Of course, this is an approximation, because the transition between correlated and uncorrelated pixels is not so distinct. In general, for a detection with a high significance we can approximate

$$\Delta V = \sigma \sqrt{NN_{dep}} \quad (2.9)$$

So the error in determining the peak flux density using the ISOFLUX method in SExtractor is given by

$$\Delta C = \sigma \ln 2 \sqrt{\frac{N}{N_{dep}}} \quad (2.10)$$

This equation can be simplified somewhat by combining it with equation 2.6:

$$\Delta C = \sigma \sqrt{\ln 2 \ln \frac{C}{T}} \quad (2.11)$$

Clearly this error increases with  $C/T$ . The relative error does decrease, since  $\lim_{C \rightarrow +\infty} \Delta C/C \propto \lim_{C \rightarrow +\infty} \sqrt{\ln C - \ln T}/C = 0$ .

Another nuisance of this method is the fact that one needs to have a priori knowledge of the compactness of the source. Only the peak flux densities of unresolved sources can be determined in this manner.

There is also an extra error in computing the volume which comes from the finite size of the pixels. This corresponds to the difference between exact integration and numerical integration. The size of this error can be computed using formulae for the trapezoidal rule in two dimensions. For one dimension, the error made by the trapezoidal integration is easy to compute, but for two dimensions the formula for that error is not easily available. We find that the typical fractional error converges to  $1e-4$  at very low thresholds. This fractional error is expressed relative to the peak of the Gauss. We derived this fractional error from our test runs in noise free maps with a pixel sampling of 4 pixels per FWHM beam, in both dimensions. The Gaussian was centered at position (0.65, 0.91), deliberately not on the center of a pixel, which corresponds to (0.5, 0.5).

In principle, one could overcome the obstacle of decreasing accuracy with increasing  $C/T$ , by setting higher thresholds for bright sources. This, however, introduces an extra source of error unless the psf is heavily oversampled.

This comes from the uncertainty in the lower limit of the numerical integration, assumed to be the threshold  $T$ . That lower limit is actually unknown; we just know that the lowest pixel values used to compute the volume are higher than  $T$ . Actually, we are integrating down to whatever the height of the Gauss is along the outer edges of the outer pixels of the source island. That error is equal to the derivative of the Gauss along the radial coordinate  $r$  times the pixel increment ( $=1$ ), so it is smallest near the top of the Gauss and far away from the center. It is largest at  $r = \pm s/\sqrt{2 \ln 2} = \pm \sigma_r$ . We can compute that error like this:

$$\Delta C = \Delta T = \frac{2Cr \ln 2}{s^2} \exp\left(-\frac{r^2 \ln 2}{s^2}\right) \simeq \frac{2T}{s} \sqrt{\ln 2 \ln \frac{C}{T}} \quad (2.12)$$

The maximum error is given by

$$\Delta C = \Delta T = \frac{C}{s} \sqrt{\frac{2 \ln 2}{e}} \quad (2.13)$$

That error is a large fraction of the peak. In our test runs we had 4 pixels per FWHM beam in both dimensions. This means  $s = 2$  and the fractional error is about 36%! This occurs at thresholds near  $T = C/\sqrt{e}$ . The error from equation 2.12 has to be added in quadrature to the error computed from equation 2.7.

If  $T$  is not much less than  $C$ , the number of pixels in the island is very limited. In this case  $T$  varies significantly as a fraction of  $C$  along the edges of the pixels. This means that formula 2.12 is only a rough estimate.

I did not find a way to tweak the ISOFLUX method in SExtractor to make it perform optimally with regard to photometry. Consequently, I conclude that there is no simple way to make the performance of SExtractor optimal with regard to photometry. This problem becomes apparent at high signal-to-noise. When using the maximum pixel method, the underperformance is obvious from figure 2.5. The deviations are smaller if the maximum pixel method is applied on images with an oversampled synthesized beam. The underestimates of the fluxes still occur, but at higher signal-to-noise. If SExtractor photometry is done with the ISOFLUX method, the relative errors decrease with signal-to-noise. However, the theoretical limit to the accuracy of photometry is an absolute error and the ISOFLUX method cannot reach this, although it does not systematically underestimate the peak flux density, like the maximum pixel method does.

As explained in paragraph 2.7.5, it is possible to do a tweaked form of moment analysis. This will do almost optimum photometry without fitting, by solving a transcendental equation. The problem that remains, even with "tweaked" moments, is the accuracy of astrometry. It is impossible to reach the theoretical limits at high signal-to-noise without fitting. Hence, the SExtractor package was excluded for use in the Transients Key Project software pipeline. The other main reason is that the SExtractor framework is such that making adjustments to the source code, e.g., changing the interface, is involved.

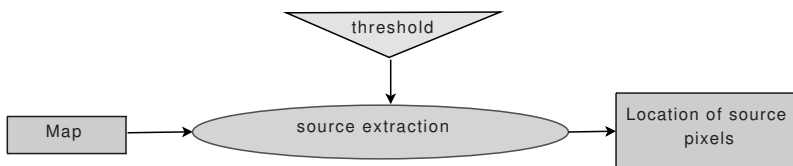
## 2.7 The TKP Python Source Extractor (PYSE)

### 2.7.1 General

This document provides a detailed description of the TKP source extraction (SE) system, revision 1325. The level of detail is such that any astronomer experienced with programming in Python should be able understand and modify the code.

### 2.7.2 Basic Idea

A flowchart describing just the source extraction system is very simple, see figure 2.6.



**Figure 2.6:** The essence of source extraction.

The essential part in source extraction is provided for by a few simple python commands, originally coded at Space Telescope Science Institute (STScI)<sup>3</sup>:

<sup>3</sup><http://stdas.stsci.edu/numarray/numarray-1.5.html/node98.html>

```
clipped = where(sci > threshold, 1, 0) (2.14)
```

```
structuring_element = [[0, 1, 0], [1, 1, 1], [0, 1, 0]] (2.15)
```

```
labels, num = nd.label(clipped, structuring_element) (2.16)
```

```
f = nd.find_objects(labels) (2.17)
```

My interest for coding source extraction routines in Python was drawn by an example from a Python tutorial by Greenfield & Jedrzejewski (2007). The readiness of Python for source extraction was one of the key reasons to choose this programming language. The routines mentioned used in equations 2.16 and 2.17 were originally only in the `stsci_python` package from STScI, they were later also included in Scipy. I will briefly describe the 4 steps corresponding to equations 2.14 through 2.17. The input is the map `sci` with pixel values, so `sci` is a two-dimensional array. The pixel values above the user-defined threshold are set to 1 while the others are set to 0. The structuring element defines which groups of pixels are considered contiguous. In this case 4-connectivity is chosen, the default in the `scipy.ndimage` package (imported as `nd`), which means that groups of pixels only connected at corners are distinct. This is not the case when 8-connectivity is chosen. The structuring element for 8-connectivity is given by `[[1, 1, 1], [1, 1, 1], [1, 1, 1]]`. Labeling of the islands is done by the `label` command, which means that each group of pixels is given a value. The `find_objects` command provides a list of slices for cutting out the source islands from the map. Here is an example.

```

sci = array([[1, 2, 2, 1, 1, 0],
            [0, 2, 3, 1, 2, 0],
            [1, 1, 1, 3, 3, 2],
            [1, 1, 1, 1, 2, 1]])

clipped = array([[0, 1, 1, 0, 0, 0],
                [0, 1, 1, 0, 1, 0],
                [0, 0, 0, 1, 1, 1],
                [0, 0, 0, 0, 1, 0]])

s = [[0, 1, 0], [1, 1, 1], [0, 1, 0]]

labels = array([[0, 1, 1, 0, 0, 0],
               [0, 1, 1, 0, 2, 0],
               [0, 0, 0, 2, 2, 2],
               [0, 0, 0, 0, 2, 0]])

sci[f[0]] = array([[2, 2],
                  [2, 3]])
sci[f[1]] = array([[1, 2, 0],
                  [3, 3, 2],
                  [1, 2, 1]])

```

Please note that the final source islands, `sci[f[0]]` and `sci[f[1]]` are rectangular with pixel values below the threshold for analysis. These pixels are to be masked when Gauss fitting is done or when moments are computed.

### 2.7.3 One step back, construction of noise and background maps

The scheme depicted in figure 2.6 is too simple in general for two reasons. First, there are often background levels in maps (positive or negative) that need to be subtracted before accurate flux measurements can be done. The cause of background levels in radio maps is due to missing spacings (Högbom 1974). This was also mentioned by Briggs, Schwab & Sramek (1999): "An undersampled large-scale emission region may introduce large undulations in image intensity that are hard to remove". These undulations can run diagonally across the image. In that case, accurate flux measurements are still possible because a background map can be computed which should be subtracted from the image.

## Background calculation

There are several ways to assess the background level in (part of) a map. The background level in a source free map is simply equal to the mean of all the pixel values, assuming it is does not vary across the map. When sources are present, it is approximated by the mode of the pixel values. This is an approximation because the mode can be shifted away from the true value, i.e., the mean background in the source free case, by pixels from weak sources and the outer pixels from strong sources.

