$\mathcal{AIPS}$ Memo 121 revised

Editing on a uv grid in $\mathcal{AIPS}$

Eric W. Greisen

April 30, 2025

Abstract

Since its beginning, $\mathcal{AIPS}$ has offered users methods to identify and delete bad data samples. Modern interferometers observe over wide bandwidths and so must include spectral regions polluted by radio-frequency interference ("RFI"). Numerous tasks, both interactive and automated, have appeared to assist the modern user in this operation. A new one, called `UFLAG`, has been written to allow the user to explore the data as it appears when gridded in the uv plane. This memo is intended to assist users in navigating the new task.

1 Introduction

The (pre-upgrade) VLA produced, by modern standards, only modest amounts of data. The visibility samples could even be examined individually with printer displays, such as those offered by `LISTR`. As data volumes grew, interactive tasks to examine data using a "television" display were required. The first of these, `TVFLG` allows the user numerous options to view single channel data on a grey-scale display with baseline number and time axes. A similar task, `SPFLG` appeared not long after to allow similar options on a display of spectral channel versus time, one baseline at a time. Graphical editors appeared in time to edit tables (`SNEDT`), to edit uv data based on table values (`EDITA`), and to edit uv data based on its values (`EDITR`), all of which are limited to a single antenna or baseline in the edit window at any one time. A powerful editing task (`WIPER`) makes a two-dimensional plot on any two physical parameters of a uv data set keeping track of the number of samples contributing to each cell in the grid and which baselines have contributed to each cell. The display is, however, simply graphical, a cell is either occupied or it isn't. Nonetheless, bad data may be identified in many cases (*e.g.*, in plotting amplitude versus baseline length), marked as bad, and then, as the data are re-read, marked with flags in an output flag table.

Recently, Golap and Momjian[1] have discussed the advantages of making a uv grid of the data in order to average over multiple data sets, save storage space, and flag grid cells differing excessively from their neighbors. From early days, $\mathcal{AIPS}$ had ways to investigate the cause of bad grid cells (which make stripes in images) through tasks `FFT` to Fourier transform the suspect image, `COMB` to make an amplitude image from the real and imaginary images, `UVFND` to display the data contributing to bad cells, and finally `UVFLG` to add the flag commands to a flag table. There is nothing automatic or simple about that process. So it was decided to investigate an interactive task to produce a generalized shortcut of the process including the capability of doing much of the flagging itself. With an interactive task, one may also investigate possible automatic or semi-automatic algorithms for identifying and flagging bad visibility samples.

This memo has been modified by the addition of menu option `OFMCOLOR` which should appear in Figures 1, 4, and 5. The original figures have been retained for the time being.

[1]Golap, K., Momjian, E. 2016, EVLA Memo #198, "MSUVBIN: A Way to Combine, Average, Flag Visibility Data"

2 Gridded UV data exploration: UFLAG

UFLAG, new in the 31DEC16 version of *AIPS*, is the result. It is a "clone" of WIPER in that it grids the uv data into an image with u and v as axes and then displays an editing menu. Also, at the end it re-reads the uv data, finds the cells to which they contribute, and, if the cells are flagged, writes appropriate flag commands to the output flag table. However, in the case of UFLAG, the images are of the scalar and vector average amplitudes of the data falling into the grid cell, the vector average phase, and the difference between scalar and vector amplitudes. A menu item allows you to select which of these you wish to examine and to use in editing at any particular time. UFLAG keeps track of which baselines contribute to each grid cell (up to 25) and will display the first two in the list during some interactive flagging operations. UFLAG also constructs lists of which visibility samples contribute to each grid cell with no limit in the length of the lists (a linked list structure is used). This allows the task to re-examine each visibility that contributed to a particular cell, allowing the user to flag some, rather than all, of the visibilities contributing to a cell. Of course, one may also flag all visibilities that contribute to a cell (as in WIPER).

One should note that the grid constructed by UFLAG is made by putting each visibility into a single cell with full weight. This "pill-box" convolution, while necessary for flagging, is a rather poor convolution for imaging purposes. High-quality convolutions, such as the spheroidal function used by default in *AIPS*, grid a sample into multiple cells in the grid with non-unity weights and make a rather confusing image when used for editing. The old FFT method suffered from this issue unless the image was heavily Cleaned before the FFT or was constructed with the pill-box convolution.

