



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.

Zeroth order validation of TKP source extraction and source measurement code

3.1 Abstract

The source extraction and source measurement modules in the TKP software pipeline were validated by processing a large number of different maps with and without artificial point sources. To make those maps, the visibilities of a certain VLA observation were replaced by Gaussian noise. The point sources, with the size and shape of the appropriate restoring beam, were inserted in the images. We tested the background and noise estimators, the deblending algorithm, the False Discovery Rate algorithm and the peak flux density and position measurements.

3.2 Description

Zeroth order validation is the first step in the validation of the TKP source extraction and source measurement software. It comprises of a series of tests on artificial maps whose properties are completely controlled and known. Sources are therefore inserted directly in the image plane, to avoid the introduction of biases from a cleaning process. Although these biases are probably small and can be quantified, we decided to conveniently avoid them at this first stage in the validation process. The next step, first order validation, would involve the insertion of sources in visibilities. The maps from those visibilities would then be cleaned using some appropriate algorithm. Subsequently, the measured fluxes and positions would be compared with the corresponding values of the inserted sources. After that, second order validation would use real maps with cleaned sources, e.g., NVSS and WENSS maps, and

compare the source parameters derived by the TKP software with the corresponding catalogues that were constructed using different source extraction packages.

3.3 General

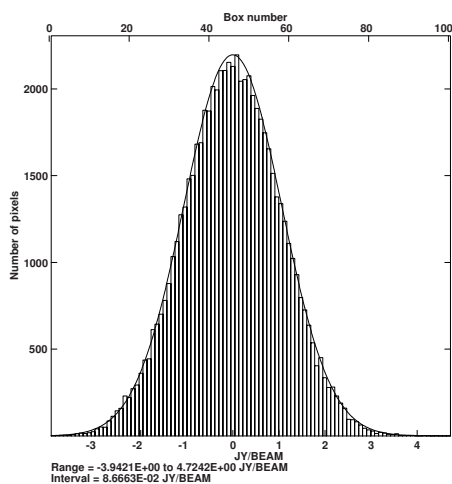
All of the output from the TKP source extraction and source measurement software, such as noise and background levels and source parameters are considered sufficiently accurate if they are the "best" estimators of the true values. However, "best" estimator as such is an ill defined concept. In general, since an extremely large number of maps will be processed by the TKP, we will want all of our estimators to be unbiased. This means that the sample estimators of a certain quantity averaged over the total number of samples, should be equal to that quantity over the complete ensemble. We also require the sample estimators to have the smallest possible mean squared error (M.S.E.). It is known from statistics that these two requirements cannot always be met simultaneously and I will discuss those cases when they occur.

3.4 Source free maps

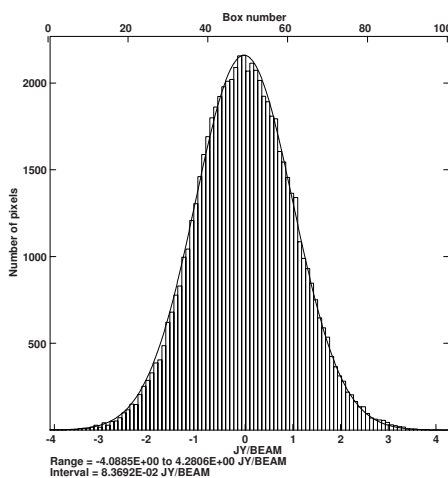
The maps were made by adding Gaussian noise to the visibilities of the dataset from the discovery observation of GCRT J1745-3009 (Hyman et al. 2005), using the AIPS task 'UVMOD'. The actual data were not used, just the (u, v) coverage of this observation. Gaussian noise can be added to the visibilities in units of Jy/weight, by setting the adverb 'FLUX' to the desired rms value. To avoid any complexity from the visibility weights, we changed them all to 1 using the AIPS task 'WTMOD'. This ensures that, for both of the polarization products, the pixel rms noise will be equal to the value of the adverb 'FLUX' divided by the square root of the number of visibilities in the case of natural weighting. We aimed for an rms noise value in Stokes I dirty images of 1 Jy beam^{-1} , so we set the adverb 'FLUX' in UVMOD to $\sqrt{2} \cdot 807417$ to make the first "random" (u, v) dataset. 807417 is the total number of visibilities, the extra factor $\sqrt{2}$ is required to attain the desired noise level in Stokes I. We made another 9996 (u, v) datasets, every new dataset was made by adding noise with a standard deviation of $\sqrt{807417}$ to $-1/\sqrt{2}$ times the previous dataset. In this way, we made sure that all 9997 datasets had the same noise level.

The dirty images were made with natural weighting by setting the adverb 'UUVWTFN' equal to 'NA' in the AIPS task 'IMAGR'. A pixel size of $12''$ in both dimensions was chosen and an image size of 256^2 pixels. The FWHM size of the dirty beam is $67.14'' \times 56.15''$, so the dirty beam was sampled with more than 4 pixels in both dimensions.

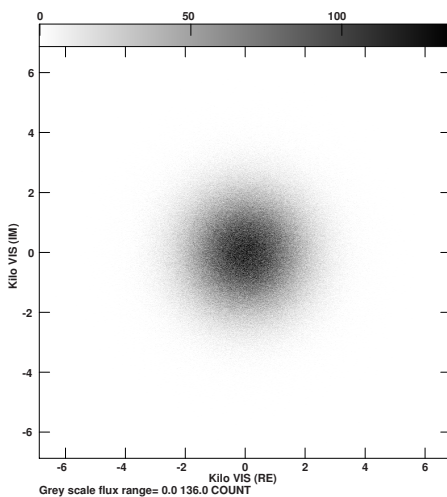
To check whether the distribution of the noise in both the visibilities and the maps is normally distributed, we made histograms of both the real and imaginary part of the visibilities for the first dataset. We have also investigated if the cumulative noise of the visibilities in the datasets was still Gaussian. This was indeed the case. However, we found that deviations from normal distributions do occur when new datasets are made by adding noise to the previous dataset times $1/\sqrt{2}$ instead of $-1/\sqrt{2}$.



(a) First image



(b) Last image



(c) 3D histogram of the real and imaginary part of the visibilities in our last source free dataset.

Figure 3.1: 3D Histogram of the last source free (u, v) dataset as produced by 'UVMOD' (bottom) and histograms of pixel values (top).

A 3D histogram of the real and imaginary part of the visibilities of the last source free dataset is shown at the bottom of figure 3.1. Histograms of the pixel values of the first and last image are also shown in that figure. This indicates that we have indeed produced Gaussian noise in our images. The noise will also be spatially correlated within the psf, as we desired.

3.5 Validation of the TKP background mean and background noise estimation

3.5.1 General

For validating the correct determination of the background mean and noise, we divided each of the 9997 source free images into 7×7 subimages, of size 32×32 pixels. Of course, since each of the images has a size of 256×256 pixels, we could have used 8×8 subimages per image, but we wanted to avoid any edge effects by excluding an edge of 16 pixels along all sides. So the total number of submaps processed is $9997 \times 49 = 489853$ and the total number of pixels processed is $5 \cdot 10^8$. These numbers are large enough to provide statistics that are accurate enough for validation.

For source free maps, the plain mean is the true background level, so this gives us a way to assert the accuracy of the TKP background level determination, which is essentially a mode estimator. Without sources, the mean should be equal to the mode. In the presence of sources the mean background should be close to the mode of all the pixel values. Thus, two approximations are involved when sources are present. The mode is estimated and the mode itself is an approximation to the true source free mean background, because pixels from weak sources and the outer pixels from strong sources can shift the mode depending on the density of sources.

3.5.2 Source free maps

We ran tests on these 489853 maps as well as on an equal number of maps of the same size with uncorrelated Gaussian noise. We also compared the TKP method with an alternative method commonly used: histogram fitting. For every subimage we made a histogram with 100 bins and fitted a 1-D Gaussian to it. Figures 3.2 and 3.3 show all the results graphically and table 3.1 contains the most important numbers. The main conclusions are that neither method shows any serious bias in determining the means or modes, but that the variance of the means is smaller when they are determined by histogram fitting. For uncorrelated noise this effect is most significant.

It is remarkable that, in the case of correlated noise, the statistics of the subimages are better described by 30 than by 50 independent elements. The FWHM synthesized beam is covered by 20.56 pixels, so if every synthesized beam would correspond to an independent element, that would amount to $1024/20.56 = 49.8$ independent elements. Evidently, the spatial correlation between pixels is significantly larger than the FWHM synthesized beam. In the TKP software, the user can enter the pixel correlation length in both dimensions. This is indeed a very important parameter, since it is used in the determination of the error

Table 3.1: Comparison of background means in source free maps as determined by TKP software and by histogram fitting.

Method	uncorrelated noise			correlated noise				
	Overall bias (Jy) (True-measured)	Significance of overall bias ¹	Variance (Jy ²)	Theoretical plain variance ² (N=1024)	Overall bias (Jy) (True-measured)	Significance of overall bias ¹	Variance (Jy ²)	Theoretical plain variance ² (N=50)
TKP	$-2.5 \cdot 10^{-5}$	-0.57	0.0044	$9.8 \cdot 10^{-4}$	$-5.7 \cdot 10^{-5}$	-0.28	0.048	0.020
Histogram fitting	$2.5 \cdot 10^{-5}$	0.57	0.0015	$9.8 \cdot 10^{-4}$	$-9.9 \cdot 10^{-6}$	-0.049	0.038	0.020

¹ As a multiple of the standard deviation of the mean for the total number of maps processed: $\frac{\sigma}{\sqrt{N=489853}} \approx \frac{Jy}{\sqrt{N=489853}}$.

² In Jy².

Table 3.2: Comparison of background variances in source free maps as determined by TKP software and by histogram fitting.