There is a direct way to find the mode, by constructing a histogram of pixel values. An exception may, however, come from maps that are confusion limited where the mode can be shifted by the presence of many sources. Another exception comes from extended sources that can dominate large part of a map. In this case the background cannot be computed reliably. The best thing to do in this case is to interpolate the background levels around this extended source or by applying a median filter that selects the most appropriate nearby background level. In other cases the overwhelming majority of all pixels are noise pixels and the mode is found at the peak of the histogram. Two difficulties remain. First the mode may be ill-defined because the histogram can be almost flat near the mode. Second, the mode is dependent on the bin size. There is no solution to the former issue, but the latter was addressed by Patat (2003), by the introduction of "The Optimal Binning Technique". Other tasks, like SFIND in Miriad, fit a Gaussian to the histogram in order to compute the noise mode and its standard deviation (Hopkins et al. 2002), without applying the Optimal Binning Technique.

In principle we could use this approach. However, we want to trace variations of the background and noise across the image. In order to do so we would have to make histograms in small subimages. Patat (2003, paragraph 3) states that it is unlikely that the statistics in such a subimage will be good enough to derive the background and noise from a histogram. Bertin & Arnouts (1996) argue that the method developed by Bijaoui (1980) is probably the most unbiased but very noisy in small samples. They also note that this method requires excessive computing time. This is actually true for all methods that make use of histograms. We therefore refrain from any method that requires histogramming maps.

Instead, we tweaked the method developed by Bertin & Arnouts (1996) for calculating the mean and standard deviation of the background noise. The original method was incorporated in the SExtractor source extraction package. It involves  $\kappa, \sigma$  clipping around the median until convergence. After each clipping, the median and standard deviation are recomputed. At the start all pixels are used to compute the median and standard deviation, after that the pixel range is set by the median  $\pm 3\sigma$ . Convergence is achieved when all pixels are within this range. We implemented  $\kappa, \sigma$  clipping slightly differently from SExtractor, by clipping  $\pm n\sigma$  around the median instead of  $\pm 3\sigma$ . Also,  $\sigma$ , the standard deviation of the clipped distribution, is corrected for "clipping bias". This correction is needed because the rms of the clipped distribution underestimates the true noise rms. Both of these "tweaks" are explained in paragraph 2.7.3.

After convergence, the mode is estimated from the following formula:

$$\text{mode} = 2.5 \cdot \text{median} - 1.5 \cdot \text{mean} \quad (2.18)$$

if the distribution of pixel values is not too skewed. This requirement is quantified by

$$|\text{mean} - \text{median}|/\sigma \leq 0.3 \quad (2.19)$$

If the distribution is strongly skewed we adopt the median, just as SExtractor does it. To my knowledge there are, however, no rigid justifications for the distinction made by the skewness criterion from equation 2.19. In fact the skewness check is a bit odd, since close to that skewness limit there is a step in the mode of size  $0.45\sigma$ . This is easily seen by setting

$$|\text{mean} - \text{median}|/\sigma = 0.3 \quad (2.20)$$

which results in

$$\text{mode} = \text{median} - 0.45 \cdot \sigma \quad (2.21)$$

if the mean is larger than the median. Intuitively, one would favour a continuous transition from the skewed to the non-skewed regime. This has not been implemented yet because it is not trivial how the appropriate formula should be derived.

Formula 2.18 differs from the usual empirical relationship ( $\text{mode} \approx 3 \cdot \text{median} - 2 \cdot \text{mean}$ ), see, e.g., Kenney & Keeping (1962), which holds for unimodal distributions of moderate asymmetry. We adopted formula 2.18 instead, because tests by Bertin & Arnouts (1996) proved it more accurate for clipped distributions.

In very old versions of SExtractor there is a distinction in calculating the mode between crowded and uncrowded fields. If  $\sigma$  changes less than 20% during the process of  $\kappa, \sigma$  clipping, the field is considered not crowded and the mean of the clipped distribution is used to estimate the mode. Later the SExtractor code was changed to use formula 2.18 or the median to estimate the mode, depending on the skewness test from formula 2.19<sup>4</sup>. The mean was no longer used to estimate the mode.

The final background map is derived by interpolating the node values from the all the subimages that constitute the map. The size of the subimages is specified by the user. The interpolation can be of any order, for instance bilinear or bicubic spline.

### Noise calculation

SExtractor determines the background characteristics by  $\kappa, \sigma$  clipping. When this has converged, i.e. when all remaining pixels in a subimage are within  $3\sigma$  from the median, the true background rms noise is calculated as the standard deviation of the clipped distribution. The boundary of  $3\sigma$  in SExtractor is somewhat arbitrary. No pixels in a source free subimage should be clipped. However, if its size is large enough there will be pixels with values that differ more than  $3\sigma$  from the median. Naively, it seems that  $3\sigma$  clipping works perfectly if the size of a source free subimage were small enough. In that case Gaussian statistics would predict that less than one pixel exceeded the  $3\sigma$  boundary. On the other hand, if the subimage were too small,  $3\sigma$  would be too large a boundary for clipping. This could result in source pixels remaining unclipped.

<sup>4</sup>See my discussion with E. Bertin at <http://terapix.iap.fr/forum/showthread.php?tid=267>

From this reasoning it can be inferred that the ideal size of a subimage for  $3\sigma$  clipping is 185, since  $3\sigma$  corresponds to 0.27% and  $185 \cdot 0.27\% = 0.5$ , i.e. "half a pixel" would be clipped. If the number of independent pixels in the subimage were significantly smaller or larger the determination of the rms noise would be biased.

In a realistic source extraction system, the size of the subimage is chosen by the user, of course. To accomodate this, we decided to implement  $\kappa, \sigma$  clipping slightly differently from SExtractor by adjusting the limit for clipping based on the number of independent pixels  $N_{indep}$  in the subimage or the number that is left after a number of clipping iterations. The limit  $n \cdot \sigma$  for clipping can be estimated because the number of source pixels is expected to be a minor fraction of the total number of pixels:

$$n \approx \sqrt{2} \cdot \text{ErfcInv}\left(\frac{1}{2 \cdot N_{indep}}\right) \quad (2.22)$$

Here  $\text{ErfcInv}$  is the inverse of the Complementary Error Function.  $N_{indep}$  is usually calculated as the total number of pixels  $N$  after zero or more clippings divided by the number of pixels  $N_{dep}$  in the synthesized beam. The clipping boundary is recomputed after each clipping iteration based on the number  $N_{indep}$  and standard deviation of the remaining pixels, using equation 2.22. If this were a plain standard deviation, as used by SExtractor, we would underestimate the true noise, so we implemented a correction for "clipping bias". This correction is easy to derive:

$$\sigma^2 = \frac{1}{\sigma \sqrt{2\pi}} \int_{-\infty}^{+\infty} x^2 e^{-\frac{x^2}{2\sigma^2}} dx \quad (2.23)$$

$$\sigma_{meas}^2 = \frac{\frac{1}{\sigma \sqrt{2\pi}} \int_{-D}^{+D} x^2 e^{-\frac{x^2}{2\sigma^2}} dx}{\frac{1}{\sigma \sqrt{2\pi}} \int_{-D}^{+D} e^{-\frac{x^2}{2\sigma^2}} dx} \quad (2.24)$$

$$\sigma^2 = \sigma_{meas}^2 \frac{\sqrt{2\pi} \text{Erf}\left(\frac{D}{\sigma \sqrt{2}}\right)}{\sqrt{2\pi} \text{Erf}\left(\frac{D}{\sigma \sqrt{2}}\right) - 2 \frac{D}{\sigma} e^{-\frac{D^2}{2\sigma^2}}} \quad (2.25)$$

Here,  $D$  is the clipping limit,  $\sigma_{meas}^2$  is the measured variance of the clipped distribution and  $\text{Erf}$  is the Error Function. The ratio  $D/\sigma$ , i.e., the clipping limit in units of the true rms noise is simply equal to  $n$  as derived from equation 2.22.

The size of the subimages correspond to the maximum area over which the noise and background are close enough to constant, from the perspective of the user. The more ambitious user will set it as small as possible in order to trace minor variations in noise and background level in the map but still large enough to do accurate statistics. "large enough" implies that  $\kappa, \sigma$  clipping can separate source pixels from noise pixels. In order to do so it is necessary that the number of source pixels is a minor fraction of the total number of pixels. If the density of the sources in some part of a map is too high, some of the noise and background node values will be unreliable. In that case, these values can be replaced by the median of the surrounding

background and noise node values. In determining the median, a median filter is applied to a two dimensional window of a size specified by the user. Alternatively, the user may set a subimage size so large that any of the subimages in the map will cover enough source free pixels. It should be clear to the user that if no precautions are taken, the noise and background will be overestimated near large concentrations of sources.