For calibration sources, the phase image in UFLAG can be of considerable interest, while for target sources it is likely to be more confusing than useful. However, in both cases, the difference image between the scalar and vector images will reveal grid cells in which the visibilities differed significantly in their phase. Note, however, this difference image is exactly zero where only one visibility sample contributes to a cell, no matter how egregious that visibility.

2.1 Inputs

There is a large number of input adverbs to UFLAG but many of them are familiar adverbs used in the normal fashion. Such adverbs select the input data set, the sources included, the Stokes value (one at a time is required), the frequency ID, times, antennas, spectral windows ("IFs"), calibration, and input flagging. This section will only highlight the new adverbs and those used in somewhat non-standard ways.

Channel selection is handled in the usual way with BCHAN and ECHAN, but every channel so selected appears in the single uv plane grid at its correct (frequency-dependent) location. That may be modified by setting NCHAV to pre-average channels before they are gridded. This is a non-standard use of NCHAV, also practiced in WIPER, but quite unlike the usage in IMAGR in particular. CHINC may also be used to limit the channels included in the grid. Adverb OUTFGVER may be used to specify a particular output flag table. It is probably best to leave this at zero to create a new flag table. Note that UFLAG will copy the input flag table (FLAGVER, if any) to a new output table, but not to a pre-existing one. Note also that UFLAG is capable of restarting, using the new flag table as an input flag table, and then creating a new output flag table. For this reason, leaving OUTFGVER zero is probably a good idea. Among other reasons, if you do not like the results of UFLAG, it is easy to delete a new flag table, but harder to edit the UFLAG entries out of an existing table.

The user controls the size of a cell in the uv plane grid with the familiar CELLSIZE (image cell size in arc seconds in x and y) and IMSIZE (number of image pixel in x and y) adverbs. The cell size in wavelengths is then $(180 * 3600)/(\pi * \mathrm{cellsize} * \mathrm{imsize})$. Note that there is no requirement in UFLAG that IMSIZE be powers of two or that the grid cell sizes be the same. UFLAG reads the data to determine the minimum and maximum values of u and maximum value of $|v|$, using the Hermitian property of visibilities to require that $v \geq 0$. This then sets the grid image size, including some pixels around the edge to improve the display and keep the tick marks from interfering with the data. Adverb DOWEIGHT controls whether the data weight or 1.0 is used to weight the data into the grid. Adverb DOCENTER simply controls the position of the CURVALUE-like display during interactive, pixel-based editing operations. A value of -1 places it at the upper left, +1 at the upper right, and 0 at the upper center. Adverb PRTLEV controls the amount of messages appearing with > 0 printing information about each cell addressed in auto-flagging operations and > 1 also producing, at the end or a restart, matrix displays of correlations flagged.

Adverb DOALL is a bit pattern that you may use to control which spectral channels, spectral windows, and polarizations are ultimately flagged when a flag command is generated. A value of zero is conservative while higher values cause additional data to be flagged.

The one new adverb is ISCALIB to tell the task whether the source included should be treated as a target or a calibration source. The latter are assumed the have amplitudes centered about some constant value and phases centered on zero. The auto flagging operations flag calibration sources for both high and low amplitudes as well as phases too far from zero. For target sources, the auto flagging routines ignore phase and low amplitudes, flagging high amplitudes only. Adverb APARM is used to control the setting of examination and clip levels in the AUTO FLAG VIS operation. See below for a description of this operation and the usage of APARM.

2.2 Running UFLAG: menus and display options

UFLAG begins by reading the data set to make sure that some data are found and to determine the size of the uv grid to be displayed. Then it compares that grid size to the size of the TV display screen. If the grid is larger than the screen, the task displays the grid image in a pixel-averaged form and requires the user to select a sub-image small enough to fit of the screen without pixel averaging. If the grid is small enough, the display will use pixel replication to blow up the display, but it always uses the same pixel replication factor on both axes. The selection of a sub-image is done by the usual TVWINDOW operation: point at the bottom left corner, hit button A or B to select the top right corner, and finally hit button C or D to exit, selecting the sub-image. The sub-image (or full image) will then be displayed including axis labels. The VECTOR average image is selected for the first display. A grey-scale step wedge will appear above the uv plane grid. It is labeled at the left with the lowest value, at the right with the highest value, and in the center with the data type and polarization displayed. The median value of the displayed image and the simple mean and rms of the image (ignoring unsampled cells) are also displayed above the step wedge. All labeling uses the first graphics plane which is usually yellow.