Method	uncorrelated noise			correlated noise				
	Overall bias (Jy) (True-measured)	Significance of overall bias ¹	Variance (Jy ²)	Theoretical plain variance ² (N=1024)	Overall bias (Jy) (True-measured)	Significance of overall bias ¹	Variance (Jy ²)	Theoretical plain variance ² (N=50)
TKP	$3.9 \cdot 10^{-3}$	62	0.002	0.0020	0.0067	23	0.036	0.040
Histogram fitting	$-9.9 \cdot 10^{-4}$	-16	0.0046	0.0020	-0.0042	-15	0.052	0.040

¹ As a multiple of the standard deviation of the variance for the total number of maps processed: $\sqrt{\frac{2(N-1)\sigma^4}{N^2+489853}} \approx \sqrt{\frac{2\sigma^4}{N^2+489853}} \approx Jy^2 \sqrt{\frac{2}{N+489853}}$.

² In Jy².

bars in fitted source parameters. The product of the two correlation lengths ($\times\pi/4$) should correspond to an area somewhat larger than the FWHM synthesized beam such that the number of independent elements matches the pixel statistics, similar to the Gaussians with $N = 30$ in Figure 3.3.

The distribution of the submap noise variances, s^2 is theoretically given by a Pearson type III distribution ¹, at least in the case of uncorrelated noise:

$$f(s^2) = \frac{\left(\frac{N}{2\sigma^2}\right)^{\frac{N-1}{2}}}{\Gamma\left(\frac{N-1}{2}\right)} (s^2)^{\frac{N-3}{2}} e^{-\frac{Ns^2}{2\sigma^2}} \quad (3.1)$$

where Γ denotes the gamma function. A plot of this function for unit variance and 20 elements per sample is shown in figure 3.4. The mean of this distribution is $\frac{N-1}{N}\sigma^2$. This result should not surprise us since we know that s^2 is a biased estimator of the true variance, σ^2 . The plain noise variances are compared with Pearson type III distributions in figure 3.5 for both correlated and uncorrelated noise. Both the theoretical curves as well as the actual noise variances were corrected for biases by multiplying the abscissa values by $\frac{N}{N-1}$. It is striking that the shape of the distribution of plain noise variances is best matched by a Pearson type III distribution with $N > 60$ although the distributions of the means in figure 3.3 were indicating that the statistics of the submaps were well described by $N \approx 30$. Nevertheless, it turns out that solving N in the equation that corrects for the bias in the variances:

$$\sigma^2 = \frac{N}{N-1} \langle s^2 \rangle \quad (3.2)$$

where $\langle \rangle$ denotes the averaging over the samples, gives $N \approx 30$ as before. In calculating N we have assumed that the true noise variance is equal to the variance of the complete ensemble of 489853 submaps:

$$\sigma^2 \approx s_{tot}^2 \quad (3.3)$$

A similar assumption was actually made in determining the biases with respect to the true mean in the sense that the true mean was assumed to be equal to the mean of the complete ensemble.

The fact that we find $N = 30$ again from equation 3.2 should not be a surprise. Consider the following equation for the variance of the complete ensemble of submaps:

$$\sigma^2 \approx s_{tot}^2 = \langle s^2 \rangle + \langle \bar{x}^2 \rangle - \langle \bar{x} \rangle^2 = \langle s^2 \rangle + \langle \bar{x}^2 \rangle - \mu^2 = \langle s^2 \rangle + \sigma(\bar{x})^2 \quad (3.4)$$

with μ is the average of all pixel values in all samples and $\sigma(\bar{x})^2$ the ensemble variance of the means. Equation 3.4 is equivalent to equation 3.2 if

$$\sigma(\bar{x})^2 = \sigma^2 / N \quad (3.5)$$

¹<http://mathworld.wolfram.com/SampleVarianceDistribution.html>

However, from the ensemble statistics of the means we have already determined $\sigma(\bar{x})^2$:

$$\sigma(\bar{x})^2 \simeq \sigma^2/30 \quad (3.6)$$

It follows that we find $N = 30$ again. Summarizing, the bias of the noise variance is well corrected by $N = 30$, while the variance of the noise variance seems to correspond to a number of independent elements per sample at least twice as large.

Histograms of measured variances calculated by the TKP software and by histogram fitting are shown in figures 3.6 and 3.7. The most important results from the validation runs are shown in table 3.2. The main conclusion is that the TKP method outperforms histogram fitting in terms of the mean squared error, i.e., the variance of the variance, but not with respect to the overall bias. The rms noise as calculated by the TKP software is significantly biased too low, but only by 0.2% in the case of uncorrelated noise and by 0.3% in the case of correlated noise. This will not compromise the overall performance of the pipeline in any way. The advantage of lower mean squared errors seems more important than the disadvantage of a larger, but still relatively small bias. In the TKP pipeline sources are detected above a threshold defined in terms of the local rms noise. A robust estimation of the local rms noise is therefore necessary since any severe underestimate of the local rms noise will cause false alerts, i.e., noise peaks will be falsely identified as sources.

For the purpose of validating the TKP noise variance estimation we have made use of the following formula for the variance of the plain variance ², s^2 :

$$\sigma^2(s^2) = 2 \frac{N-1}{N^2} \sigma^4 \simeq \frac{2}{N} \sigma^4 \quad (3.7)$$

$$\sigma(s^2) \simeq \sqrt{\frac{2}{N}} \sigma^2 \quad (3.8)$$

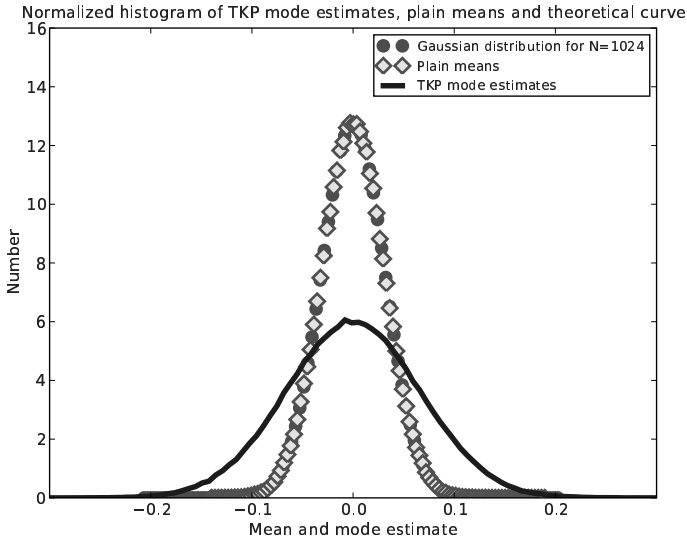
The latter expression is used to determine the significance of the overall bias.

3.5.3 Maps with sources

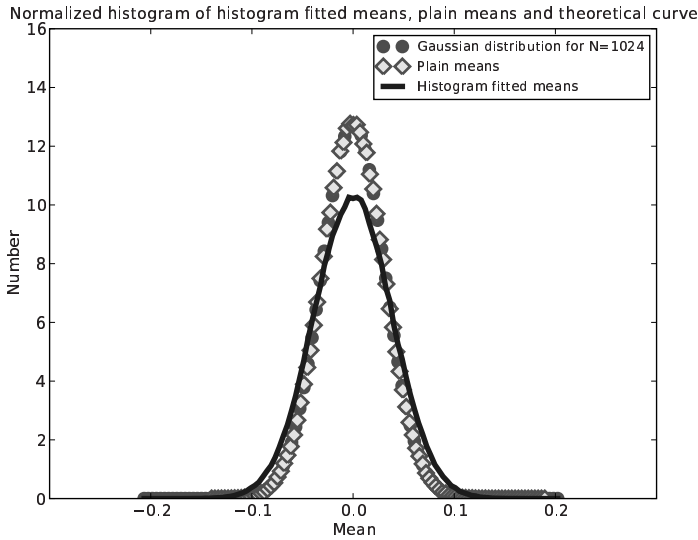
With sources added to the maps, determining the rms and mean of the background becomes more challenging. When the number of source pixels approaches the number of noise pixels, distinguishing between them becomes increasingly difficult. At some point it will be clear that determining the background characteristics will become impossible, e.g., in the case of a "classical" confusion limited map (Hogg 2001; Condon 1974). Here, it is our goal to assess the proper operation of the TKP software for some reasonable concentration of sources in a map, i.e. when the number of source pixels is still a minor fraction of the total number of pixels.

We added sources to the 9997 maps of the previous section, with fluxes randomly distributed between 0 and 5 Jy. The source positions were also randomly chosen. We did three test

²From <http://mathworld.wolfram.com/SampleVarianceDistribution.html>

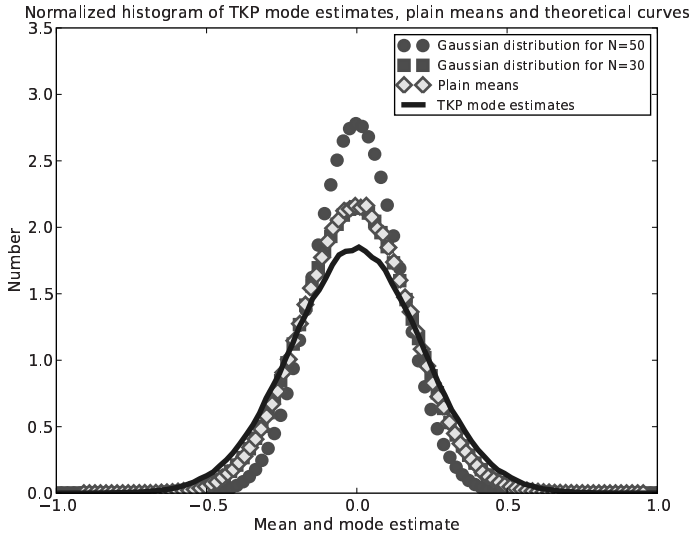


(a) TKP method for 1024 uncorrelated pixels per submap. Naturally, the plain means completely coincide with the theoretical distribution.

