A multi-scale approach to automated feature extraction from HI data cubes

P. Kemper[1]
supervision by J.M. vd Hulst[1] and T. Oosterloo[2]

[1] Kapteyn Astronomical Institute, P.O. Box 800, 9700 AV Groningen, the Netherlands
[2] ASTRON, P.O. Box 2, 7990 AA Dwingeloo, the Netherlands

Abstract. Due to the increasing sizes of datasets (three-dimensional spectral datasets from a radio interferometer) produced by modern astronomical observatories, feature extraction in a highly automated fashion becomes more and more a necessity. Therefore a multi-scale approach to automated feature extraction based on the wavelet transform and morphological operators, has been implemented. The discrete wavelet transform is performed by the *à trous* algorithm. A multi-scale approach allows for object extraction independent of size and shape (adaptive filtering). The model has been tested successfully on astronomical datasets, in both the spatial as well as in the visibility domain.

1. Introduction

Astronomical signal is always affected by noise. This noise can be either of two things. The first is the 'background noise' that arises in the receiving telescope itself, due to noise from objects in the field of view, and due to noise contributions from the atmosphere. The second type of noise is in fact spurious signal. It is very strong and sometimes also very extended which makes it easy for a computer to be mistaken as true astronomical signal. To extract the true astronomical signal from measurements of the sky and to remove the spurious signal, some information about a radio synthesis telescope is required first.

1.1. Visibility data

A radio synthesis telescope is a system consisting of multiple receivers (telescopes), which are interconnected. Such a system is called an interferometer, which makes spectroscopic observations of a small part of the sky. The signals of the receivers are correlated to form Fourier transformed images of the sky. So the combined interferometer measurements construct an image as if it was measured by a large single aperture antenna. These data are called the *visibility data.* They are a sequence of quasi-monochromatic images, usually referred to as channels. They are sampled as a function of element separation b (also called baselines), time t and frequency ν. These data are in the *visibility domain* (also called the *measurement* domain). The earth rotation changes the orientation of each baseline (pair of telescopes) with time. So each b,t pair can be translated into a unique position in the aperture plane (the Fourier transform of the image plane). This position has coordinates u,v which is why visibility data is sometimes referred to as u,v data, as it will be referred to throughout this paper. An example of visibility data with Radio Frequency Interference (RFI) can be seen in figure 1. As is the case here, interference is often a narrow band signal of short duration. Therefore its signature is that it is narrow in ν,t space (right panel) and broad in b space. In figure 2 interference in the *spatial domain* (also called the *image* domain) is visible. It is the Fourier transform of the visibility data of figure 1, after the latter have been regrided (i.e. transformed in u, v coordinates). The interference is clearly visible as strong stripes, which means that the interference was there only for a short time (the pattern in the image can be thought of as the convolution of the signal from the sky, with the PSF sampled over only that part of the u,v plane samples during the time the interference was present).

1.2. Image data

By Fourier transforming the *visibility data*, one recreates the brightness distribution of the same small part of the sky consisting of channels each at a slightly different wavelength. All of these images together represent the same part of the sky, looked at at different wavelengths. The reason why different wavelengths are important is the following.

The signal to be detected in the data cube corresponds to the signal from celestial objects. If one is looking for interstellar hydrogen gas in e.g. galaxies, then the 21-cm line emission from neutral atomic hydrogen (HI) must be detected. This can be done by visually inspecting the data at various spatial resolutions, but with the large amounts of

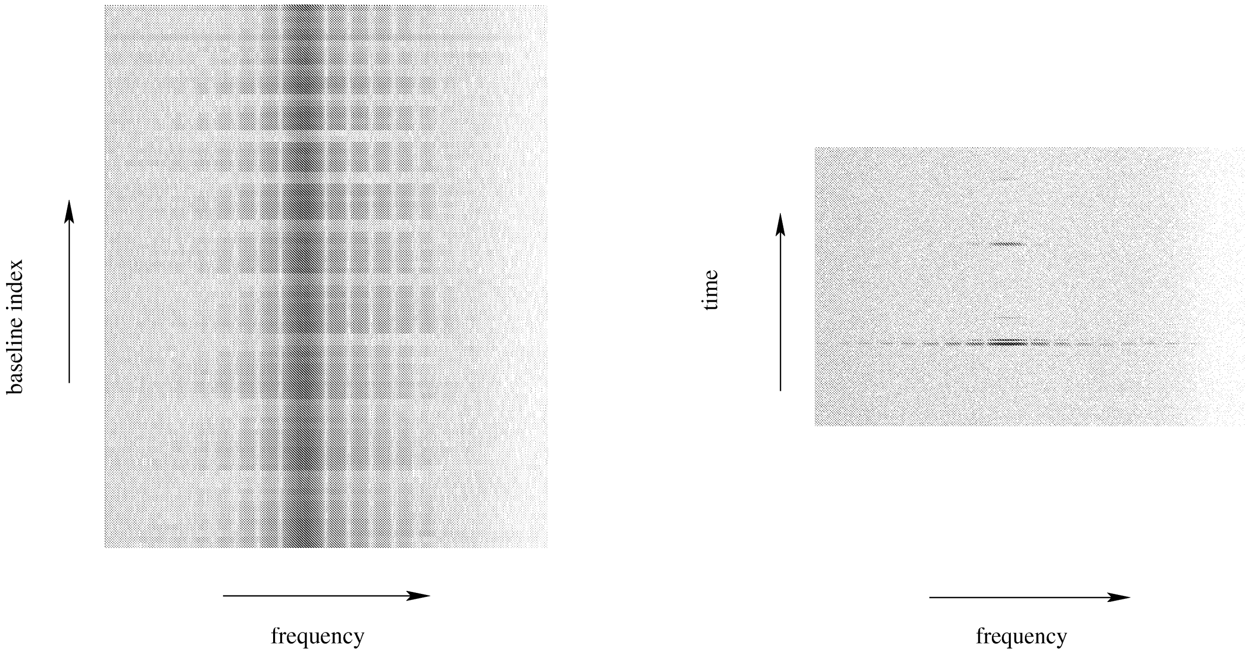

Fig. 1. Gray scale image of u,v data (visibility data) in the ν,b plane (left panel) and in the ν,t plane (right panel). Strong interference from a GPS satellite is visible.

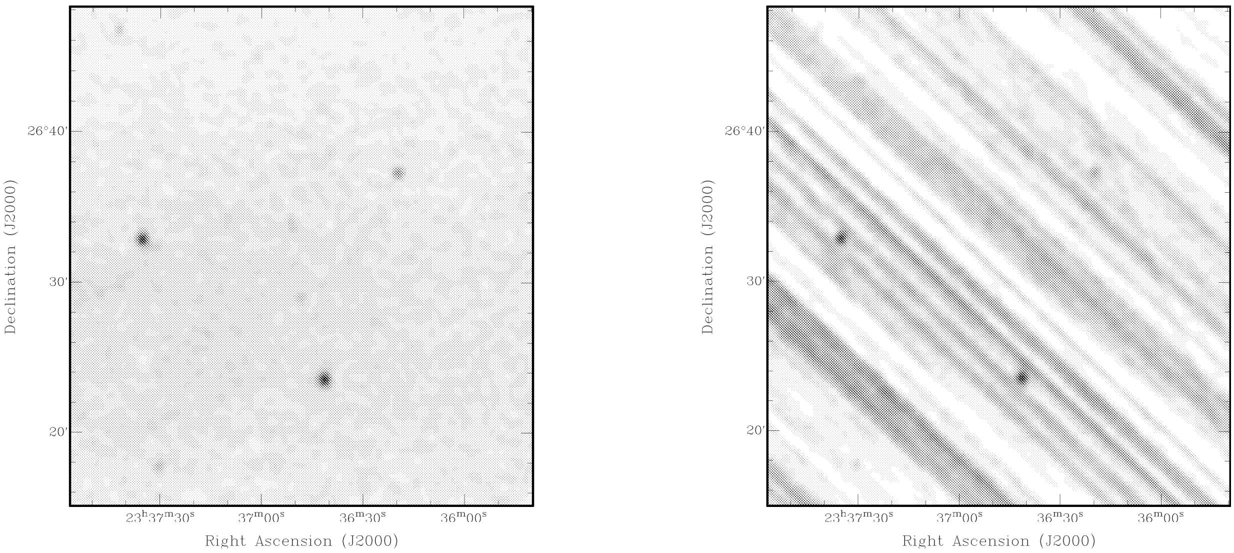

Fig. 2. Image obtained by Fourier transforming the data shown in figure 1. The left panel shows an interference free frequency channel, the right panel shows a channel that was affected by interference.

data created by modern telescopes this is no longer possible. Fortunately, galaxies have the characteristic that they rotate (else they would collapse). Because of the Doppler effect (a change in frequency of emitted waves produced by motion of an emitting source relative to an observer), a different wavelength (or frequency) corresponds to a different velocity. This way the rotation of the object shifts the hydrogen emission in frequency. This way a set of images is created. So this three-dimensional data set has two spatial axes (right ascension and declination) and one velocity axis. Such a data set is called a *data cube*.

So, in terms of morphology of real data, one is looking for small coherent structures, sheared a bit in the frequency (velocity) coordinate. This can be used to detect the real signal.

1.3. The problem

Now the question arises, how to separate true HI signal from system and background noise and spurious signal in a data cube, in an automated way. Several methods have been used in the past, some of them are still being used today (Perley et al., 1986). These methods apply to image data.

– Cutoff method (Rogstad & Shostak, 1971) (also referred to as hard thresholding). To exclude data points

with no HI signal, a cutoff in intensity is applied. Usually this cutoff is set to 2 or 3 times the standard deviation of the background noise in the data.
This method has one serious disadvantage. When determining the integrated properties of the emission at each data point in a rotating object (the zeroth, first, and second moments of the velocity profile, i.e. the integrated emission, its velocity and the velocity dispersion), the calculated moments are subject to systematic effects depending on the cutoff value used (the moments are biased).

- Window method (Bosma, 1978). One can also set a cutoff in velocity, to overcome the biases introduced by the 'cutoff' method. This way the influence of noise peaks outside the HI emission range is eliminated. The size of the window (the acceptance gate) is increased until the mean intensity of the background data points converges.
- Interactive study. This means that each channel is studies carefully on a graphics display.
  This method works very well, but it is far too time consuming for large databases. The size of a data cube can vary from 128 × 128 × 32 elements (pixels) to really large sizes of 1024 × 1024 × 1024 pixels or even larger. With the need for better and bigger telescopes (like the LOFAR telescope (LOw Frequency ARray) for instance where 25,000 antennas produce a datastream of 2 Gbit/sec each), the amounts of data each telescope produces increases accordingly. So the need for automated feature extraction also increases accordingly.
- Fit the data. The data is fitted to a preconceived shape. This method locates in each channel the highest point and then fits a certain profile (i.e. Gaussian or Voight profile) to the data around those points (van der Kruit & Shostak, 1982).
  The problem with this method is that the assumed functional form can be incorrect. If the profile one is looking for is very extended it becomes very hard (if not impossible) to choose the right shape to fit.
- Hybrid method. First smoothing the data to a lower resolution, then using the 'cutoff' or 'window' method gives good results.
  The problem with this method is that you have to know a priori the scale of the signal you are looking for. Since this is unknown, different smoothing functions must be tested to bring out the HI in different parts of the galaxy.

The problems that arise by these conventional approaches can be solved by *multi-resolution filtering.* This is an extension of the 'hybrid' method. We don't know a priori the scale of the signal we are looking for. Unspecified astronomical patterns can occur at any scale, and in all directions. Multi-resolution operators make it possible to smooth the data to different resolutions and then apply filtering methods to each of these smoothed cubes (or to the differences between each next level of smoothed cubes).

In this paper a multi-scale approach to automated feature extraction for both image data as well as for visibility data is presented. It describes a sequence of operations required for automated image analysis, and for automated spurious signal (RFI) removal in visibility data. It incorporates adaptive filtering techniques applied to radio data cubes in both the spatial as well as the visibility domain, based on the *à trous* wavelet transform and morphological filters like openings and closings.

1.4. Goal of this project

The goal of this project is to extract true astronomical features from image data cubes and to extract spurious signal from visibility data in a highly automated fashion. This is done by setting the mask (or 'flag table') of a data cube. This can be an arbitrary data cube with different sorts of objects of various sizes and scales. Different kinds of algorithms must be applied to select the significant signal in the data cube. Significant means that the value of the pixels in the data cube at certain positions is above a given detection limit. If a certain pixel is significant or not, the mask of the data cube must be set accordingly. This mask is a (binary) data cube itself, with the same dimensions as the first. For image data, all the significant pixels in the mask must be set to 1, so that only those remain visible. For u, v data it is the other way around, since in this case one wants to filter the spurious signal (RFI). The mask of the u, v data is usually referred to as the flag table and the masking itself is called 'flagging'.

Manual feature extraction is still a very good, if not the best method but it is by far too time consuming nowadays. So if a completely automated feature extraction method can be created that can even come close to that then a large step forward is made.

1.5. Approach

There are many different adaptive filters. The particular filter used here is based on the *à trous* algorithm. It works basically like this. First the data is smoothed with a certain small smoothing kernel. Then the difference is taken between data cubes of each two successive smoothing levels. These differences are called *wavelet cubes.* The inverse transform is very straightforward. One simply adds all the different wavelet cubes to the final smoothed data cube to obtain the reconstructed data cube. Before this reconstruction is done however, the adaptive filter is applied to the wavelet cubes. It includes estimating the background noise level for each wavelet cube and retain only those data that are above a certain clip level. The other data can be removed from the cube (effectively altering the original data by using only certain wavelet coefficients) or a mask can be set (thus retaining the original data).

Determining the right clipping level is not as straightforward as is seems. Usually it is set to a multiple of the standard deviation (σ) of the background noise, like 3σ or

5σ. However, as described above, the calculated moments of the data are subject to systematic effects depending on the cutoff value used. One does not know a priori the cutoff value to choose. Therefore the more adaptive *False Discovery Rate* (FDR) is implemented.

Second of all, estimating the background noise level is a problem. To do so, one must calculate the standard deviation over all pixels that are part of the background noise. But it is not known which pixels are noise, this is exactly what a clipping procedure tries to find. So it is very important to have a good model of the background noise in the data cubes. For image data, the characteristics of the noise are known only to first order. The basic contributor is the thermal noise of the radio receivers. To first approximation, this is white, Gaussian noise and similar for each telescope combination. There are additional sources of noise however, such as instrumental defects and characteristics of the distribution of emission in the sky. Therefore it is very important to characterize the noise properties for each observation independently. To test if indeed the distribution function of the background noise in image data has a Gaussian shape the Kolmogorov-Smirnov test has been applied.

2. The Kolmogorov-Smirnov test

To test if indeed the background noise in a data cube is Gaussian, the Kolmogorov-Smirnov one-sample test for goodness of fit is used (i.e. Siegel (1956)). The test is concerned with the degree of agreement between the sample distribution of a set of sample values (observed scores, in the case of a data cube these are the values of all of the pixels) and some specified theoretical distribution. The test determines if the scores from the sample can reasonably be thought to have come from a population having the theoretical distribution. Note that the theoretical distribution should not be estimated from the sample distribution, which is done here (if the hypothesized distribution is estimated from the data, then the KS-test is not accurate). If the KS-test is done after the masking is applied however, then the result should be much more reliable.

The KS-test is implemented as follows. For each map (channel) in the data cube a histogram is created that shows the counts of each pixel value in a range from the minimum to the maximum pixel value. To this histogram (which is expected to be Gaussian) a Gaussian curve is fitted (parameters: maximum counts, position of maximum, standard deviation and offset).

Next, both the data from the sample data as well as from the theoretical data (the fitted Gaussian curve) is added cumulatively to create two cumulative distributions (S(x) and F(x) respectively). These distributions are scaled to within the range of zero to one. The Kolmogorov-Smirnov test focuses on the largest of the deviations **D** (the maximum absolute difference) between the normalized cumulative histogram S(x) and the uniform cumulative distribution function F(x). So the statistic concerned is

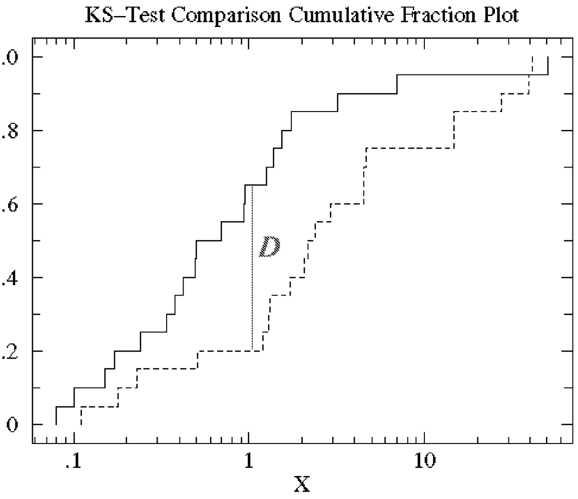

Fig. 3. Graphical representation of the Kolmogorov-Smirnov statistic D (the maximum absolute difference between the normalized cumulative histogram and the uniform cumulative distribution function).

$$D = \text{maximum}|F(x) - S(x)|. \qquad (1)$$

The critical values of D for a sampling distribution are tabled. For large samples the values of D can be found in table 1. A graphical representation of the KS-test can be seen in figure 3.

Level of significance for D = maximum $|F(x) - S(x)|$

0.20	0.15	0.10	0.05	0.01
$\frac{1.07}{\sqrt{N}}$	$\frac{1.14}{\sqrt{N}}$	$\frac{1.22}{\sqrt{N}}$	$\frac{1.36}{\sqrt{N}}$	$\frac{1.63}{\sqrt{N}}$

Table 1. Critical values of D in the Kolmogorov-Smirnov one sample test for sample size N larger than 35. Adapted from Massey (1951).

Sample Data Cube (size: 256)

significance	0.20	0.15	0.10	0.05	0.01
'D'	0.067	0.071	0.076	0.085	0.102
Maximum 'D'	0.003				

Table 2. Results of the Kolmogorov-Smirnov test applied to a sample data cube (256 × 256 × 256). the maximum value of D is 0.003, so the probability of occurrence of the furthest outliers is higher than 20 percent so the assumption of a Gaussian distribution of the background noise is justified.

In table 2 the Kolmogorov-Smirnov has been applied to a sample data cube of*ugc 5253* of dimensions 256 × 256 × 256 (of which a channel can be seen in figure 5). For all the channels in this sample, the maximum value of D does not become larger than 0.003, which means (as can be seen from table 1) that the probability of occurrence of the furthest outliers is higher than 20 percent. In such a case it is evident that the assumption of a Gaussian distribution of the background noise is justified. The statistic D for this same sample data cube is plotted in figure 4. It shows that the statistic D stays close to zero until approximately bin number 100. There it goes up and down, i.e. becomes positive *and* negative, for about 50 bins and then it stabilizes again. The fact that the statistic D is positive and negative throughout each channel proves that the sample values deviate randomly from the theoretical Gaussian distribution. The plot also shows that the value of D becomes largest at channel 55, which is exactly the channel where most of the HI signal is in the sample data cube.

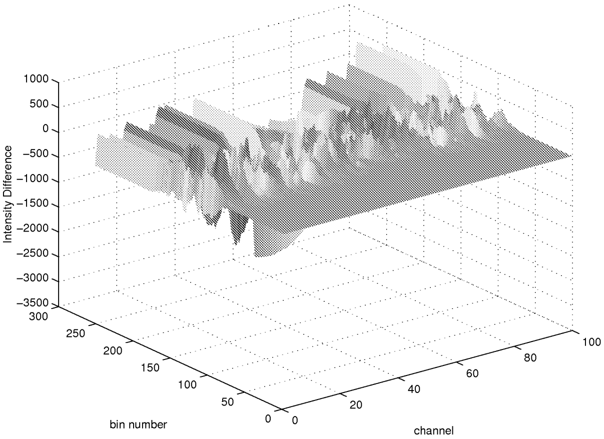

Fig. 4. Values of the Kolmogorov-Smirnov statistic D for a sample data cube. The histograms created from the sample data (for each map) contain 265 bins.

3. The wavelet transform

3.1. The à trous wavelet transform

There are several ways of, in some way, enhancing an image. Popular tools to use are the Fourier transform, as well as the wavelet transform. These are versatile tools, which can be used for image compression, object detection, large scale structure analysis, and many other tasks. When the wavelet transform is used for image restoration, it can be used for multi-resolution analysis, smoothing, and background noise removal. Decomposing an image into multiple resolutions provides a way of treating objects of different sizes separately.

The wavelet transform used here is the efficient *à trous* ("with holes") one (Holschneider & Tchamitchian, 1990). The more widely used Mallat transform was not employed, since it prioritizes non-isotropic structures in the data (Starck et al., 1995). This can be used to identify RFI in u, v data. The same can be done with the *à trous* algorithm however, by using an uneven amount of convolutions in each direction.

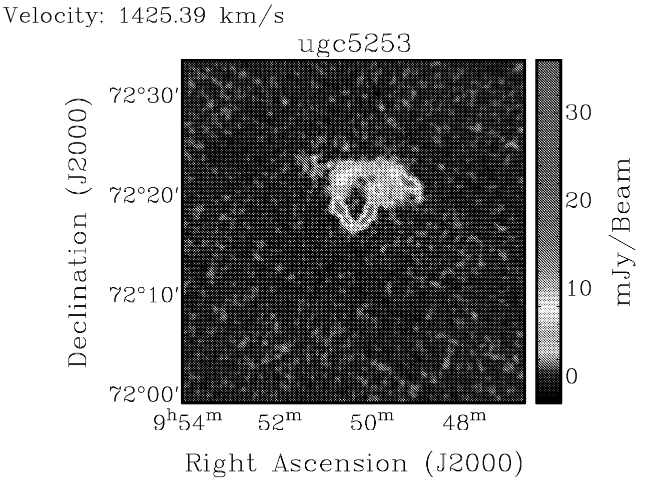

Fig. 5. A channel of a data cube of *ugc 5253*. This cube has dimensions 256 × 256 × 31. The bright green, yellow and red pixels are part of the signal that one wants to include. This signal is at least as strong as 2 mJy/beam. The distance between pixels in the x-, y-, and z-direction is 8.0 $''$, 8.0 $''$and 16.5 km/sec respectively. The beam size is $39.53'' \times 33.82''$, so the number of pixels/beam-width is approximately 5.

The sampled data is considered, $\{c_0(k)\}$. It is assumed that the sampled data are the scalar products at pixels k of the function $f(x)$ with a scaling function $\phi(x)$ which corresponds to a low-pass filter.

$$c_0(k) =< f(x), \phi(x-k) > . \tag{2}$$

The smoothed data $c_i(k)$ at a given resolution i and at a position k is the scalar product

$$c_i(k) = \frac{1}{2^i} < f(x), \phi(\frac{x-k}{2^i}) >, \tag{3}$$

where $f(x)$ is the original image. This product is consequently obtained by the convolution

$$c_i(k) = \sum_l h(l) c_{i-1}(k + 2^{i-1} l), \tag{4}$$

where h is a discrete low pass filter associated with the scaling function. Now the discrete wavelet transform, for a resolution level i is

$$w_i(k) = c_{i-1}(k) - c_i(k), \tag{5}$$

which has the same number of pixels as the image.

It is important to note that the distance between levels increases by a factor 2 from one scale to the next. So the distance between the central pixel (of h) and the adjacent ones is 2^{i-1}. The coefficients $h(k)$ derive from the scaling function $\phi(x)$,

$$\frac{1}{2}\phi(\frac{x}{2}) = \sum_{l} h(l)\phi(x-l). \tag{6}$$

In practice, computing the wavelet transform associated with the choice of h goes as follows:

1. The level i is set to zero and the data is taken as $c_i(k)$.
2. i Is incremented by 1, and the data $c_{i-1}(k)$ is discretely convolved with the low pass filter h. The distance between the central pixel and the adjacent ones is 2^{i-1}.
3. The discrete wavelet transform of level i is obtained by $w_i(k) = c_{i-1}(k) - c_i(k)$.
4. If i is less than the number of wavelet planes p that are required, then go to step 2.
5. The wavelet transform of the data is represented by $W = \{w_1, w_2, ..., w_p, c_p\}$.

The way to handle the boundaries is chosen to be continuity, so $c(k+N) = c(N)$.

To recreate the original image, c_o, in terms of the wavelet coefficients is now given as follows. The final smoothed image c_p is added to all the differences w_i (called *wavelet planes*):

$$c_0(k) = c_p + \sum_{i=1}^{p} w_i(k). \tag{7}$$

In one dimension, for the *à trous* algorithm, the convolution mask (the low pass filter h) is chosen as $\frac{1}{16}(1,4,6,4,1)$, which is associated with a B-spline scaling function $\phi(x)$ of degree 3. $B_3(x)$ is close to a Gaussian function and the results are quasi-isotropic. Other scaling functions can be used as well. The discrepancy to a Gaussian would be even smaller with $B_5(x)$. In that case the wavelet can be considered as isotropic (Bijaoui & Rue, 1995). However, since more coefficients are needed to define $B_5(x)$, it takes more computation time, which is scarce. Of course the described algorithm above is easily extended to two- or three-dimensional space.