The final noise map is derived in the same way as the background map, by the interpolation of the node values. It is not advised to use bicubic spline to interpolate the noise grid. Although the noise grid will only have positive values, a bicubic spline interpolation can result in negative(!) values in the final noise map if there are large differences between contiguous nodes. A bilinear interpolation of the noise grid, on the other hand, will never result in negative values.

## 2.7.4 Thresholding

### General

For all types of thresholding there is the problem that a population of sources with fluxes just below the threshold will either be missed or overestimated in a time sequence of images. A consequence of this is that the average estimated flux of these sources over all positive detections will be too high. This is known as Malmquist bias or truncation bias.

Apart from this problem, the observed distribution of source fluxes in an image can generally not be assumed to be representative of the true source distribution. In the unrealistic case where a complete population of sources is detected well above the threshold will the means of the observed and measured distributions agree. The shapes of the true and measured distributions will generally not agree, not even in the latter case. The problem of recovering the true distribution of fluxes that have been scattered by Gaussian noise was first addressed by Eddington (Eddington 1913, 1940). The average shift as a function of flux resulting from the convolution of the intrinsic distribution with Gaussian noise is referred to as Eddington bias. It was emphasized by Teerikorpi (2004, paragraph 3) that the bias itself is a threshold independent effect and that it vanishes not only if the intrinsic distribution is flat, but also when it is a linear function of the true flux. The reason for this is that the convolution of a linear function with a symmetric Gaussian does not change the function. "Threshold independent" here means that the bias occurs at all flux levels, even well above the detection threshold, but only if one considers the sources that appear between certain flux limits. Again, the bias vanishes if one averages over a population that has been detected completely.

The intrinsic distribution can be recovered from the observed distribution as a series expansion of the observed distribution (Eddington 1913, 1940; Teerikorpi 2004). In general, however, there is the practical problem of determining higher order derivatives of the observed distribution accurately. Chandrasekhar & Münch (1950) faced a similar problem when trying to derive the true distribution of the rotational velocities of stars from a sample of the apparent rotational velocities. It is much easier to derive moments of the true distribution than the true distribution itself. Strictly speaking, the latter is impossible in the case of Eddington bias since the accuracy in determining derivatives decreases with the order of the derivative.

Hogg & Turner (1998) solved a slightly different problem: given a measured flux  $F$  with

signal-to-noise  $r = F/\sigma$ , what is the most likely true flux  $S_{ML}$ , assuming that it belongs to a class of sources with a known (true) flux distribution? If the (true) cumulative number of sources  $N$  increases down to lower fluxes as the flux to the power  $-q = d \log N / d \log S$ , the most likely true flux  $S_{ML}$  is given by the maximum of the conditional probability  $P(S|F)$ : given a measured flux  $F$ , what is the most likely true flux. They use:

$$P(S|F) = \frac{P(F|S)P(S)}{P(F)} \quad (2.26)$$

$$P(F|S) \propto e^{-\frac{(S-F)^2}{2\sigma^2}} \quad (2.27)$$

$$P(S) \propto S^{-(q+1)} \quad (2.28)$$

$$P(S|F) \propto S^{-(q+1)} e^{-\frac{(S-F)^2}{2\sigma^2}} \quad (2.29)$$

The value  $S = S_{ML}$  for which  $P(S|F)$  is maximal is then easy to find:

$$\frac{S_{ML}}{S_0} = \frac{1}{2} + \frac{1}{2} \sqrt{1 - \frac{4q+4}{r^2}} \quad (2.30)$$

This is equation 4 from Hogg & Turner (1998). From this equation these authors conclude that flux measurements done at signal-to-noise ratios of 4 or less are practically useless. From equation 2.30 one may be tempted to infer that, if one knew the intrinsic, unbiased, distribution of sources, it could be reconstructed by applying equation 2.30 to all measured fluxes. This statement is incorrect. If we were to apply equation 2.30 to all measured fluxes in an image than it is easy to show that the maximum likelihood concept is inappropriate, for instance by considering a bimodal intrinsic source distribution:

$$\frac{dN}{dS} \propto C_{left} \delta(S - S_{left}) + C_{right} \delta(S - S_{right}) \quad (2.31)$$

with

$$C_{left} > C_{right} \quad (2.32)$$

where  $\delta$  indicated the Dirac  $\delta$  function. Now, the most likely true flux value for any measured flux will always be  $S_{left}$ . This implies that the average of the most likely true fluxes also equals  $S_{left}$ . This is not equal to the average true flux,  $(C_{left}S_{left} + C_{right}S_{right}) / (C_{left} + C_{right})$ . Hogg & Turner (1998) acknowledge that confidence intervals are more robust than maximum likelihood estimates, but refrain from using them because of a normalization problem. It is a pity that more authors have copied their method, see, e.g., Herranz et al. (2006); Wang (2004).

If one wants to avoid flux biases, instead of deriving the most likely true flux for every measured flux  $F$  one should calculate the expected value  $S_{exp}(F_{meas}^i)$ , like this:

$$S_{exp}(F_{meas}^i) = \frac{\int_0^{+\infty} S P(S|F_{meas}^i) dS}{\int_0^{+\infty} P(S|F_{meas}^i) dS} = \int_0^{+\infty} S P(S|F_{meas}^i) dS \quad (2.33)$$

since  $\int_0^{+\infty} P(S|F_{meas}^i) dS = 1$ . If the differential source counts are well described by a power-law,  $dN/dS \propto S^{-(q+1)}$ , we can write:

$$S_{exp}(F_{meas}^i) = \int_0^{+\infty} S P(S|F_{meas}^i) dS = \frac{\int_0^{+\infty} S^{-q} e^{-\frac{(S-F_{meas}^i)^2}{2\sigma^2}} dS}{\int_0^{+\infty} S^{-(q+1)} e^{-\frac{(S-F_{meas}^i)^2}{2\sigma^2}} dS} \quad (2.34)$$

For  $q > 0$  even, the nominator can be expressed analytically, but not the denominator and vice versa for  $q$  odd. So either the nominator or the denominator have to be evaluated numerically, or both if  $q$  is not an integer. There is another complication, because the integral in the nominator diverges for  $q > 0$ . Evidently, the true fluxes cannot be described as a true power law down to zero flux, because the integrated flux would diverge and there is not an infinite amount of flux in the universe. This was also noted by Hogg & Turner (1998). Ideally, the power law as a means of describing the true source distribution should be replaced by a more realistic, normalizable, function. This is more accurate than setting a lower limit  $> 0$  to the integrations in the nominator and denominator of equation 2.34. However, if the contribution from the faintest sources can be neglected, the latter approach will be sufficiently accurate.

Correcting individual fluxes according to equation 2.34 is appropriate, in the case that one knows the true flux distribution, the method of Hogg & Turner (1998) is not. Still, the average of all corrected fluxes,  $\overline{S_{exp}}$ , will differ from the average flux of the underlying distribution,  $\overline{S_{true}}$ , because the measured fluxes are a specific sample of all measurable fluxes. They are all above a threshold,  $T$ . In the limiting case,  $\lim_{T \rightarrow -\infty} \overline{S_{exp}}$ , we find:

$$\lim_{T \rightarrow -\infty} \overline{S_{exp}} = \lim_{T \rightarrow -\infty} \frac{\int_{F=T}^{F=+\infty} \int_{S=0}^{S=+\infty} S P(S|F) dS P(F) dF}{\int_T^{+\infty} P(F) dF} \quad (2.35)$$

$$= \int_0^{+\infty} S P(S) dS = \overline{S_{true}} \quad (2.36)$$

This is not found if the fluxes are corrected as prescribed by equation 2.30.

We have considered implementing a correction for Eddington bias, using equation 2.34, but it seemed more appropriate as a post processing step, because it requires assumptions about the detected sources. For the software validation, we will just have to check if the measured fluxes are biased apart from Eddington and Malmquist (truncation) bias. When running these tests, we are in the advantageous position that we have perfect knowledge of the intrinsic

distribution of sources that we insert in the image plane. We derive the theoretical, Eddington and Malmquist biased, mean measured flux  $\overline{F}$  by integrating over  $P(F)$  and comparing it with the actual measurements,  $\overline{F}_{meas} = \sum_{i=1}^M F_{meas}^i / M$ .

$$\overline{F} = \frac{\int_{F=T}^{F=+\infty} F P(F) dF}{\int_T^{+\infty} P(F) dF} \quad (2.37)$$

$$= \frac{\int_{F=T}^{F=+\infty} F \int_{S=0}^{S=+\infty} P(F|S) P(S) dS dF}{\int_{F=T}^{F=+\infty} \int_{S=0}^{S=+\infty} P(F|S) P(S) dS dF} \quad (2.38)$$

$$= \frac{\int_{F=T}^{F=+\infty} F \int_{S=0}^{S=+\infty} e^{-\frac{(S-F)^2}{2\sigma^2}} P(S) dS dF}{\int_{F=T}^{F=+\infty} \int_{S=0}^{S=+\infty} e^{-\frac{(S-F)^2}{2\sigma^2}} P(S) dS dF} \quad (2.39)$$