The following menu appears in graphics plane two (usually green) at the left of the screen whenever you need to select some operation.

ABORT	Exit the task, writing no more flag tables
FLAG + EXIT	Exit the task, writing flag tables
FLAG + REPEAT	Write current flagging information, restart with that flag table as input
FLAG + SWITCH	As FLAG+REPEAT, but also switch Stokes
OFF ZOOM	Reset zoom factor to one
TVZOOM	Set zoom factor and center for some displays
OFF TRANS	Initialize black-and-white transfer function
OFF COLOR	Turn off any pseudo-coloring
TVTRANSF	Adjust black-and-white transfer function
TVPSEUDO	Color contours of a variety of types
TVPHLAME	Flame-like pseudo-coloring
OFMCOLOR	Many standard color tables (OFMs) available
SET PIXRANGE	Type in grey-scale intensity range for display for the image type being viewed
LOAD *xxxx*	Re-load image with *xxxx* transfer function
VIEW *zzzz*	Re-load with image type *zzzz*
VIEW ONLY	To set the intensity range visible in the display, then may flag
VIEW ALL	To view all intensities now available
FLAG PTS OFF	To turn the display of flagged cells off or back on

This menu includes the familiar image display options. For more accurate flagging, you may zoom the image interactively including the center about which the zoom is done. The image is shown in this zoom during enhancement operations, window setting, flagging/unflagging operations, and the visibility data "examination" function. Otherwise the image is shown without zoom. Note that `SET WINDOW` allows you to select a sub-image which will be shown in a pixel-replicated fashion up to the replication factor that still fits on the TV display. This may be a better way to see a portion of the image "zoomed." Other familiar display options allow you to enhance the image interactively in black and white and in pseudo-colorings of various sorts and then to turn these enhancements off again.

`SET PIXRANGE` allows you to type in the desired intensity range to be displayed with actual data values above the range getting full intensity and those below the range getting the minimum value. The range set is for the image type currently displayed and does not affect the other types. It may be reset by entering a minimum value $\geq$ the maximum value. `LOAD` *xxxx* allows you to re-load the image with the specified transfer function. The value of *xxxx* that appears in the menu is the next one in the sequence `LIN`, `LOG`, `SQRT`, and `LOG2`. Thus, if the display has been loaded with a linear transfer function (as is done when it starts), then `LOG` will appear in the menu.

Similarly, `VIEW` *zzzz* allows you to select which image to view and to use in editing from the sequence `PHASE`, `SCAL-VEC`, `SCALAR`, and `VECTOR`. Again, if `VECTOR` has been loaded (which is how the task begins), then `PHASE` will appear in the menu. The option `FLAG PTS OFF` appears when flagged grid cells may be shown with marks in graphics plane four (usually cyan). Since large numbers of these can be distracting, the option allows you to turn off the graphics plane. If you do, the menu option changes to `FLAG PTS ON`. The remaining options in this menu will be discussed in later sections.

The following menu appears in graphics plane two (usually green) at the right of the screen whenever you need to select some operation. Some of these options appear only when appropriate conditions occur

`SET WINDOW`	Select a sub-image for more detailed viewing
`RESET WINDOW`	Return to viewing the full image
`FLAG POINT`	Flags cells one at a time on buttons A, B, C
`FLAG AREA`	Flags rectangles of cells set in the usual manner
`FLAG FAST`	Flags any cell at which the cursor is pointed
`FLAG BASELINE`	Records flagging for a pair of antennas and marks cells appropriately
`UNFLAG POINT`	Unflags flagged cells one at a time with buttons
`UNFLAG AREA`	Unflags flagged cells in rectangles
`UNFLAG FAST`	Unflags any flagged cell at which cursor is pointed
`UNFLAG BASEL`	Removes record of flagging for a pair of antennas and unmarks cells appropriately
`EXAMINE VIS`	Displays all visibilities contributing to interactively selected cells, allows some of them to be flagged with the remaining cell value (if any) restored
`USER FLAG VIS`	Auto-flags visibilities based on value ranges entered on the terminal
`AUTO FLAG VIS`	Auto-flags visibilities based on value ranges set by the image statistics
`FLAG ALL`	Flags all cells in the `VIEW ONLY` range
`FLAG ALL WIN`	Flags all cells in the `VIEW ONLY` range in the current sub-image
`UNDO FLAG ALL`	Unflags a previous `FLAG ALL` range
`UNDO FLAG WIN`	Unflags a previous `FLAG ALL` range but only in the current sub-image