Corresponding to the scaling function $\phi(x)$ is the wavelet function $\psi(x)$,

$$\frac{1}{2}\psi(\frac{x}{2}) = \phi(x) - \frac{1}{2}\phi(\frac{x}{2}). \tag{8}$$

Note that, for the *à trous* transform, the normalization condition $\int|\psi(x)|^2dx$ does not hold (Maisinger et al., 2004). This means that the transform of Gaussian white noise (which is not always the type of noise present, for instance with natural weighting (section 6.1.3)) does not have a constant dispersion in all wavelet domains. This means that it is very important to recalculate the dispersion of each wavelet plane when setting a masking threshold so it can not be estimated from the original image or previously calculated wavelet planes.

The *à trous* algorithm has the very nice property that each wavelet plane has exactly the same number of pixels as the original image. This is a restriction on the use of this particular wavelet transform approach for image compression. However, working with each wavelet plane individually is very intuitive. An example of the *à trous* algorithm in action can be seen in figures 6 and 7 which show a channel of a data cube, three wavelet planes of that same channel, and the residual image. It is not necessary to compute a fourth wavelet plane, since the signal-to-noise ratio (which increases with scale) is already sufficiently high in the third plane. Also, as can be seen in figure 7, wavelet plane w_i contains more negative signal if i, i.e. the number of iterations, becomes larger. This can have negative side effects.

The *à trous* algorithm is a powerful algorithm for the following reasons:

- The computational requirement is reasonable.
- The algorithm is easy to program.
- The transform is practically isotropic.
- Compact scaling functions can be used.
- The reconstruction algorithm is trivial.
- The transform is known at each pixel. This allows detection without any error and without interpolation.
- The evolution of the transform can be followed from one scale to the next.
- Invariance under translation is completely verified.

(Starck et al., 1995).

The next step is to remove or suppress background noise from the individual wavelet planes as well as from the residual image, before reconstructing the image. This reconstruction can be done like

$$c_0(k) = c_p + \sum_{i=1}^{p} \alpha(w_i(x,y), \sigma_i) w_i(k), \tag{9}$$

where σ_i is the standard deviation of the background noise at scale i, and α is a weighting function that could look like

$$\alpha(a, \sigma_i) = \begin{cases} 1 & \text{if } |a| \geq 3\sigma_i \\ 0 & \text{if } |a| < 3\sigma_i \end{cases} \tag{10}$$

This technique has been characterized as *hard thresholding* (or 'cutoff' method, see section 1.3). It is more likely however, that adaptive thresholding leads to better results (i.e. more power in relation to the number of false discoveries). Therefore the false-discovery rate (FDR) is implemented (see section 4).

3.2. Multi-resolution from the median

Other multi-resolution tools can be used, of which one is presented here. One would prefer that a positive structure in the image does not create negative values in the multi-resolution space, as the *à trous* wavelet transform does.

In the case of the *à trous* algorithm, the wavelet function is a B_3 spline, which takes the shape of the difference

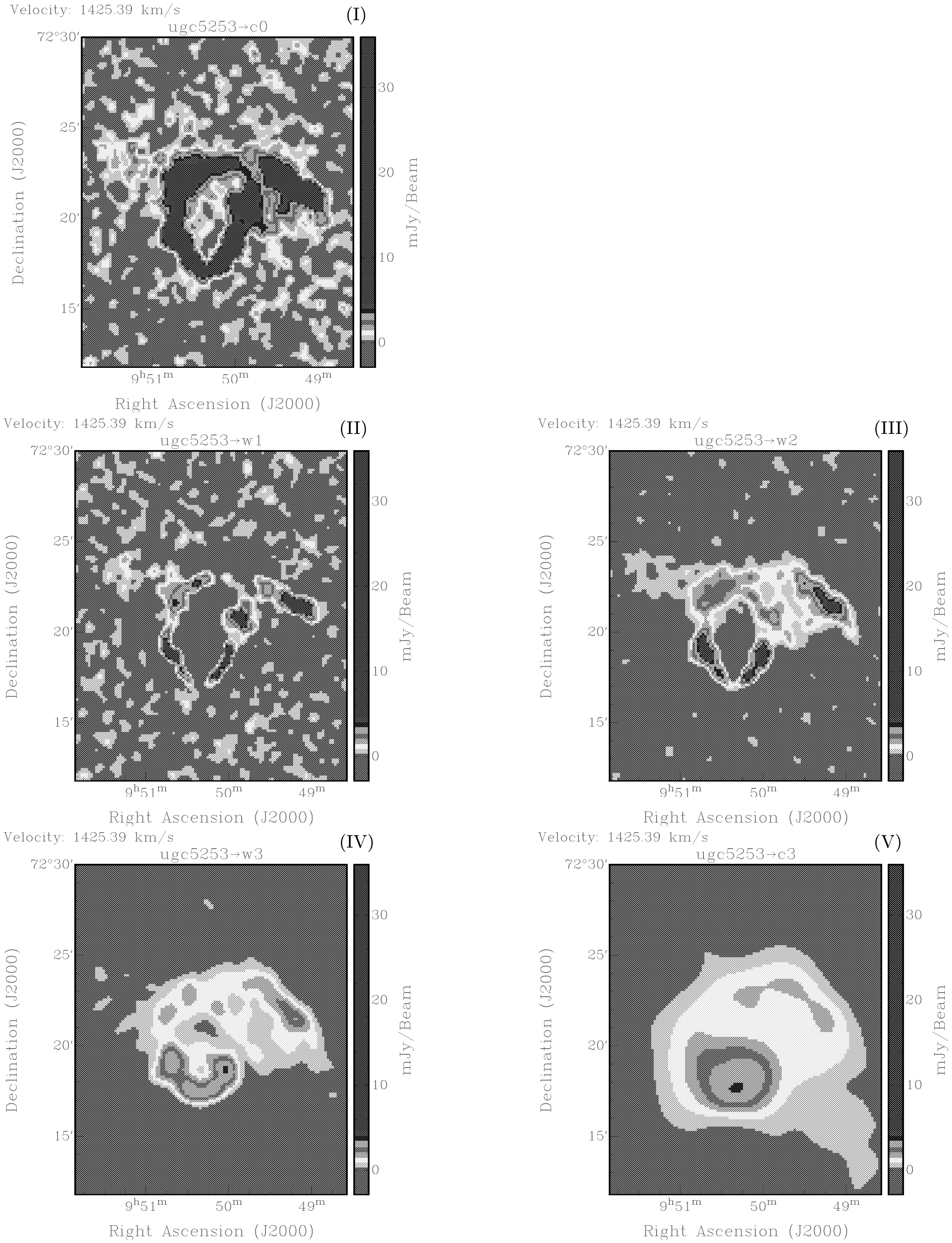

Fig. 6. A zoomed in part of the same channel of the data cube of *ugc 5253* as in figure 5. Image (I) is the original data. Images (II), (III) and (IV) are the first, second and third wavelet cubes respectively. The first wavelet cube contains the highest frequency details, the last wavelet cube contains the lowest frequency details, i.e. the most extended signal. Image (V) shows the final smoothed image. If the lower four images are added together, then the original image is recreated. A different color scheme of these same four cubes can be seen in figure 7, to show where the negative signal is.

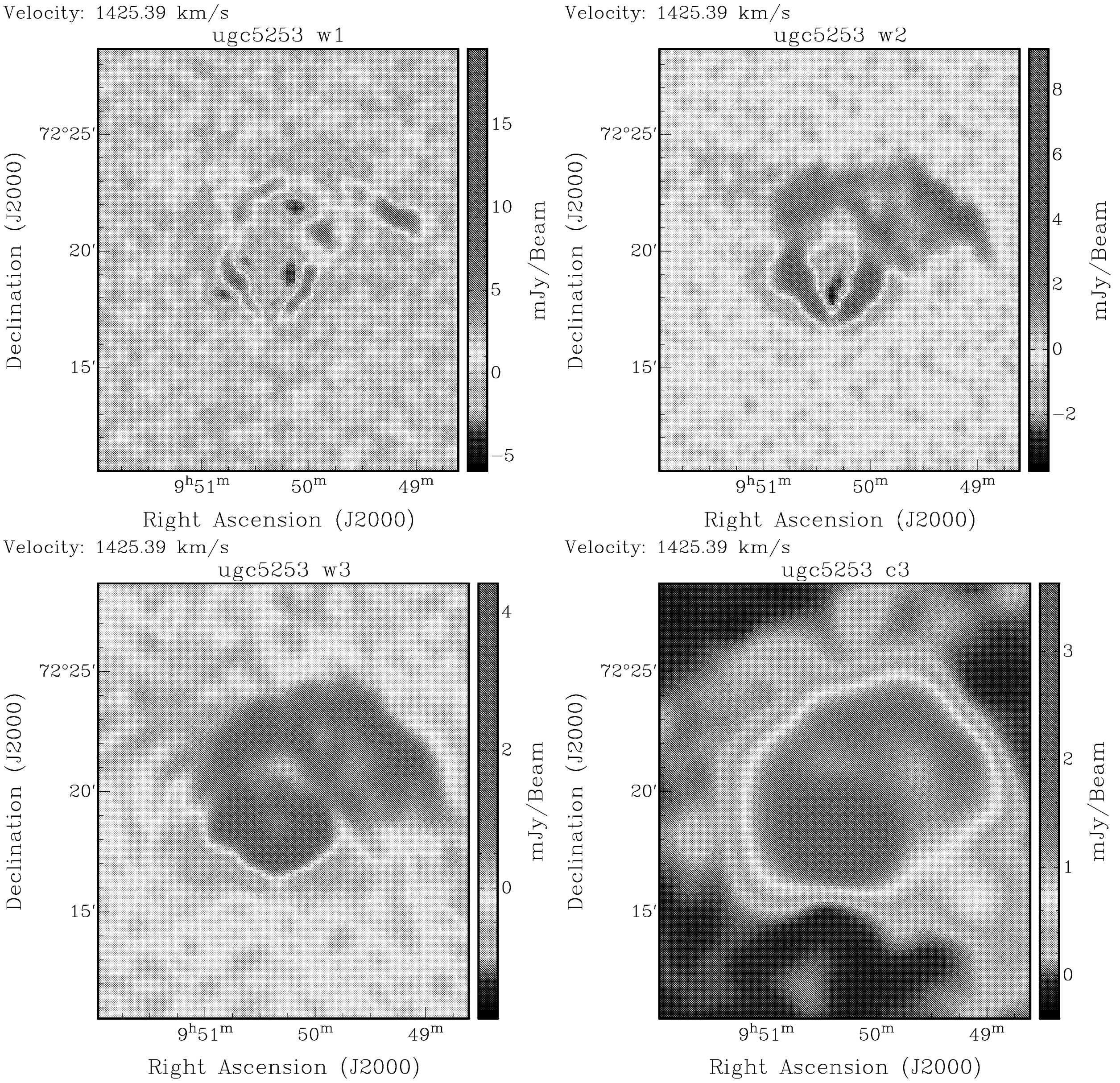

Fig. 7. The same channel of the data cube of *ugc 5253* as in figure 6. These figure have different color schemes to show the negative signal. From left to right, from bottom to top the panels show the first, second and third wavelet cube and the final smoothed image. The blue parts in the first, the blue and green parts in the second and the green parts in the third image are negative signal. It is clearly visible that wavelet plane w_i contains more negative signal if i, i.e. the number of iterations, becomes larger.

between two Gaussians G1 and G2, with the width of G2 being twice that of G1 (Starck et al., 1997). This implies that if there is a detection, then a peak at the corresponding wavelength λ_i and two smaller peaks at the opposite sign symmetrically displaced from λ_i will be present. The shape of the B_3 spline can be seen in figure 12.

By modifying the *à trous* algorithm, the median filter can be used (Starck et al., 1995). It simply uses a mask of a certain size, in which the median of each pixel is evaluated. After smoothing the data in this way, the wavelet planes are, again, simply the difference between two adjacent scales. Next the size of the mask is increased and the process is repeated until the desired number of wavelet planes is calculated. The main advantages of such an algorithm is that it can work at intermediate scales (by increasing the size of the mask by a non-integer value) and that there are no negative values created around positive values. However, the big disadvantage of such an algorithm is that it takes much more computation time. This problem can be partially solved (by using decimation, see Starck et al. (1995)), but such a transform does not replace the wavelet transform, it can only complement it.

4. The false-discovery rate

A recurrent statistical problem in astrophysical data analysis is to decide whether data are consistent with the predictions of some theoretical model. One can test whether the data and model differ overall using a statistical hypothesis test, such as the χ^2 test if the data are uncorrelated. Such a test does not specify where or how model and data differ. Instead, multiple hypothesis tests must be performed, one at each wave number. One may, for instance, declare a test significant if the discrepancy between the data and model spectrum at a certain wavenumber is greater than twice or three times the standard error of the measurement. The higher the cutoff, the smaller the probability of a false discovery. But this also severely reduces the probability of correctly detecting real deviations (source pixels). Throughout this paper, the term "source pixel" is used to mean a pixel in an image that is above some threshold and thereby assumed to be part of a true source (an actual astronomical object like a star or galaxy whose properties are of interest). All the other pixels in the image are referred to as "background pixels". So what is needed is an effective method for multiple testing that improves the probability of correct detections over methods in current use (like hard thresholding) while still controlling the probability of a false discovery. Therefore the false-discovery rate method (FDR) is chosen. FDR has three key advantages over other existing methods (Miller et al., 2001):

- FDR has a higher probability of correctly detecting real observations between model and data.
- It controls a scientifically relevant quantity; the average fraction of false discoveries over the total number of discoveries.
- Only a trivial adjustment to the basic method is required to handle correlated data.

There are two types of errors that can be made when deciding whether a test statistic is significant or not:

(I) incorrectly identifying a background pixel as source (a false discovery).

(II) incorrectly identifying a source pixel as background (a false rejection).

There is always a trade-off between the number of false discoveries (type I error) and the power (1 minus the probability of type II errors).

The only parameter the FDR uses is the significance level α. Using $\alpha = 0.05$ is equivalent to using a "2 σ" threshold. A useful quantity to compute is the p-value. It is defined as the probability, when the pixel is background, of obtaining a test statistic that is at least as extreme as the observed test statistic. So the p-value for pixel i is the probability that a background pixel will have intensity I^i or greater,

$$p^i_{val} = \int_{I^i}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{I-\mu_{back}}{\sigma_{back}}\right)^2\right] dI \qquad (11)$$

From this equation it is immediately apparent that it is important to have the right model for the background noise in the data, since the p-values depend on it (the σ and μ in equation 11 are the parameters that describe the assumed Gaussian distribution of the background noise). A graphical representation of the important variables can be seen in figure 9. Note that the calculation of the p-values is only in two dimensions for a reason (In the case of image data these dimensions are right ascension and declination, in the visibility domain these two dimensions are baseline and time). Namely, the background noise in a data cube is only assumed to be constant in each separate channel. It can vary in the velocity direction.

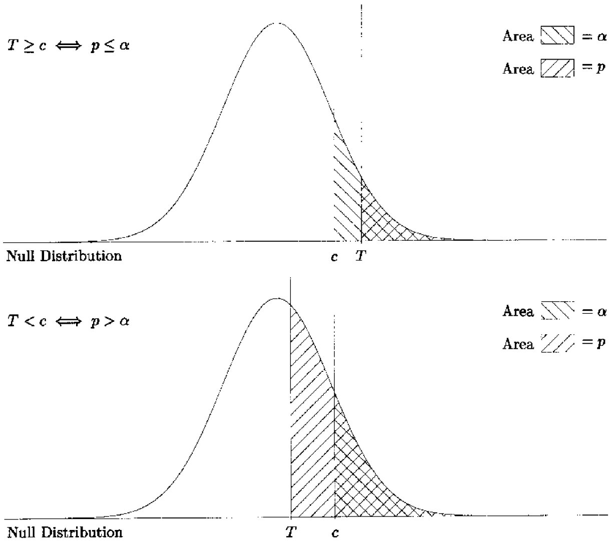

Fig. 9. Schematic illustration of the relationship between the p-value p and the type I error probability α and the equivalent relationship between the test statistic T and the critical threshold c. The curves in both panels represent the distribution of the test statistic, assuming that the pixel is background. In the upper panel, the test statistic T is larger than the critical threshold c, leading to identifying the pixel as source. In the lower panel it is the other way around, leading to identifying the pixel as background.

Most thresholding techniques are designed to control the number of type I errors (false discoveries):

Naive multiple testing (hard thresholding).
Use the same threshold (e.g. 2 σ) for every test. This same threshold is used regardless of the number of tests. Unfortunately, this leads to a higher than expected rate of making type I errors, since the chance of at least one false detection over all pixels is much larger than α. Since it's impossible to know the number of real sources, the level of contamination remains unknown.

Three-sigma multiple testing.
The same technique as naive multiple testing, except for the choice of a higher cutoff (e.g. 3 σ). This adjustment reduces the average number of false discoveries relative to naive multiple testing. However, the probability of making a false discovery stays typically much larger than the de-

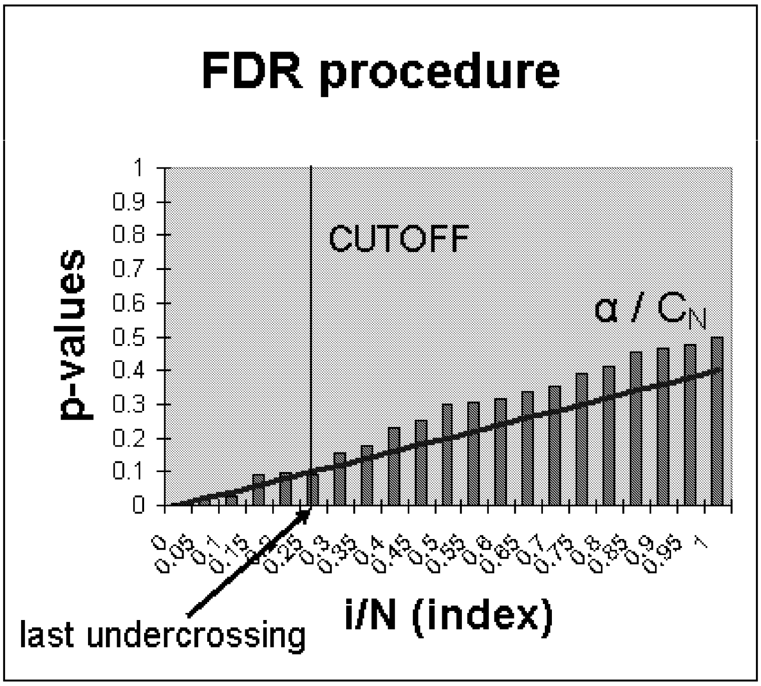

Fig. 8. Graphical representation of the *False Discovery Rate* procedure. The p-values along the y-axis are plotted against their index. The blue line crossing the origin is for α=0.40 (which is normally a very high cutoff value), and the black vertical line shows the p-cutoff value. Al the tests (i.e. pixels) to the left of the vertical line are rejected (labeled as source pixels) with a false-discovery rate of 0.40.

sired significance level α. Moreover, the chance of making a type II error (a false rejection) becomes big.

The Bonferoni method.
Define any pixel who's p-value is less than $\alpha'_N = \alpha/N$ as source pixel, where N is the number of tests. In this case, there is a strong constraint on the number of false discoveries. However, the probability of erroneously identifying source pixels as background pixels (type II error) goes to 1 as the number of tests N becomes large. So, the Bonferoni threshold identifies no false discoveries, at the expense of detecting few real sources.

5. The FDR procedure

5.1. The False Discovery Rate

The above thresholding techniques are most common and allow one to specify the number of acceptable false discoveries, but they fail to control the relevant quantity: the fraction of false discoveries over the total number of discoveries (the False Discovery Rate). It is easier to evaluate the number of source pixels than it is to know the number of background pixels a priori. Suppose that a 3σ threshold is applied to an image of 1000 × 1000 pixels, in which the background noise is completely Gaussian. Then the 3σ threshold ensures that 0.1% of the total number of *pixels* (which is 10^6) are falsely discovered. So on average, 1000 pixels are above the threshold. If there are real sources present, this means that 1000 pixels are discovered (selected). If there are many source pixels, then this threshold may be quite adequate. But if there are only 2000 pixels measured as source pixels, then half the detections are false! It is much easier to keep the number of false discoveries, on average, over the total number of discoveries small, since such a threshold *can* be specified a priori. So, the FDR is defined as

$$FDR = \frac{N_{\text{false discoveries}}}{N_{\text{discoveries}}} = \frac{N_{\text{false discoveries}}}{N_{\text{false discoveries}} + N_{\text{true discoveries}}}. \tag{12}$$

The FDR controls the most relevant errors, namely the proportion of errors among those pixels that are identified as source. Therefore the FDR attains higher power than the Bonferoni method.

First select a significance level α such that $0 \leq \alpha \leq 1$ (if α=0.10 then a tenth of the discoveries may be false discoveries). The FDR procedure described below guarantees that

$$< FDR > \ \leq \alpha, \tag{13}$$

where the expectations (in angle brackets) represent ensemble averages over replications of the data. The primary advantage of specifying an FDR α value over choosing a 3 or 5 σ threshold is that the FDR threshold is adaptive. It will assume a different value depending on the source intensity distribution relative to the background (Miller et al., 2001).

Let $P_1, ..., P_N$ denote the p-values from the N tests (pixels), arranged from smallest to largest. Now calculate

$$d = \max\left\{ j : P_j < \frac{j\alpha}{c_N N} \right\}, \tag{14}$$

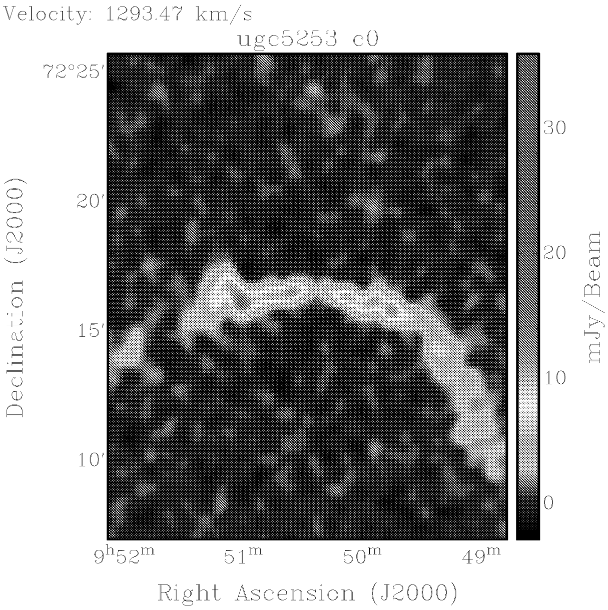

Fig. 10. A channel of a data cube of *ugc 5253*. Further info such as Right Ascension, declination and velocity have not been omitted, although they could have been since the idea behind multi-resolution filtering is that signal on *any* scale can be found. The cyan, bright green, yellow and red pixels are part of the signal that one wants to include.

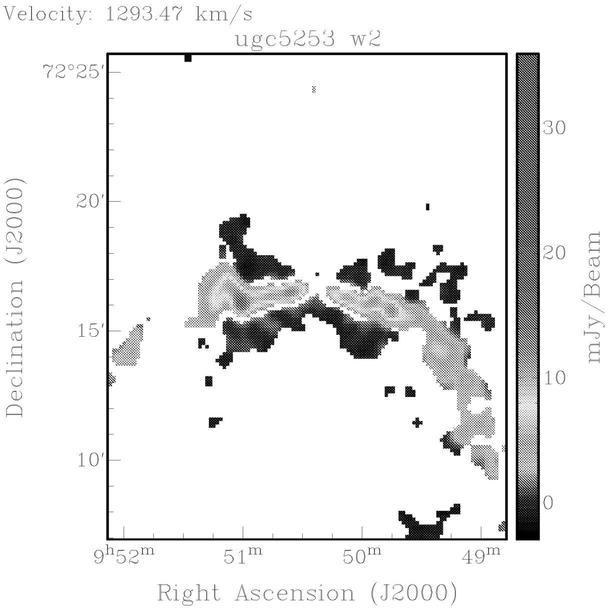

Fig. 11. The same channel of a data cube of *ugc 5253* as in figure 10. Only those pixels that are above the FDR threshold in any of the wavelet planes are shown. The mask of the other pixels is set to zero (the white pixels). The wavelet transform is the so called *à trous* wavelet transform and consists of 2 wavelet planes. The strong negative signal (above the FDR threshold) is also taken into account. It can be seen that not only the true source (the light from *ugc 5253*) but also some other, even negative signal is shown. The gap between these two parts is also visible at some points.

where the constant c_N is defined below. Now define all pixels with p-values less than or equal to P_d as source pixels. Graphically, this procedure corresponds to plotting all the P_j versus j/n. Then superimpose the line through the origin with slope α/c_N, and find the last point at which P_j falls below the line. This can be seen in figure 8.