If all the inserted sources had the same flux  $S_{ins}$ , i.e., the flux distribution is a  $\delta$ -function, Eddington bias vanishes, but Malmquist bias does not. We find:

$$P(S) = \delta(S - S_{ins}) \quad (2.40)$$

$$\int_0^{+\infty} P(F|S) P(S) dS = P(F|S_{ins}) \propto e^{-\frac{(S_{ins}-F)^2}{2\sigma^2}} \quad (2.41)$$

$$\frac{\overline{F}}{S_{ins}} = \frac{\int_{F=T}^{F=+\infty} F e^{-\frac{(S_{ins}-F)^2}{2\sigma^2}} dF}{S_{ins} \int_{F=T}^{F=+\infty} e^{-\frac{(S_{ins}-F)^2}{2\sigma^2}} dF} \quad (2.42)$$

$$= \frac{2\sigma e^{-\frac{(S_{ins}-T)^2}{2\sigma^2}} + \sqrt{2\pi} S_{ins} \text{Erfc}\left(\frac{T-S_{ins}}{\sqrt{2}\sigma}\right)}{\sqrt{2\pi} S_{ins} \text{Erfc}\left(\frac{T-S_{ins}}{\sqrt{2}\sigma}\right)} \quad (2.43)$$

with  $\text{Erfc}$  the complementary error function. Consider the special case  $S_{ins} = T = n\sigma$ , when one inserts sources at the  $n\sigma$  threshold. In that case we find:

$$\frac{\overline{F}}{S_{ins}} = \frac{2 + n\sqrt{2\pi}}{n\sqrt{2\pi}} \quad (2.44)$$

This ratio equals  $\approx 1.16$  for  $n = 5$ . In the limit  $n \rightarrow \infty$ ,  $\overline{F}/S_{ins} \rightarrow 1$ . But the absolute correction  $\overline{F} - S_{ins}$  does not vanish. For  $n \rightarrow \infty$ ,  $\overline{F} - S_{ins} \rightarrow \sigma\sqrt{2/\pi}$ . This shows why Malmquist bias cannot be ignored near thresholds, not even if these thresholds are very high with respect to the noise. This, of course, assumes that photometry can be done with an accuracy of about  $\sigma$ , the rms noise. If photometry is compromised by other, larger, errors, Malmquist bias may not become apparent.

More generally, if we test the software by inserting a cumulative source distribution  $N(S) \propto S^{-q}$ , with differential source counts  $dN/dS \propto S^{-(q+1)}$  in the image plane, the average measured flux should be close to  $\overline{F}$ , as given by:

$$\overline{F} = \frac{\int_{F=T}^{F=+\infty} F \int_{S=-\infty}^{S=+\infty} S^{-(q+1)} e^{-\frac{(S-F)^2}{2\sigma^2}} dS dF}{\int_{F=T}^{F=+\infty} \int_{S=-\infty}^{S=+\infty} S^{-(q+1)} e^{-\frac{(S-F)^2}{2\sigma^2}} dS dF} \quad (2.45)$$

In general,  $\overline{F}$  cannot be expressed in analytic form. For testing, it is much simpler and more accurate to take into account the discrete sampling of the inserted sources:

$$P(S) = \sum_{i=1}^N \delta(S - S_{ins}^i) / N \quad (2.46)$$

$$\overline{F} = \frac{\sum_{i=1}^N 2\sigma e^{-\frac{(S_{ins}^i - T)^2}{2\sigma^2}} + \sqrt{2\pi} S_{ins}^i \text{Erfc}\left(\frac{T - S_{ins}^i}{\sqrt{2}\sigma}\right)}{\sqrt{2\pi} \sum_{i=1}^N \text{Erfc}\left(\frac{T - S_{ins}^i}{\sqrt{2}\sigma}\right)} \quad (2.47)$$

Equation 2.47 shows that the validation of flux measurements is straightforward if all inserted sources are well above the detection threshold. This is the way the validation runs will be set up. For simplicity, we will insert sources with equal fluxes in each map. This is described by equation 2.43 which reduces to

$$\frac{\overline{F}}{S_{ins}} = 1 \quad (2.48)$$

when all sources are inserted well above the threshold.

### Plain thresholding

Once a background and noise map have been computed the background map is subtracted from the original (calibrated and cleaned) radio image. The noise map is multiplied by a user specified number to make a threshold map for finding sources. The values of pixels (after background subtraction) have to be higher than the local threshold level to be selected as source pixels. This is accomplished by equation 2.14, with "threshold" not a number, but a 2-D array, the threshold map. In SExtractor, one may also set a fixed threshold in Jy/beam, independent of the local rms noise. We have dropped the option of a fixed threshold across the map because it did not seem useful.

Imposing a  $n\sigma$  threshold implies that the fraction of false source pixels in the image will be smaller than  $\text{Erfc}(n/\sqrt{2})$ , averaged over a very large number of images. For the NVSS (Condon et al. 1998), for example, which has a 2mJy/beam limit for catalogued sources and a close to uniform rms noise, one can calculate the fraction of false source pixels relative to the

total number of pixels in the NVSS maps. That fraction is not easily translated to a number of false sources in the NVSS.

In fact, two separate threshold maps may be applied simultaneously in the TKP software, the analysis threshold map and the detection threshold map. First we select all source pixels above the analysis threshold and then drop all islands that do not have peak pixels above the detection threshold. This is not supplied for in SExtractor or in SAD (AIPS), but Rengelink (1998) used it, too.

### The False Discovery Rate (FDR) algorithm

The criterion for controlling the False Discovery Rate was invented by Benjamini & Hochberg (1995). It was used for source detection in astronomical images a few years later (Miller et al. 2001; Hopkins et al. 2003). It was also implemented in the source detection task SFIND 2.0 (Hopkins et al. 2002) in the MIRIAD reduction package. The user should enter a maximum allowed fraction of "false positives", i.e., noise peaks erroneously interpreted as sources. The implemented False Discovery Rate (FDR) algorithm divides the radio map (with the background subtracted) by the noise map. This normalized map is then used to calculate the detection threshold, as explained in Appendix B of Miller et al. (2001). The noise map is multiplied by this threshold to make a threshold map for selecting source islands. On average the fraction of false source pixels will be lower than the user given maximum fraction. Again, as with plain thresholding, this refers to source pixels, not to sources, strictly speaking. Hopkins et al. (2002) have shown that the FDR algorithm is not as accurate with regard to sources as with regard to source pixels.

In default mode, the FDR algorithm makes no use of a separate analysis threshold. Nonetheless, this option is also available, like in SFIND. However, the maximum allowed fraction of false source pixels is not guaranteed when the analysis threshold is lower than the FDR threshold, as shown by Hopkins et al. (2002).

### Deblending

The lowest pixel values of the islands will be above the threshold for analysis while the peak pixel value will also be above the threshold for detection. If two or more sources are close enough, they will initially be detected as one source. The deblending algorithm implemented in the TKP pipeline uses subthresholds and connectivity as defined by the structuring element to separate sources analogous to SExtractor (Bertin 2006, paragraph 6.4). In SExtractor and in the TKP software, the user specifies the number of subthresholds exponentially spaced between the lowest and the highest pixel value. In both deblending algorithms, at each subthreshold level the structuring element is used to deblend the subislands from one of the lower subthresholds, although SExtractor uses 8-connectivity only.

There are, however, differences in the codes. The most important difference is that in SExtractor there is a subsequent procedure that descends back from the "tips of the branches" down to the "trunk". At each junction threshold it checks if the flux above that level in that branch (subisland) is larger than some user given fraction of the total flux in the composite object. If

this is the case **and** there is at least one more branch that also satisfies this condition at the same junction threshold, then those branches are identified as separate sources. The condition for a minimum flux in a branch ensures that noise spikes are not falsely identified as separate sources. In the TKP software, we have attempted to accommodate for this without a separate procedure. "On the way up" there are simultaneous checks for connectivity and sufficient flux. Again, this is specified by the user as some fraction of the total flux. Unless there are at least two subislands (from the same split), both with sufficient flux above that subthreshold, nothing is done and the algorithm leaps to the next subthreshold. If the subislands are sufficiently bright **and** have peak values above the detection threshold they are identified as separate sources, while the residual pixels from the deblending of the "parent" are discarded. The procedure continues up to the highest subthreshold or until some subthreshold is reached above which there is insufficient flux in all of the remaining subislands.

It is worth noting that the user given number of subthresholds, `DEBLEND_NTHRESH`, is an important parameter for deblending. If it is too small, it will slow down the source extraction process without achieving anything extra. If it is too large, the deblending algorithm may miss a source.