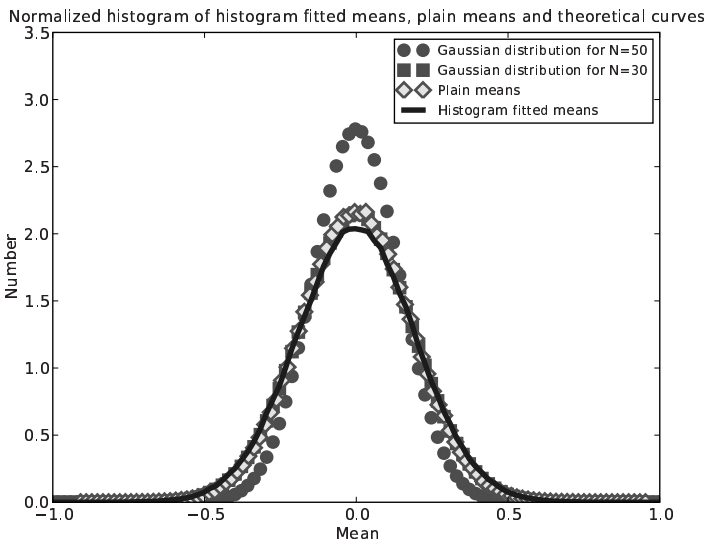


(b) Histogram fitting method for 1024 uncorrelated pixels per submap. Naturally, the plain means completely coincide with the theoretical distribution.

Figure 3.2: Comparison of background level determination methods, for uncorrelated noise in the source free case.



(a) TKP method for 1024 correlated pixels per submap, the FWHM beam-size is 20.56 pixels. The plain means completely coincide with the theoretical distribution for N=30.



(b) Histogram fitting method for 1024 correlated pixels per submap, the FWHM beamsize is 20.56 pixels. The plain means completely coincide with the theoretical distribution for N=30.

Figure 3.3: Comparison of background level determination methods, for correlated noise in the source free case.

Table 3.3: Comparison of background characteristics in maps with sources as determined by TKP pipeline software.

Average number of sources	Background mean estimates			Background variance estimates				
	Overall bias (Jy) (True-measured)	Significance of overall bias ¹	Variance (Jy ²)	Theoretical plain variance (N=50) ²	Overall bias (Jy) (True-measured)	Significance of overall bias ³	Variance (Jy ²)	Theoretical plain variance (N=50) ²
100	-0.073	-361	0.052	0.020	-0.13	-444	0.060	0.040
200	-0.15	-732	0.058	0.020	-0.25	-867	0.088	0.040
400	-0.31	-1523	0.072	0.020	-0.53	-1846	0.171	0.040

¹ As a multiple of the standard deviation of the mean for the total number of maps processed: $\frac{\sigma}{\sqrt{N=489853}} \approx \frac{1Jy}{\sqrt{N=489853}}$.

² In Jy².

³ As a multiple of the standard deviation of the variance for the total number of maps processed: $\sqrt{\frac{2(N-1)Jy^4}{N^2+489853}} \approx \sqrt{\frac{2Jy^4}{N^2+489853}} \approx Jy^2 \sqrt{\frac{2}{N+489853}}$.

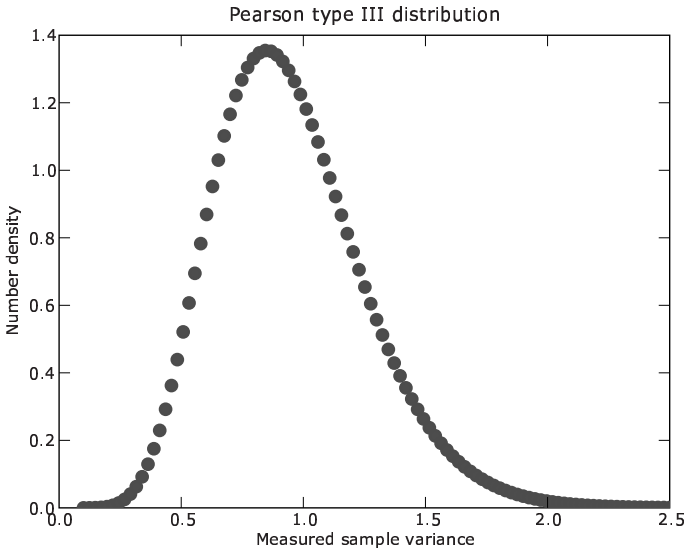
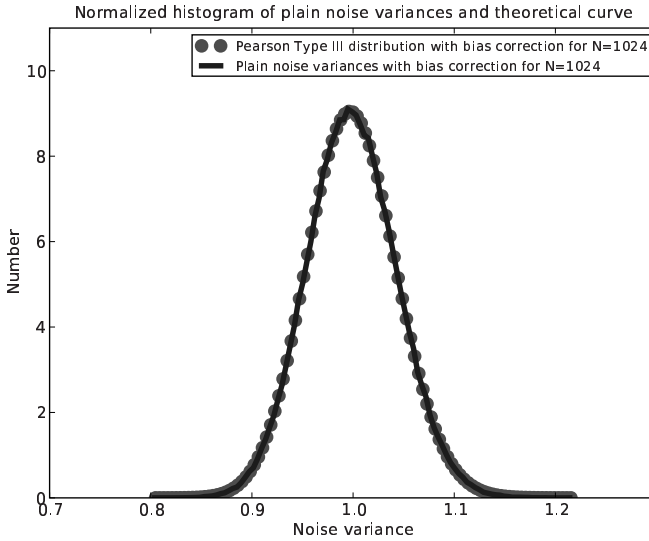


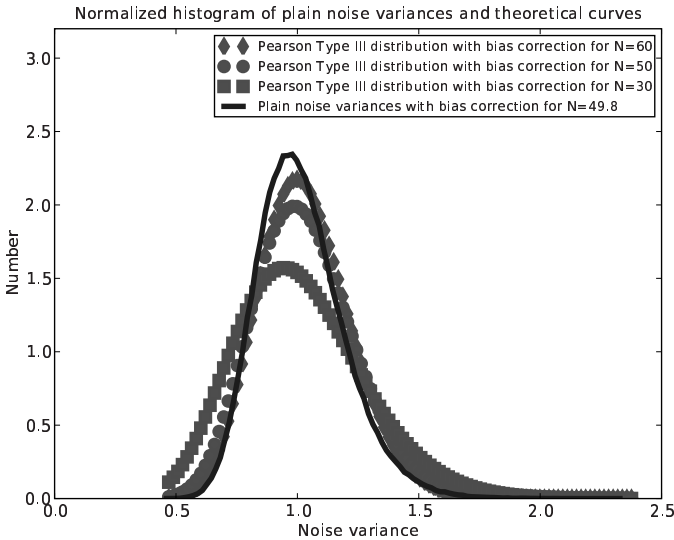
Figure 3.4: The distribution of sample variances is a Pearson type III distribution. This plot is for $N = 20$ elements per sample and for unit variance, $\sigma^2 = 1$. The mean of this distribution equals $\frac{N-1}{N}\sigma^2$.

runs, one with 100 inserted sources per map, one with 200 and one with 400. The results for all three runs are shown in table 3.3 and figures 3.8, 3.9 and 3.10. It is evident that both the biases and mean squared errors become significant when the average number of sources per map is 400. Both the background level and the noise are severely overestimated. The number of pixels in a map, in units of the number of pixels in the FWHM synthesized beam, is $49 \cdot 1024/20.56 = 2440$. In units of truly independent pixels, as we have calculated above, this number is about $49 \cdot 30 = 1470$. Thus, 400 sources represents a fraction of 27% of the total number of independent pixels. Apparently this fraction is too high to determine the background characteristics accurately.

It remains to be investigated where this error comes from. Is the mode shifted away from the true mean background or is the TKP mode estimator not accurate enough? In the former case histogram fitting will not give better results. In the latter case, it may be more accurate. Histogram fitting has the disadvantage that the bin size should be chosen carefully and that, in the presence of sources, only the highest part of the histogram should be used for fitting. Alternatively, one could decide to use only the negative part of the histogram for fitting. This reduces contamination by source pixels in Stokes I images. Note that negative pixel values can still have a small contribution from a source. Also, in radio maps there are often negative pixel values near sources from cleaning artefacts or negative sidelobes. Problems with accuracy remain if the histogram does not have a sharp peak. According to Patat (2003, paragraph 3) the statistics in small subimages are probably not good enough to derive background characteristics from histograms. Also, histogramming is a computationally expensive

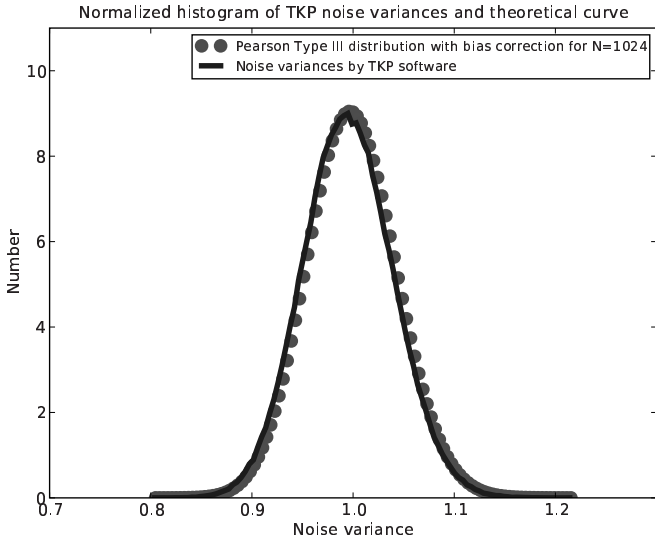


(a) Plain noise variances for 1024 uncorrelated pixels per submap. Naturally, this completely coincides with the theoretical distribution.

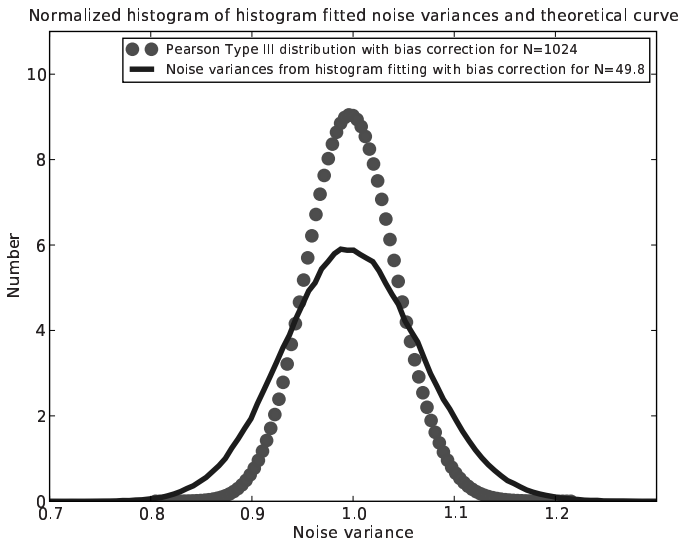


(b) Plain noise variances for 1024 correlated pixels per submap, the FWHM beamsize is 20.56 pixels. Also shown are Pearson type III distributions for various N .