5.2. The Constant c_N

If the p-values are based on statistically independent tests, then $c_N = 1$. If the tests are dependent, then

$$c_N = \sum_{i=1}^{N} i^{-1}. \qquad (15)$$

Most radio images show some degree of correlation between pixels but tend not to be fully correlated. This means that the intensity of a given pixel is not influenced by that of every other pixel in the image. Therefore an intermediate estimate for c_N is chosen that reflects the level of correlation present in the image. This is related to the synthesized beam size (or Point Spread Function (PSF)). Now define "partially correlated" coefficient $c_N = \sum_{i=1}^{n} i^{-1}$ where n is an integer number of pixels representing the PSF ($n = PSF_{area}$) (Hopkins et al., 2002).

Calculating the right number of pixels representing the PSF goes as follows. The integral over the synthesized beam

$$I_{beam} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e^{-\frac{1}{2}(\frac{x}{\sigma_x})^2-\frac{1}{2}(\frac{y}{\sigma_y})^2} dxdy \qquad (16)$$

in two dimensions, so that $I_{beam}(x = 0, y = 0) = 1$. The average value of pixels in the image is assumed to be $\mu = 0$ (which is should be). Then

$$PSF_{area} = I_{beam} = \sqrt{2\pi}\cdot\sigma_x\cdot\sqrt{2\pi}\cdot\sigma_y = 2\pi\sigma_x\sigma_y. \qquad (17)$$

Also we have that the FWHM (Full Width Half Maximum) = $2.354\cdot\sigma$ so

$$PSF_{area} = \frac{2\pi\cdot FWHM_x\cdot FWHM_y}{(2.354)^2\cdot cell^2} \qquad (18)$$

where *cell* is the image cell size in arcsec.

Using the form for C_N described here is not a rigorous result of the formal FDR proof. It is a 'compromise' estimate that seems mathematically reasonable and gives reliable results in practice (Hopkins et al., 2002).

5.3. Negative signal

When applying the FDR procedure to the wavelet cubes, the question rises whether to include negative signal as well or not. Suppose the data contains a point source, which can be represented by a Gaussian. Then the smoothing done by the wavelet transform will broaden this Gaussian, though retaining the total power, so the peak of the smoothed Gaussian will be lower. The wavelet plane is the difference between these two Gaussians. This will

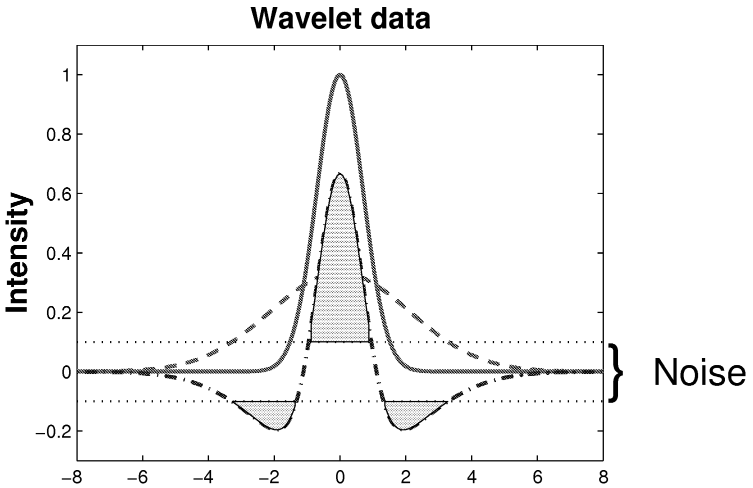

Fig. 12. Graphical representation of a point source (the red, solid Gaussian), the smoothed point source (the green, dashed Gaussian) and the difference (the wavelet plane; the blue, dash-dotted line). The difference is the shape of the effective point spread function for the B_3 spline used in the *à trous* wavelet transform. The filled area represents the pixels identified as source. Including the negative signal (the 2 lower filled areas) will identify too many pixels as source pixels. In this particular case one does not want to identify any pixels as source that are at $|x| >\sim 2$. Note the gaps between the positive signal and the negative signal in the wavelet plane.

be a function that looks like the original Gaussian, but it will have two tails that are negative. A graphical representation of the Gaussian, the smoothed Gaussian and the difference (the wavelet plane) can be seen in figure 12. This negative signal is indeed signal in the smoothed Gaussian. But this is not (completely) the case for the original Gaussian. So when a filter is applied to the wavelet plane that includes the strong negative signal (above the FDR threshold), too much area around the true signal is identified as source. In between the positive signal and the negative signal in the wavelet plane, there will also be a gap. Therefore the negative signal is not taken into account. An example of the consequences of taking negative signal of image data into account, can be seen in figures 10 and 11. The first is a channel of a data cube of *ugc 5253*. The latter is that same channel, to which a masking operation is applied (all the signal in any wavelet plane that is above the FDR threshold (α=0.05 in this particular case) for that plane is retained).

6. Mask enhancements

Once a certain mask has been made for a given data cube, there are several ways of enhancing it. This means that it is assumed, that after a certain selection method has been applied such as hard thresholding or the FDR method, *not* all signal has been found. These selection methods are not perfect. That is because they select only the *strong* signal. They will look over weak, but extended emission just above the background noise level. Weak emission that surrounds strong emission is likely to be part of the true emission.

Part of the solution to this problem lies in the wavelet transform. The combination of the latter with the FDR method can produce much better results. The FDR method should be used to extract the strong features in the image data at all scales. The other part of the solution is to let complementing signal selection methods find the described weaker signal. These are presented here.

6.1. Region selection

The signal selection method presented here is based on the so-called *morphological watersheds*, which is a segmentation procedure. The latter will be explained first, to describe the concept of a watershed.

6.1.1. The watershed transform

The concept of watersheds is based on the representation of an image in two spatial coordinates and one gray level dimension. Now there are three different types of points (Gonzalez & Woods, 2002):

- **a)** points that belong to a regional minimum;
- **b)** points at which a drop of water, if placed at the location of any of those points, would fall with certainty to a single minimum.
- **c)** points at which a drop of water, if placed at the location of any of those points, would be equally likely to fall to more than one such minimum.

For a particular regional minimum, the set of points satisfying condition b) is called the *watershed* of that minimum. The set of points satisfying condition c) form crest lines on the topographic surface and are called *watershed lines*. The principal objective of a segmentation algorithm based on the watershed concept is to find the watershed lines. The concept of how to do this is simple: Suppose that a hole is punched in each regional minimum and that the entire topography (the image) is flooded with water from below by letting water rise through the holes at uniform rate. When the rising water in two or more watersheds is about to merge, a dam is built to prevent the merging. When the water has reached the highest mountain top, all the dams form the watershed lines (the continuous boundaries extracted by the watershed segmentation algorithm).

6.1.2. The watershed-equivalent transform

A variant to the watershed algorithm is implemented. It is a kind of *region growing* (Gonzalez & Woods, 2002), which is part of the real watershed transform. In this case not the regional minimums, but the regional maximums are important. This algorithm has to work on a predefined mask. So local maximums are only searched for in regions of the image where the mask is set to one (regions that contain emission). From these maximums, a downhill search is performed, in 2 dimensions. It searches for all the points with monotonically decreasing values in the image, only taking those pixels with values above a certain clipping level (threshold), for instance 0.5σ. This is to prevent that too many pixels are included. For all the pixels that are found, the corresponding mask pixels are set to be included. This way an object is formed that is very much like a watershed, except that it starts from the top and expands downward instead of upward. Just as with the watershed algorithm, different watersheds do *not* merge this way. An example of the result of the algorithm can be seen in figure 13. Why the results are more pronounced with this data cube than with the one in figure 5 is explained in section 6.1.3. A graphical representation of the algorithm itself can be seen in figure 14.

Region growing is, in principle, a procedure that groups pixels of subregions into larger regions based on predefined criteria. The basic approach is to start with a set of 'seed' points. These regions can grow by appending neighboring pixels to them that have similar properties as the seed. Such an algorithm can be improved by additional criteria. For instance, one can look at all the pixels that have been added so far and create criteria based on their value, average or the shape of the grown region. Note that for the above described algorithm the region itself can in fact also decrease depending on the value of the threshold. This is because the seed points are taken only as the local maximums.

The fact that watersheds do not merge is to prevent each watershed from including noise peaks that are just above the chosen threshold. Once the local maximums are found, the algorithm searches for adjacent pixels with monotonically decreasing value. If the pixel values do not have to decrease monotonically, then the watersheds will merge. The result can be seen in figure 15. It shows that by far too many pixels are included, since half of the channel has been included. The problem is that too much signal is above the threshold of 0.5σ (so watersheds can merge this way).

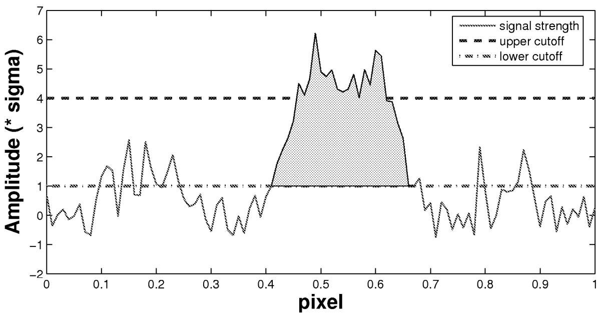

Fig. 14. Graphical representation of the watershed-equivalent algorithm. First, all the peaks above some high threshold (blue dashed line) are selected. From there, a downhill search is performed to select all adjacent monotonically decreasing signal that is above some lower threshold (red, dash-dotted line) that can be 0.5 or 1 σ for instance. This way the extended emission (colored area) can be completely selected.

6.1.3. Weighting functions

It must be noted that for the data cube in the previous example (section 6.1.2) a different weighting function has been used than for the data cube in figure 5. When sampling the u, v data, a weighted sampling function can be written as

$$W(u,v) = \sum_{k=1}^{M} R_k T_k D_k \delta(u - u_k, v - v_k) \tag{19}$$

(Perley et al., 1986). The coefficients R_k, T_k and D_k are weights assigned to the visibility points. The R_k are weights that indicate the reliability of the k^{th} visibility datum. The tapers T_k are used to weight down the data at the outer edge of the u, v coverage. Thus they suppress small-scale side-lobes and increase the beam width. The density weights D_k are used to offset the high concentration of u, v tracks near the origin, and to lessen the side-lobes caused by gaps in the coverage. In other words, the D_k simulate a more uniform u, v coverage if needed. They compensate for the clumping of data in the u, v plane by weighting by the reciprocal of the local data density. Two kinds of weighting are commonly applied:

- *natural weighting*, $D_k = 1$
- *uniform weighting*, $D_k = \frac{1}{N_s(k)}$,

where $N_s(k)$ is the number of data points within a symmetric region of the u, v plane. This region is centered on the k^{th} data point and has a characteristic width s, which might be the radius of a circle or the width of a square. The data cube in figure 13 uses natural weighting. This means that all points are treated alike, and gives the best signal-to-noise ratio for detecting weak sources. However, since the u, v tracks tend to spend more time per unit area near the u, v origin, natural weighting emphasizes

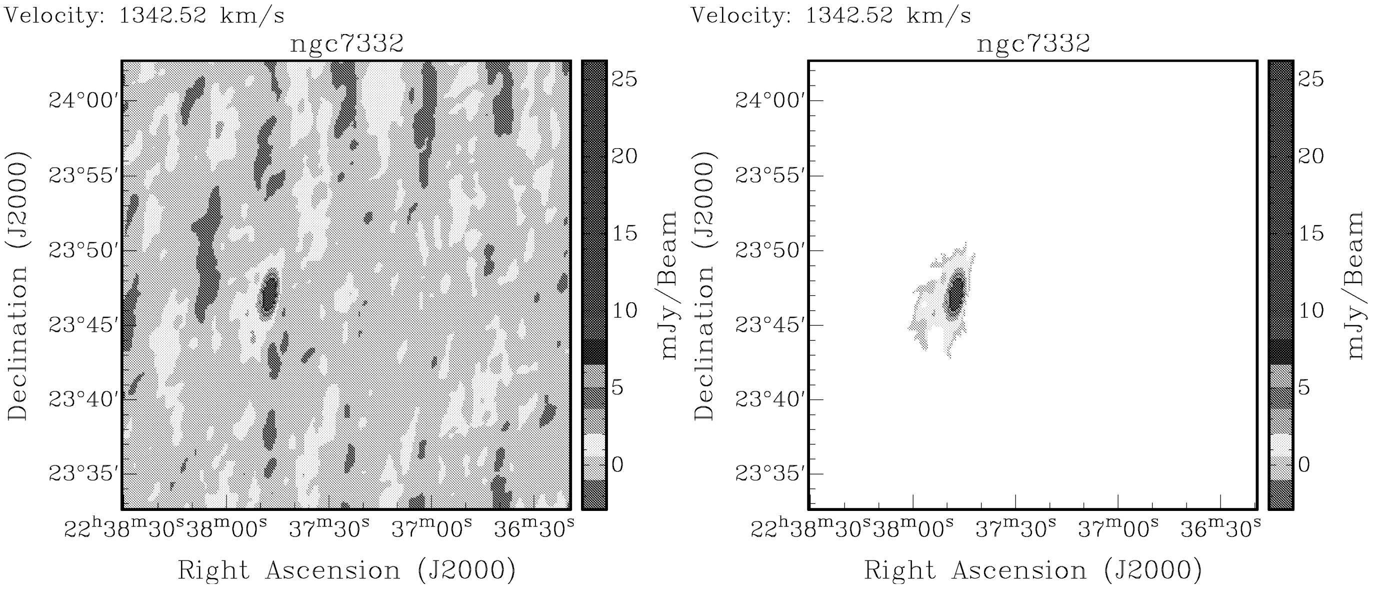

Fig. 13. A channel of a data cube of *ngc 7332*. The dark green, blue and black pixels is the strong signal that the FDR method will define as signal. The right panel shows the result of the combination of the FDR method and the watershed-equivalent algorithm, which have been applied to all the data cubes (including the wavelet cubes and the first smoothed data cube) and all the masks have been added. It shows that the lower light green area close to the selected (dark) area has been missed (i.e. has not been merged with) by the watershed-equivalent algorithm.

the data from the short spacings. This tends to produce a beam with a broad, low-level plateau which is undesirable when imaging sources with both large and small scale structures. Since the data cube in figure 13 has been natural weighted, the watershed-equivalent algorithm is especially useful because of the high signal-to-noise ratio for detecting weak sources.

6.2. Morphological operators

Another way of enhancing the mask is using morphological operators. Two of those are implemented, namely *opening* and *closing*. In order to explain these two morphological operators, *dilation* and *erosion* must be explained first.

6.2.1. Dilation and Erosion

Sets in mathematical morphology represent objects in an image. In binary images (such as the mask-planes of a data cube), the sets in question are members of the 2-D integer space Z^2, where each element of a set is a tuple (2-D vector) whose coordinates are the (x, y) coordinates of a black pixel in the image (Gonzalez & Woods, 2002). Suppose we have two sets, A and B, in Z^2. B is reflected about its origin, $\hat{B}$, and shifted by z, $(\hat{B})_z$. Then the *dilation* of A by B is the set of all displacements z such that $(\hat{B})_z$ and A overlap by at least one element. Set B is commonly referred to as the *structuring element.* Note that when B is a symmetric set with respect to its origin, $\hat{B} = B$. It is obvious that dilation expands set A.

Again consider two sets, A and B, in Z^2. Now the *erosion* of A by B is the set of all points z such that $(B)_z$ is *completely* contained in A. This way the set A can only shrink. Graphical representations of both *dilation* and *erosion* can be found in Gonzalez & Woods (2002).

6.2.2. Opening and Closing

The *opening* of set A by structuring element B is the erosion of A by B, followed by a dilation of the result by B. So at first the set A shrinks a bit, and then it expands again. This generally smooths the contour of an object. Also, it eliminates thin protrusions and breaks narrow strips, hence the term opening.

The *closing* of set A by structuring element B is the dilation of A by B, followed by an erosion of the result by B. This way, closing also tends to smooth sections of contours. However, as opposed to opening, closing generally fuses narrow breaks, eliminates small holes and fills gaps in a contour.

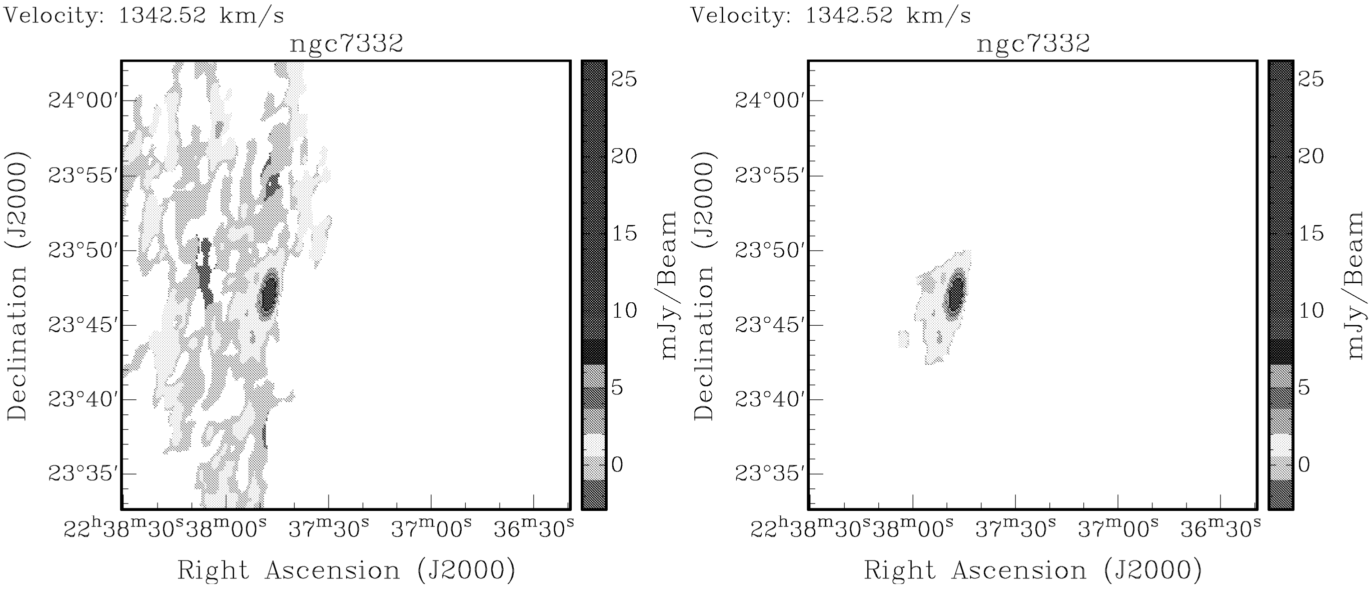

Fig. 15. The same channel of the same data cube as in figure 13. The watershed-equivalent algorithm has been applied. In the left panel, the constraint that when expanding the mask only decreasing-value pixels are included, has been removed. It shows that by far too many source pixels are included. Thus, the mask of the right image of figure 13 is used. First an opening and a closing have been applied to get rid of the rough edges. Next another step of the watershed-equivalent algorithm has been applied. Then, again an opening and a closing have been applied. It shows that, in comparison to figure 13, the relatively strong source that is nearby the originally selected source, has been selected this time (right panel).

7. Implementation for the spatial domain

The implementation of all the different algorithms and the order in which they are executed is discussed here. First, an overview of the complete implementation is given. Next, all the different parts of the implementation are explained in more detail, giving examples of the results and the reasons why certain choices are made.

Note that this approach to automated feature extraction works best on extended sources. Galaxies have the characteristic that they rotate, so they are sheared a bit in the frequency. This does not hold for point sources however, so for these sources the three-dimensional *à trous* wavelet transform is not useful.

7.1. Overview

The full implementation looks like this (only for image domain data cubes):

1. The mask of the original data cube, called c_0, is reset.
2. As many wavelet transforms are done on c_0 as the user wants wavelet cubes.
3. A sequence of algorithms is executed on c_1 and all wavelet cubes. First, the FDR selection method is used and then the watershed-equivalent procedure is executed.
4. All of the masks are combined, to form the first mask.
5. An opening and a closing are performed.
6. The new mask undergoes another step of the watershed-equivalent algorithm.
7. Again an opening and a closing are performed.
8. All the objects that do not "span" a certain range of channels are removed.
9. The data cube and its new mask are written.

7.2. Explanation

1. The mask of the original data cube, called c_0, is reset. So the values of all the pixels in the mask are set to 1. This means that the whole data cube becomes visible when viewed with a viewer.

2. As many wavelet transforms are done on c_0 as the user wants wavelet cubes. Because all of these cubes are needed later, they are all stored on the hard drive. This step creates $\{c_0, ..., c_p\}$ and $\{w_0, ..., w_p\}$, where p is the number of iterations of the wavelet transform. The first wavelet cube, w_1, contains the highest frequency details of the original images. The last, w_p (equation 7), contains the lowest frequency details. So the most extended features of the images are stored in this wavelet cube. The final smoothed images c_p are stored in the original data cube.

An important choice to make is the number of wavelet planes that will be created. It is common to create 3 wavelet planes for an image. The images used (astro-

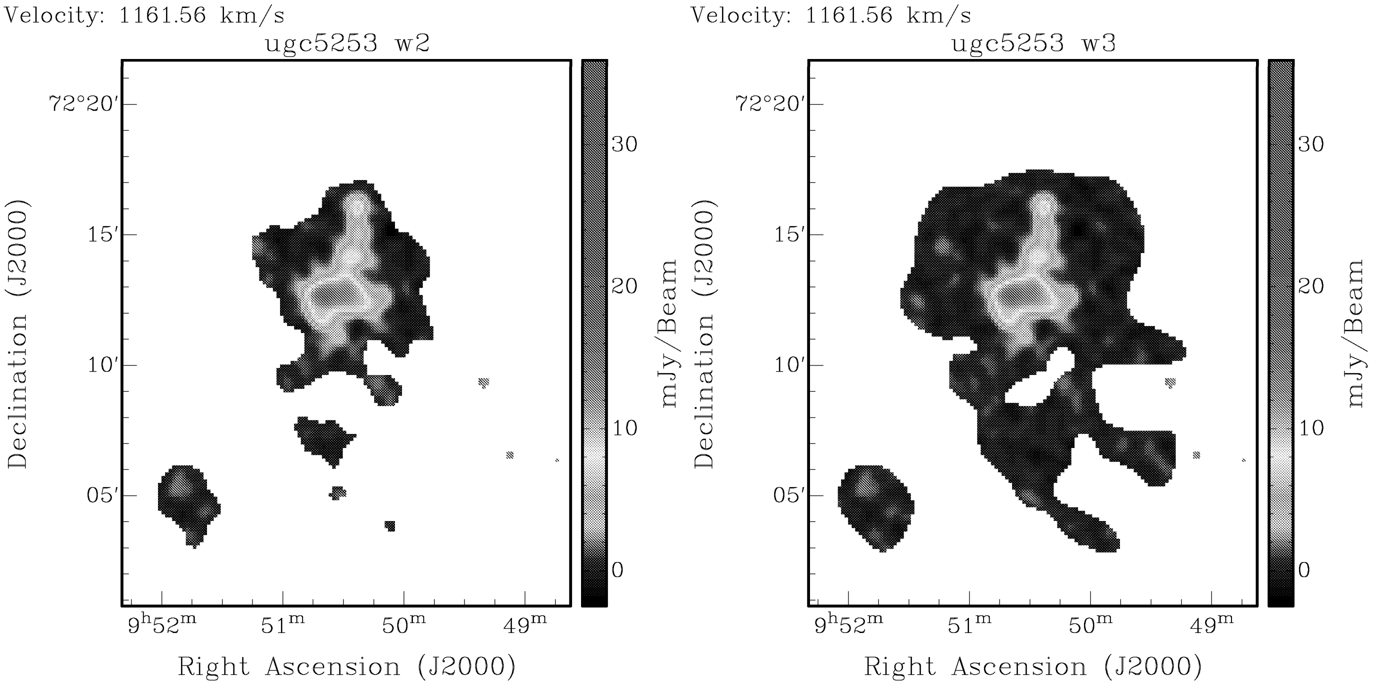

Fig. 16. A channel of a data cube of *ugc 5253*. The left image has been wavelet-transformed in 2 levels, in 3 dimensions. Both of the wavelet cubes, as well as the final smoothed cube have been searched for astronomical signal with the FDR selection method. The value for the significance level α is 0.05. Next, all of the masks have been combined, simply by adding them. The white pixels were not selected by the FDR method. The green and red visible parts are the strongest signals, at least 2 mJy/beam. Now some of the blue pixels must be included (so that there is a blue strip of approximately one synthesized beam around the stronger green and red signal). Most of the dark blue pixels shouldn't be included however, since these are negative pixels (negative mJy/beam). The right image shows the same procedure with three wavelet planes. It can easily be seen that even more (dark) blue pixels are now visible. This is because the strong signal of channels close to this channel has been smoothed and thereby entered this channel. It is obvious that this way too many pixels have been falsely identified as source.

nomical images) contain a lot of information in the low frequencies. The more iterations (convolutions) are done, the less information the next wavelet plane s will contain. On the other hand, the signal-to-noise ratio increases with scale, and from the third plane it is sufficiently high ((Starck & Murtagh, 1994)). However, in practice it appears that when a third wavelet plane is also calculated, too much data will be falsely identified as source. This can be seen in figure 16. This is because the 3 dimensional wavelet transform will smooth and spread strong signal in one channel into adjacent channels.