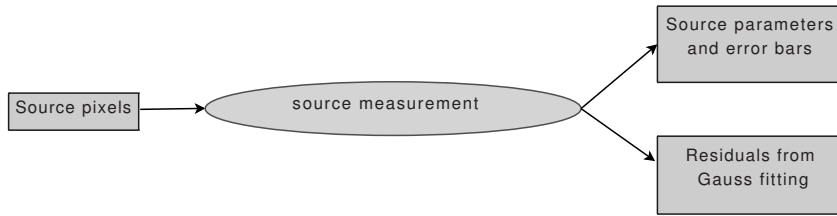
It is easy to show that two equally high circular Gaussians with FWHM size  $2s$  can make a saddle point if their separation is at least  $s\sqrt{2/\ln 2} \approx 1.7s$ . If sources are to be separated by the use of connectivity, saddle points are required. Consequently, the deblending capacities of both the TKP software and SExtractor are rather limited. More refined algorithms for deblending, like the simultaneous fitting of multiple Gaussians were, of course, considered, but they tend to slow down the source measurement process significantly. It is hard to write code that can separate between blended and extended sources. Consequently, in the case of fitting multiple Gaussians, this results in wasting many cpu cycles on the deblending of extended sources. In the case of our deblending algorithm, far less processing time is wasted on this. Although the multiple Gaussian fit can separate sources that are closer together and although it is probably more accurate in individual cases, it seems to lack the robustness required for real time processing. Besides that, when fitting multiple Gaussians simultaneously, it seems hard to provide accurate error bars on the fitted parameters because the calculation of the error propagation is complex. Also, the reported error bars of the fitted parameters from the AIPS tasks 'IMFIT', 'JMFIT' and 'SAD' seem to be underestimated, when multiple Gaussians are fitted simultaneously.

## 2.7.5 Source Measurement

### General

Once islands of source pixels have been selected, they can be measured, as depicted in figure 2.7. Measurement in our context essentially means describing the source pixel values as well as possible by a Gaussian with six parameters: the peak flux density, right ascension, declination, semi-major axis, semi-minor axis and the position angle of the semi-major axis, east from local north. Those parameters can be derived either from moment analysis or from least squares fitting.

In general seven quantities and their error bars are to be derived from each source: besides



**Figure 2.7:** Source measurement involves the computation of source parameters such as shape and peak flux density for every island of pixels.

the six parameters mentioned, also the integrated flux density and its error bar. The integrated flux density is a dependent quantity, it can be derived from the peak flux density, the size of the axes and the size of the synthesized beam.

Apart from these fourteen quantities, a measure of the quality of the fit, the fit residual, is also returned. The fit residual is actually a small 2-D array with the fit subtracted from the source pixels. The user may also request for a residual map, where, for each island, the computed Gaussian has been subtracted from the data. The computed Gaussian is sometimes derived from moment analysis if Gaussian fitting did not yield adequate results.

The source measurement for the NVSS (Condon et al. 1998, paragraph 5.2.1) was performed with a maximum size for the fitted FWHM major axis equal to three times the size of the (circular) synthesized beam. If residuals were too high, up to four Gaussians were fitted simultaneously. **It is essential to note that different choices have been made for the TKP pipeline.** There is no upper limit to the fitted size. When residuals from a Gaussian fit to an island or subisland are high, there is no attempt for the simultaneous fitting of more than one Gaussian. This choice reflects one of the basic software requirements: speed. A consequence of these choices is that residuals, in units of the rms noise, will be higher than for the NVSS. However, if pixels at the location of extended sources were excluded, the residuals, in units of the rms noise, should be about equal to the NVSS residuals.

## Moment analysis

**Calculating the Gaussian model parameters** There are seven "moments" to be computed: the peak flux density, the integrated flux, the x and y position, the semi-major and the semi-minor axes and the position angle of the semi-major axis.

The peak flux is the value of the maximum pixel,  $P_{max}$ , with a correction for the fact that the peak is not located exactly at the center of a pixel. We can calculate an average correction, `fudge_max_pix`, hereafter  $f_{mp}$ , assuming that the peak is located at a random position on the pixel and that the source is a point source, like this:

$$f_{mp} = \int_{y=-0.5}^{y=0.5} \int_{x=-0.5}^{x=0.5} \exp(\ln(2) \left[ \frac{(\cos(\theta_{sb})x + \sin(\theta_{sb})y)^2}{m_{sb}^2} + \frac{(\cos(\theta_{sb})y - \sin(\theta_{sb})x)^2}{M_{sb}^2} \right]) dx dy \quad (2.49)$$

where  $M_{sb}$ ,  $m_{sb}$  and  $\theta_{sb}$  are the semi-major and semi-minor axes and the position angle of the synthesized beam, respectively. The correction is usually of the order of a few %, such that  $1.0 < f_{mp} < 1.1$ . We adopt a peak flux density of  $C = P_{max} \cdot f_{mp}$  if the source island consists of only one pixel. The only flaw of this method is that it does not take account of a detection bias: given that that pixel is above the detection threshold the source is more likely to be located close to the center of the pixel. If it were not, it would not have been detected. In order to take account of this effect, one has to make assumptions about the distribution of sources as a function of their flux. This has not been implemented. Rengelink (1998) adopted a fixed value for  $f_{mp} = 1.06$  for all the WENSS maps, although the size of the synthesized beam varied with declination.

The center position is calculated as follows:

$$\text{xbar} = \bar{x} = \frac{\sum_{i \in S} I_i x_i}{\sum_{i \in S} I_i} \quad (2.50)$$

$$\text{ybar} = \bar{y} = \frac{\sum_{i \in S} I_i y_i}{\sum_{i \in S} I_i}. \quad (2.51)$$

where  $\sum_{i \in S}$  indicates a summation over all the pixels that belong to the source, i.e., the island with pixel values above the threshold for analysis and with the highest pixel value above the threshold for detection, disconnected from other islands. It can also be a subisland if the island has been deblended.

The position angle of the semi-major axis, measured counterclockwise from the y-axis, is given by:

$$\tan 2\theta = \frac{2\overline{xy}}{\overline{x^2} - \overline{y^2}} \quad (2.52)$$

Of course, there are two solutions in the interval  $]-\pi/2, 3\pi/2]$  for  $2\theta$ , namely  $2\theta$  and  $2\theta + \pi$ , so for  $\theta$  the solutions are  $\arctan(2\overline{xy}/(\overline{x^2} - \overline{y^2}))/2$  and  $\arctan(2\overline{xy}/(\overline{x^2} - \overline{y^2}))/2 \pm \pi/2$ . If the first solution has the opposite sign as  $\overline{xy}$ , we choose that. If not, we select from the second solutions a value of  $\theta$  in the interval  $]-\pi/2, \pi/2]$ .

The lengths of (half of) the axes are given by:

$$\frac{\overline{\text{semimajor}}^2}{2 \ln(2)} = \frac{M^2}{2 \ln(2)} = \frac{\overline{x^2} + \overline{y^2}}{2} + \sqrt{\left(\frac{\overline{x^2} - \overline{y^2}}{2}\right)^2 + \overline{xy^2}} \quad (2.53)$$

$$\frac{\overline{\text{semiminor}}^2}{2 \ln(2)} = \frac{m^2}{2 \ln(2)} = \frac{\overline{x^2} + \overline{y^2}}{2} - \sqrt{\left(\frac{\overline{x^2} - \overline{y^2}}{2}\right)^2 + \overline{xy^2}} \quad (2.54)$$

Note that both the semi-major and the semi-minor axis differ from the description in the SExtractor manual (Bertin 2006) by a factor  $\sqrt{2 \ln(2)}$ . Formulas 2.53 and 2.54 underestimate the length of the axes because of the nonzero cutoff at the threshold as set by the user. The easiest way to see this is by using a coordinate system in which the semi-major axis is aligned with the y-axis, so  $\theta = 0$ . This means that also  $\overline{xy} = 0$ . We can now compare  $\overline{x^2}$  and  $\overline{y^2}$  for zero and nonzero threshold. In the noise-free case, one has:

$$\overline{x^2} = \frac{\int_{y=0}^{y=y_{max}} \int_{x=0}^{x=x_{max}} C x^2 \exp(-\ln(2)[\frac{x^2}{m^2} + \frac{y^2}{M^2}]) dx dy}{\int_{y=0}^{y=y_{max}} \int_{x=0}^{x=x_{max}} C \exp(-\ln(2)[\frac{x^2}{m^2} + \frac{y^2}{M^2}]) dx dy} \quad (2.55)$$

with  $x_{max}$  and  $y_{max}$  such that

$$x_{max} = m \sqrt{-\frac{\ln \frac{T}{C}}{\ln 2} - \frac{y^2}{M^2}} \quad (2.56)$$

$$y_{max} = M \sqrt{-\frac{\ln \frac{T}{C}}{\ln 2}} \quad (2.57)$$

with  $T$  the threshold for source detection.

The denominator in equation 2.55 can be shown to be  $\pi M m (C - T) / \ln(2)$ . When integrating over  $x$  and then over  $y$  one finds  $m$  times a function that depends only on  $M$ ,  $C$  and  $T$ . If  $M = m$  the integration is straightforward in polar coordinates and one finds  $\pi M^2 (C - T) / \ln(2)$ . It then follows that this function must be  $\pi M (C - T) / \ln(2)$ .