Figure 3.5: Plain noise variances compared with theoretical distributions in the source free case.

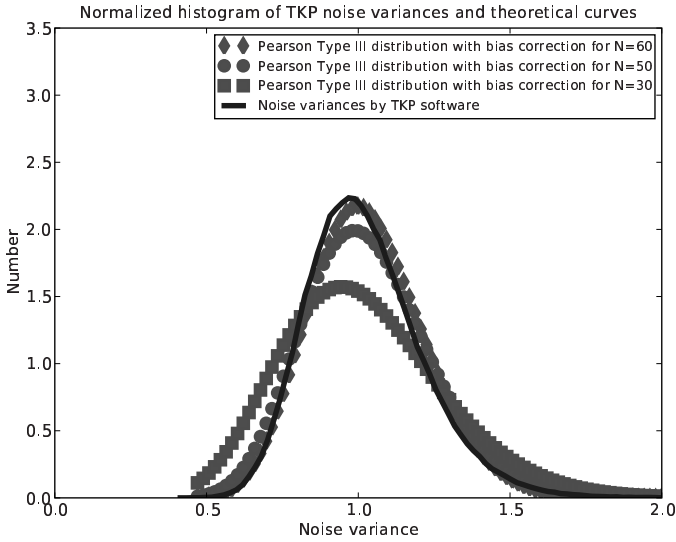


(a) TKP method for 1024 uncorrelated pixels per submap.

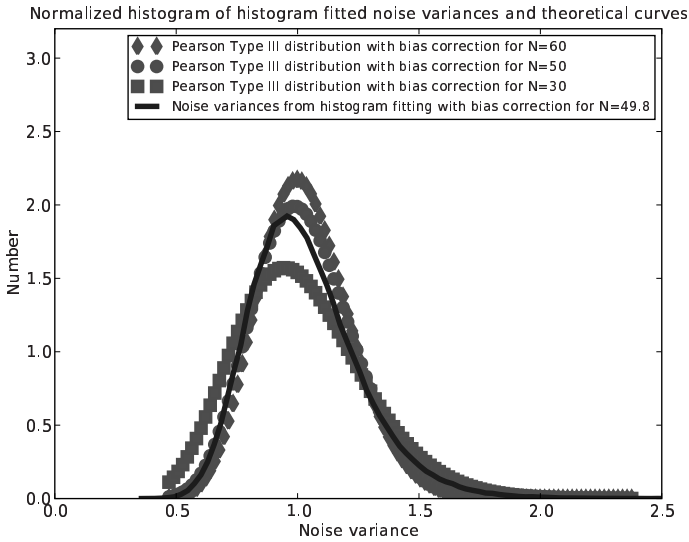


(b) Histogram fitting method for 1024 uncorrelated pixels per submap.

Figure 3.6: Comparison of noise variance determination methods, for uncorrelated noise in the source free case.

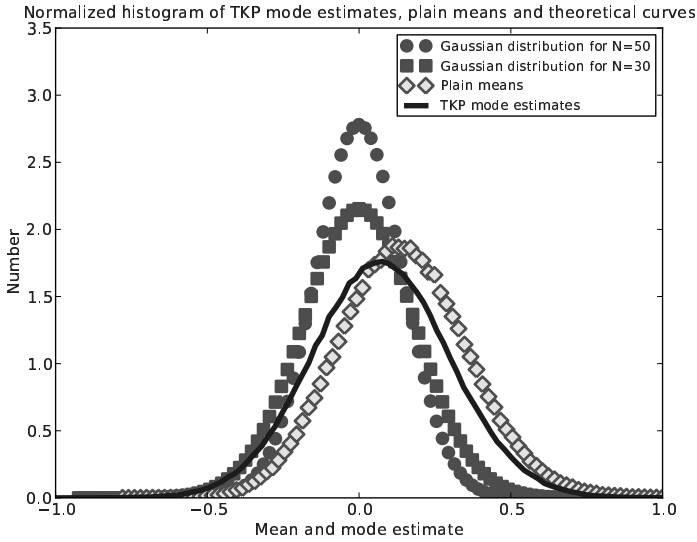


(a) TKP method for 1024 correlated pixels per submap, the FWHM beam-size is 20.56 pixels.

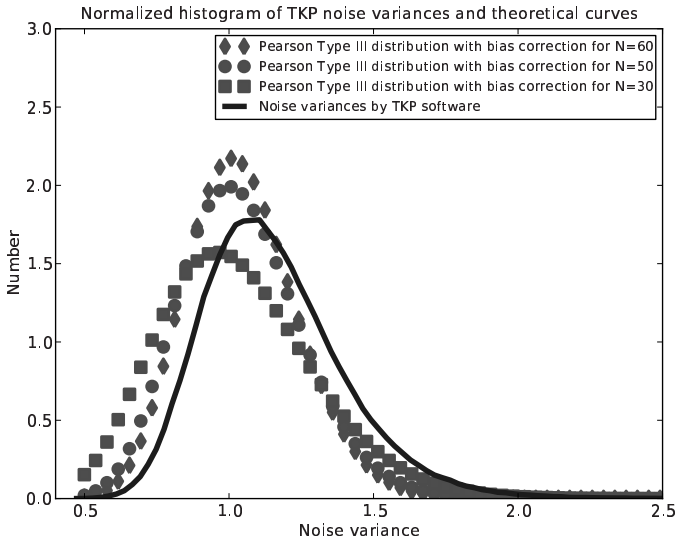


(b) Histogram fitting method for 1024 correlated pixels per submap, the FWHM beamsize is 20.56 pixels.

Figure 3.7: Comparison of noise variance determination methods, for correlated noise in the source free case.

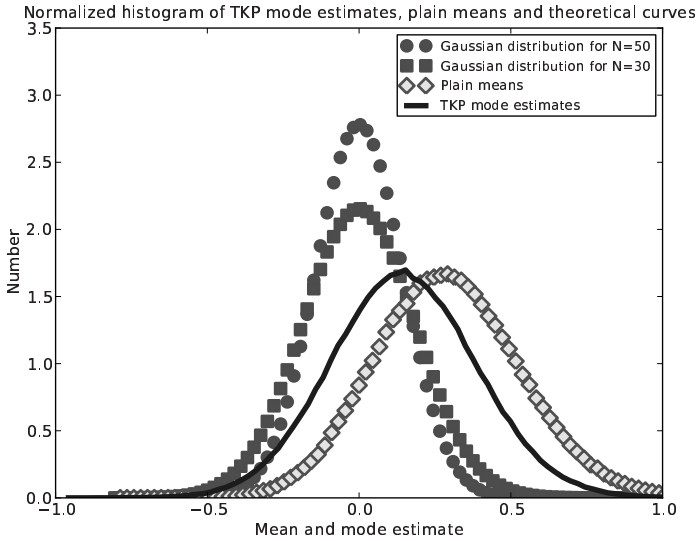


(a) TKP mean background estimates, theoretical curves and plain means for an average of 100 sources per map.

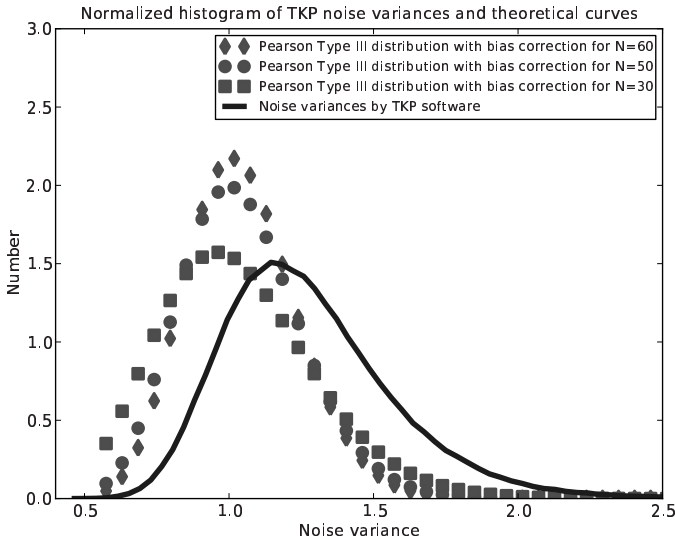


(b) TKP noise variance estimates and theoretical curves for an average of 100 sources per map.

Figure 3.8: TKP background level and noise estimates with 100 sources per 256^2 pixel map. The FWHM beamsize is 20.56 pixels.

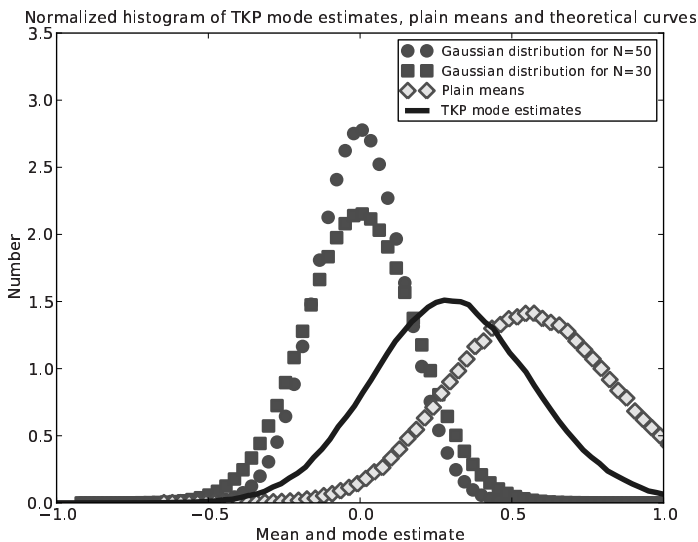


(a) TKP mean background estimates, theoretical curves and plain means for an average of 200 sources per map.