3. A sequence of algorithms is executed on c_1 and all wavelet cubes. First, the FDR selection method is used and then the watershed-equivalent procedure is executed.
   - First, the False Discovery Rate selection method is used to set the mask of c_1 (see section 4 and 5). This mask is used for the original data cube c_0. This is the kind of mask that is very often used (applying a cutoff method to the smoothed data to set the mask of the original data). Sometimes this already gives nice results, but it always misses out on the weak, extended sources. An example of the result of this step can be seen in figure 17. It shows that the faint signal in the upper left corner close to the strong signal has been missed out. This channel will be used in the next steps for further visual support of the explanation of the different steps in the whole procedure.

When applying the FDR method, the p-values are calculated, based upon a model for the background noise. This model is created as follows. First an approximation of the standard deviation σ is made, by calculating the squared average pixel value of all the pixels **in each channel separately**. The latter is very important, since the characteristics of the background noise can vary with each different channel. This is because the system temperature of the telescopes varies a little with frequency. Now all the pixels with values between -2 σ and +2 σ are put into a histogram. This is to leave out as many true source pixels as possible. Then a Gaussian curve is fitted to the pixels in the histogram. This gives the best fit for the true standard deviation. The value for the average pixels value is set to 0. This provides a model for the Gaussian noise.

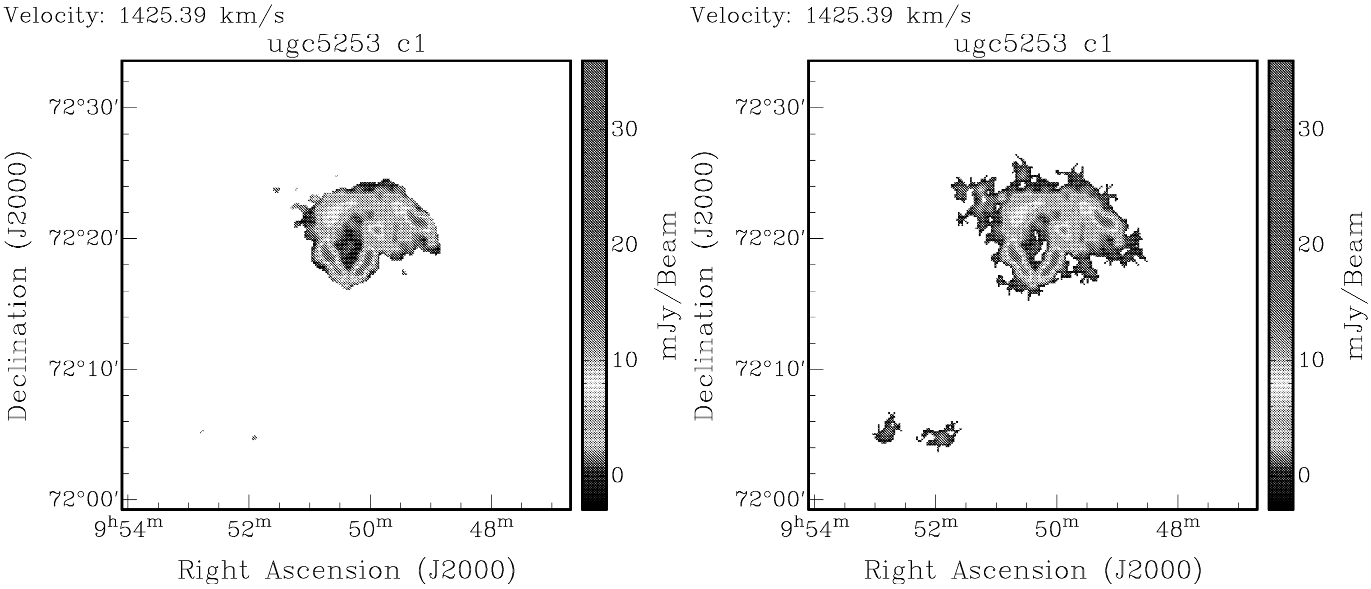

Fig. 17. The same channel of the data cube of *ugc 5253* as shown in figure 5. The FDR method has been applied to the smoothed data of this channel (c_0) to set the mask. It shows that the faint signal in the upper left corner close to the strong signal (as can be seen in the upper left panel of figure 6) has been missed out. The right image shows the result of expanding this mask with the watershed equivalent algorithm. More details of this step, as well as the next steps, can be seen in figure 21.

This model for the background noise can immediately be used as the theoretical distribution in the Kolmogorov-Smirnov one-sample test for goodness of fit (section 2). It shows that for each channel in the tested data cubes, the Kolmogorov-Smirnov statistic D stays so low, that the equivalent probability of occurrence of the furthest outliers is higher than 20 percent so the assumption of a Gaussian distribution of the background noise is justified.

There are other ways of finding the right model for the background noise. One other way is by using the median to get the standard deviation. The median also looks at negative values. For pure Gaussian distribution, it holds that $\mu = 0.6744897\sigma$, since the result of calculating equation (11) for $z = \frac{I-\mu}{sigma} = 0.6744897$ is 0.5. The median is less sensitive to outliers than the average is, so using the median to find the standard deviation is a more robust way than using the squared average pixel value to find an approximation of the standard deviation. Since the data will be fitted afterward anyway, it is doubtful however, if this will improve the results. It is also possible to execute the FDR method twice: once to give a good approximation of the background noise, and once to use the pixels that are not selected as signal as input for the model for the background noise. It is obvious that this method will eventually converge to the best results. It takes up a lot of computation time however, since finding the p-values is computationally expensive, as well as sorting them is.

- Next, the watershed-equivalent procedure (see section 6.1.1) is executed in two dimension, which expands the mask of c_0. Three dimensions is also possible, but this clearly selects too much signal if two succeeding channels are not correlated enough (the distance in velocity between channels is too large). In other words, the data are not isotropic in three dimensions, because the pixels are more correlated in the spatial axes than in the velocity axis. This should be better after a Hanning smoothing of the data in the velocity direction, since then half the power of each channel will also be in its neighbors. The threshold is set to 0σ. This makes sure that a strip of weak signal of approximately one synthesized beam around the stronger signal is included. The result of this step can be seen in figure 17.

The same sequence of algorithms described above, is also executed to set and expand the masks of all the wavelet cubes. This is a very essential step in the process, since the watershed-equivalent algorithm will find the extended (very) weak signal around stronger signal, in the lower frequency wavelet cubes. An example of this can be seen in figure 18 and, which show the first three wavelet cubes after the sequence

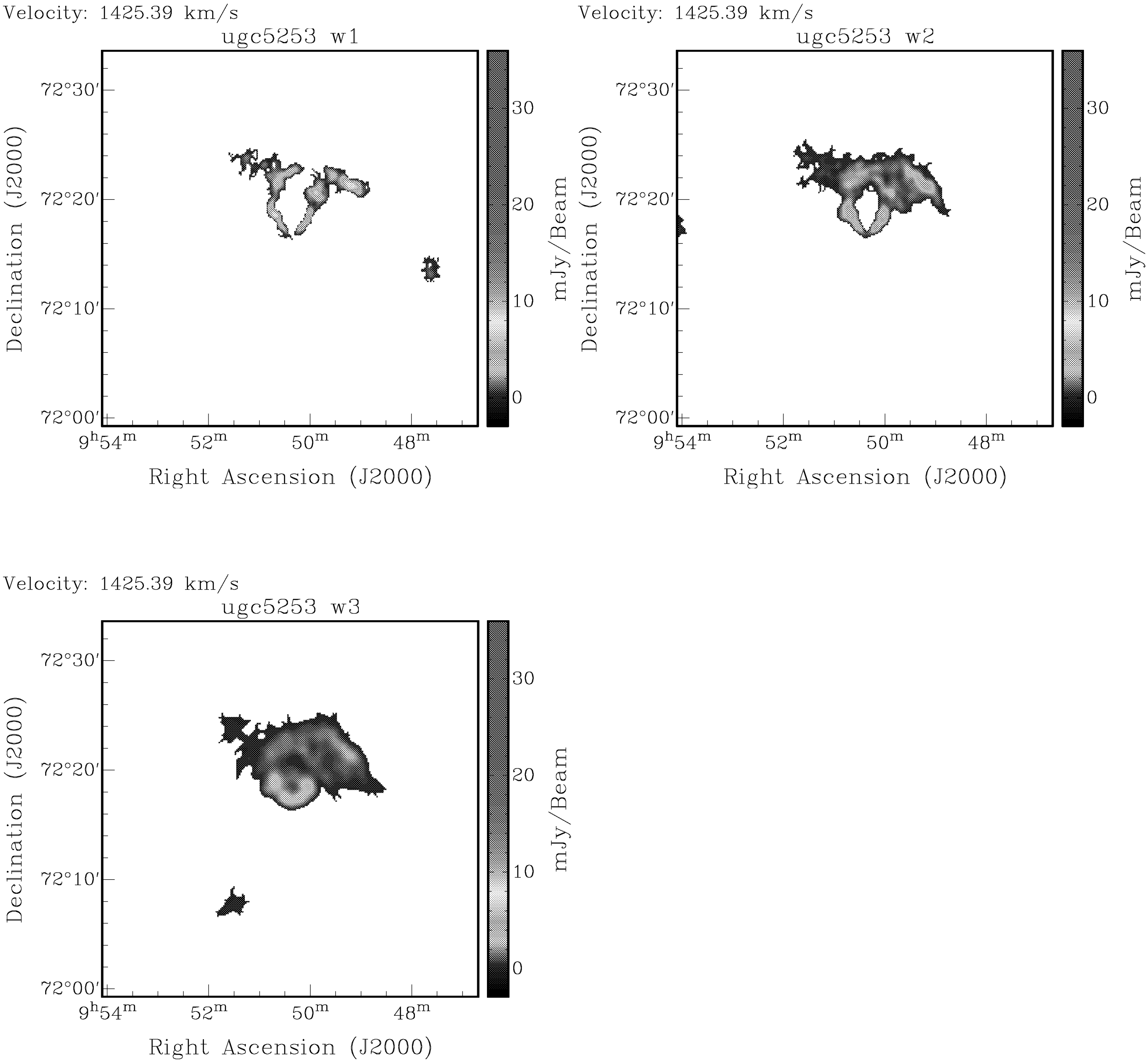

Fig. 18. The same channel of the data cube of *ugc 5253* as shown in figure 17. The FDR method followed by the watershed equivalent algorithm have been applied to the first, second and third wavelet cubes (upper left, upper right and lower left images). When the masks are compared it shows that sometimes different sources show up only in one single wavelet cube. Also it can be noted that the mask of the second wavelet cube was able to pick up the faint signal in the upper left corner close to the strong signal (which was not picked up by the first smoothed cube (figure 17)). The third wavelet cube is shown just for completeness. More details of this step can be seen in figure 21.

of algorithms.

it shows that sometimes different sources show up only in one single wavelet cube. Also it can be noted that the mask of the second wavelet cube was able to pick up the faint signal in the upper left corner close to the strong signal (which was not picked up by the first smoothed cube (figure 17)). The third wavelet cube is shown just for completeness. It can be seen that another object is visible in this wavelet plane, that does not show up in the first two cubes. This is a very weak signal, but it is visible over several channels at exactly the same position. That is why it shows up in the third wavelet cube. This is likely to

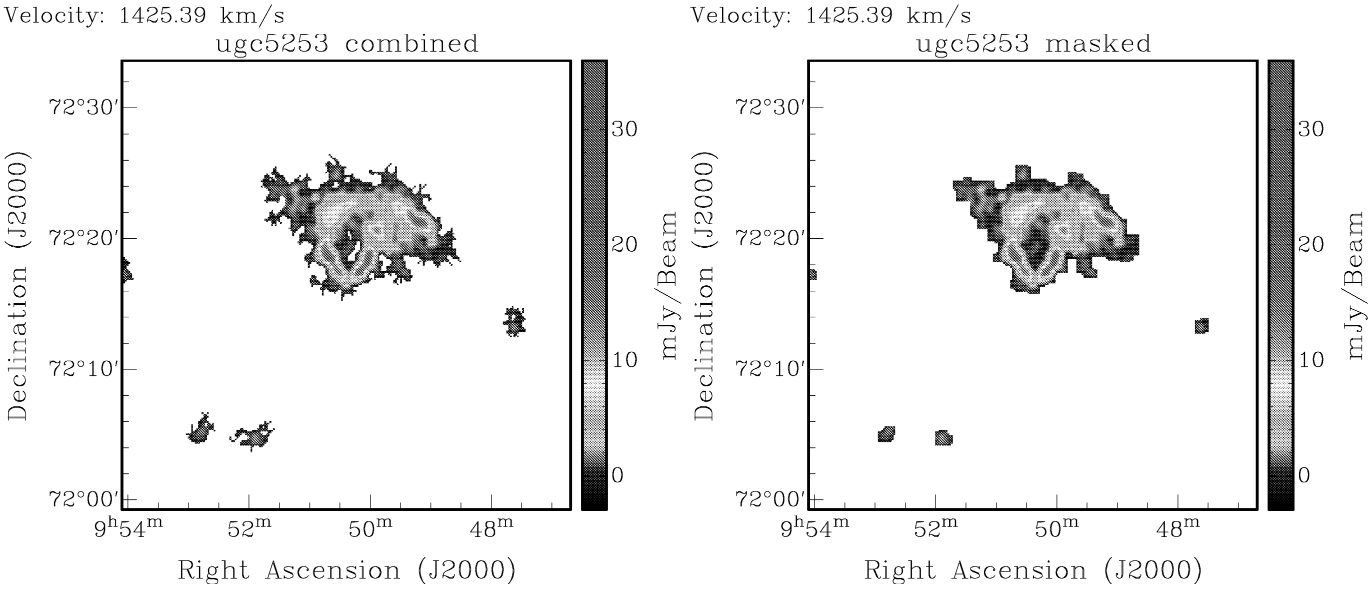

Fig. 19. The left panel shows the result of adding the mask of the right image in figure 17 to all the masks in figure 18. The right panel shows the result of applying an opening and a closing to this mask. All the "fingers" are removed. The diameter of the structuring element is 5 pixels, which is approximately the number of pixels/beamwidth. More details of this step can be seen in figure 21.

be RFI. If one wants to find these particular sources, it is possible to indeed include the third wavelet cube.

4. All of the masks (of c_0 and all the wavelet cubes) are combined, to form the first mask (figure 19). This is a very rough mask, with a lot of gaps, noise peaks (which show up mostly in w_1) and long fingers(parts of an object close to the edge of the object).
When combining these masks, a choice needs to be made on how to do this. What is done here, is simply adding all the masks. This means that if one of the masks at a certain pixel is set to one, then the resulting mask will also be a one at that pixel. There are different ways of doing this of course. One other way is to include a pixel in the mask if it is visible in at least two other masks. A stronger constraint would be that at least two "touching" masks must both be visible. This means that wavelet planes w_i and w_{i+1}, or w_p and c_p must both be visible. Such criteria alone appear not to be sufficient, so more complicated combination rules must be made up in order to get better results than simply adding all masks. It might be a step forward to use some information a priori. For instance, a point source will be smeared when smoothing the data. This means that it should be most prominently visible in the highest frequency wavelet plane. So when looking for point sources specifically, this could be a good criterion when adding the masks. A more extended source on the other hand, should be more prominently visible in the lower frequency wavelet planes.

5. An opening and a closing are performed, to get rid of the unwanted features described above (see figure 19). The operations are performed as described in section 6.2, with a circular structuring element. The radius of the structuring element is a parameter that can be adjusted. If one wants a smoother final mask, the radius must be chosen larger. Instead of choosing the radius however, it would be better to set it depending on the number of pixels per beamwidth. Suppose the beam has a width of 40″and the distance between two adjacent pixels in the data cube is 8″, then real astronomical signal must have a width of at least 5 pixels. So the radius of the structuring element must be at least $2\frac{1}{2}$, so that spurious signal in an image smaller than this size will be filtered out of the mask. In practice, a radius of approximately this size gives nice results, as can be seen in the right panel of figure 19.

The opening and closing performed here, are done in two dimensions (the spatial dimensions). Using three dimensions is also possible, but this gives terrible results. On one hand, signal will be included (because of the dilations performed) because an adjacent channel contains significant structures. On the other hand, some signal will be left out (especially at the borders of structures in a particular channel) because the cube used as structuring element does not fit in three dimensions. Again, as with the watershed-equivalent algorithm, the difference in correlation in spatial and velocity direction is too large so the data is not isotropic enough.

6. The new mask undergoes another step of the watershed-equivalent algorithm, to find sources that are nearby the originally selected sources (by the FDR procedure) and relatively strong.
   By using the watershed-equivalent algorithm, there is a specific kind of signal that will not be included, but which one probably wants to include. This is signal that was originally *not* included in the mask (by the first thresholding algorithm, i.e. the FDR method). It lies close to a stronger signal that *was* included. So the first step of the watershed-equivalent algorithm will miss out on this signal because of the criterion that neighboring pixels must have monotonically decreasing value. This new signal can be found by *another* step of the watershed-equivalent algorithm.
   Suppose we start with the resulting data cube of figure 13. This figure shows that this specific type of signal has not been selected. First, an opening and a closing are performed to expand the mask "into" the new area, as is described in the last step. This is necessary to find the new local maximum inside this new area. Then the watershed-equivalent algorithm is applied again, this time with a threshold of 0.5σ. This threshold is necessary, because else too much signal will be included. The value of this threshold is rather arbitrary though, and should be chosen depending on what one thinks should be the strength of a source at its borders.

   This step should in fact be combined with the first step of the watershed-equivalent algorithm to speed things up. Therefore another, more complicated region growing algorithm must be created.

7. Again an opening and a closing are performed to remove unwanted features. These features have appeared again, because the watershed-equivalent algorithm has been applied again. The result of this step can be seen in figure 15. Note that this mask included signal that was just above the level of the background noise. By setting the threshold of the last step higher, less signal would be selected.

8. All the objects that do not "span" a certain range of channels are removed. Since it is possible to calculate the minimal range in km/sec an astrophysical object must span in the data cube, this method can remove spurious signal that span less than this minimum. In other words, the signal has to be above a certain cutoff level in more than one consecutive channel.

   First the definition of an object must be given. A general idea for such a definition lies in the connectivity property (i.e. (Starck & Bijaoui, 2000)). Consider two structures in two successive channels, S^1 and S^2. Each structure corresponds to a region in each image where the signal is significant (its value is above a given detection limit, so the mask at each of the pixels in these structures is set to 1). Now S^1 is said to be connected to S^2 if the maximum position of the structure S^1 (except for the third coordinate; the channel) is contained in S^2. All the structures together that are connected (either upward or downward) can be called an object. Structure S^1 is already connected to S^2 if they simply overlap in velocity direction, if the channels are completely correlated.

   Now that an object is defined, it is possible to look at he maximum and minimum channel of all the pixels that are contained in the same object. By taking the difference, one knows the number of channels an object "spans" (in the channel direction). This way it is possible to select for certain kinds of objects. For instance, there is a physical lower limit to the dispersion of HI gas in galaxies of about 8 km/sec (Kamphuis (1993)). That means that if, for instance, the velocity difference between channels is 2 km/sec, then a physical astronomical object must span at least 4 channels. Also, isolated structures (objects that span only one channel) can be removed this way.

   It is also possible to search for objects in the wavelet transform space (WTS). This way objects are defined in each channel separately. This is useful if one wants to include structures in objects in wavelet planes only if they are connected. However, this way it is also possible to miss out on signal that is only prominently visible in one certain wavelet plane (see for instance figure 18). And that might be just the signal one is looking for.

9. The data cube and its new mask are written. Note that the data itself has not been changed. Only the mask has been set. So the whole procedure is easily reversed if necessary. The result of the complete procedure for *ngc 7332* can be seen in figure 20 and for *ugc 5253* it can be seen in figure 21.

7.3. Summary

The first thing that can be said is that the *à trous* wavelet transform is capable of extracting true astronomical features at all scales in the spatial domain. This can be done with the help of the False Discovery Rate. Because of its adaptive nature, it correctly controls the fraction of false discoveries over the total number of discoveries. To ensure that the right model for the background noise is indeed Gaussian white noise, the Kolmogorov-Smirnov one-sample test for goodness of fit proved to be helpful.

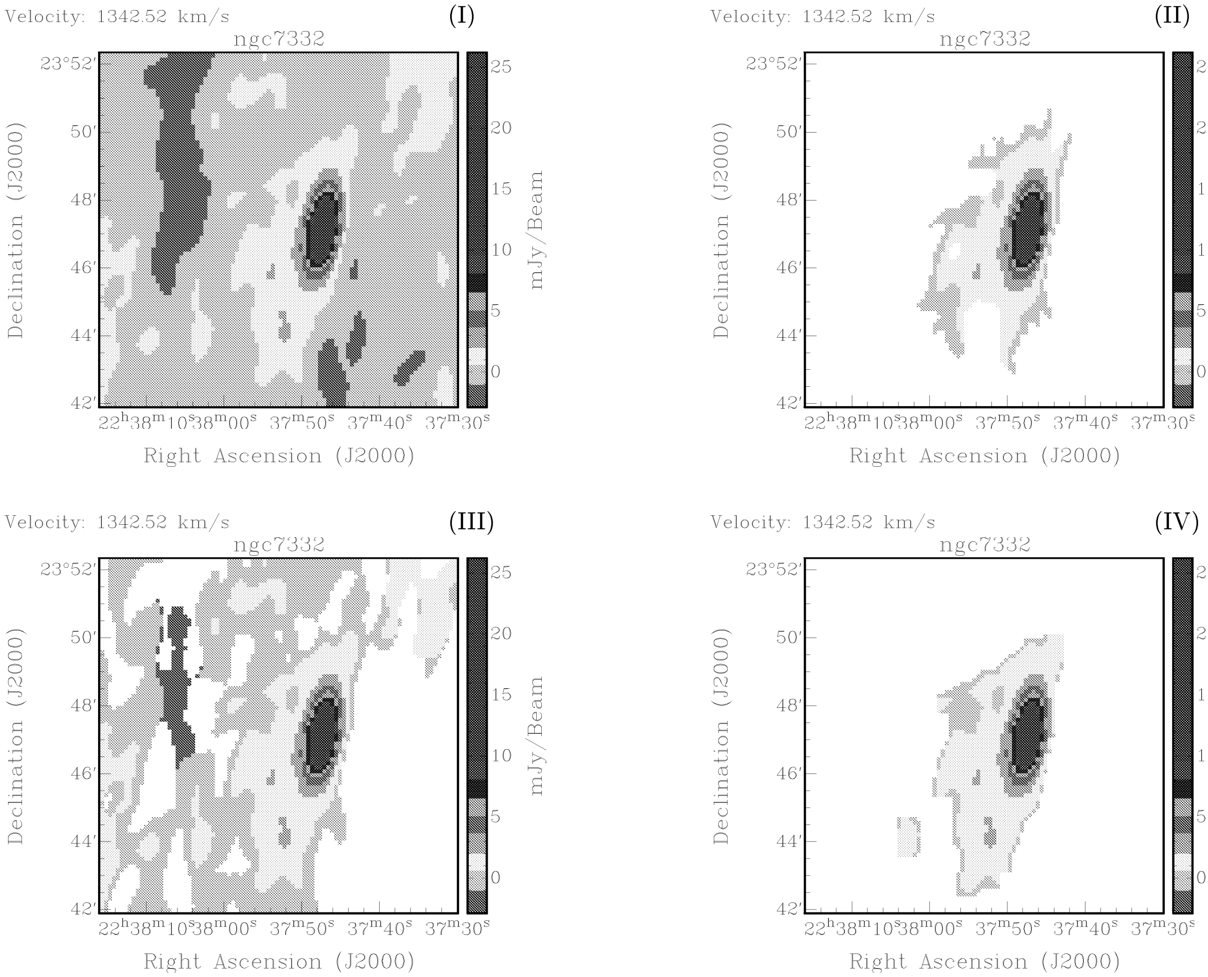

Fig. 20. Overview of a zoomed in part of the same channel of the same data cube as in figure 13. Image (I): the original data. Image (III): Applying the watershed-equivalent algorithm without the constraint (pixel values must be higher than a certain threshold when expanding the ask). Image (II): The watershed-equivalent algorithm applied with the constraint. Image (IV): the final result.

Second of all, the watershed-equivalent algorithm is a fast and easy way to improve the mask, i.e. include or exclude certain groups of pixels based on morphological or intrinsical properties.

Lastly, a first and simple comparison with conventional masking methods shows that the new procedure is capable of automatically extracting all the signal that probably is of true astronomical origin without including background noise peaks that are just above the threshold.

8. Filtering in the Visibility Domain

It is possible to define a mask (the flag table) for the u, v data in the same way as it is for the image data. Note that in this paper only the amplitude part of the visibility data is considered. Of course it is also possible (and perhaps much more effective) to consider both amplitude and phase u, v data, but this is the first step. Also using the phase data can be very effective since sometimes noise (or spurious signal) can be averaged out over time due to its varying phase.