For calculating the nominator, it is easiest to change variables, replacing  $y$  by  $u = y \sqrt{m/M}$  and then transform to polar coordinates:  $x^2 + u^2 = r^2$ ,  $x = r \cos(\phi)$ ,  $u = r \sin(\phi)$ . We then find:

$$\overline{x^2} = \frac{m^2}{2 \ln(2)} \left(1 + \frac{\ln(\frac{T}{C})}{\frac{C}{T} - 1}\right) \quad (2.58)$$

and

$$\overline{y^2} = \frac{M^2}{2 \ln(2)} \left(1 + \frac{\ln(\frac{T}{C})}{\frac{C}{T} - 1}\right) \quad (2.59)$$

We correct for the nonzero threshold by dividing the size of the axes as derived from equations 2.53 and 2.54 in this way:

$$m_{corr} = \frac{m}{\sqrt{1 + \ln(\frac{T}{C})/(\frac{C}{T} - 1)}} \quad (2.60)$$

$$M_{corr} = \frac{M}{\sqrt{1 + \ln(\frac{T}{C})/(\frac{C}{T} - 1)}} \quad (2.61)$$

This results in  $m_{corr} > m$  and  $M_{corr} > M$ .

Note that the position angle of the semi-major axis is not affected by the cutoff at the threshold, as can be seen from equation 2.52. One can always rotate the  $(x,y)$  coordinate system such that  $\bar{x}\bar{y} = 0$  and then rotate the coordinates back. Now, for  $\bar{x}\bar{y} = 0$  the cutoff does not play a role, so it does not play a role in any coordinate system.

After the semi-major and semi-minor axes and the position angle have been determined, the peak flux density,  $C$ , could be determined like this:

$$C = P_{max} \cdot \exp(\ln(2) \left[ \frac{(\cos(\theta)\Delta x + \sin(\theta)\Delta y)^2}{m_{corr}^2} + \frac{(\cos(\theta)\Delta y - \sin(\theta)\Delta x)^2}{M_{corr}^2} \right]) \quad (2.62)$$

Here,  $\Delta x = \bar{x} - x_{max}$  and  $\Delta y = \bar{y} - y_{max}$  are the  $x$  and  $y$  offsets from the center of the peak pixel, at  $x_{max}, y_{max}$ . If the source island consists of only one pixel, we do not apply equation 2.62. In that case we just keep  $C = P_{max} \cdot f_{mp}$ .

The integrated flux,  $F$ , would then be simply calculated as follows:

$$F = C \cdot \frac{M_{corr} m_{corr}}{M_{sb} m_{sb}} \quad (2.63)$$

However, this approach is not 100% correct since both  $m_{corr}$  and  $M_{corr}$  use  $C = P_{max} \cdot f_{mp}$  instead of equation 2.62. But the latter equation, on the other hand, uses  $m_{corr}$  and  $M_{corr}$ . This can be solved because there are in fact three equations for the three unknown variables,  $m_{corr}$ ,  $M_{corr}$  and  $C$ :

$$M_{corr}^2 = m_{corr}^2 \frac{M^2}{m^2} \quad (2.64)$$

$$m_{corr}^2 = \frac{m^2}{1 + \ln(\frac{T}{C})/(\frac{C}{T} - 1)} \quad (2.65)$$

$$\begin{aligned} C &= P_{max} \cdot \exp\left(\frac{\ln(2)}{m_{corr}^2} [(\cos(\theta)\Delta x + \sin(\theta)\Delta y)^2 + (\cos(\theta)\Delta y - \sin(\theta)\Delta x)^2 \frac{m^2}{M^2}]\right) \\ &= P_{max} \cdot \exp\left(\frac{\epsilon}{m_{corr}^2}\right) \end{aligned} \quad (2.66)$$

The latter two equations can be combined into one transcendental equation for the peak flux density:

$$\ln\left(\frac{C}{P_{max}}\right) = \epsilon \frac{1 + \ln\left(\frac{T}{C}\right) / \left(\frac{C}{T} - 1\right)}{m^2} \quad (2.67)$$

Once the peak flux density has been determined, the axes and the integrated flux follow from equations 2.60, 2.61 and 2.63. This is the optimum way for calculating the moments. From tests we found that the peak flux densities derived in this manner are almost as accurate as from Gauss fitting. Figure 2.5 shows that maximum pixel method is too coarse at high signal to noise. Now that we have an almost optimum method for determining peak flux densities without fitting, it may seem that Gauss fitting is no longer needed. This is not the case. For astrometry, we found that the barycenter position from the method described above, which we call "tweaked moments", is still outperformed by Gauss fitting. Plain moments are used as input for Gauss fitting, but both moments and fitted parameters should be catalogued. Occasionally, the Levenberg-Marquardt algorithm for fitting fails. Very rarely it converges to a runaway solution. In those cases the user of the catalogue should be able to revert to the results from moment analysis. Ideally, this should be "tweaked moments" instead of plain moments, but at present "tweaked moments" has not been implemented yet.

**Error bars** The theoretical variances in multipole moments in the presence of correlated noise are not given in textbooks or anywhere on the web, as far as I can tell, so I tried to deduce these variances myself. The barycenter position  $(\bar{x}, \bar{y})$  is given by equations 2.50 and 2.51. With all pixels  $i$  independent, the uncorrelated noise  $\sigma_i$  gives the following variance in  $\bar{x}$ :

$$\sigma(\bar{x})^2 = \sum_{j \in S} (d_j \bar{x})^2 = \sum_{j \in S} \left( d_j \frac{\sum_{i \in S} I_i x_i}{\sum_{i \in S} I_i} \right)^2 = \sum_{j \in S} \left( \frac{\partial \frac{\sum_{i \in S} I_i x_i}{\sum_{i \in S} I_i}}{\partial I_j} dI_j \right)^2 \quad (2.68)$$

$$= \sum_{j \in S} \left( \frac{x_j \sum_{i \in S} I_i - \sum_{i \in S} I_i x_i}{(\sum_{i \in S} I_i)^2} dI_j \right)^2 = \sum_{j \in S} \left( \frac{x_j - \bar{x}}{\sum_{i \in S} I_i} dI_j \right)^2 \quad (2.69)$$

$$= \frac{\sum_{j \in S} (x_j - \bar{x})^2 \sigma_j^2}{(\sum_{i \in S} I_i)^2} = \frac{\sigma(n)^2}{(\sum_{i \in S} I_i)^2} \sum_{j \in S} (x_j - \bar{x})^2 \quad (2.70)$$

where we have assumed that the noise  $dI_j = \sigma_j = \sigma(n)$ , the local rms background noise, does not vary significantly over the source. This equation for the position variance is implemented in the SExtractor package (Bertin 2006, equation 32).

If we want to take account of correlated noise, we'll have to alter these equations slightly. Instead of

$$\sigma(\bar{x})^2 = \frac{\sigma(n)^2}{(\sum_{i \in S} I_i)^2} \sum_{j \in S} (x_j - \bar{x})^2 = \frac{\sigma(n)^2}{(\sum_{i \in S} I_i)^2} N \overline{\sum_{j \in S} (x_j - \bar{x})^2} \quad (2.71)$$

where we have denoted the number of pixels in the source by  $N$ , we should write:

$$\sigma(\bar{x})^2 = \frac{\sigma(n)^2}{(\sum_{i \in S} I_i)^2} N^2 \overline{\sum_{j \in S} (x_j - \bar{x})^2} \quad (2.72)$$

We want to replace the latter expression in terms of the peak flux density  $C$  and the detection threshold  $T$ . It is relatively easy to show that

$$\overline{\sum_{j \in S} (x_j - \bar{x})^2} = \frac{\theta_m^2}{16 \ln 2} \ln \frac{C}{T} \quad (2.73)$$

$$\overline{\sum_{j \in S} (y_j - \bar{y})^2} = \frac{\theta_M^2}{16 \ln 2} \ln \frac{C}{T} \quad (2.74)$$

where, again, we have assumed that the y-axis is aligned with the major axis of the elliptical gaussian. It is also possible to express  $N$ , the number of pixels, in terms of  $C$  and  $T$  if we neglect the finite size of the pixels and use the well known formula for the area of the ellipse to replace  $N$ :

$$N = \frac{\pi \theta_M \theta_m}{4 \ln 2} \ln \frac{C}{T} \quad (2.75)$$

Finally, the denominator in equations 2.71 and 2.72 can be replaced by the formula for the volume of a gaussian with peak height  $C$  and minimum height  $T$  when, again, we neglect pixellation effects:

$$\sum_{i \in S} I_i = \frac{\pi \theta_M \theta_m}{4 \ln 2} (C - T) \quad (2.76)$$

If we compile the latter equations, we find the theoretical limits for the accuracy of the barycenter method in the presence of partially correlated noise:

$$\frac{\sigma(n)^2 \theta_m}{4 \pi \theta_M (C - T)^2} \ln^2 \frac{C}{T} \leq \sigma(\bar{x})^2 \leq \frac{\sigma(n)^2 \theta_m^2}{16 \ln 2 (C - T)^2} \ln^3 \frac{C}{T} \quad (2.77)$$

$$\frac{\sigma(n)^2 \theta_M}{4 \pi \theta_m (C - T)^2} \ln^2 \frac{C}{T} \leq \sigma(\bar{y})^2 \leq \frac{\sigma(n)^2 \theta_M^2}{16 \ln 2 (C - T)^2} \ln^3 \frac{C}{T} \quad (2.78)$$