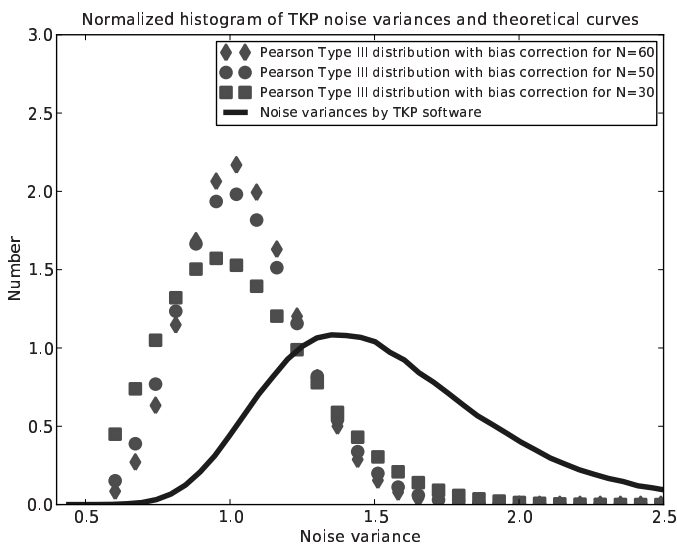


(b) TKP noise variance estimates and theoretical curves for an average of 200 sources per map.

Figure 3.9: TKP background level and noise estimates with 200 sources per 256^2 pixel map. The FWHM beamsize is 20.56 pixels.



(a) TKP mean background estimates, theoretical curves and plain means for an average of 400 sources per map.



(b) TKP noise variance estimates and theoretical curves for an average of 400 sources per map.

Figure 3.10: TKP background level and noise estimates with 400 sources per 256^2 pixel map. The FWHM beamsize is 20.56 pixels.

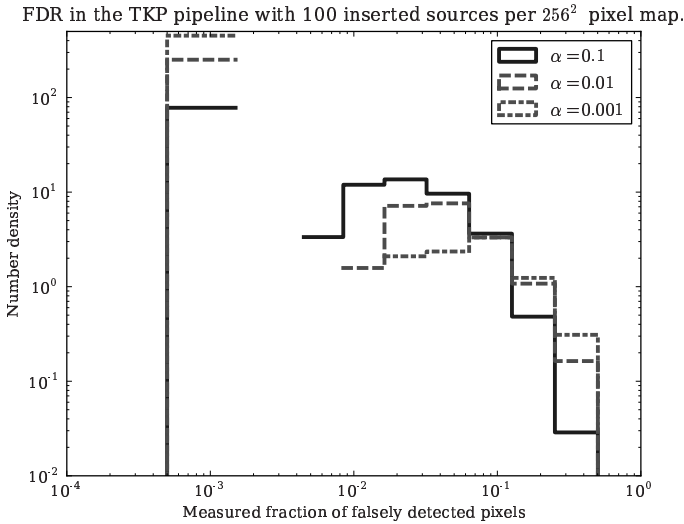


Figure 3.11: The distribution of the number of falsely detected pixels divided by the total number of detected pixels with 100 inserted sources per 256^2 pixel map. All three histograms are normalized equally. In order to represent, in this loglog plot, the bin with most counts, i.e., with zero falsely detected pixels, we have chosen to center the bins on the right bin edges. The left bin contains all fractions in the interval $[0, 1e-3]$. All other bins have equal size on a log scale.

process. We expect that improvements could come from an asymmetric clipping algorithm, possibly similar to the one proposed by Ratnatunga & Newell (1984). Perhaps the simple adjustment of clipping only the pixels that are larger than the median+ 3σ and **not** the pixels that are smaller than the median- 3σ will remove most of the bias if the algorithm is applied to properly cleaned maps where negative sidelobes are negligible.

3.6 False Discovery Rate algorithm

The False Discovery Rate algorithm was derived relatively recently (Benjamini & Hochberg 1995) and implemented in SFIND 2.0, the MIRIAD source extraction algorithm (Hopkins et al. 2002). The prime advantage of it is that the maximum number of falsely detected source pixels is expressed as a fraction of the total number of detected pixels. For the TKP, this can be translated into a maximum allowed fraction of false alerts for the discovery of new transients, which is a useful quantity. Without this algorithm, the threshold would have to be set as a multiple of the local rms noise ($n\sigma$), which can be converted to a number of false positives only after the total number of pixels processed has been determined. It is a slight disadvantage of the FDR algorithm that it refers to source pixels, not sources. Hopkins et al. (2002) pointed out that the FDR algorithm provides a less than strict constraint with

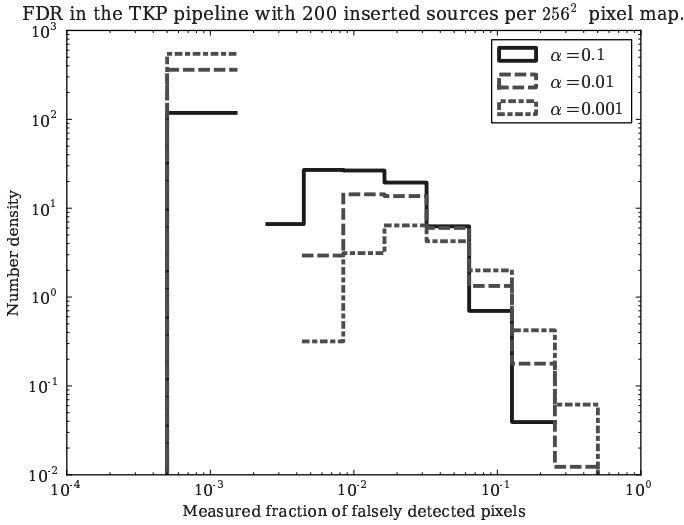


Figure 3.12: The distribution of the number of falsely detected pixels divided by the total number of detected pixels with 200 inserted sources per 256^2 pixel map. All three histograms are normalized equally. In order to represent, in this loglog plot, the bin with most counts, i.e., with zero falsely detected pixels, we have chosen to center the bins on the right bin edges. The left bin contains all fractions in the interval $[0, 1e-3]$. All other bins have equal size on a log scale.

respect to sources but that it is still a quite good estimator in that respect. These authors also note that FDR loses some of its fidelity when contiguous pixels with values between the detection threshold and the (lower) analysis threshold are included. For simplicity, we have only considered source pixels in our tests and we have always set the analysis threshold equal to the detection threshold.

It should be clear that the maximum allowed fraction refers to ensemble averages (Miller et al. 2001). This means that in the processing of individual maps the fraction of falsely detected source pixels can exceed the overall maximum fraction.

Before we discuss the results from our tests it should be noted that it is not precisely α , the maximum allowed fraction of false positives, that is entered in the algorithm, but rather α/C_N . The quantity C_N expresses the degree of correlation between the pixels:

$$C_N = \sum_{i=1}^N \frac{1}{i} \quad (3.9)$$

where N is the total number of pixels in the image if all pixels are fully correlated. The usual way of processing radio maps is to set C_N equal to the number of pixels in the FWHM synthesized beam. We have chosen a slightly larger area of pixels corresponding to the true correlation length. In this way, we again have that the number of independent pixels, as cal-

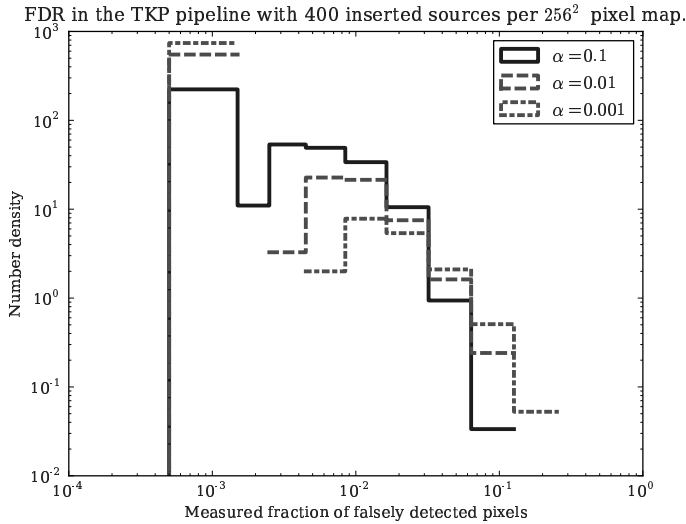


Figure 3.13: The distribution of the number of falsely detected pixels divided by the total number of detected pixels with 400 inserted sources per 256^2 pixel map. All three histograms are normalized equally. In order to represent, in this loglog plot, the bin with most counts, i.e., with zero falsely detected pixels, we have chosen to center the bins on the right bin edges. The left bin contains all fractions in the interval $[0, 1e-3]$. All other bins have equal size on a log scale.

culated from the bias of the sample variances, times this area equals the total number of pixels in the map.

There is a subtlety with respect to the definition of correctly and falsely detected source pixels. A correctly detected pixel can be associated with an inserted source. This means that the detected pixel must be above one of the source pixels. However, the inserted sources are Gaussians which, strictly speaking, extend out infinitely, although we have inserted them as 32^2 pixel arrays. It is unclear how "associated" is defined in the tests run by Hopkins et al. (2002). We have chosen a 5^2 pixel area centered on the source to distinguish between falsely detected, unassociated pixels and correctly detected, associated pixels. More concretely, the code used for our tests makes use of the `watershed_ift` algorithm from Scipy. `watershed_ift` uses a watershed from markers algorithm as described by Felkel et al. (2002). Thus the association is determined by 25 equal markers centered on every inserted source. This number is somewhat arbitrary but it takes care of the few rare cases where the peak pixel of a source is missed due to a negative noise peak while adjacent pixels were in fact detected.

It should be clear that `watershed_ift` also marks all the contiguous pixels as "correct" pixels if they were detected. SFIND 2.0, which was tested by Hopkins et al. (2002), also discards all pixels from fits that did not converge. We did not copy that in our tests.