This brings us to the next remark, namely that only the total intensity (Stokes parameter I) is taken into account at first. The frontend feeds of the WSRT consist of two perpendicular probes for the detection of linearly polarized radiation (a dipole cross) designated X and Y. By taking a look at the difference between XX and YY of each telescope combination (Stokes parameter Q), one should be able to spot the polarized RFI much more easily.

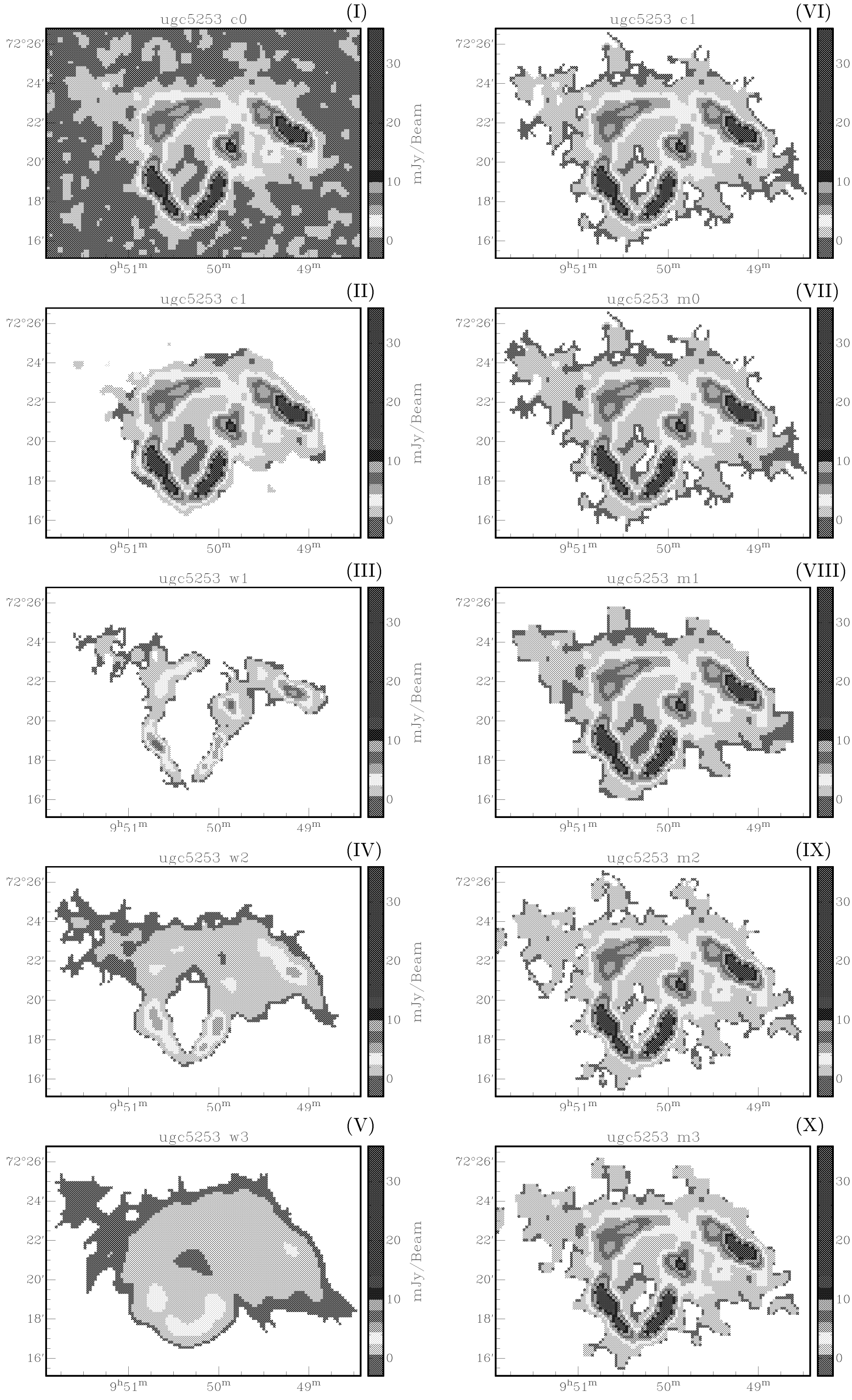

Fig. 21. Overview of a zoomed in part of the same channel of the data cube of *ugc 5253* as shown in figures 5 and 17 to 19. Image (I): original image c_0. Image (II): first smoothed cube c_1. Images (III) to (V): FDR clipping method and mask expansion with watershed equivalent algorithm applied to first, second, and third wavelet cube (w_1, w_2, w_3) respectively (and the data is the wavelet data itself, not the original data). Image (VI): same procedure applied to c_1. Images (VII) to (X): Combined mask of image (II), (III) and (VI) m_0, opening and closing of this mask m_1, another step of the watershed equivalent algorithm m_2, and again an opening and closing of this mask m_3 (which is the final mask). It can be argued whether image (X) shows better results than image (VIII), but the smoothness of the last mask can be adjusted by the size of the used structuring element for the opening and closing.

HI line emission (the 21 cm. line of the hydrogen spin flip) is unpolarized. In contrast, radio interference is mostly polarized. The scattered light from the sun is a good example of that.

8.1. Differences with image data

Compared with image data, there are three big differences. First of all, when filtering in the visibility domain, the filtering is the same for all pixels in the image data. This means that if one pixel in a u, v data plane is excluded, this affects *all* pixels in the image data plane (because it is the Fourier transform). So if RFI can be removed at a certain position in an u, v data plane, then the complete image plane can be improved at once. RFI is a good example of spurious signal.

The second big difference between filtering in visibility space and filtering in image space is that distinguishable objects look different. In the image domain one looks for true astronomical signal, while in the visibility domain it is much easier to look for RFI. This is due to the discontinuous character of RFI. As can be seen in figure 1 its signature is that it is narrow in ν,t space (right panel) and broad in b space (sometimes RFI is random in ν,t space). If an image in the u, v cube is smoothed in all directions but the time axis, then the signal-to-noise ratio for all structures of short duration will be increased. So typical RFI of short duration can be selected this way. The same method can be applied for interference at a certain frequency that is continuously present.

The third and most important difference is the following. In the image domain, techniques have been applied to separate the true astronomical signal from the noise. This means that the signal for every pixel itself has not been changed. However, if one blanks pixels in the visibility domain then the value of the pixels in the image obtained by the Fourier transform *will* be affected. This means that, if done right, flagging pixels in the visibility domain can improve the complete resulting image.

8.1.1. Selecting a cutoff

The first step in extracting spurious signal from u, v data is calculating some kind of threshold. For the image data, the False Discovery Rate calculated this threshold. For the u, v data, however, this is impossible. When considering u, v data, it is impossible to truly select any pixel as source pixel or as background pixel. The definition of the FDR is 'the average fraction of false discoveries over the total number of discoveries'. But in the case of u, v data, there isn't truly such thing as a false discovery. Every pixel in the u, v data contains *both noise and signal.* Therefore it is important to be very careful when selecting signal. The second problem with FDR is that the p-values can only be calculated with a uniform Gaussian model for the background (see equation 11). In u, v data, the background is not uniform (see section 8.1.2). The choice made here is therefore to calculate an average standard deviation for every pixel in the complete data cube. The idea is then to first select only a small part of the strong signal (in the case of u, v data this should be RFI) by using a very high threshold and then to define the spurious signal more decently with the use of morphological operators.

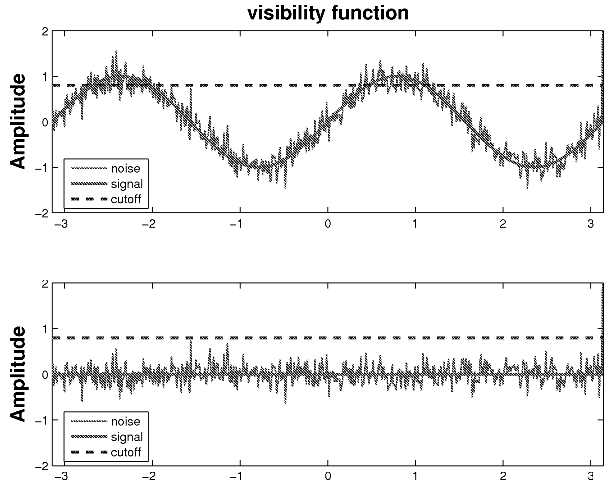

Fig. 23. The visibility function for a certain telescope spacing. For a point source, this will be a sinusoidal function. The upper plot shows that, because of the varying amplitude, pixels (green, thin line) should not be tested against the same threshold (blue, dotted line) over time. Instead, the running mean over a short period of time (red, thick line) should be subtracted first (lower plot).

8.1.2. Visibility functions

All the telescopes in a synthesis array are pointed towards a phase center. Due to the varying direction of this phase center, a correlator will give a sinusoidal output in the case of a perfect point source. The visibility function of two antennae is, for a point source, a sinus, with a period that is inversely proportional to the baseline length. So for smaller baselines, the amplitude of the interferometer output will vary more slowly. For extended sources, the visibility function can have a complex shape. It will be a time-varying function that depends on the shape of the source.

Because of this effect of varying amplitude over time, it is not wise to select a threshold at the u, v data directly. This can be seen in figure 23. For a certain threshold, data at times when the visibility function peaks are more easily selected then data at other times (in other words, the data cubes c_i pick up by far too much signal, see upper figure). This can also be seen in figure 22, which shows the data of one baseline in the first smoothed cube c_1. When the output is corrected for these visibilities (lower figure), only the truly outstanding pixels will be selected.

One way to correct for the visibilities is to look at the average value of the pixels over all frequencies and over a

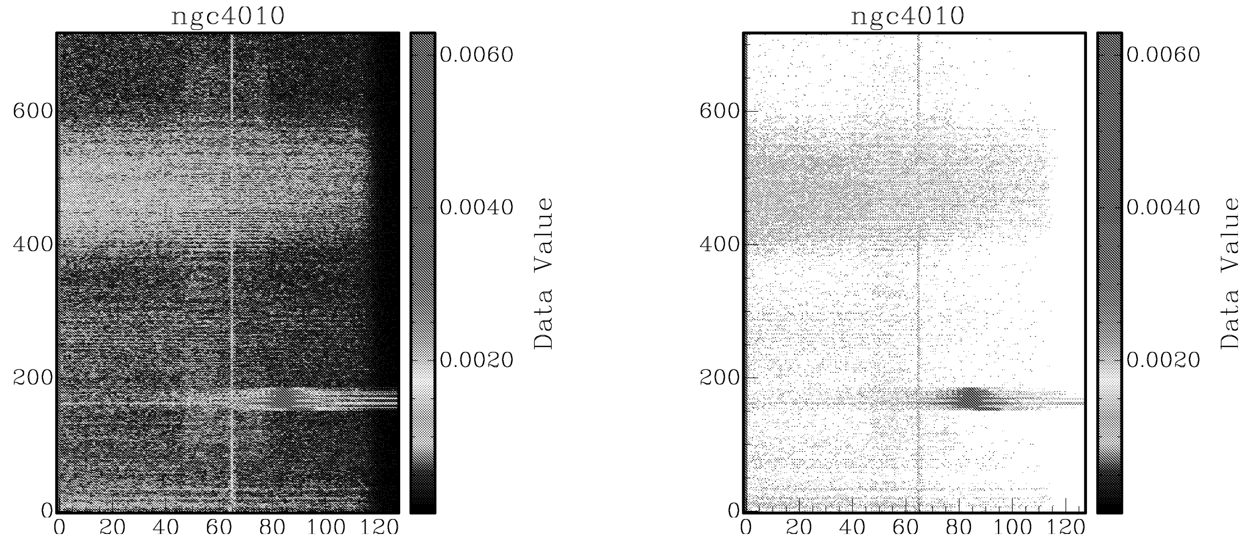

Fig. 22. The first smoothed u,v data (c_1) of a very short baseline (of antennae 9 and A of the WSRT) of *ngc 4010*. The horizontal axis represents the different frequencies (in total 128), the vertical axis represents the time (12 hours, with 60 seconds per sample, giving 720 time samples in total). The third dimension holds the 91 baselines. This will be the same for following figures of u, v data planes, unless stated otherwise. The left panel shows the unmasked map. The red blob is a small cube of really strong RFI that is visible on most baselines (at least on all the short ones, which is characteristic for RFI). The right panel shows the masked map, meaning that all pixels below a certain threshold have been flagged as bad. (5σ). It can be seen that the amplitude averaged over all frequencies varies very slowly over time for this short baseline. At those moments where it is highest, the pixel values can easily get higher than the threshold (right panel). This should be corrected for by subtracting the averaged amplitude (over time) before setting a threshold (this is graphically explained in figure 23).

short period of time (say ten minutes) for each baseline individually, and then to subtract this average. Another (and very convenient) way of doing this is to only take a look at the wavelet planes and not the u, v data directly (data cubes c_0, c_1, ...). So only the wavelet planes (cubes w_1, w_2, ...) will be used to set the mask of c_0.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

-1	2	-1
-1	2	-1
-1	2	-1

-1	-1	-1
2	2	2
-1	-1	-1

Table 3. Example of a mask that can be used to find strong lines in a plane. The parameter $R = \sum_{i=1}^{9} w_i u_i$ (where w_i are the weights of the mask (upper left figure) and u_i are the values in the plane centered on w_5) is calculated for every pixel in the plane. Next a pixel is though of as part of a line (in the case of the upper right mask in the vertical direction; in the case of the lower right mask in the horizontal direction) if $R \leq T$ where T is a certain threshold (Gonzalez & Woods, 2002).

8.1.3. Defining spurious signal

There are three kinds of strong 'objects' in u, v data that are most prominently present and that can be pointed out as spurious signal (RFI) with high probability.

1. Rather isotropic blocks of very strong data (so prominently visible in baseline, frequency and time space), that are best visible at short baselines and that usually last for a short period of time (like a half an hour). The interference that arises due to scattered light from the sun is a typical example of this kind of RFI.
2. A plane of very strong data that peaks in the frequency axis. Again, the short baselines are most sensitive to this kind of RFI because of their very low fringe rate. At the beginning and end of the measurement, the longer baselines appear shorter to the source (since it is closer to the horizon). Everywhere in between, the phase changes so rapidly for the longer baselines, that the RFI is averaged out.
3. A variant of the previous kind of RFI, this time only visible in all baselines with one certain antenna. The shape is the same, but instead of a real plane it is more a collection of strong lines that last for a certain amount of time.

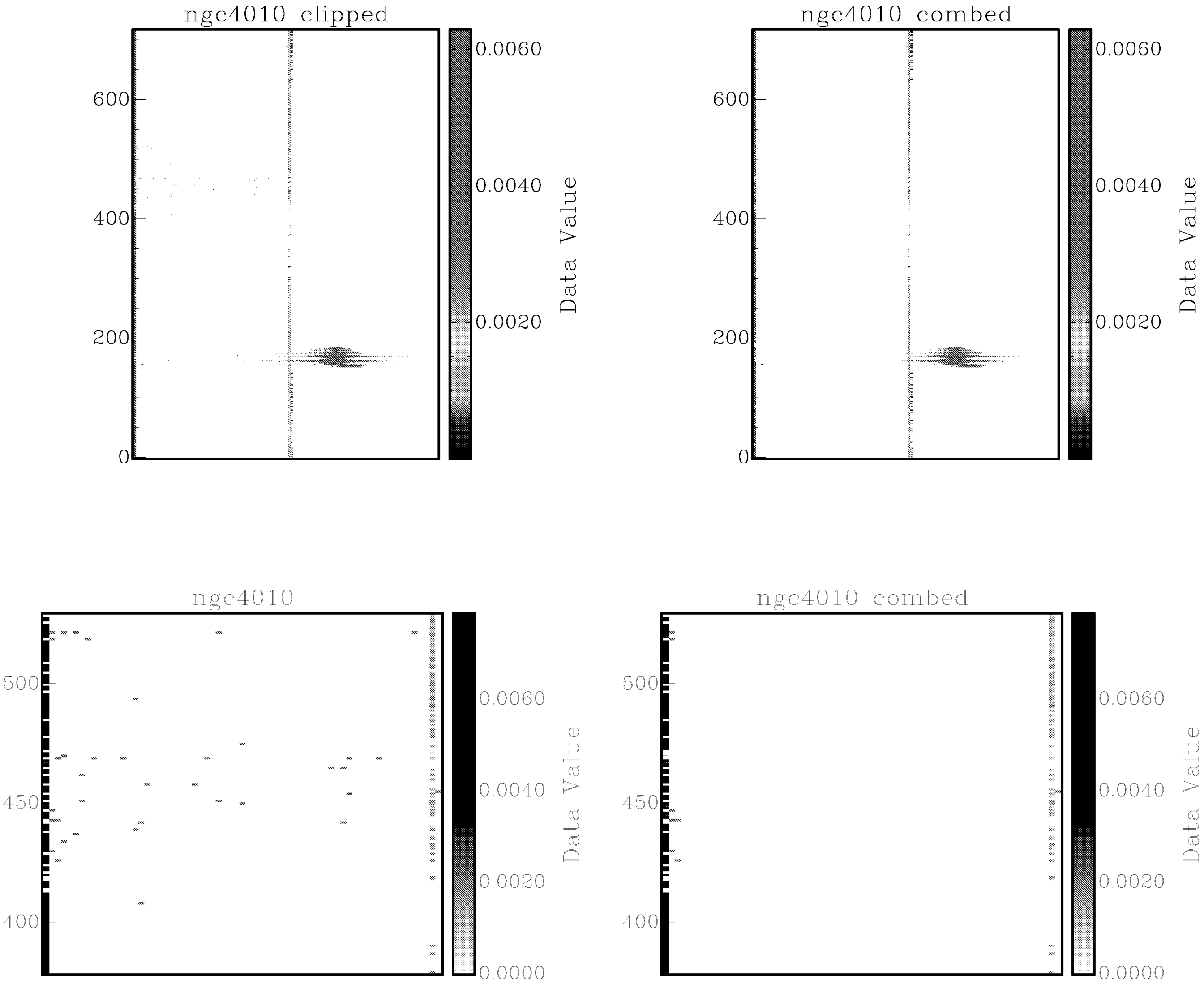

Fig. 24. The same baseline plane as in figure 22. The mask has first been set by selecting all the pixels in the wavelet planes (two) with values above a certain clipping level. From this (upper left panel) it can be seen that mostly only spurious signal has been selected. The greenish stripe in both upper panels has been selected because of the wavelet transform: the baselines after this baseline (the next longer baselines) have strong RFI peaks at this frequency. Upper panels: the left panel shows a few loose points that are most likely real signal, but have also been selected. Note that the first few channels contain very high pixel values. The right panel is the result of combing the mask (i.e. removing small groups of pixels not larger than 10 pixels) so that only the pixels that are very likely to be RFI have been selected. Lower panels: these show a zoomed-in part of the baseline plane.

8.2. Mask enhancements

There are several ways of improving the flag table of a u, v data cube, just as with image data. In some case the same algorithm can be used, but with different parameters. The biggest difference is that the shape of the spurious signal one is looking for is different. So some different algorithms are needed also.

8.2.1. Selecting strong lines

In order to find strong lines in a certain frequency plane of the u, v data, a new algorithm is needed. The algorithm presented here makes use of the fact that strong lines are lines, because the two adjacent rows in the plane are significantly weaker. Thus, by looking at the slope of the lines perpendicular to these strong lines, a sudden increase and decrease can be noted. This is the same thing as saying that the derivative at the line is very high. For this, a one dimensional Laplacian filter can be used as the two in table 3 (on the right).

The length of the mask can also be varied. This means that instead of only three, five or seven or even more rows can be used. The advantage of this is that longer lines with gaps in them will be selected completely. The disadvantage is that one really strong pixel can make all the adjacent pixels above and below be selected, even though they should not be.

8.2.2. Region growing

In a typical WSRT antennae layout, there are 9 baselines with length 144 meters and 8 baselines with length 288 meters, and so on. This means that the sun will leave blocks of RFI in the u, v data, as can be seen in figure 29. Due to the characteristic shape of these blocks (which are clearly visible at all frequencies), the second wavelet cube does not pick up enough RFI. The third wavelet cube does so, but it also selects too much pixels of which it is impossible to say that they are RFI. Therefore the following algorithm is implemented.

The concept of region growing was explained in section 6.1.2. In the case of u, v data, region growing can be used to expand the mask that is already set to remove gaps and bridges in the mask that arise due to the use of the *à trous* wavelet transform (section 5.3). The algorithm works as follows,

- The set of seed points is chosen as the complete mask of a previous step. This way the region itself can only grow, it never becomes smaller (as opposed to the watershed-equivalent algorithm in section 6.1.2).
- As long as the next neighboring pixel in the *baseline direction* (both ways) has a value higher than some chosen threshold, it will be included in the mask. This threshold must be high, like 6 or 7 σ. Expanding in the baseline direction only (as is done in the wavelet transform) makes this algorithm sensitive for typical RFI-like structures.

This algorithm will find strong signal, but only in the neighborhood of signal that was already selected by previous masking steps.

9. Implementation for the Visibility domain

The implementation of the different algorithms and the order in which they are executed on the u, v data is discussed here. As with the spatial data, first an overview of the complete implementation is given. Next, all the different parts of the implementation are explained in more detail, giving examples of the results and the reasons why certain choices are made.

9.1. Overview

The full implementation looks like this (only for visibility domain data cubes):

1. The mask of the original data cube, c_0, is reset.
2. The baseline planes are ordered from smallest to largest baseline.
3. The *à trous* wavelet transform is done.
4. A mask is set for all of the wavelet cubes by clipping all values below a certain averaged clipping level.
5. The masks of the wavelet cubes are combined.
6. Real small objects are removed from the mask.
7. All the strong lines in each frequency plane (channel) are selected.
8. The mask is expanded with a region growing algorithm.
9. The mask is inverted.
10. Gaps in the mask are filled.
11. The baseline planes are reordered and the data cube and its new mask are written.

9.2. Explanation

1. The mask of the original data cube is reset. This means that all that is done next, is easily traceable. It is also possible of course to add the flagging done by this procedure to that of another, perhaps interactive flagging procedure.

2. The baseline planes are ordered from smallest to largest baseline. This way it is easier to detect structures in the data cube. The length of a baseline corresponds to the size of the structure the interferometer is sensitive to. Large, extended structures will show up more prominently on the shorter baselines.

3. The *à trous* wavelet transform is done. This time it is done only in 1 direction, not isotropic. As said before, the signature of RFI is (mostly) that it is narrow in ν,t space and broad in b space. Therefore it is enough to smooth the data in just the baseline direction. This is to select as less (bad) data as possible. For the same reason, it is usually enough to set the number of wavelet cubes that are created to two.

   If the *à trous* transform *is* done isotropically, then the worst results arise when RFI at one single frequency is

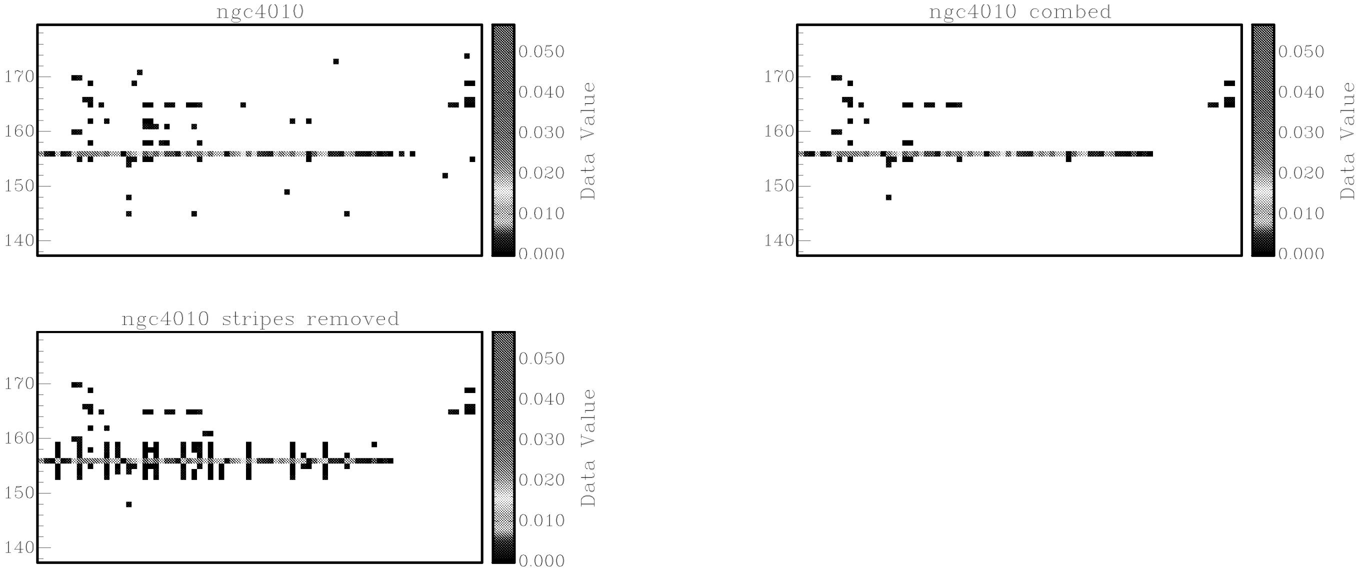

Fig. 25. A part of a frequency plane of *ngc 4010* (baselines on horizontal axis with smallest baselines left; time on vertical axis in minutes, every 60 seconds samples where taken). The upper left panel shows the mask that has been selected by the wavelet transform procedure. The upper right panel shows the result of combing this last mask (in three dimensions) for groups smaller than 10 pixels. The lower left panel shows the result of selecting strong stripes in the new mask, using a filter mask (such as the two right masks in table 3) of 7 rows. It is clearly visible that this last procedure can create unwanted features of that same size of 7 rows around strong, isolated pixels.

present. The data will be smoothed over the adjacent frequency planes, so in the original data good signal will be identified as RFI.