Of course, noise is not either completely correlated or completely uncorrelated. The approach I have used in section 2.6 in analysing the ISOFLUX method in SExtractor is to consider all  $N_{dep}$  pixels within the "correlated area" completely correlated and outside that completely uncorrelated. Applying that here essentially translates into replacing the factor  $N$  in equation 2.71 or the factor  $N^2$  in equation 2.72 by  $(N \bmod N_{dep})^2 + (N - (N \bmod N_{dep}))N_{dep}$ . For high enough signal to noise  $(N \bmod N_{dep})^2 + (N - (N \bmod N_{dep}))N_{dep}$  can be approximated by  $NN_{dep}$ . We will use this approximation from now on. This means that we will validate the results from moments analysis in the TKP software pipeline by comparing the differences between the measured positions and the true positions with the following theoretical position variances:

$$\sigma(\bar{x})^2 = \frac{\sigma(n)^2}{16(C-T)^2} \frac{\theta_m \theta_b \theta_B}{\theta_M} \ln^2 \frac{C}{T} \quad (2.79)$$

$$\sigma(\bar{y})^2 = \frac{\sigma(n)^2}{16(C-T)^2} \frac{\theta_M \theta_b \theta_B}{\theta_m} \ln^2 \frac{C}{T} \quad (2.80)$$

where we have replaced  $N_{dep}$  by  $\pi\theta_b\theta_B/4$ , with  $\theta_B$  and  $\theta_b$  the correlation lengths along the minor and major axis of the ellipse, respectively. Generally,  $\theta_b \simeq \theta_m$  and  $\theta_B \simeq \theta_M$  are good approximations, but we will not use them to avoid loss of generality. It is non trivial to derive the variances of other quantities, like the peak, the axes and the position angle, in the same way. Instead, we try the same approach as Condon (1997) by relating all the variances to a generalized signal to noise,  $\rho$ :

$$\rho^2 = \frac{4}{\ln 2} \frac{\theta_m \theta_M}{\theta_b \theta_B} \frac{(C-T)^2}{\sigma(n)^2 (\ln C - \ln T)^2} \quad (2.81)$$

Now we can derive all variances from these equations:

$$\frac{\sigma(C)^2}{C^2} = \frac{2}{\rho^2} \quad (2.82)$$

$$= 8 \ln(2) \frac{\sigma(\bar{y})^2}{\theta_M^2} = 8 \ln(2) \frac{\sigma(\bar{x})^2}{\theta_m^2} \quad (2.83)$$

$$= \frac{\sigma(\theta_M)^2}{\theta_M^2} = \frac{\sigma(\theta_m)^2}{\theta_m^2} \quad (2.84)$$

$$= \frac{\sigma(\phi)^2}{2} \frac{(\theta_M^2 - \theta_m^2)^2}{(\theta_M \theta_m)^2} \quad (2.85)$$

The integrated flux and its relative variance can then be obtained from the previously derived parameters and their variances, in the same way as from the fitted parameters in the next paragraph:

$$I = C \frac{\theta_m \theta_M}{\Theta_m \Theta_M} \quad (2.86)$$

$$\frac{\sigma(I)^2}{I^2} = \frac{\sigma(C)^2}{C^2} + \frac{\theta_B \theta_b}{\theta_M \theta_m} \left[ \frac{\sigma(\theta_M)^2}{\theta_M^2} + \frac{\sigma(\theta_m)^2}{\theta_m^2} \right] \quad (2.87)$$

where the clean beam FWHM minor and major axes are denoted by  $\Theta_m$  and  $\Theta_M$ , respectively.

The ensemble averaged fitted peak is biased (too high, see Refregier & Brown 1998, paragraph 3.2, for some background). The correction for this bias is shown in equation 2.96. For the peaks from the maximum pixel method, with the fudge factor from equation 2.49, or from the yet unimplemented "tweaked moments", we do not expect that any bias correction is necessary, but this remains to be verified.

## Gauss fitting

**Calculating the Gaussian model parameters** Gauss fitting can do more accurate astrometry than moments, as shown in figure 2.4. However, Gauss fitting may sometimes fail while moments can always be calculated. In all cases the moments are used as initial values for Gauss fitting. The actual fit on the data is performed by `scipy.optimize.leastsq` on the errorfunction, i.e. the difference of the Gauss and the data. This routine uses a modification of the robust Levenberg-Marquardt algorithm. It is just a wrapper around MINPACK's LMDIF and LMDER algorithms (Moré 1977). These algorithms do not provide for constraints on the fit, contrary to the procedure for the NVSS catalogue (Condon et al. 1998). Consequently, the boundary condition that the fitted semi-major and semi-minor axes can never be less than the corresponding axes of the clean beam, as applied in the production of the NVSS catalogue, is not forced.

**Error bars** Formulae for the errors from Gauss fitting were first derived by Condon (1997):

$$\frac{\sigma(C)^2}{C^2} = \frac{2}{\rho^2} \quad (2.88)$$

$$= 8 \ln(2) \frac{\sigma(y_0)^2}{\theta_M^2} = 8 \ln(2) \frac{\sigma(x_0)^2}{\theta_m^2} \quad (2.89)$$

$$= \frac{\sigma(\theta_M)^2}{\theta_M^2} = \frac{\sigma(\theta_m)^2}{\theta_m^2} \quad (2.90)$$

$$= \frac{\sigma(\phi)^2}{2} \frac{(\theta_M^2 - \theta_m^2)^2}{(\theta_M \theta_m)^2} \quad (2.91)$$

and

$$\sigma(\alpha)^2 = \sigma(x_0)^2 \sin(\phi)^2 + \sigma(y_0)^2 \cos(\phi)^2 \quad (2.92)$$

$$\sigma(\delta)^2 = \sigma(x_0)^2 \cos(\phi)^2 + \sigma(y_0)^2 \sin(\phi)^2 \quad (2.93)$$

Here  $\sigma$  indicates standard deviation and  $C$ ,  $I$ ,  $(\alpha, \delta)$ ,  $\theta_M$ ,  $\theta_m$  and  $\phi$  indicate the fitted peak flux density, integrated flux, position, FWHM major and minor axis and position angle, respectively.  $(x_0, y_0)$  is the fitted position in pixel coordinates. The y-axis is implicitly assumed to be aligned with the major axis of the elliptical Gaussian.  $\rho$  is a generalized "signal to noise", which is given by

$$\rho^2 = \frac{\pi}{8 \ln 2} \frac{\theta_M \theta_m C^2}{\sigma(n)^2} \quad (2.94)$$

if there is no pixel to pixel correlation of  $\sigma(n)$ , the background rms noise. Condon (1997) also derived semi-quantitative formulae for the variances in the presence of correlated noise, which were generalized by Hopkins et al. (2003, equation 41) for a non-circular synthesized beam:

$$\rho^2 = \frac{\theta_M \theta_m}{4\theta_B \theta_b} [1 + (\frac{\theta_B}{\theta_M})^2]^{\alpha_M} [1 + (\frac{\theta_b}{\theta_m})^2]^{\alpha_m} \frac{C^2}{\sigma(n)^2} \quad (2.95)$$

Here, the area of noise correlation is assumed to have a Gaussian shape and  $\theta_B$  and  $\theta_b$  are its FWHM major and minor axes. Generally, the synthesized beam is a good estimate of the noise correlation area.  $(\alpha_M, \alpha_m) = (1.5, 1.5)$  for amplitude errors and  $(\alpha_M, \alpha_m) = (2.5, 0.5)$  for the errors on  $x_0$  and  $\theta_m$ .  $(\alpha_m, \alpha_M) = (0.5, 2.5)$  for errors on  $y_0$ ,  $\theta_M$  and  $\phi$ . Equation 2.95 is formally exact only in the limit of high signal to noise, unfortunately (Condon 1997). Consequently, the validation of the correctness of the error bars on fitted parameters is only possible at high signal to noise.

The positional variances in the NVSS (Condon et al. 1998, equation 25) differ from the equations above by a factor 2, taking into account the very dirty synthesized beam of VLA snapshots.

Note that the catalogued value for the fitted peak,  $C_c$ , is corrected for bias from the local noise gradient:

$$C_c = C - \frac{\sigma(n)^2}{C} \quad (2.96)$$

The integrated flux and its relative variance are given by:

$$I = C_c \frac{\theta_m \theta_M}{\Theta_m \Theta_M} \quad (2.97)$$

$$\frac{\sigma(I)^2}{I^2} = \frac{\sigma(C)^2}{C_c^2} + \frac{\theta_B \theta_b}{\theta_M \theta_m} \left[ \frac{\sigma(\theta_M)^2}{\theta_M^2} + \frac{\sigma(\theta_m)^2}{\theta_m^2} \right] \quad (2.98)$$

where  $\Theta_m$  and  $\Theta_M$  indicate the minor and major FWHM clean beam axes, respectively.