We have done three test runs with a number of 100, 200 and 400 point sources inserted at random positions in each map. We again used our 9997 source free maps of 256^2 pixels for each

Table 3.4: Ensemble averaged fraction of falsely detected pixels

Sources inserted	$\alpha = 0.1$	$\alpha = 0.01$	$\alpha = 0.001$
100	0.011	0.0040	0.0019
200	0.0060	0.0023	0.0011
400	0.0028	0.0010	0.00047

run. The rms noise in these maps is 1.0 Jy. The fluxes of the inserted sources were randomly distributed between 0 and 5 Jy. The results are presented in figures 3.11, 3.12 and 3.13 and in table 3.4. The main conclusion that can be drawn is that a fraction of less than 0.1% false positives is generally not achieved. Only in the case where many (400) sources are inserted are the FDR promises actually kept. Hopkins et al. (2002) did not run tests for $\alpha = 0.001$, so this makes it hard to compare. If pixels that do not give converging fits are left out, as these authors did, the fraction of falsely detected pixels will likely drop. We tried to derive the cause of the discrepancy by also running tests on source free maps. For $\alpha = 0.1, 0.01$ and 0.001 we found that the fraction of maps with any pixel above the FDR threshold was 2%, 0.7% and 0.17%, respectively. The latter fraction decreased by only 0.02% when we changed the correlation length to include all pixels in the maps. This makes it unlikely that the correlation length was chosen too small. Thus it remains unclear why the FDR algorithm gives poor results when the allowed FDR is set very low ³.

3.7 Deblending

The performance of deblending algorithms naturally depends on the separation between sources, on the fluxes of the blended sources with respect to the detection threshold, but also on the ratio of the fluxes of two neighbouring sources if the algorithm requires saddle points. Saddle points emerge at some minimum separation. We are interested in the limiting case where the minimum between two peaks is not yet apparent. Instead, we have a flat plateau near the weakest source which can be found by setting the first and second derivatives of the blended source to zero:

$$\Sigma = Ce^{-\frac{x^2}{2\sigma^2}} + Be^{-\frac{(x-D)^2}{2\sigma^2}} \quad (3.10)$$

$$\frac{d\Sigma}{dx} = 0 \quad (3.11)$$

$$\frac{d^2\Sigma}{dx^2} = 0 \quad (3.12)$$

We have assumed that the blended source is composed of two point sources with peak values C and B and separation D . In the case of point sources the sources they have the same

³See, however, the remark in paragraph 3.8

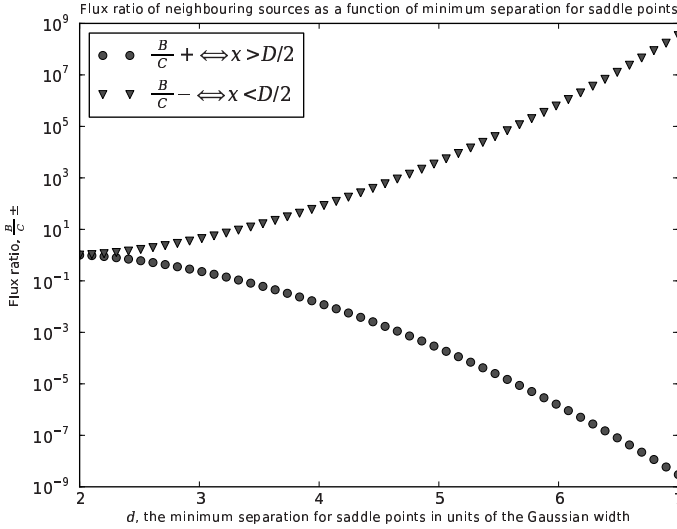


Figure 3.14: The sum of two Gaussians (with equal width) show saddle points at some minimum separation which is a weak function of the flux ratio of the sources.

width σ as the psf.

When we work out these equations, we find

$$x_{\pm} = \frac{D \pm \sqrt{D^2 - 4\sigma^2}}{2} \tag{3.13}$$

$$\frac{B}{C} \pm = \frac{D \pm \sqrt{D^2 - 4\sigma^2}}{D \mp \sqrt{D^2 - 4\sigma^2}} e^{\mp \frac{D\sqrt{D^2 - 4\sigma^2}}{2\sigma^2}} = \frac{d \pm \sqrt{d^2 - 4}}{d \mp \sqrt{d^2 - 4}} e^{\mp \frac{d\sqrt{d^2 - 4}}{2}} \tag{3.14}$$

With $d \equiv D/\sigma$. The functional dependence between d and $B/C \pm$ has been depicted in figure 3.14. It is clear that the minimum separation depends weakly on the flux ratio. A separation of ≈ 6 Gaussian widths is required for a flux ratio of $10^6 : 1$. Higher flux ratios should not be expected, because this ratio exceeds the highest dynamic range ever achieved in radio astronomy ($5 \cdot 10^5 : 1$ is the highest according to Walker (2006)).

Validation of the deblending algorithm was run on 9800 of our 256^2 pixel maps, with 1 Jy beam^{-1} rms noise and 64 pairs of sources per map. The pairs were inserted on a uniform grid to avoid any overlap between pairs. We chose a 10σ detection threshold and a 3σ analysis threshold. The flux of the weakest source was always 20 Jy and the flux ratios were logarithmically spaced between 1 and 1000. For every flux ratio 20 maps were processed successively, each map having a fixed separation between the two sources in each of the 64 pairs. The separations corresponding to each of the 20 maps were linearly spaced between 2 and 7 Gaussian widths. Since the psf is elliptical, we have chosen to align the sources in each

Table 3.5: Performance of TKP deblending algorithm for different values of DEBLEND_MINCONT

	0.1	0.05	0.01	0.005	0.001
% extra sources deblended	6.9	14.8	31.6	37.5	43.7
% of incorrect deblendings per source	$6.9 \cdot 10^{-3}$	$2.1 \cdot 10^{-2}$	0.88	1.6	5.6
figure of merit	6.9	14.7	28.5	31.9	23.7

pair along their minor axes and use the Gaussian width along the minor axis of the ellipse, 1.987 pixels, as a unit for separation.

The high detection threshold ensured that no detection could be a noise peak, which simplified the validation of the deblending algorithm. The user sets the deblending parameters DEBLEND_NTHRESH and DEBLEND_MINCONT as in SExtractor (Bertin 2006, paragraph 6.4) runs. We set DEBLEND_NTHRESH = 32, the default value in SExtractor. This determines the number of subthresholds between the smallest and largest pixel value of the composed object, logarithmically spaced. We ran tests for five different values of DEBLEND_MINCONT. This parameter ensures that only subislands with significant flux will be regarded as separate sources. It works like this: connectivity is determined at subsequent subthreshold levels, starting at the lowest level. When more than one subisland is found, the fluxes of the new-found "branches" above the subthreshold are divided by the flux of the original blended source. At least two ratios have to be above DEBLEND_MINCONT before the corresponding subislands will be considered separate. Also, the peak values of those subislands have to be above the detection threshold. Pixels belonging to subislands that do not satisfy both of those conditions are neglected. The results for DEBLEND_MINCONT = 0.1, 0.05, 0.01, 0.005, 0.001 are shown in table 3.5.

The "% extra sources deblended" is the % of extra sources that are found relative to plain source extraction. Plain source extraction can, of course, identify the different components of a blended object if they are not connected at the level of the analysis threshold. Thus, plain source extraction already does some deblending. It should be noted that these and other numbers in table 3.5 depend on the specific range of flux ratios and separations of the inserted pairs of sources. The "% of incorrect deblendings per blended source" is the rate at which blended sources are decomposed in more than two components, which is, of course, incorrect in these specific test runs. The "figure of merit" applies a penalty for those incorrect deblendings, such that one incorrect deblending crosses off one correct deblending. If the figure of merit is defined in this way, we find that DEBLEND_MINCONT = 0.005 is best. This is also the optimum value found from SExtractor test runs (Bertin 2006, paragraph 6.4).

It should be noted that DEBLEND_MINCONT sets a lower limit on the flux ratio of the faintest source to the brightest source in a composed object that can be deblended. That lower limit is somewhat larger than DEBLEND_MINCONT because the significance criterion of the deblending algorithm compares the flux of a subisland **above the subthreshold** with the total flux of the composed object.

Without a deblending algorithm, the numbers of detected sources are depicted in figure 3.15. For DEBLEND_MINCONT = 0.05 and 0.005 we have depicted the numbers of detected sources

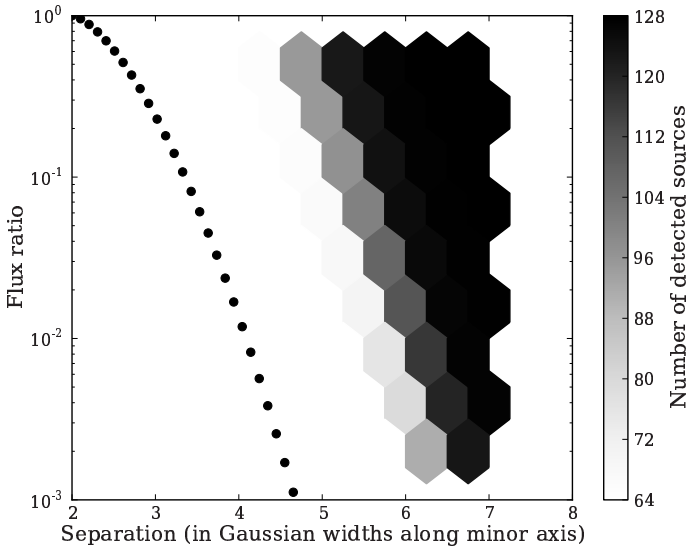


Figure 3.15: Distribution of detected sources without a deblending algorithm. The dots indicate the minimum separation required for saddle points (equation 3.14).

divided by these numbers without a deblending algorithm. It is evident from figure 3.16 that if the flux ratios are very small, deblending is done very poorly for `DEBLEND_MINCONT = 0.05`. In figure 3.17 we see more complete deblending for these small flux ratios. It should be kept in mind, however, that this comes at the price of a larger number of erroneously deblended objects, with more than two components, as quantified in table 3.5.