4. A mask is set for all of the wavelet cubes by clipping all values below a certain averaged clipping level. This average is created as follows. First all the sigmas (standard deviations) for all channels are calculated. Then an average sigma is formed including all channels except the first and last 5 (these often contain very bad data). Next an average is formed over all channels that have sigmas below the first average (to exclude peaks). This way one channel with real high values cannot increase the average sigma too much. The idea is to take a low and save threshold and to expand the RFI selection (flagged as bad) more carefully.

   In contrast to the image data, strong negative signal is also included. As is described in section 5.3, this leads to selecting too much signal in image data, for a three-dimensional wavelet transform. This negative side effect is not visible for the one dimensional transform.

5. The masks of the wavelet cubes are combined. This mask is the basis of the mask for the real data, c_0. The result so far can be seen in the left panels of figure 24.

6. Real small objects (in three dimensions) are removed from the mask. When handling u, v data, it is very hard (or even impossible) to classify a single pixel. Pixels need to be classified depending on the nature of there neighbors. Because of this, it is best not to include single strong pixels (or real small groups of ten or twenty pixels) in the mask. The obvious result of this step can be seen in the right panels of figure 24.

   Note that for the u, v data, no opening and closing is performed. This is because these operations will include too less or too much signal and the u, v data should be treated with more care.

7. All the strong lines in each frequency plane (channel) are selected. This is done with the algorithm described in section 8.2.1. The number of rows in the mask is chosen as seven, since this gives best results. As described, the use of this algorithm comes with a price. This can be seen in figure 25. More importantly, this algorithm also selects signal of which it is certain that it is RFI. This can be seen in figure 27.

8. The mask is expanded with the region growing algorithm described in section 8.2.2. The result can be seen in figure 29 (lower left panel). The combination of the *à trous* wavelet transform and this algorithm is very powerful. The first can spot extended sources at all scales, the latter fills the gaps and creates bridges in the mask that arise due to the creation of negative signal in the wavelet planes.

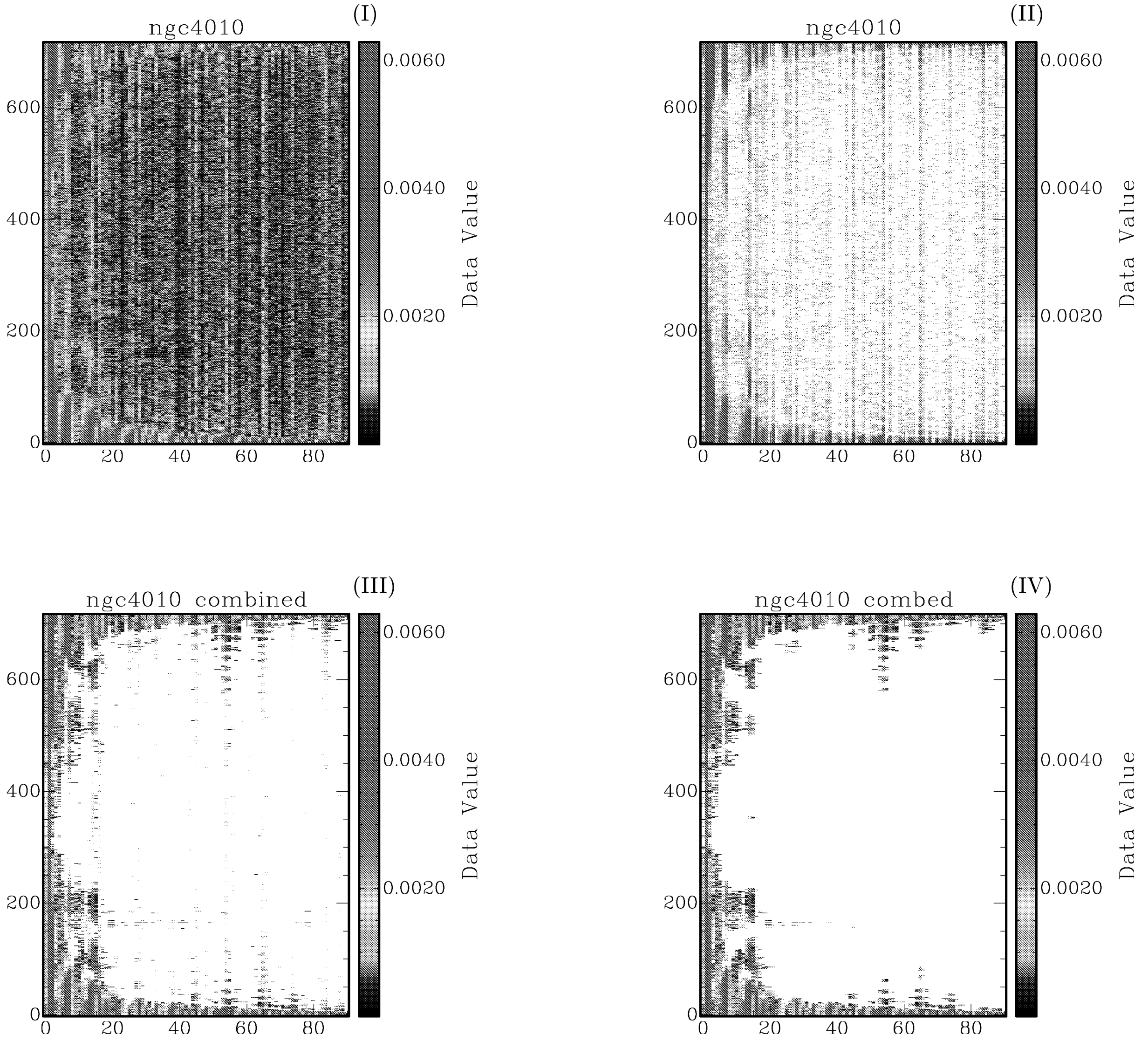

Fig. 26. A frequency plane of *ngc 4010* (baselines on horizontal axis with smallest baselines left; time on vertical axis). Image (I) shows the original frequency plane. The red pixels are so strong that they must be RFI. Also, the shape of this 'object' is very characteristic for RFI. The short baselines have a very low fringe rate, so that the RFI is visible. At the beginning and end of the measurement, the longer baselines appear shorter to the source (since it is closer to the horizon). Everywhere in between, the phase changes so rapidly for the longer baselines, that the RFI is averaged out. Image (II) shows the result of setting a threshold for the original data, which clearly selects by far too many pixels. Image (III) shows the result of adding all masks created by the wavelet planes. The is the basis for the new mask. Image (IV) shows the result up until the combing step of the whole procedure.

As described in section 9.2, it is usually enough to create only two wavelet cubes for the u, v data, because the masking procedure applied to the third wavelet cube includes too much real signal. The third wavelet cube will filter more RFI though. The difference between using this third wavelet cube and using only two wavelet cubes and applying the region growing algorithm can be seen in figure 30.

This figure shows that even when only two wavelet planes are created and the region growing algorithm is not applied, part of the true source (the rings in the image) is also removed (upper right panel). This is logical, since (in this case, with uncalibrated data) the shortest few baselines show interference that last for the complete measurement time of 12 hours for this particular frequency. So removing even part of this RFI in the u, v data will also remove part of the true source in the resulting image data. In theory one

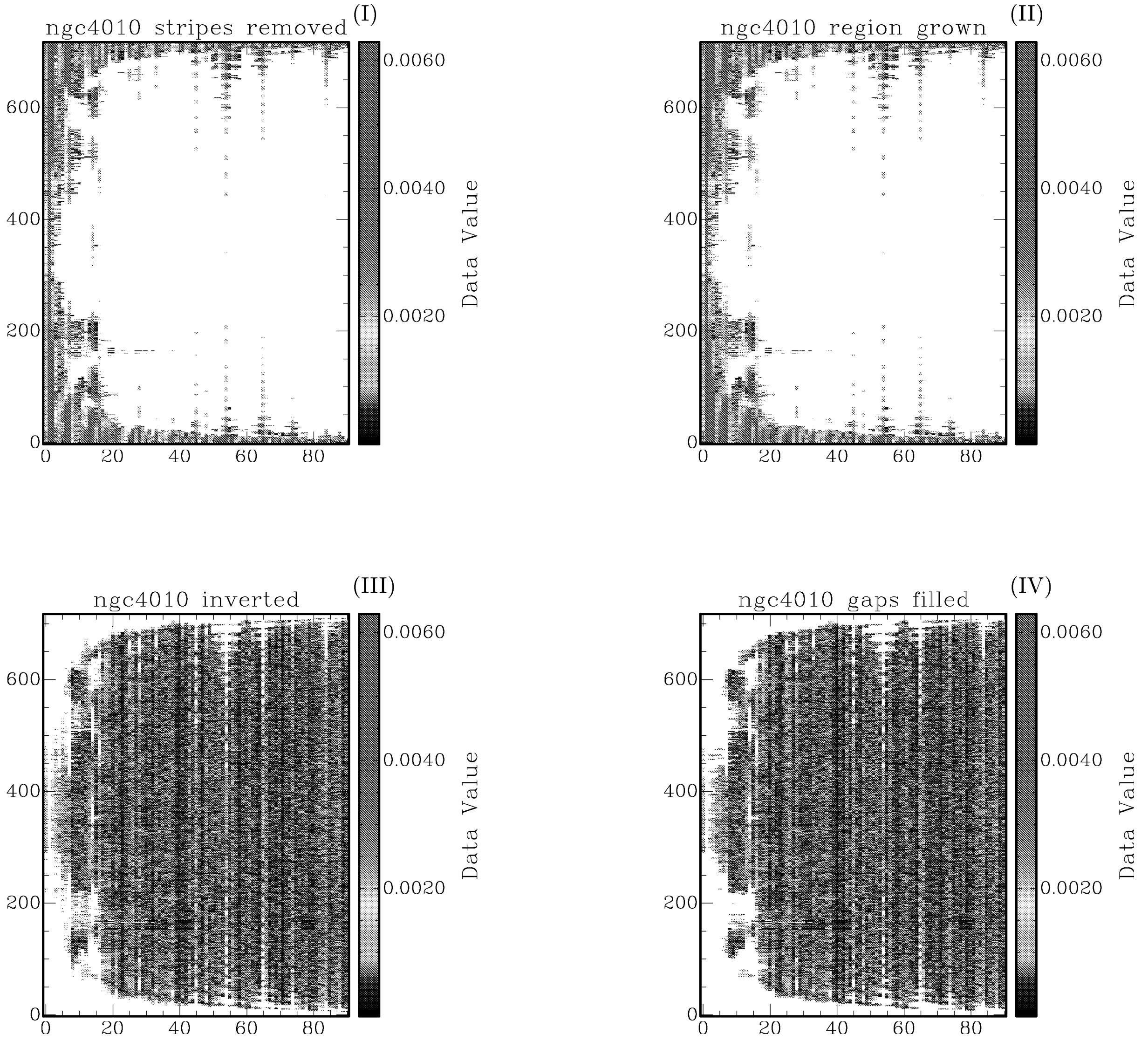

Fig. 27. The same frequency plane of *ngc 4010* as the one in figure 26. Image (I) shows the result of selecting strong stripes in the new mask, using a filter mask (such as the two right masks in table 3) of 7 rows. It is clearly visible that this last procedure can select some strong stripes in the frequency planes, that are clearly RFI. Image (II) shows the result of expanding the mask with the region growing algorithm described in section 8.2.2. Next, the mask is inverted (Image (III)). There are still some 'floating' strong pixels left in the part of the mask that has been flagged as bad. These can be removed (see section 9.2), to get the final mask that can be seen in image (IV).

would only want to remove the RFI from the 'bad' data and leave the pixels flagged as good. But it is impossible to know what part of a data value is RFI and what part is signal.

The most important result from this figure is the following. The third wavelet cube was able to remove the last parts of the horizontal stripes that are visible in the image data. However, it also selects even more true signal. The region growing algorithm removes *only* extra RFI.

9. The created mask selected as much of the visible RFI as possible. Now the mask is inverted (all the flags unequal to zero are set to zero and visa versa), because in the end, we do not want to include RFI in the mask. The resulting mask can be seen in the lower left panel of figure 27.

10. There still is some strong signal in the mask left, that is easily identified as RFI. Therefore all the

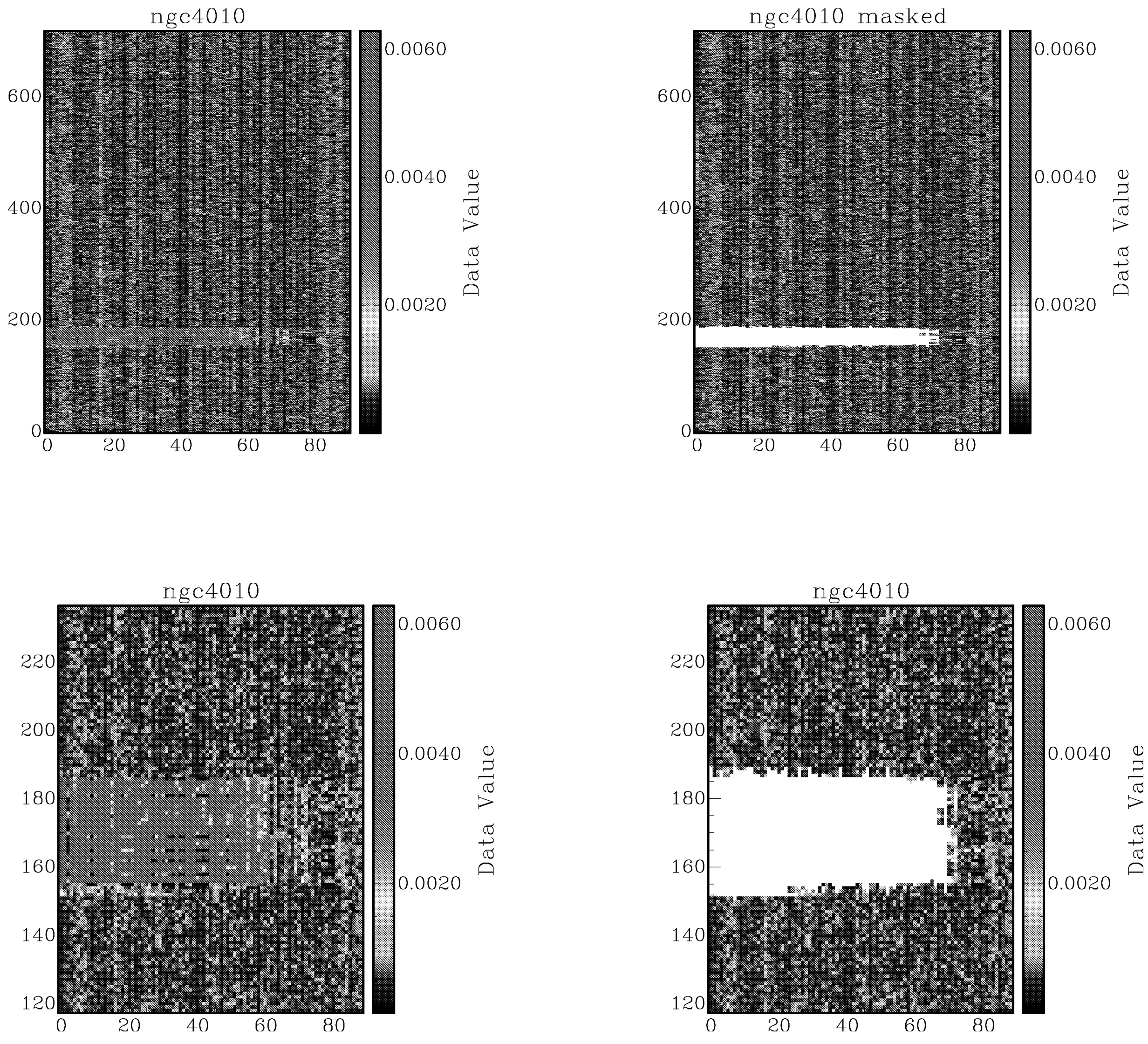

Fig. 28. A frequency plane of *ngc 4010*. The left images show the original data. The right images show the result of all the masking procedures as described in section 9.1. The lower images show only a part of all the time samples for more detail.

gaps in the mask are filled. This means the following. For every pixel in the mask, the mask values of the surrounding eight pixels in the frequency plane are added up. If this value is lower than 3 (so at most 2 adjacent pixels have been flagged as good), then the mask of the centered pixel is flagged as bad. This removes single isolated unflagged pixels from the mask as well as stripes that are surrounded by bad pixels. The result of this last step can be seen in figure 27.

11. The baseline planes are reordered, to make up for the second step. This is to make sure that the right data is at the right place again. Next, the data cube and its new mask are written. Note that the actual data itself has not been changed. Now images can be made from the visibilities (the u, v data), by the Fourier transform (using all the right weights, gains, etc.). The corresponding image of the frequency plane in figure 26 (upper left panel) can be seen in the upper left panel of figure 32. The corresponding image of that same frequency plane after the flagging has been done (lower right panel of figure 27), can be seen in the right panel. The improvement is obvious.

The strong stripes in the unflagged image must be due to errors in the early beginning, or in the end of the measurement, since they are very much vertically aligned. This can also be seen in the visibility domain, which shows even more clearly where the RFI is

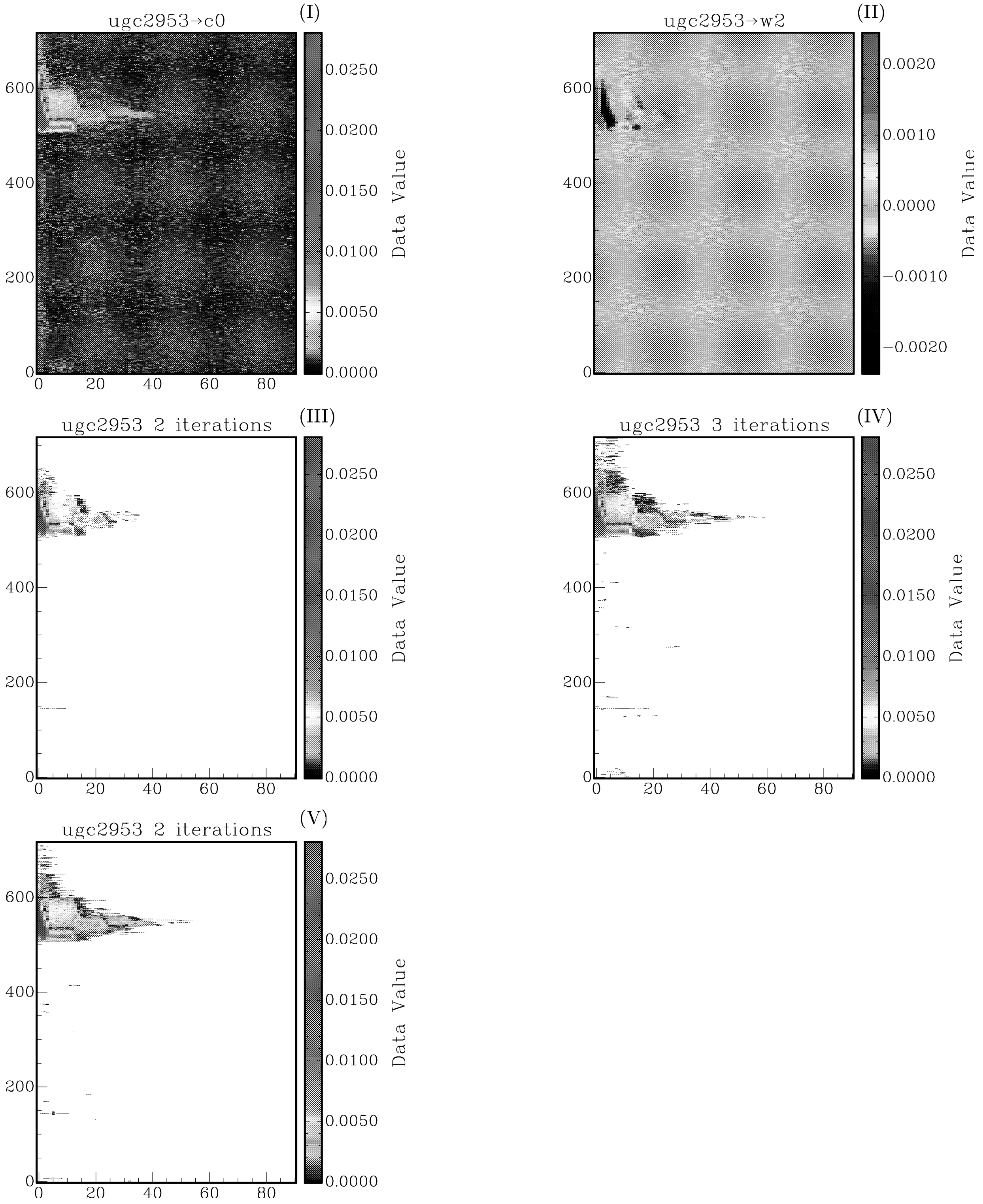

Fig. 29. A frequency plane of *ugc 2953*. Image (I) is the original data, which shows some clear blocks of RFI from the sun. These blocks are characteristic for an interferometer with many redundant baselines (like the WSRT). Image (II) shows the second wavelet cube. It has strong positive, as well as strong negative values. Due to the structure of the blocks of RFI, not all of it will be visible in the *à trous* wavelet space. This can be seen in image (III) which shows the data after the clipping procedure. Image (IV) shows that the third wavelet cube is able to pick up much more of the RFI at the cost of removing too much good signal (which in this case is visible as dark blue pixels). Some gaps in the blocks of RFI still remain. The mask in image (III) can be extended by the region growing algorithm in the baseline direction (as described in section 8.2.2). For a used clipping level of 7σ this results in image (IV). Almost all of the RFI that is visible with the eye has been selected and also much less good signal is selected in comparison with the mask set for the third wavelet plane (image (IV)).