For the NVSS (Condon et al. 1998, paragraph 5.2.5), optimal solutions are derived in the case where the source is unresolved in one or two dimensions. In these cases the variances of the peak flux density and the integrated flux are smaller, which is a consequence of the reduced number of degrees of freedom in the Gaussian fit. It is not necessary to redo the fit. The best estimates for both  $C_c$  and  $I$  can be derived by adjusting the results from the initial fit with all (=6) degrees of freedom (Condon 1997). We have not incorporated this in the TKP software pipeline, mainly because it is hard to make a clear distinction, from the results of a fit, between resolved, partially resolved and completely resolved sources. Of course, it is easy to make this distinction a posteriori, for instance if a later observation with a better resolution did not resolve the source. In that case, it is possible to rederive the optimum values for the peak flux density and the integrated flux as an image post processing step, using only the catalogued values for the source parameters. It is, however, unclear to me whether the solution for the position could be improved by redoing the fit with only 3 free parameters. Since in the error matrix (Condon 1997, equation 10) there are no source parameters that correlate or anticorrelate with position, I am inclined to state that a 3 parameter fit would not improve the accuracy of the position.

Condon (1997) derived the equations for the variances in the presence of correlated noise in a heuristic manner. A rigorous approach was pursued by Refregier & Brown (1998). They found formulae for errors in elliptical Gaussian fits which involve the noise Auto Correlation Function (ACF). Their formulae are complex and cumbersome to implement because the noise ACF in general cannot be approximated easily by an analytic expression. Nonetheless, it is conceivable that the Condon formulae will not suffice in the long term and that indeed some measurement of the noise ACF will be needed to calculate the error bars accurately.

## Deconvolution from the restoring beam

**Deconvolved parameters** The fitted axes and position angle are convolved with the restoring beam. As noted in paragraph 2.7.5, fitting in the TKP software is unconstrained, contrary to the NVSS (Condon et al. 1998) procedure. This means that the fitted axes may be smaller than the restoring beam axes. Consequently, deconvolution from the restoring beam is not always possible. The deconvolved axes and position angle are computed in AIPS (Greisen 2003) using the module DECONV.FOR. This module is available in the TKP pipeline. The equations for the deconvolved shape parameters,  $\vartheta_m$ ,  $\vartheta_M$  and  $\varphi$  are:

$$\begin{aligned}
\beta^2 &= (\theta_M^2 - \theta_m^2)^2 + (\Theta_M^2 - \Theta_m^2)^2 \\
&\quad - 2(\theta_M^2 - \theta_m^2)(\Theta_M^2 - \Theta_m^2) \cos(2(\phi - \Phi)) \\
2\vartheta_m^2 &= (\theta_M^2 + \theta_m^2) - (\Theta_M^2 + \Theta_m^2) - \beta \\
2\vartheta_M^2 &= (\theta_M^2 + \theta_m^2) - (\Theta_M^2 + \Theta_m^2) + \beta \\
\varphi &= \frac{1}{2} \arctan\left(\frac{(\theta_M^2 - \theta_m^2) \sin(2(\phi - \Phi))}{(\theta_M^2 - \theta_m^2) \cos(2(\phi - \Phi)) - (\Theta_M^2 - \Theta_m^2)}\right) + \Phi
\end{aligned}$$

Here, we have used  $\Theta_m$  and  $\Theta_M$  to denote the restoring beam FWHM axes. The clean beam position angle is indicated by  $\Phi$ .

---

## Bibliography

---

- Y. Benjamini, & Y. Hochberg, *Controlling the False Discovery Rate: a Practical Approach to Multiple Testing*, *J.R. Stat. Soc.* **57**, 289-300, 1995.
- E. Bertin, & S. Arnouts, *SExtractor: Software for Source Extraction*, *A&AS* **117**, 393–404, 1996.
- E. Bertin, *SExtractor v2.5 User's Manual*, draft version, Institut d'Astrophysique & Observatoire de Paris, 2006.
- A. Bijaoui, *Sky Background estimation and Application*, *A&A* **84**, 81–84, 1980.
- D.S. Briggs, F.R. Schwab & R.A. Sramek, in *Synthesis Imaging in Radio Astronomy II*, ed. G.B. Taylor, C.L. Carilli, & R. A. Perley (San Francisco, CA: ASP), 301, 1999.
- S. Chandrasekhar & G. Münch, *On the integral equation governing the true and the apparent rotational velocities of stars*, *ApJ* **111**, 142, 1950.
- J.J. Condon, *Errors in Elliptical Gaussian Fits*, *PASP*, **109**, 166-172, 1997.
- J.J. Condon, W.D. Cotton, E.W. Greisen, Q.F. Yin, R.A. Perley, G.B. Taylor, & J.J. Broderick, *The NRAO VLA Sky Survey*, *AJ* **115**, 1693, 1998.
- A.S. Eddington, *On a Formula for Correcting Statistics for the Effects of a known Probable Error of Observation*, *MNRAS* **73**, 359-360, 1913.
- A.S. Eddington, *The correction of statistics for accidental error*, *MNRAS*, **100** 354, 1940.
- E.B. Fomalont in *Synthesis Imaging in Radio Astronomy II*, ed. G.B. Taylor, C.L. Carilli, & R. A. Perley (San Francisco, CA: ASP), 301, 1999.
- P. Greenfield & R. Jedrzejewski, *Using Python for Interactive Data Analysis*, <http://stdsdas.stsci.edu/perry/pydatatut.pdf>, 2007.
- E.W. Greisen, 2003, in *Information Handling in Astronomy - Historical Vistas*, ed. A. Heck (Astrophysics and Space Science Library, Kluwer Academic Publishers, Dordrecht, Netherlands), 285, 109.
- D. Herranz, J.L. Sanz, M. López-Caniego & J. González-Nuevo, *A Bayesian Approach To*

- Flux Correction In Extragalactic Source Detection*, in *The 2006 IEEE International Symposium on Signal Processing and Information Technology*, vol. **1**, 541-544, 2006.
- J.A. Högbom, *Aperture Synthesis with a non-regular distribution of interferometer baselines*, *A&AS* **15**, 417 (1974).
- D.W. Hogg, & E.L. Turner, *A Maximum Likelihood Method to Improve Faint-Source Flux and Color Estimates*, *PASP* **110**, 727-731, 1998.
- A.M. Hopkins, C.J. Miller, A.J. Connolly, C. Genovese, R.C. Nichol, & L. Wasserman, *A New Source Detection Algorithm Using the False-Discovery Rate*, *AJ* **123**, 1086-1094, 2002.
- A.M. Hopkins, J. Afonso, B. Chan, L.E. Cram, A. Georgakakis, & B. Mobasher, *The Phoenix Deep Survey: The 1.4 GHz MicroJansky catalog*, *AJ* **125**, 465-477, 2003.
- J.F. Kenney & E.S. Keeping in *Mathematics of Statistics*, Pt. 1, 3rd ed. Princeton, NJ: Van Nostrand, pp. 50-54, 1962.
- J.D. Kraus, *Radio Astronomy*, *Cygnus-Quasar Books*, Powell, Ohio, 1986.
- C.J. Law, *Outline of the Transients Detection Pipeline and its Database Interactions*, TKP internal documentation, 2006.
- C.J. Miller, C. Genovese, R.C. Nichol, L. Wasserman, A. Connolly, D. Reichart, A. Hopkins, J. Schneider, & A. Moore, *Controlling the False Discovery Rate in Astrophysical Data Analysis*, *AJ* **122**, 3492-3505, 2001.
- N.R. Mohan & H.J.A. Röttgering, *Source Detection and Source Measurement for LOFAR Images*, LOFAR internal documentation, 2005.
- J.J. Moré, *The Levenberg-Marquardt algorithm: Implementation and Theory*, *Numerical Analysis*, in *Lecture Notes in Mathematics* ed. G.A. Watson, **630**, 1977.
- F. Patat, *A robust algorithm for sky background computation in CCD images*, *A&A* **401**, 797-807, 2003.
- A. Refregier, & S.T. Brown, *Effect of Correlated Noise on Source Shape Parameters and Weak Lensing Measurements*, <http://arxiv.org/pdf/astro-ph/9803279v1>, 1998.
- R.B. Rengelink, *The Westerbork Northern Sky Survey: The Cosmological Evolution of Radio Sources*, PhD Thesis, *Rijksuniversiteit Leiden*, 1998.
- J. Swinbank, *Transients Key Project: Pipeline Data Flow*, TKP internal documentation, 2007.
- P. Teerikorpi, *Influence of a generalized Eddington bias to galaxy counts*, *A&A* **424**, 73-78, 2004.
- Q.D. Wang, *Correction for the flux measurement bias in X-ray source detection*, *ApJ* **612**, 159-167, 2004.