3.8 Determination of peak flux densities and positions

For the validation of the correct determination of fluxes and positions, we used 9980 of our 256^2 pixel maps, with 1 Jy beam^{-1} rms noise. All 64 sources in a single map were inserted with equal flux on a regular grid. The values `BACK_SIZEX` and `BACK_SIZEY` that determine the size of the background window for mean background and background noise interpolation across the map, were set to 256, which results in a flat background across the map. Both the analysis threshold and the detection threshold were set to 6 times the rms noise, to avoid picking up noise peaks which would complicate the validation.

It is interesting to do some checks on the number of pixels above a certain level. We have processed a total of $9980 \cdot 256^2 = 6.54 \cdot 10^8$ pixels in our validation runs. If all of these pixel were independent, then we could expect less than one pixel, 0.645 to be precise, above 6 times the rms noise, i.e., 6 Jy. We found that, in fact, there were a total of 147 pixel values from 90 maps above 6 Jy, with the highest pixel value between 8 and 9 Jy! Of course, the

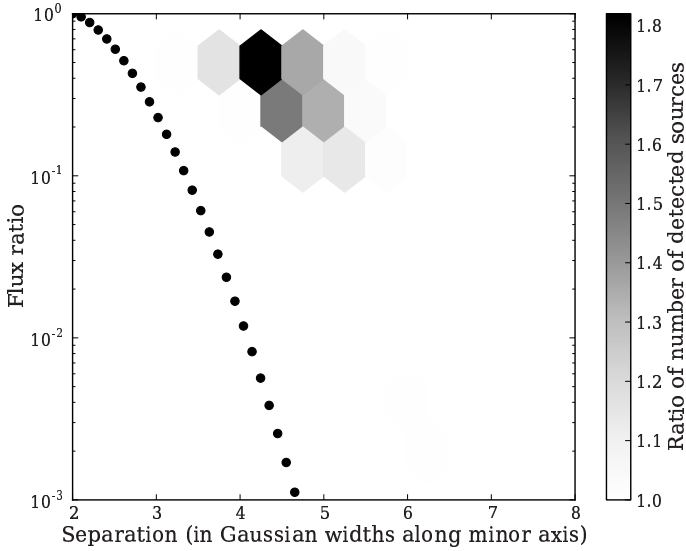


Figure 3.16: Distribution of detected sources when the deblending algorithm is run with $\text{DEBLEND_MINCONT} = 0.05$ divided by detected sources without a deblending algorithm. The dots indicate the minimum separation required for saddle points (equation 3.14).

pixel values are correlated in each map, but even so, for any number of uncorrelated pixels per map the ratio of the actual to the expected number of very high pixel values remains equal to $147/0.645 = 228$. This shows that, in fact, there are strong deviations from normal distributions in the pixel values, although the visibility noise was strictly Gaussian. As we know, the Fourier Transform (FT) of a Gaussian is also a Gaussian, so if the pixel distribution deviates from a normal distribution this must arise from the fact that we have applied a Fast Fourier Transform (FFT) with interpolation and limited sampling instead of applying a true FT to an infinitely extending and complete (u,v,w) space. In any case, this is something that needs to be accounted for when searching for transient sources. It involves a deeper understanding of the noise properties of maps, which is beyond the scope of this thesis. However, we may now have found an explanation for some unsatisfactory results in the FDR validation runs. When the rate of false positives was set very low ($\alpha = 0.001$), the number of falsely detected pixels was slightly higher than allowed, also in the case of source free maps. It should be kept in mind that the FDR algorithm **assumes** that the background noise has a normal distribution. If this is not actually the case, FDR cannot be expected to work correctly.

For the purpose of testing the accuracy of the measurements by the TKP software we have chosen the source fluxes to be linearly spaced between 10 and 1000 Jy. The lower limit of 10 Jy, 4σ above the detection threshold, was chosen to avoid any considerable truncation bias in the flux measurements. Also, the error bars that we use for the fitted parameters are given by the Condon formulae (Condon 1997). They are valid only at high signal-to-noise. The

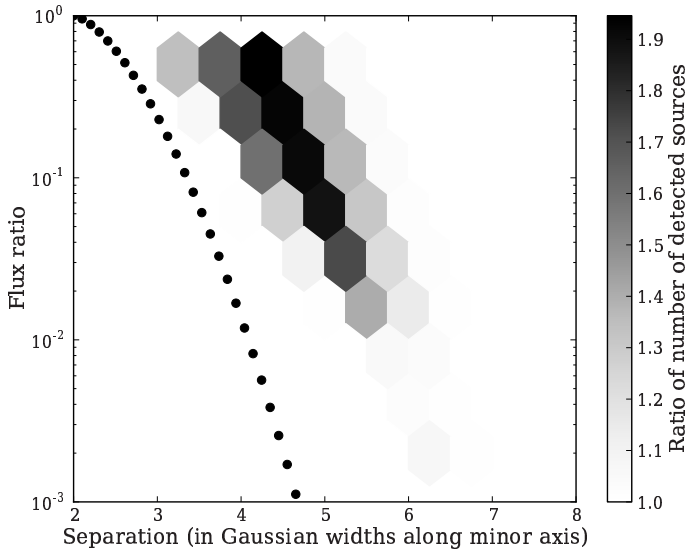


Figure 3.17: Distribution of detected sources when the deblending algorithm is run with $\text{DEBLEND_MINCONT} = 0.005$ divided by detected sources without a deblending algorithm. The dots indicate the minimum separation required for saddle points (equation 3.14).

upper limit of 1000 Jy was chosen to cover a wide range of signal-to-noise ratios. We chose a flat flux distribution for the inserted sources to give equal weight to all signal-to-noise ratios in our statistics. In 10 of the 90 maps with noise peaks above 6σ , the number of detected sources exceeded 64. Those maps were not used for our validation statistics. In the other 80 maps, the slight overestimate of the mean background and noise caused by the 64 inserted sources, filtered out those noise peaks at the 6σ detection threshold.

Figure 3.18 shows a histogram of the peak flux density measurements in the remaining 9970 maps, where the differences between the calculated peaks and the true peaks have been divided by their reported errors. It is immediately obvious that the distribution from our measurements is narrower than it should be. Most likely, the reported errors are overestimated. If the reported errors were 14% larger overall, the measured distribution would match the theoretical curve. It may be that the errors are smaller than expected since we have inserted sources in the image plane, which is somewhat artificial. More realistic would be to insert sources in the visibilities and then clean the images. This has the large disadvantage that the cleaning process is complicated and can introduce unknown biases, e.g., because it is unclear at what level cleaning should be stopped. Thus, this approach hinders a straightforward validation.

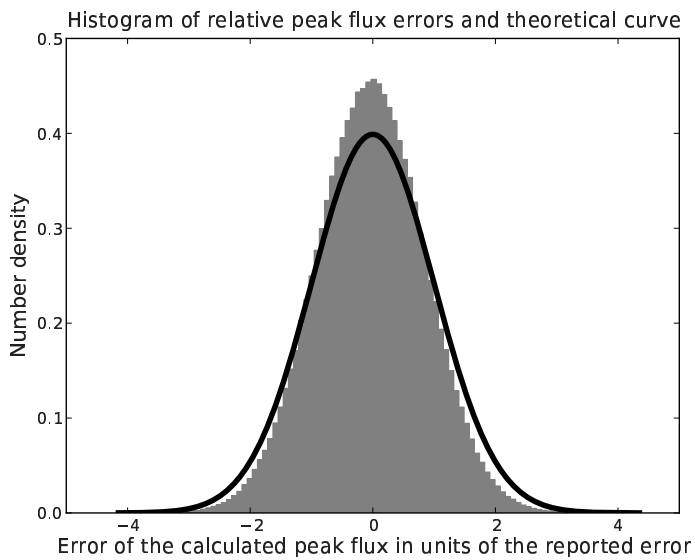


Figure 3.18: Relative peak flux errors of 638080 sources in 9970 maps by TKP pipeline software as a normalized histogram and the theoretical curve, i.e., a standard normal distribution.

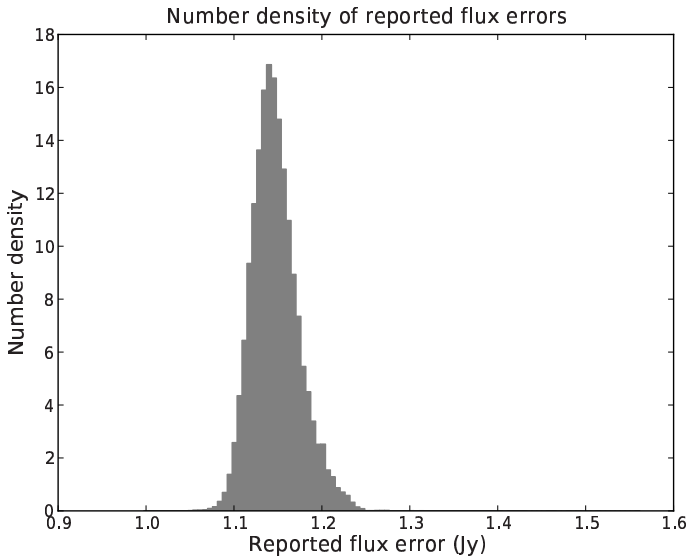


Figure 3.19: Distribution of reported peak flux errors from 638080 source measurements in 9970 maps by TKP pipeline software as a normalized histogram.

Figure 3.19 shows the number density of the reported flux errors. They are very close to the rms noise (1 Jy beam^{-1}), as expected from theory (Condon 1997).

For the validation of position measurements, we had to add some lines of code to the flux measurement validation script. The positions of the inserted sources were converted to celestial coordinates by new code, i.e. code that was independent from the conversion code in the TKP pipeline software. Also, since the order of the output source measurements was unknown and each measurement had to be coupled to the corresponding inserted source, code for correct reordering had to be applied. This, of course, would have been unnecessary if only one source were inserted per map.