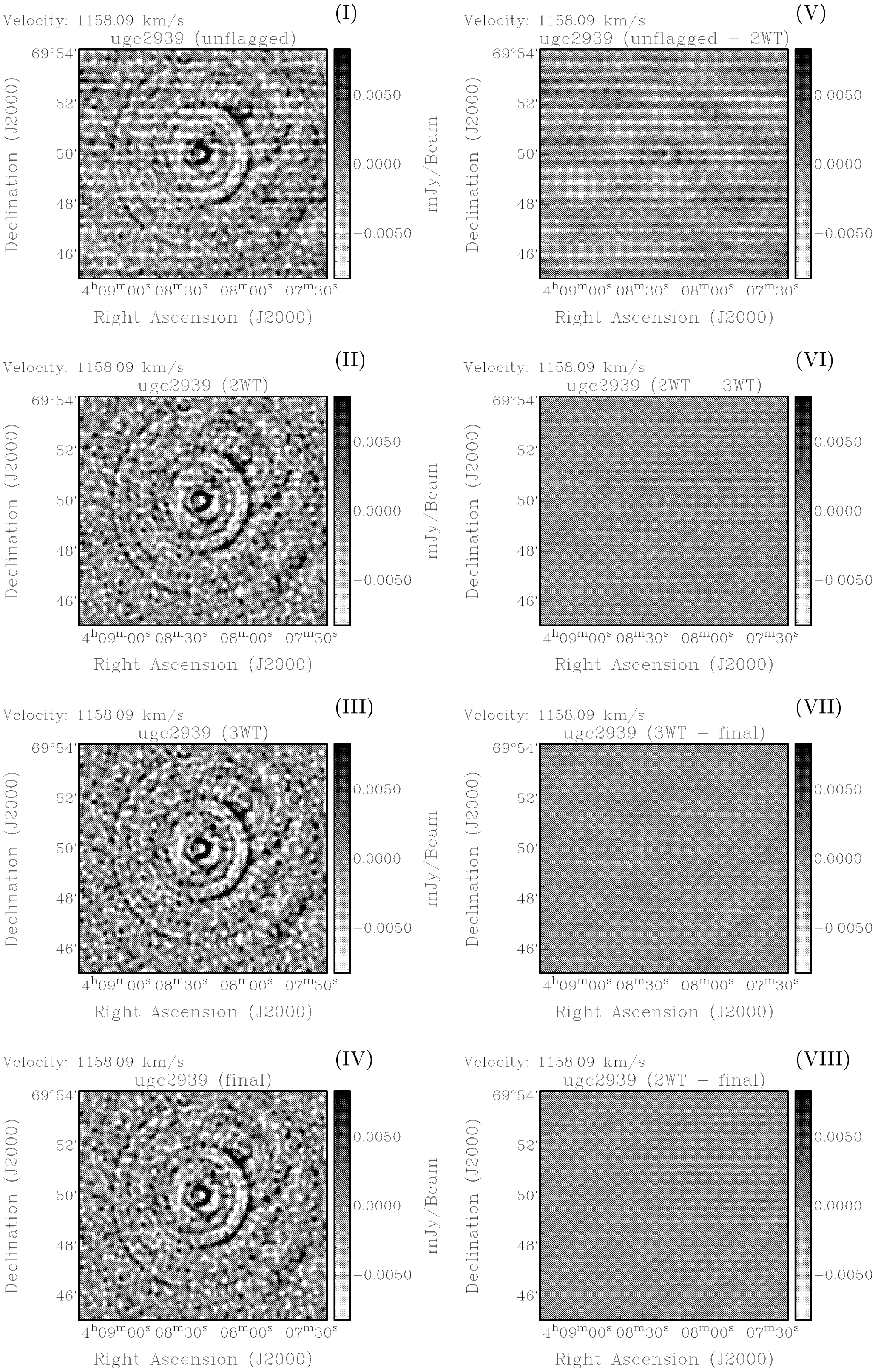

Fig. 30. A zoomed in part of a spatial image of a data cube of *ugc 2953* (the complete channel can be seen in figure 33). Image (I) shows the original data. Image (II) shows the result of the complete flagging procedure as described in section 9.1 except for the region growing step, with only 2 iterations of the wavelet transform and the same holds for Image (III) with 3 iterations. Image (III) clearly shows less horizontal stripes than image (II). Image (IV) shows the result of the complete flagging procedure, including the region growing step. Image (V) shows the difference of images (I) minus (II). Most of the RFI has been removed, including part of the true source. Some horizontal stripes remain however, which can be seen in image (VI) which shows image (II) minus (III). This image also shows some rings which means that more of the true source has also been flagged when creating three wavelet cubes instead of only two. Image (VII) shows the difference (III) minus (IV).Image (VIII) shows the difference (II) minus (IV), which shows that the extra flagging done in comparison to image (II) removed practically only extra RFI, not signal.

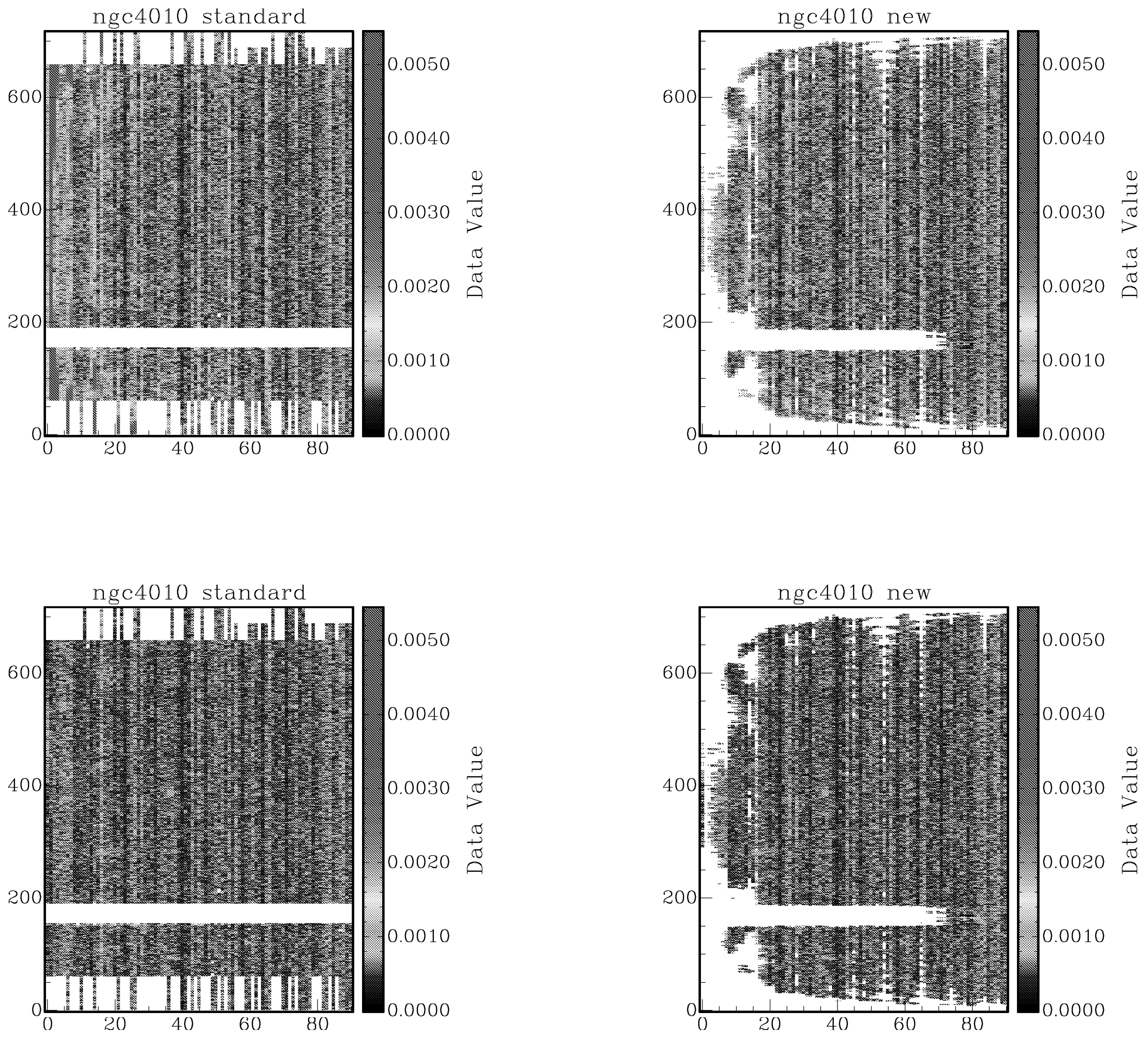

Fig. 31. The upper images show the same channel of the data cube of *ngc 4010* as in figure 26. The lower images show the same channel of the data cube of *ngc 4010* as in figure 28. The left images show a standard masks. The right images show the masks created with the new procedure.

strongest.

Another example of the result of the complete procedure can be seen in figures 28 and 32. The first shows the u, v data before flagging (left panels) and after flagging. This is a clump of RFI that lasts for about a half an hour, and which is visible at more than sixty baselines. By performing the wavelet transform in just the baseline direction (horizontally), the resulting mask is very tight (right panels). The second figure (lower panels) shows the corresponding image at that frequency.

There is one last step that could be included in the whole procedure, namely giving the same u, v coverage to every frequency. In some cases this is important when the continuum must be subtracted off the data. The way to do this is saying that if at a certain baseline and time at any of the frequencies the pixel is flagged as bad, all the frequencies at that point must be flagged as bad. Of course, this means that too much true signal will be removed. A comparison between a “standard” method and the new method can be seen in figure 31, where this last step *has* been performed.

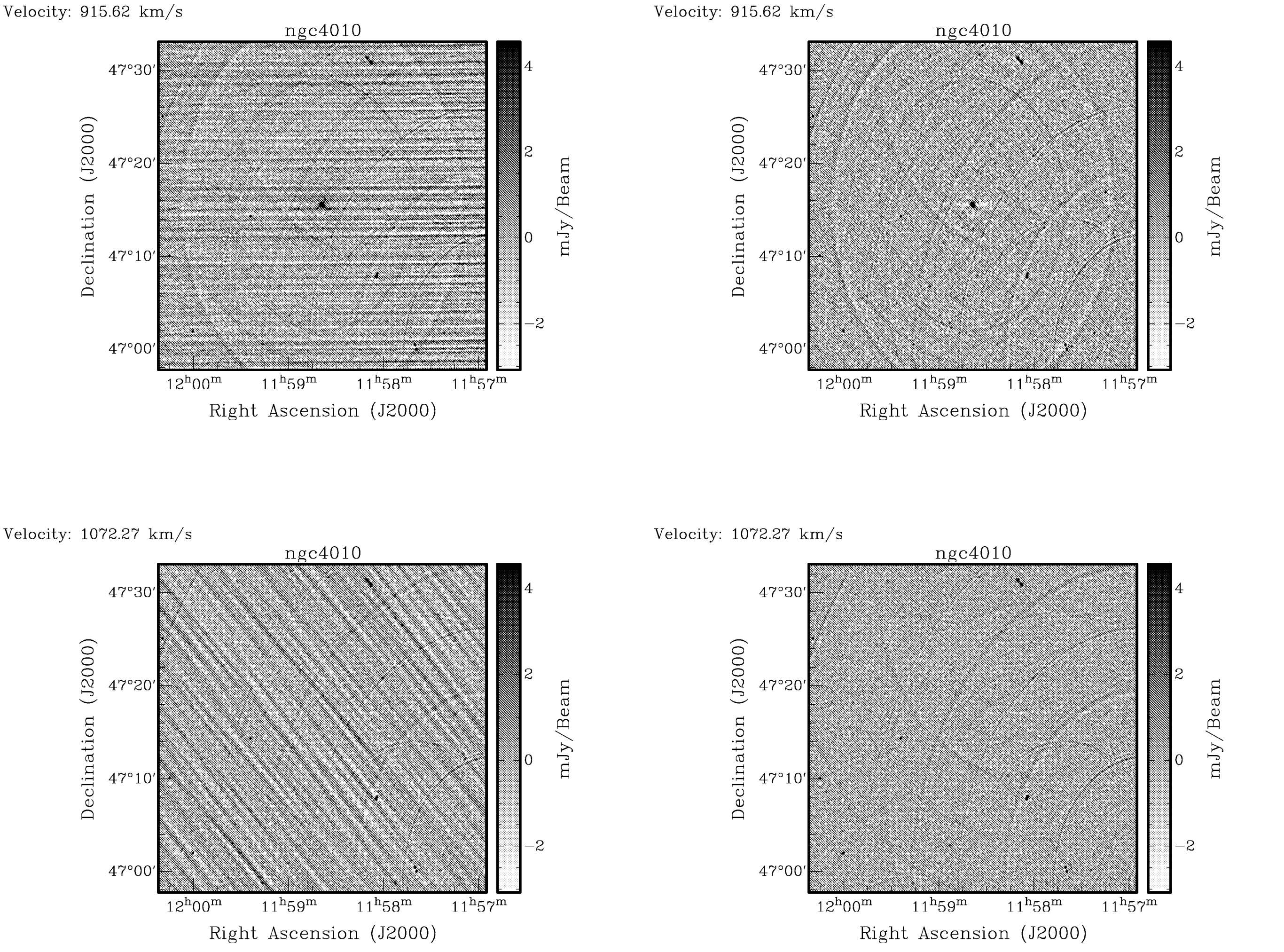

Fig. 32. Two spatial images (channels) of a data cube of *ngc 4010*. The upper images correspond to the frequency channel of figure 26. In the left images, all the flags have been set to good (so the RFI has been included). The strong stripes in the upper left unflagged image must be due to errors in the early beginning, or in the end of the measurement since they are very much vertically aligned. The upper right image shows the result after flagging the RFI as bad data (as in image (IV) of figure 27). The lower images correspond to the frequency channel of figure 28. Again, the strong stripes indicate RFI of short duration. The lower right image shows the result after flagging the RFI as bad data (corresponding to the right images of figure 28).

10. Results

If the final result for the data of *ugc 5252* in figure 21 (lower right panel) is compared to the unflagged image in figure 5 then it is clearly visible that the new procedure is capable of selecting all the weak signal around the stronger signal that was originally selected by the FDR method. The watershed-equivalent algorithm plays a large role in this, as well as the *à trous* wavelet transform which detects signal at any scale. This weaker signal can be as low as the average background noise level in each channel. The False Discovery Rate guarantees high power in relation to the number of false discoveries. So the combination of these two algorithms can find the borders (of source and background) of the true sources. This is also very well visible in the final result for the data cube of *ngc 7332*, as can be seen in figure 20.

The final result for *ngc 4010* as can be seen in figures 27 and 28 shows that the new procedure also works well on u, v data. Again, the *à trous* wavelet transform is a helpful tool for signal extraction, because it is good at detecting discontinuities. In this domain a region growing algorithm proves to be very effective too, because both algorithms make use of the characteristic shape of RFI. Also see figure 29 for the final mask for the u, v data for *ugc 2953*.

The resulting images for *ngc 4010* are created by convolving the u, v data (i.e. doing the Fourier transform) with the right weights. For two of the channels the result can be seen in figure 32. The characteristic stripes of RFI of short duration have been removed. The final result for three channels of *ugc 2953* can be seen in figure 33. Both RFI that peaks at a single frequency as well as RFI that spans a range of baselines (like RFI from the sun) have been removed successfully.

Appendix A shows another overview of the resulting images of *ngc 4010*. Figure A.1 shows a channel that is hardly not affected by RFI at all. A standard flagging procedure for u, v data (see section 9.2) removes too much good signal, which can be seen in the middle right panel. The new flagging method hardly flags anything at all for this channel, as it should, which can be seen in the lower right panel. Figure A.2 Shows a channel that is very badly affected by RFI of short duration. The standard flagging method creates an image in which a two dimensional sinusoidal landscape remains. This is also removed by the new procedure (lower left panel). Sometimes the difference between the standard and new flagging procedures is very minimal, as can be seen in figure A.3. Figure A.4 shows a channel in which the standard flagging procedure makes the RFI even more pronounced.

The standard way of masking extended image data is the following. First, the data is smoothed to a lower resolution with a convolution by a Gaussian. The Full Width Half Maximum (FWHM) of the Gaussian is chosen as double the FWHM of the original data. Next, a threshold for the smoothed data is chosen *by hand* that includes as much true signal (or what appears to be true signal) as possible without including noise peaks. This threshold is used for the original data. When data cubes become very large, there is not enough time to inspect each channel separately. Therefore, one threshold can be set for the whole data cube. For the same channel of *ugc 5253* as in figure 5, the result of this step can be seen in figure 34. The upper right panel shows that a very tight mask misses out on the weaker extended signal that is very pronounced in the smoothed data. The lower right panel shows that a lower threshold easily picks up background noise peaks.

The next step is to create a zeroth moment map (a "total HI map"). This is the first moment of the data with respect to the velocity axis, i.e. the total intensity as a function of position on the sky. For *ugc 5253*, the comparison between creating a zeroth moment map with the standard masking method and the new method can be seen in figure 35. It shows that the map created with the standard method and high threshold selects the signal very tightly. It shows no apparent background noise peaks, but the chance of missing out on some of the true signal is large. The map created with the standard method and low threshold (half of the strong threshold) selects much more of the signal close to the strongest signal (in the center of the galaxy), but also a lot of noise peaks. The map created with the new method has only the good aspects. How low the signal gets at the borders of source and background depends on the used parameters of the new method.

11. Summary

The goal of this project was to extract true astronomical features from image data cubes and to extract spurious signal from visibility data in a highly automated fashion. This goal has been achieved, which can be concluded from the following results.

First of all, the *à trous* wavelet transform is very suitable for detecting true astronomical signal at all scales in both the spatial and the visibility domain. For the latter the transform makes it easy to spot discontinuities, which is very characteristic for RFI.

Second of all, the False Discovery Rate proves to be very useful to select source pixels for the spatial data, because of its adaptive nature. Every channel in an image data cube can be treated separately, to guarantee high power in relation to the number of false discoveries. For visibility data it proves to be suited to take a very low average threshold, to ensure a careful approach.

Once a basic mask has been set for either spatial or visibility data, region growing algorithms are a fast and easy way to improve this mask, i.e. include or exclude adjacent groups of pixels based on certain morphological or intrinsical properties. This is the third important conclusion. Such an algorithm must be used carefully with the right boundary conditions. Sometimes this is accomplished most easily if certain aspects of the signal one is looking for are known a priori.

Furthermore, when setting the flags for visibility data, the new procedure is capable of removing both the RFI that is responsible for the characteristic strong stripes in the resulting images, as well as the RFI that spans a range of baselines (like the radiation from the sun).

Lastly, a first and simple comparison with conventional masking methods of spatial data shows that the new procedure is capable of automatically selecting all the signal that probably is of true astronomical origin without including background noise peaks that are just above the threshold.

12. Discussion

There are several things that can be done to improve the results.

- Very weak signal that is just above the average noise level in a channel, but which is also very extended, can be spotted by the eye immediately. It is possible to create an algorithm that searches for these kind of 'plateaus', see for instance Starck & Bijaoui (2000).

Such an algorithm does not select significant pixels based on their high value, but based on the adjacent group of pixels. The whole group must have positive value, and the group itself must have a minimal size.

- A test can be implemented that checks if indeed the mask that has been set for the image data, is a good mask. For instance by applying the KS-test on the data whose mask is set to zero before, and after the masking procedure. The mean should be lower in this residual (which should only contain noise) and the KS statistic D should also be lower.
- When performing the *à trous* wavelet transform one can choose to use an uneven number of convolutions in a certain direction to select a particular kind of signal. This is especially useful when searching for RFI in the u, v data. This is a relatively unexplored area. In this paper there has only been taken a look at doing the wavelet transform in 1 direction, and in 2 different directions, which gave unwanted results. Much more combinations are possible, like XXY, XYY, XYZZ and so forth.
- The watershed equivalent algorithm can be improved by adding additional criteria when expanding the regions. For instance, when expanding, one could take a look at the group of pixels already selected and use new information like the average, standard deviation, median and shape of the group.
- There are many different kinds of wavelet transforms that all have their own positive and negative properties. Some of them might not produce negative signal, some of them might be much better at finding specific kinds of structures when compared to the *à trous* wavelet transform. It must be kept in mind however, that this specific transform was chosen for several reasons (see section 3.1).
- RFI is scattered radiation. Therefore it is polarized. By taking a look at the difference between XX and YY of each telescope combination (Stokes parameter Q), one should be able to spot the polarized RFI easily. The scattered light from the sun produces this kind of RFI. Characteristic for the sun is that it is suddenly visible at sunrise which leaves its signature as a sharp edge in the Q data. This should be spotted easily in the wavelet cubes. To calculate Q, the data need to be calibrated first, however. And calibration can only be done after the RFI has been removed.
- Because of the wavelet transform, data is smoothed, sometimes in 1 direction in particular (for the u, v data). So if one very average baseline plane is in between two baseline planes with some very strong RFI, then part of the middle baseline plane will also be selected. This could be corrected for.
- The complete transform should in fact be done on complex data. By incorporating the phase data the noise can add up, where else it might not be visible. The procedure does not change for complex data, the results might however.
- In some baseline planes features of RFI remain unflagged, even if these would be selected manually. This is for instance the case in the baseline plane in figure 22. The relatively strong horizontal lines next to the really strong lines are impossible to select just by defining a threshold.
- The wavelet transform can also be done after some previous flagging. So for instance data that is really strong (10σ and up) can first be flagged as bad and from there on these pixels are treated as zero-valued. The advantage of this approach is that really strong RFI will not be smoothed by the wavelet transform and then will be selected as true signal.
- The new procedure can be tested on a fake data cube with known noise.

Acknowledgements. First of all I would like to thank Thijs v.d. Hulst and Tom Oosterloo for their help and support, and the fact that they always had or created time for me to discuss matters concerning this research. I would also like to thank all the students and some PhD students at the Kapteyn Astronomical Institute for their help on certain software problems and discussions about the quality of certain algorithms, and of course their company.

References

Bijaoui, A. & Rue, F. 1995, Sig. Proc., 46, 345

Bosma, A. 1978, Ph.D. Thesis

Gonzalez, R. C. & Woods, R. E. 2002, Digital image processing SE (New Jersey: Prentice-Hall, Inc., 2002)

Holschneider, M. & Tchamitchian, P. 1990, Les ondelettes en 1989 (Ed. P.G. Lemarié - Springer Verlag, 1990)

Hopkins, A. M., Miller, C. J., Connolly, A. J., Genovese, C., Nichol, R. C., & Wasserman, L. 2002, AJ, 123, 1086

Kamphuis, J. 1993, Ph.D. Thesis

Maisinger, K., Hobson, M. P., & Lasenby, A. N. 2004, MNRAS, 347, 339

Massey, F. J. 1951, J. Amer. Statist. Ass., 46, 70

Miller, C. J., Genovese, C., Nichol, R. C., Wasserman, L., Connolly, A., Reichart, D., Hopkins, A., Schneider, J., & Moore, A. 2001, AJ, 122, 3492

Perley, R. A., Schwab, F. R., & Bridle, A. H., eds. 1986, Synthesis imaging

Rogstad, D. H. & Shostak, G. S. 1971, A&A, 13, 99

Siegel, S. 1956, Nonparametric statistics for the behavioral sciences (International Student Edition - McGraw-Hill Series in Psychology, Tokyo: McGraw-Hill Kogakusha, 1956)

Starck, J. & Murtagh, F. 1994, A&A, 288, 342

Starck, J., Siebenmorgen, R., & Gredel, R. 1997, ApJ, 482, 1011

Starck, J.-L. & Bijaoui, A. 2000, in ESA SP-455: ISO Beyond Point Sources: Studies of Extended Infrared Emission, 53–+

Starck, J.-L., Murtagh, F., & Bijaoui, A. 1995, in ASP Conf. Ser. 77: Astronomical Data Analysis Software and Systems IV, 279–+

van der Kruit, P. C. & Shostak, G. S. 1982, A&A, 105, 351

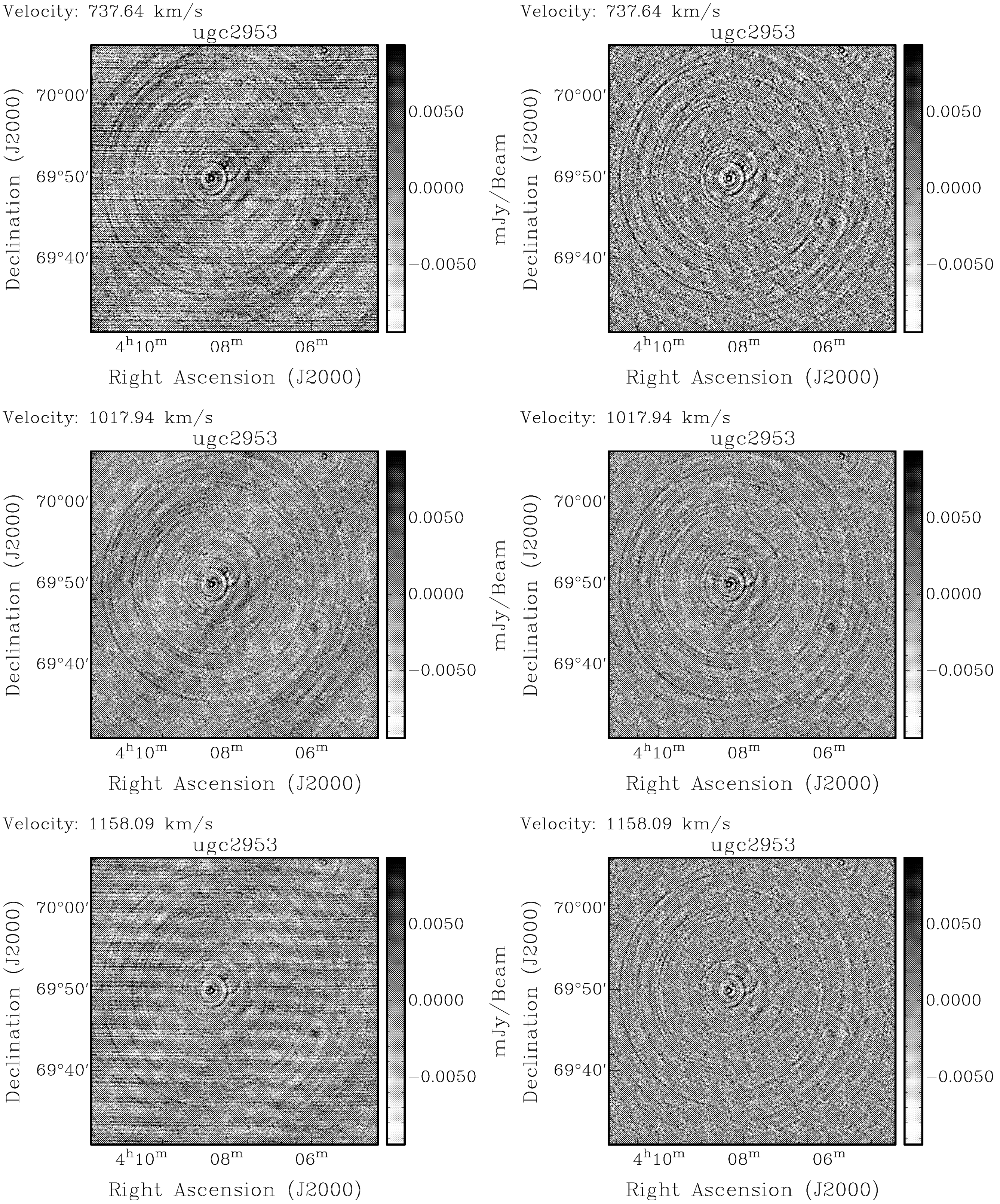

Fig. 33. Three spatial images of a data cube of *ugc 2953*. This is uncalibrated data. The left images show the unflagged data, the right images show the corresponding flagged data. The upper and lower images correspond to frequency planes with clear RFI at the end and beginning of the measurement (much like the RFI visible in figure 26). The middle images show a frequency plane which is only affected by RFI that lasts for about an hour, but which is visible at all frequencies. The broad, dark diagonal features are visible in all left images. This is the signature of the sun.

