Memo: Galaxy shifting pipeline

Ed Elson (ed.elson@icrar.org)

May 2011

This memo describes some of the technical details of the Python pipeline developed and used to "shift" real galaxies in order to simulate them being more distant than they truly are. The pipeline consists of 7 main steps:

1. Specify input parameters.
2. Import galaxy header and image information.
3. "Shift" the galaxy by to a larger distance.
4. Convolve the galaxy with a Gaussian kernel in order to degrade its spatial resolution.
5. Spatially re-sample the cube.
6. Spectrally re-sample the cube.
7. Implement a noise cut in order to remove most of the non-galactic emission from each channel.

Each of these steps is described in further detail below.

1 Specify input parameters

The pipeline works with several user-defined variables. These variables are specified in a separate text file that is read into the pipeline. A typical input file would be constructed as follows:

```
DDO154_NA_CUBE_THINGS.FITS
4.3
300
1.0 2.0 5.0 10.0 50.0 75.0
30.0
3.0
10.0
5.0
```

The details for each row are as follows:

1. Name of FITS cube to be processed.
2. True distance of the galaxy. Used mainly for the naming of output files.
3. Pixel length of an N x N sub-region of each channel to be processed. This option is made available for the case in which the galaxy emission occupies only a small central region of the cube.

4. A list of distance shift factors. For the example above, the galaxy will be processed at distances of 4.3, 8.6, 21.5, 43, 215, 322.5 Mpc.

5. HPBW of the new (ASKAP) beam in units of arcsec. This determines the spatial resolution of the processed cubes.

6. Number of flux density standard deviations used for the noise cut.

7. Pixel scale (arcsec/pixel) of the new cube.

8. Channel width (m/s) of the new cube.

2 Importing the FITS cube

The pipeline uses the PyFits module to import the cube. The header and image information are imported into separate variables, the latter being a NumPy array. Only a central sub-region of the cube* is retained for further processing. Specific header information from the cube is saved into various variables. Some basic details of the cube are printed to the screen/terminal. An example of the screen output for one full execution of the pipeline is:

```
Galaxy name: DDO154
Cube dimensions: (57, 600, 600)
Data type: float32
Original pixel scale: 1.50000003648 arcsec/pixel
BMAJ (arcsec): 14.09
BMIN (arcsec): 12.62
BPA (angular degrees): -34.0
Real galaxy distance (Mpc): 4.3
Shifting to a distance (Mpc) of: 8.6
Image data are NOT scaled
New pixel scale of shifted cube: 0.75000001824 arcsec/pixel

Spatial re-sampling...
New cube dimensions:  (57, 45, 45)

Spectral re-sampling...
New cube dimensions:  (9, 45, 45)

Noise properties:
mean (Jy/beam): -0.000740293449488
standard deviation (Jy/beam): 0.0577681602785
RMS (Jy/beam): 0.0577729034786

Output filename: DDO154_NA_CUBE_THINGS_8.6Mpc.FITS
```

*As specified by line 3 of the input file.

3 Shifting the galaxy

For any given distance shift factor SF[†], a variable called "scale" is defined as scale=1/SF. The galaxy is "shifted" by changing its pixel-scale header keywords. CDELT1 and CDELT2 become CDELT1*scale and CDELT2*scale, respectively. CRVAL3 becomes CRVAL3/scale, this is adjusts the central velocity of the cube in accordance with Hubble's law.

The flux density of every cube pixel is scaled by scale^2. This has the affect of modulating the flux densities as 1/D^2 as the cube is moved to larger distances.

4 Convolution

The input cubes have a spatial resolution that is typically higher than what the ASKAP resolution will be ($\sim 30'' \times 30''$). The flux in each channel therefore has to be convolved with an appropriate Gaussian kernel in order to degrade the spatial resolution. The discrete convolution operation is defined as

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n-m]. \quad (1)$$

It can be shown, however, that a convolution x(t) * y(t) in time/space is equivalent to the multiplication X(f) Y(f) in the Fourier domain, after appropriate padding (necessary to prevent circular convolution). Since multiplication is more efficient (faster) than convolution, the pipeline exploits the FFT to calculate the convolution of the channels.