Most of these functions will be discussed in later sections. The `SET WINDOW` function is the familiar `TVWINDOW` interaction in which the first button A or B push sets the lower left corner, then the next button A or B sets the upper right corner, until you have selected the desired sub-window for viewing. Then buttons C or D exit the function with the selected sub-image promptly loaded to the TV display. The largest pixel replication factor which allows the sub-image to fit on the screen will be used. Certain menu operations are limited to the viewed sub-image and the use of pixel replication makes it easier to select desired cells for examination or flagging. `RESET WINDOW` selects the full image, if it will fit on the display without pixel averaging. If it does not fit, a pixel-averaged image will be displayed and you will be directed to choose a sub-image that will fit.

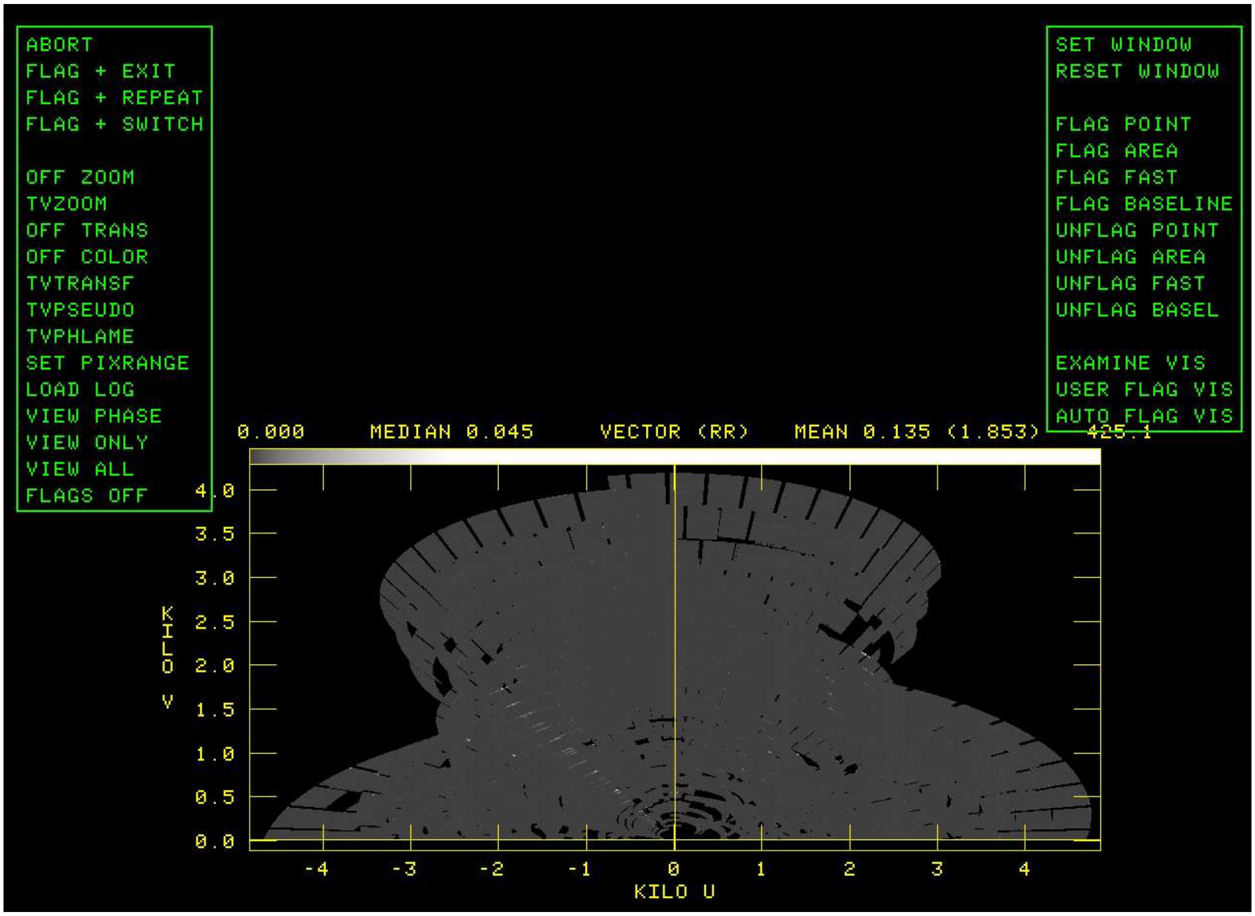

Figure 1: Initial `UFLAG` display. This example is a target source showing all 64 spectral channels of one spectral window and is badly affected by RFI. The black-and-white transfer function has been adjusted to show all visibilities in light grey with the bright white cells in need of flagging. The `31DEC25` option `OFMCOLOR` is not shown here.

The `FLAG ALL` option appears only when a `VIEW ONLY` operation is in effect for the current display image type. `FLAG ALL WIN` appears in the same circumstance when the viewed image is a sub-image. `UNDO FLAG ALL` appears only when the current display image type has had a `FLAG ALL` operation performed. `UNDO FLAG WIN` appears in the same circumstance when a sub-image is displayed. See below for further explanation of the `VIEW ONLY` function.

A sample of the initial display is shown in Figure 1 [2].

2.3 Basic flagging

Interactive flagging operations in the second menu include `FLAG POINT` which allows you to point at a cell to be flagged and then hit buttons A, B, or C to flag that cell. Button D exits the operation. During this operation, a display at the top of the window shows the cell coordinates in pixels and in u and v, the cell value, and up to two baselines which contribute to the cell (with a + if there are more). This is illustrated in Figure 2. The value code letter is `S` for scalar amplitude, `A` for vector amplitude, `P` for phase, and `D` for the

[2] For legibility in printed copies, the figures shown in this memo have been constructed with `CHARMULT 2` to double the character size and with uv cells somewhat larger than desirable in order to avoid overlapping the menus and data.