The results for the right ascension are depicted in figures 3.20 and 3.21. From figure 3.20 we, again, have to conclude that the reported Right Ascension errors are overestimated, by about 23%, since the distribution is narrower than a normalized Gaussian. The reported position variances, as implemented in the TKP software, are copied from the NVSS (see equation 25 in Condon et al. 1998). They are twice the theoretical values (Condon 1997), taking into account the very dirty synthesized beam of a VLA snapshot. Most likely, this is the cause of the overestimate, since the (u,v) coverage used to produce the artificial noise maps for these validation runs, corresponds to about 6h of VLA synthesis and not to a snapshot. The theoretical values, on the other hand, would be too small. A value in between 1 and $\sqrt{2}$ times the theoretical position standard deviation would be most appropriate.

The declination errors are depicted in figures 3.22 and 3.23. From these graphs, similar

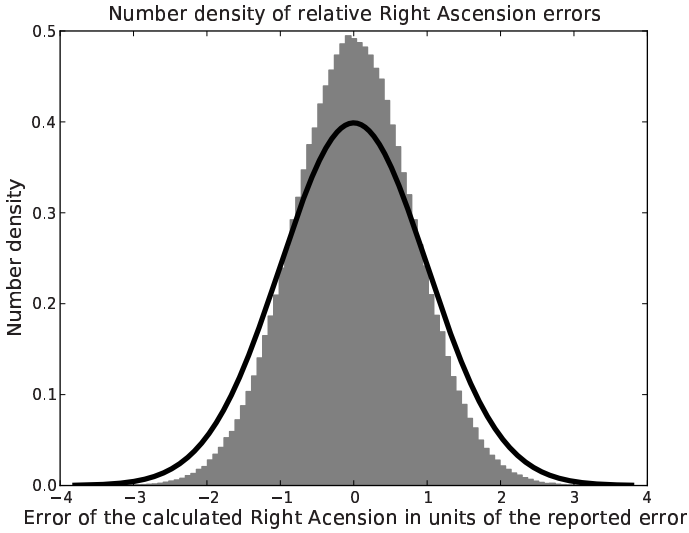


Figure 3.20: Relative Right Ascension errors of 638080 sources in 9970 maps by TKP pipeline software as a normalized histogram and the theoretical curve, i.e., a standard normal distribution.

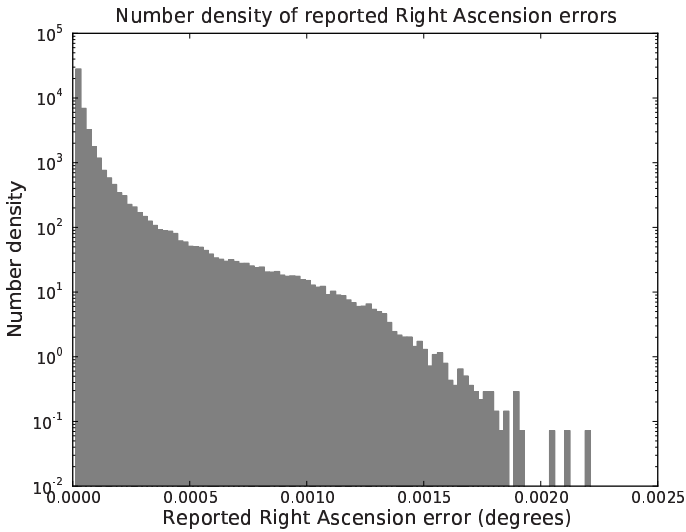


Figure 3.21: Reported Right Ascension errors from 638080 source measurements in 9970 maps by TKP pipeline software as a normalized histogram.

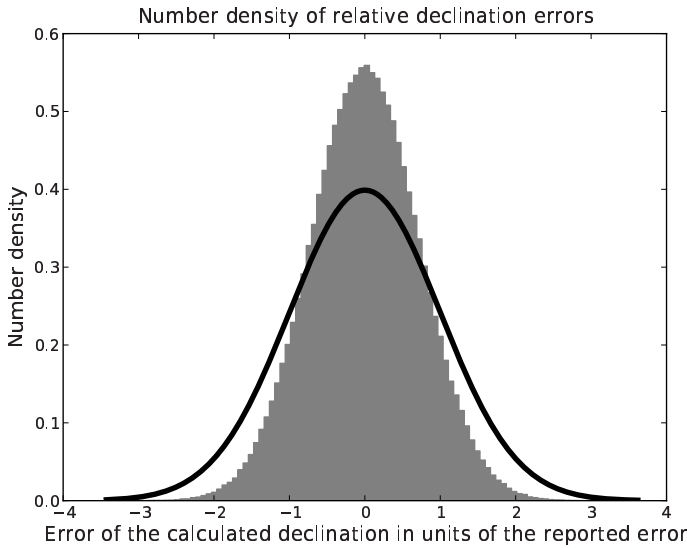


Figure 3.22: Relative declination errors of 638080 sources in 9970 maps by TKP pipeline software as a normalized histogram and the theoretical curve, i.e., a standard normal distribution.

conclusions can be drawn as from the Right Ascension statistics. The reported declination errors are 40% too large. In other words, if the reported declination errors were decreased by 40% overall, the observed distribution in figure 3.22 would match the theoretical curve.

Table 3.6 lists the outliers and biases of both the flux and position measurements. We have processed 638080 sources, so we expect our highest outlier at about the 4.7σ level. This is not what we see for the positions, indicating, again, that the position errors have been overestimated. The flux outliers are also smaller than expected, but not by so much.

The positions are measured without any considerable bias. The fluxes, however, are systematically low. This is due to the overestimate of the background level from κ, σ clipping. Table 3.3 shows an overestimate in the mean background of 0.073 Jy when 100 sources are inserted. From this table, one would expect an overestimate in the mean background of about 0.046 Jy when 64 sources are inserted. Consequently, when the error in the background mean has been corrected, one expects an overall bias in the fluxes of only +0.013 instead of -0.033. Of course, a more exact approach would involve redoing the calculations of paragraph 3.5.3 for exactly the set of sources we have used here.

3.9 Conclusions

We have processed large numbers of maps with and without sources. The TKP source extraction and source measurement software met the requirements with regard to robustness: we

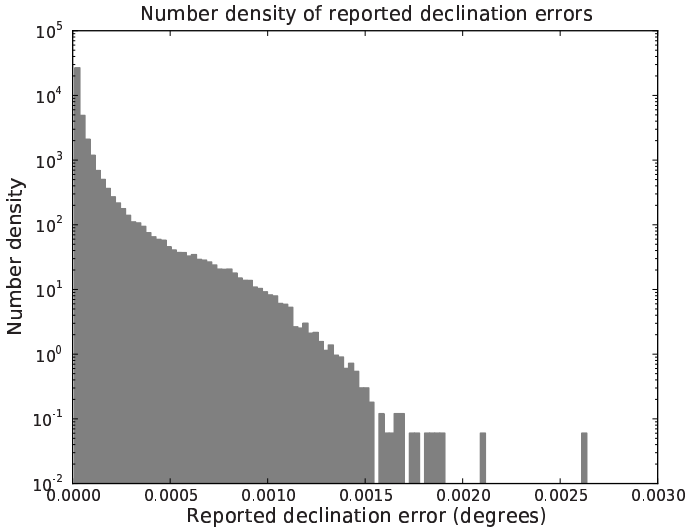


Figure 3.23: Reported declination errors from 638080 source measurements in 9970 maps by TKP pipeline software as a normalized histogram.

Table 3.6: Outliers and biases in TKP flux and position measurements

	highest negative outlier	highest positive outlier	overall bias	significance of overall bias (σ)
(measured - true peak flux)/reported error	4.3	-4.2	-0.033	26
(measured - true R.A.)/reported error	3.8	-3.8	0.0022	1.8
(measured - true declination)/reported error	3.6	-3.4	0.00084	0.67

did not encounter any halts. Speed requirements were met: these 256*256 pixel maps were processed in less than 1s per map. Errors in the determination of background and noise levels increased with the fraction of source pixels. Possibly, the application of an asymmetric clipping algorithm could improve the performance, especially towards confusion limited maps. The deblending of sources was optimum for DEBLEND_MINCONT = 0.005, which resulted in 37.5% extra sources deblended and an error rate of 1.6% per source. The FDR algorithm performed satisfactory down to a fraction of less than 1% false positives. The discrepancy for $\alpha = 0.001$ can possibly be explained by non-Gaussianity in the distribution of the background pixels. Source positions were measured accurately. Source fluxes were biased low. This reflects the overestimate of the background level which increases with the fraction of source pixels. Again, performance could possibly be improved by a more advanced clipping

algorithm.

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.
- J.J. Condon, *Confusion and flux-density error distributions*, *ApJ* **188**, 279 (1974).
- J.J. Condon, *Errors in Elliptical Gaussian Fits*, *PASP* **109**, 166 (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).
- P. Felkel, M. Bruckwschwaiger & R. Wegenkittl, *Implementation and complexity of the watershed-from-markers algorithm computed as a minimal cost forest*, in *The EURO-GRAPHICS conference*, Manchester, UK, 2001.
- D.W. Hogg, *Confusion errors in astrometry and counterpart association*, *AJ* **121**, 1207 (2001).
- 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 catalogue*, *AJ* **125**, 465–477, 2003.
- S.D. Hyman, T.J.W. Lazio, N.E. Kassim, P.S. Ray, C.B. Markwardt, & F. Yusef-Zadeh, *Nature* **434**, 50, 2005.
- 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.

- F. Patat, *A robust algorithm for sky background computation in CCD images*, *A&A* **401**, 797-807, 2003.
- K.U. Ratnatunga, & E.B. Newell, *The reduction of panoramic photometry. III - an asymmetric clipping algorithm*, *AJ* **89**, 176, 1984.
- 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.
- P. Teerikorpi, *Influence of a generalized Eddington bias to galaxy counts*, *A&A* **424**, 73-78, 2004.
- C. Walker, *High Dynamic Range Imaging*, presentation at the *Tenth Summer Synthesis Imaging Workshop*, Albuquerque, New Mexico, 2006.