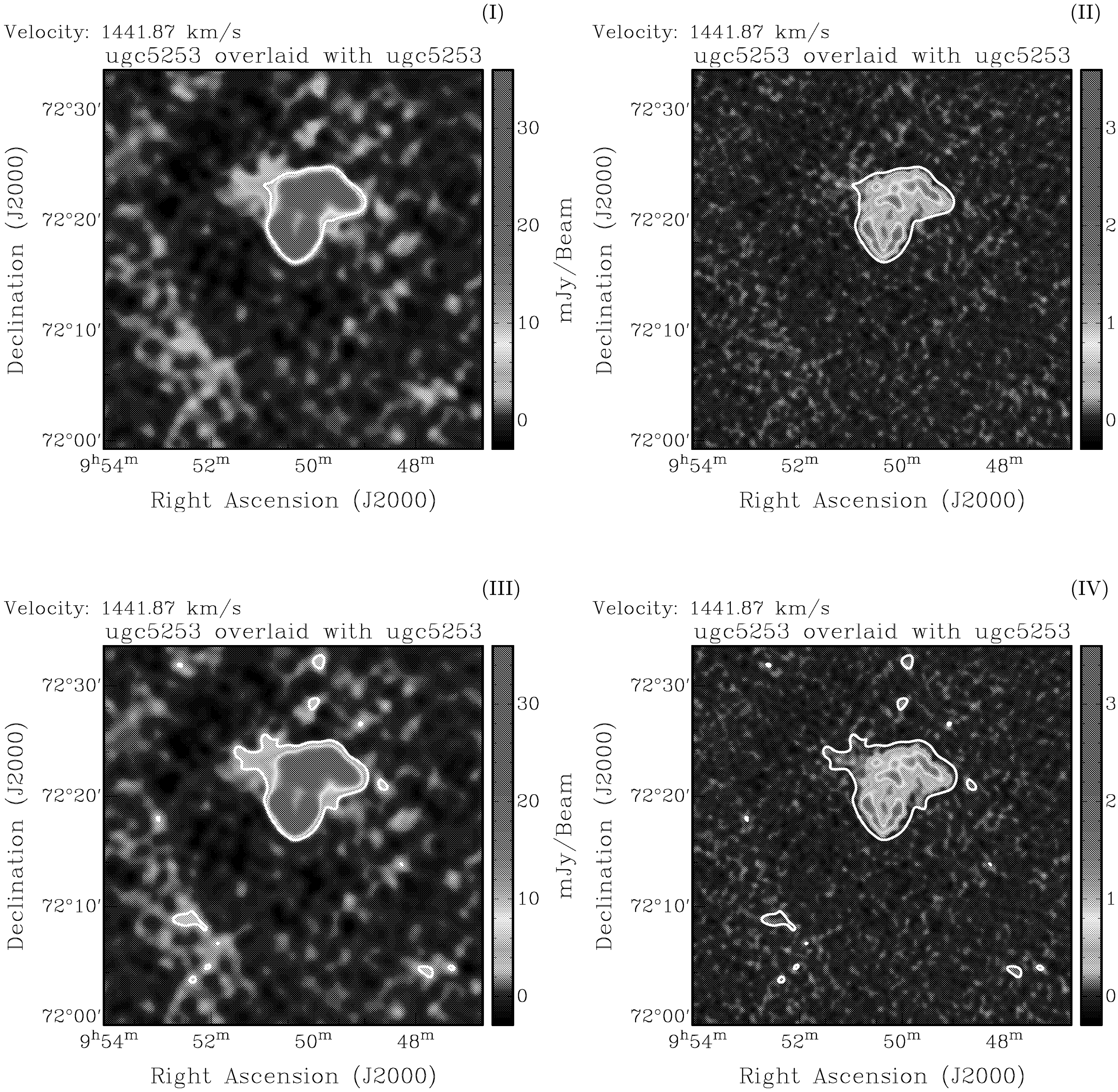

Fig. 34. The same channel of *ugc 5253* as in figure 5. The left images show the data convolved by a Gaussian of double FWHM. Image (I) has a contour overlaid that includes only the strongest signal. Image (III) has a contour overlaid that includes much more of the weaker extended signal close to this stronger signal. The right images show the original data with contours overlaid of those same two levels.

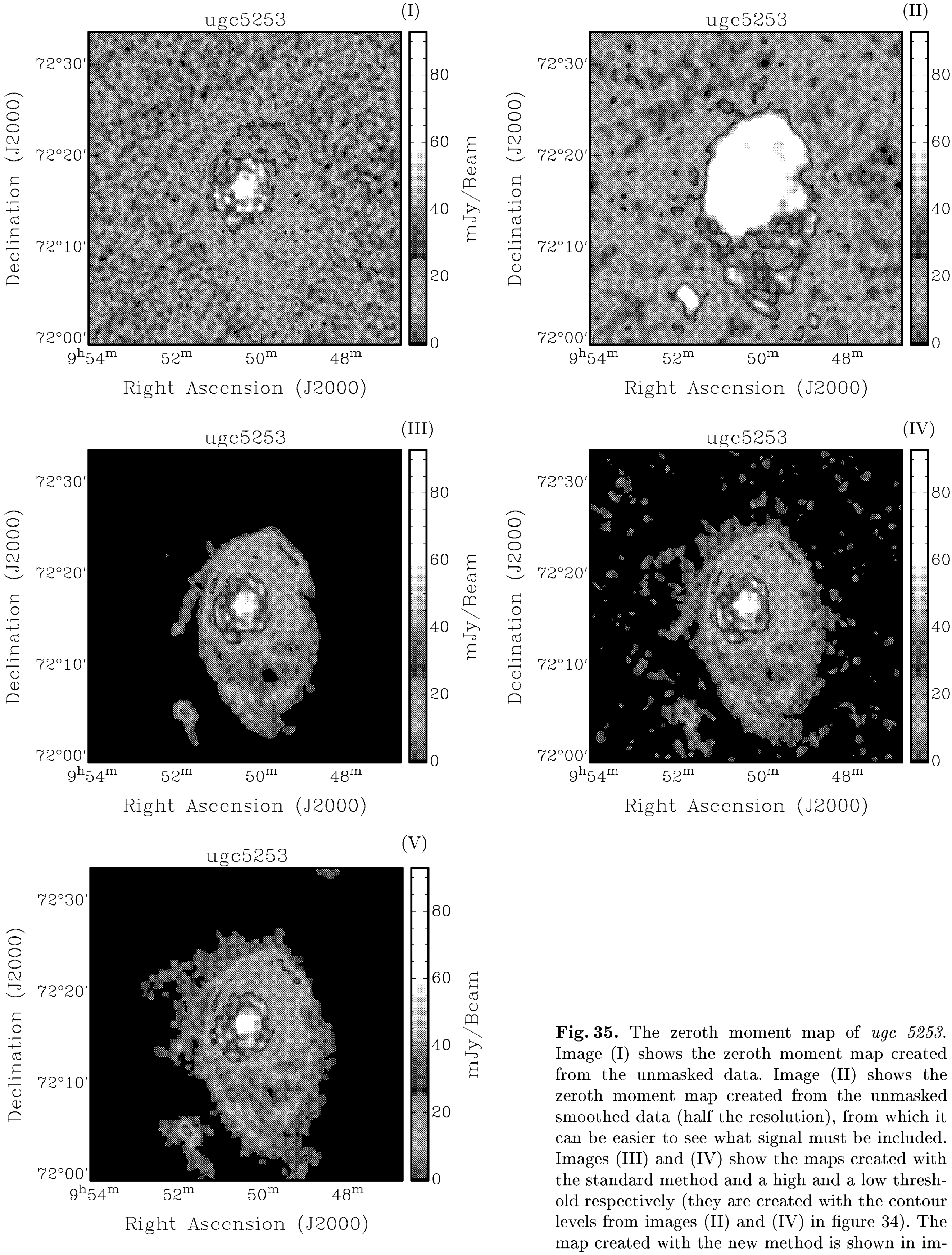

Fig. 35. The zeroth moment map of *ugc 5253*. Image (I) shows the zeroth moment map created from the unmasked data. Image (II) shows the zeroth moment map created from the unmasked smoothed data (half the resolution), from which it can be easier to see what signal must be included. Images (III) and (IV) show the maps created with the standard method and a high and a low threshold respectively (they are created with the contour levels from images (II) and (IV) in figure 34). The map created with the new method is shown in image (V).

Appendix A: Results NGC 4010

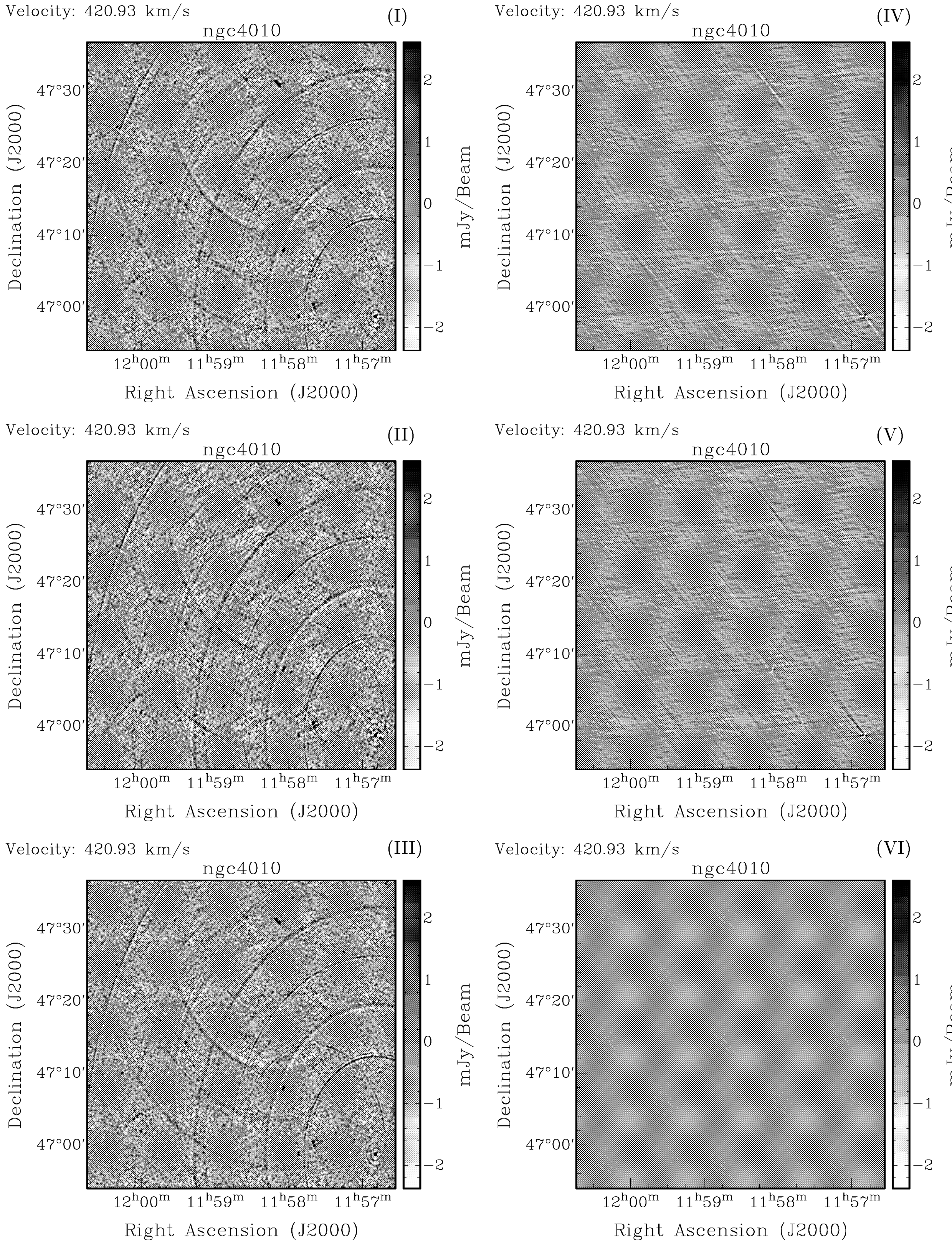

Fig. A.1. A channel of *ngc 4010*. Image (I): unflagged image, image (II): standard flagged image, image (III): new flagged image, image (IV): image (II) minus (III), image (V): image (I) minus (II), image (VI): image (I) minus (III).

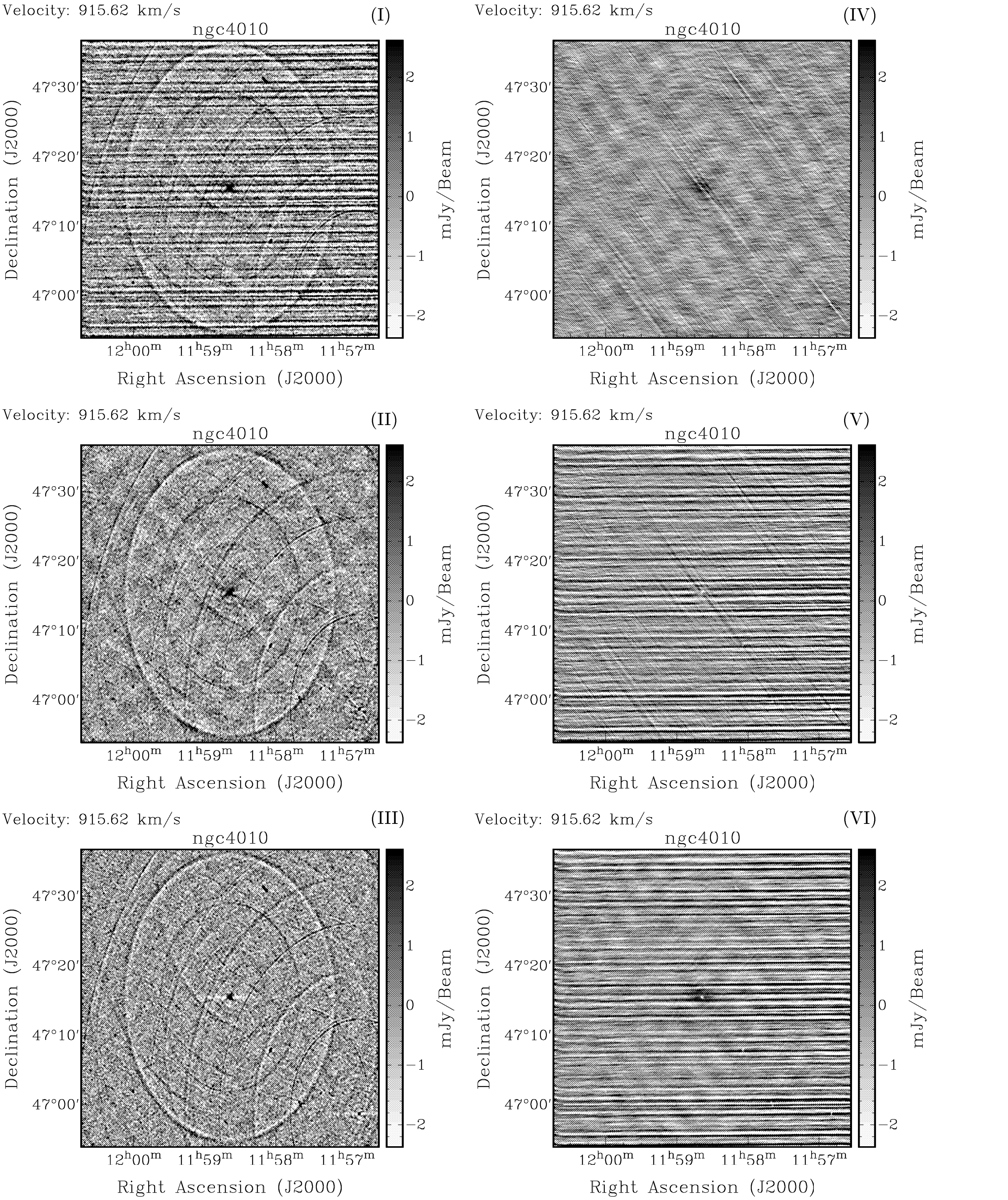

Fig. A.2. A channel of *ngc 4010*. Image (I): unflagged image, image (II): standard flagged image, image (III): new flagged image, image (IV): image (II) minus (III), image (V): image (I) minus (II), image (VI): image (I) minus (III).

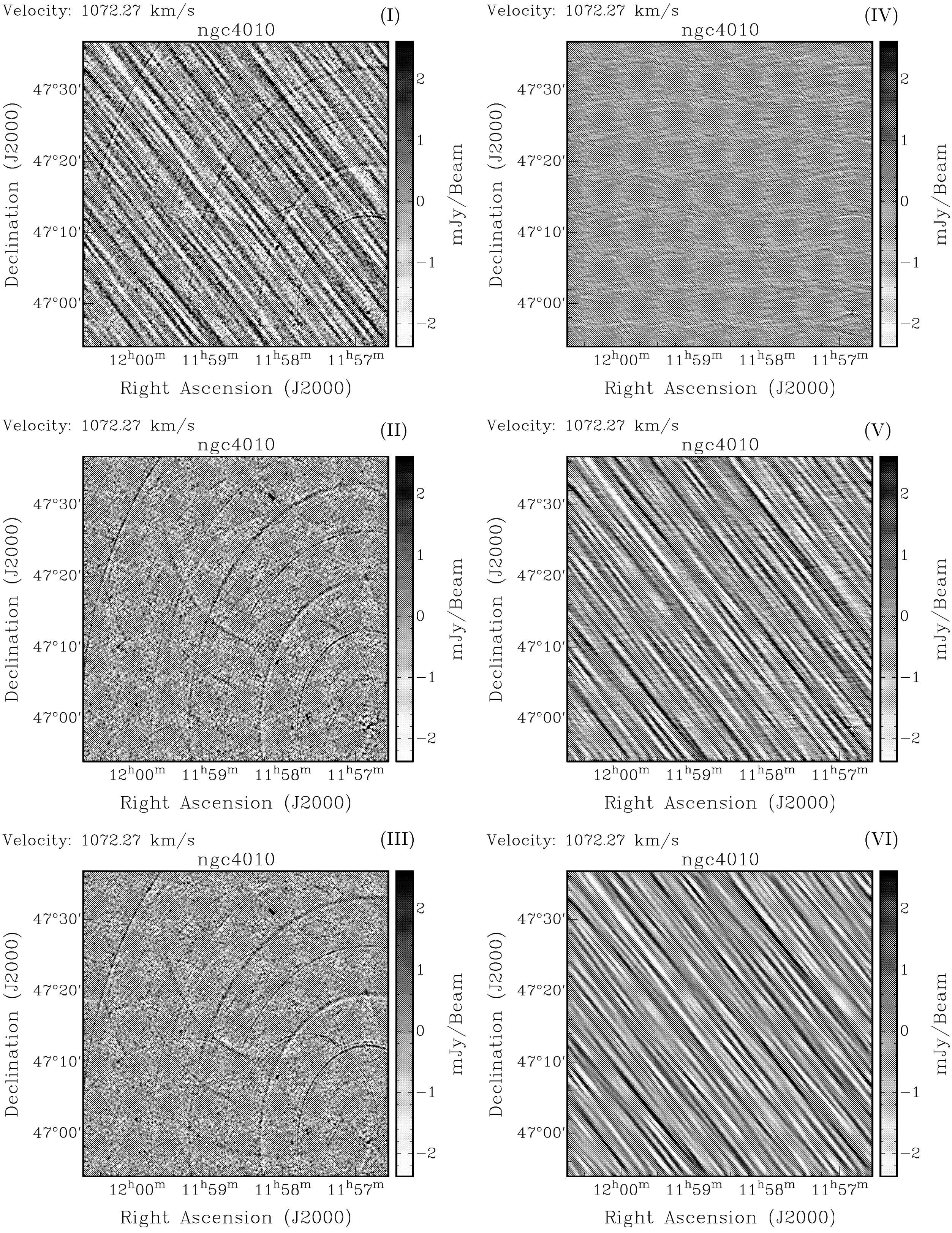

Fig. A.3. A channel of *ngc 4010*. Image (I): unflagged image, image (II): standard flagged image, image (III): new flagged image, image (IV): image (II) minus (III), image (V): image (I) minus (II), image (VI): image (I) minus (III).

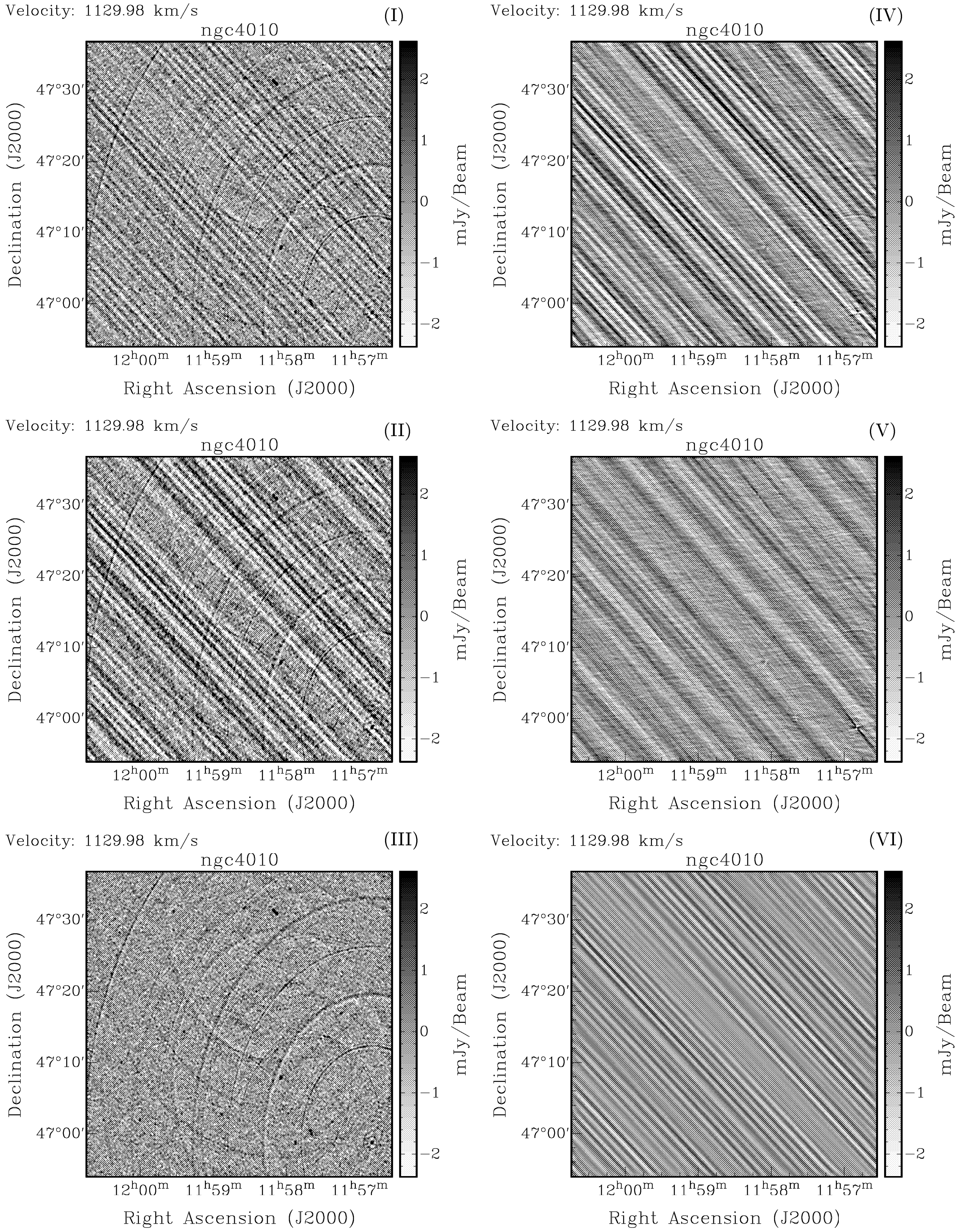

Fig. A.4. A channel of *ngc 4010*. Image (I): unflagged image, image (II): standard flagged image, image (III): new flagged image, image (IV): image (II) minus (III), image (V): image (I) minus (II), image (VI): image (I) minus (III).