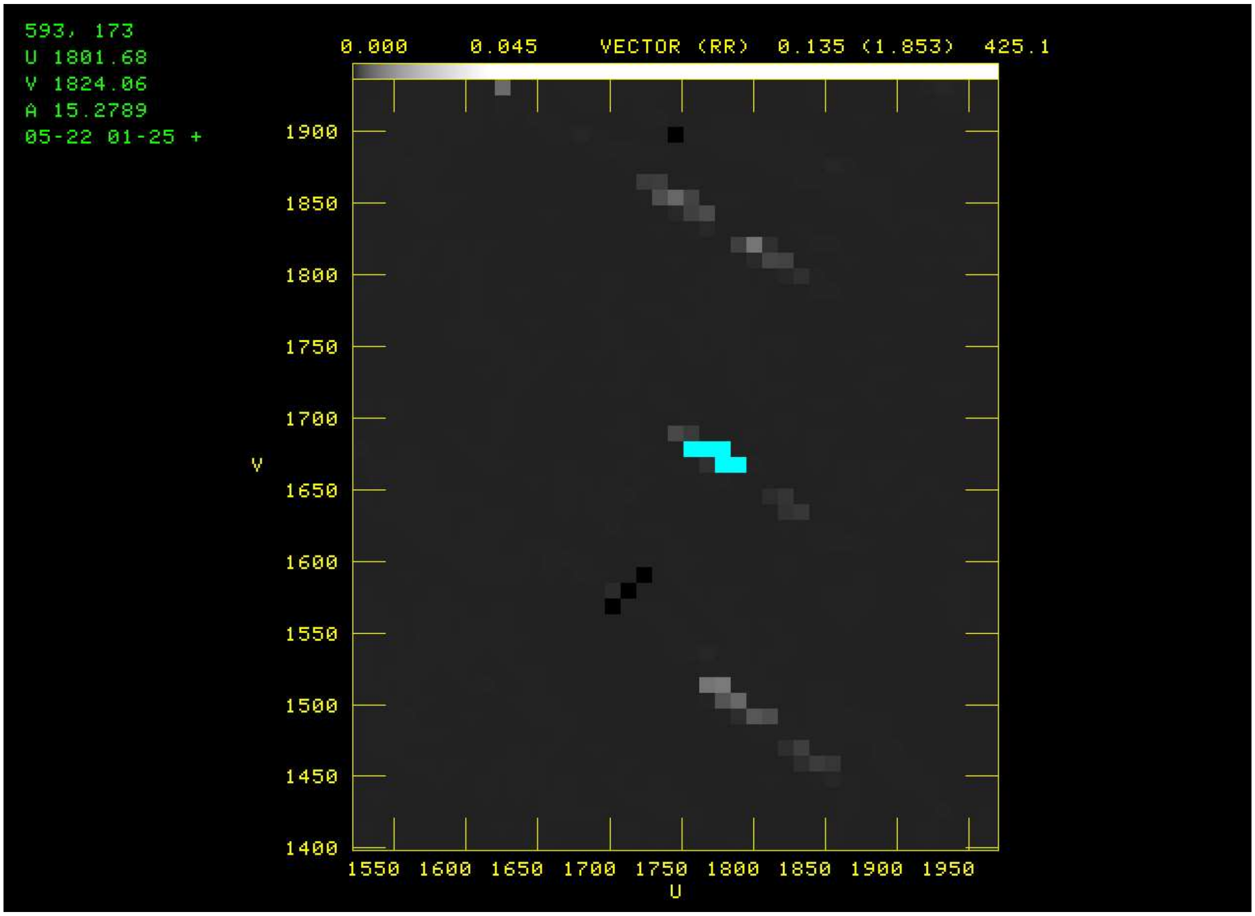

Figure 2: `UFLAG` display during `FLAG POINT` using a sub-image of the data set shown in Figure 1. Five cells have already been flagged.

difference between scalar and vector amplitude. A faster version of this function is offered by `FLAG FAST`. The fast operation is started by pointing at a sample and hitting button A, B, or C. From then, any cell at which you point with the cursor will be flagged until you hit button D. Note that you "point" at a cell by holding down the left mouse button. The fast operation then consists of clicking the left mouse button over all cells you wish to delete. The coordinates and value for the cells selected are displayed.

If a whole rectangular area of the grid needs to be flagged, use the `FLAG AREA` operation. It begins by letting you point at the lower left corner. Then hit button A to select the bottom left and to start setting the upper right corner. After that, hit button A to switch between setting the two corners, button B to flag the current area and continue to set another area, and button C to flag the current area and exit. Hit button D to exit without flagging the current area.

The `UNFLAG POINT`, `UNFLAG FAST`, and `UNFLAG AREA` operate in the same way to undo the flagging of selected cells. Doing an `UNFLAG` operation on a cell which has not been flagged has no effect.

The `FLAG BASELINE` menu item lets you type into the terminal the numbers of two antennas whose baseline you wish to flag. If one of the numbers is zero, all of the baselines to the other antenna number are marked to be flagged. When you do this, the images are changed by flagging only those cells affected solely by the flagged baseline(s). The flag is written to the output flag table when you exit as a flag for all sources, spectral channels, spectral windows, and times for the specified baseline(s). This is pretty drastic and so should be reserved for special, egregious cases. `UNFLAG BASELINE` undoes all (or part) of a previous `FLAG`