The beam parameters of the cube (BMAJ, BMIN) are used to calculate the HPBW of the Gaussian convolution kernel. The HPBW along the major and minor axes of the kernel are given by,

$$\theta_{\rm kernel}^{\rm MAJ} = (\theta_{\rm ASKAP}^2 - {\rm BMAJ}^2)^{1/2} \quad (2)$$

and

$$\theta_{\rm kernel}^{\rm MIN} = (\theta_{\rm ASKAP}^2 - {\rm BMIN}^2)^{1/2}, \quad (3)$$

respectively. $\theta_{\rm ASKAP}$ is the HPBW of the circular ASKAP beam. These values are converted from units of arcsec to pixels using the pixel scale of the shifted cube. A two-dimensional Gaussian convolution kernel for the cube is then created by passing $\theta_{\rm kernel}^{\rm MAJ}$ and $\theta_{\rm kernel}^{\rm MIN}$ to the routine gauss_kern. This routine also receives a beam position angle an input, used to essentially construct a rotation matrix that will rotate the kernel as required. The size (number of pixels) of the kernel is made to match the size of a single channel in the cube.

Having constructed the Gaussian convolution kernel, a loop is used to pass each channel of the cube to the routine CONVOLVE so that it may be convolved with the kernel. Because the image provided to the Fourier transform is neither infinite nor periodic, a false cyclic behaviour of the convolution process will be induced. This is avoided by extending the image passed to CONVOLVE to twice its size. The extended region is filled with zeros. Because the FFT prefers having samples of powers of two, the image is further extended up to the next power of two. Once this padding of the image is complete, the extended image is convolved with the Gaussian kernel. After convolution, the extended (padded) portion of the image is discarded and the real part of the resulting FFT returned. An example of a convolved image is shown in Fig. 1.

Note that the definition of brightness temperature is not preserved during convolution, e.g. a point source of flux X mJy at a particular position does not contain the same flux value X at that position after convolution. To maintain the definition, the flux density of every pixel in the smoothed cube is divided by

[†]From line 3 of the input file.

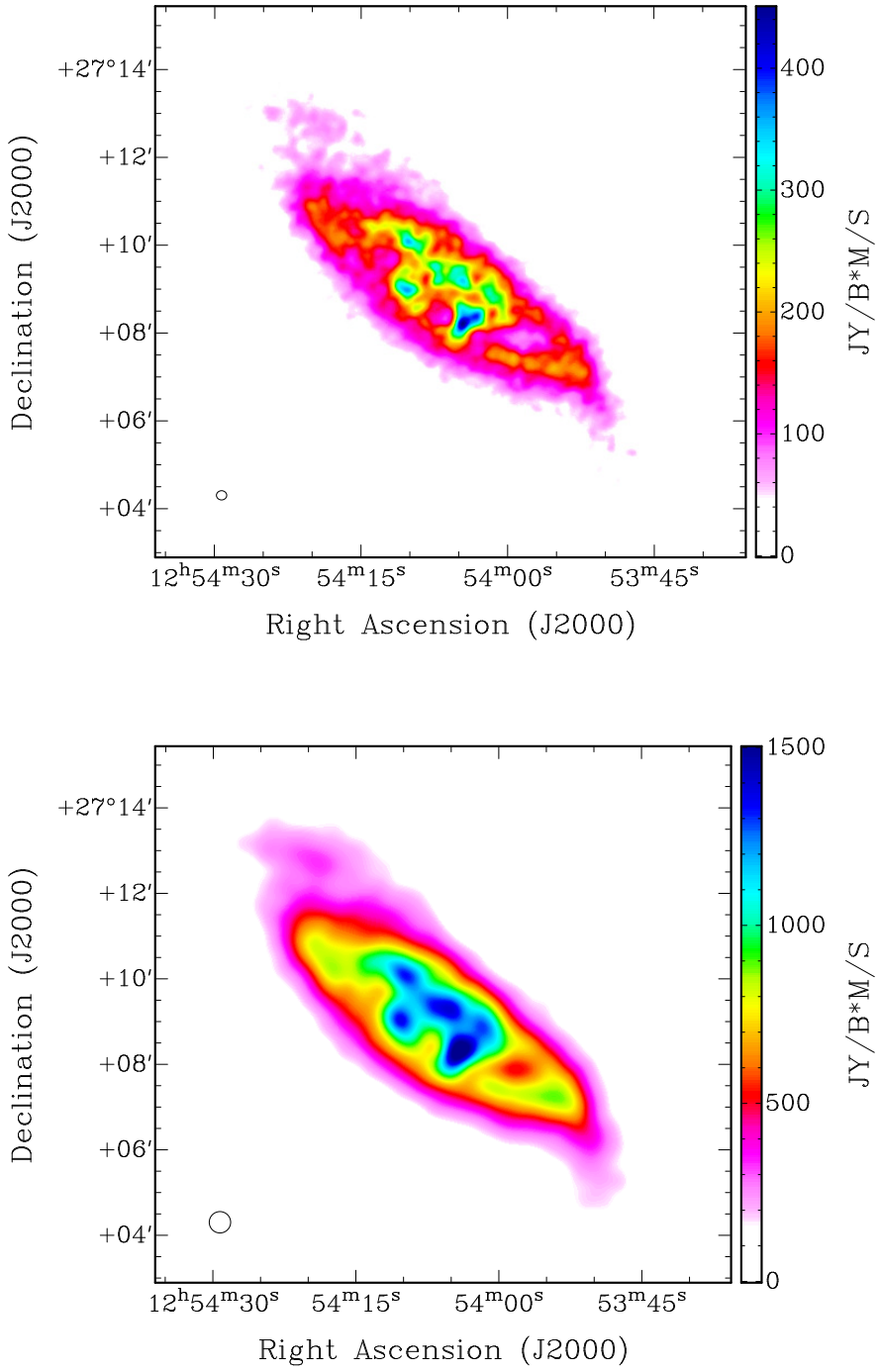

Figure 1: Comparison between the THINGS moment zero map of DDO 154 at a spatial resolution of 14.09″ × 12.62″ (top panel) and the smoothed version of the map at a spatial resolution of 30″ × 30″ (bottom panel) after convolution with a Gaussian kernel. In each panel the beam is shown in the bottom left corner.

"scale_factor". In addition to the convolution kernel, the beam parameters of the cube (BMAJ, BMIN) are used to create a second two-dimensional Gaussian. This Gaussian is convolved with the convolution kernel to produce a third two-dimensional Gaussian. The number that is needed to scale the central value of this Gaussian back to unity is "scale_factor".

5 Spatial Re-sampling

Each channel of the shifted, smoothed, noise-cut cube is passed to the routine RESAMPLE to be spatially re-sampled to a user-specified pixel scale[§]. The routine determines the ratio of the new (desired) pixel size P_{new} to the old pixel size P_{old}. This ratio is stored in a variable called "ratio". The value of element [x,y] in the re-gridded channel map is assigned to the weighted mean flux density of elements [x:x+ratio, y:y+ratio] in the original channel map passed to the routine. The weights are equal to $1/D^2$, where D is the linear distances of each contributing pixel to the centre of the larger re-gridded pixel.

The flux densities of the final re-sampled cube are scaled up by the ratio of its total flux density to the total flux density of the original cube. This ensures that no flux is "lost" during the re-gridding process.

6 Spectral Re-sampling

In addition to being spatially re-sampled, the cube can also be spectrally re-sampled in order to modify its channel spacing[¶]. This task is carried out by the routine RESPEC which operates in an analogous manner to the RESAMPLE routine. The number of channels of the re-sampled cube is determined by calculating the ratio of the old channel spacing to the new one (stored in a variable "ratio") and then multiplying this number by the original number of channels. The flux density of each element in the new cube is calculated as the flux density of the corresponding spatial element in the original cube averaged over a number of channels equal to "ratio". A weighted mean is again used. The same total flux correction as described above for the RESAMPLE routine is carried out by the RESPEC routine to ensure that the total flux before and after spectral re-sampling is conserved.

7 Noise cut

Finally, a simple noise cut is carried out in order to remove most of the non-galactic emission in each channel. A noise-filled channel is selected to have its RMS flux density determined. All pixels with a flux density value less than C times the RMS flux density are blanked[‡].

8 Pipeline implementation

The pipeline has been used to produce a suite of $\sim$ 5600 artificially shifted galaxies. Inputs from three large H I surveys of galaxies in the nearby universe have been used for this purpose. The surveys are:

- The H I Nearby Galaxy Survey (Walter et al. 2008),
- The Local Volume H I Survey (Koribalski; 2008, 2010),

[§]Line 6 of the input file.
[¶]The new channel spacing is specified in line 7 of the input file.
[‡]The value of C is specified in line 5 of the input file.

- Westerbork observations of neutral Hydrogen in Irregular and SPiral galaxies (Van der Hulst et al. 2001).

Semi-analytic models of the properties and distribution of galaxies were used to determine exactly how the various galaxies should be modified in order to generate a sample of artificial galaxies that are representative of the nearby universe. All of the "cubelets" generated by the pipeline will be assimilated to form a single $\sim 5^\circ \times 5^\circ \times 300$ MHz cube that will serve to simulate a full ASKAP observation. All of the galaxies generated by the pipeline will be made available to the ASKAP team.