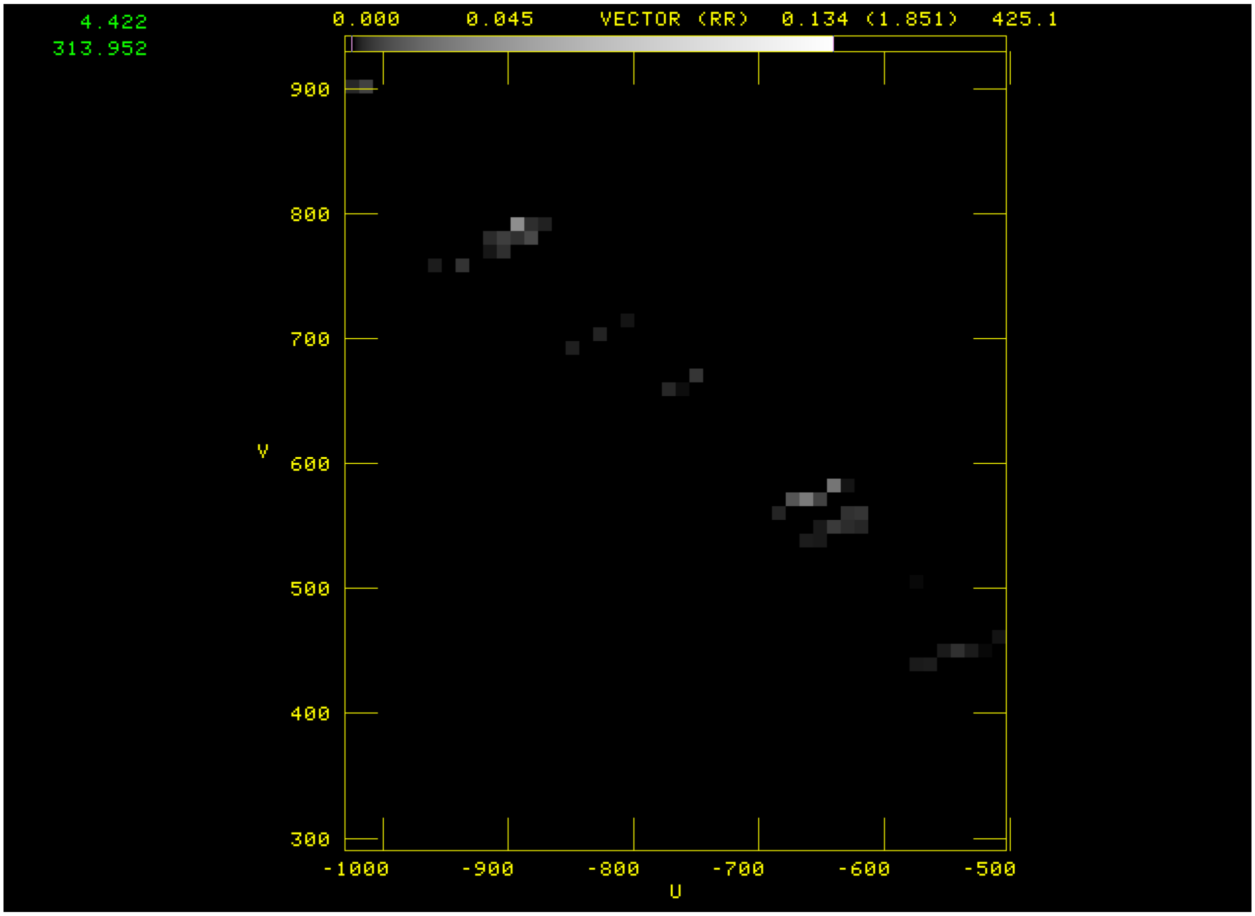

Figure 3: `UFLAG` display during `VIEW ONLY` using a sub-image of the data set shown in Figure 1. Flagged cells are not visible and cells with value less than 4.4 or greater than 314 Jy (!) are also not visible.

`BASELINE`. You can, for example, flag all baselines to antenna 18 and then unflag baseline 1-18.

2.4 VIEW ONLY flagging

The `VIEW ONLY` operation introduces a form of clipping. As illustrated in Figure 3, selecting this menu item produces an interaction using the step wedge at the top of the displayed image. Hit button A or B to switch between setting of the top value and the bottom value of the image value range you wish to view. The current value range is shown at the top of the TV display. During the interaction, the black-and-white image transfer function is used to make all values outside the range completely black. If you hit button C, these values are used to control a full image load making all excluded image values invisible. Hit button D instead to exit the operation without setting a "view-only" range and state.

When a view-only state is set, the menu items `FLAG ALL` and, if you are viewing a sub-image, `FLAG ALL WIN` are shown as illustrated in Figure 4. Note that in the view only state, the only cells shown are those in the range that was set and that this includes flagged cells as well as unflagged cells. Selecting `FLAG ALL` will proceed to flag all pixels in the selected value range over the full image, while `FLAG ALL WIN` will only flag cells in the selected sub-image. If one of these is selected, the operation and its values are recorded in a list separated by image type.

You may turn off the view-only state with the menu item `VIEW ALL`. The view-only state is also reset

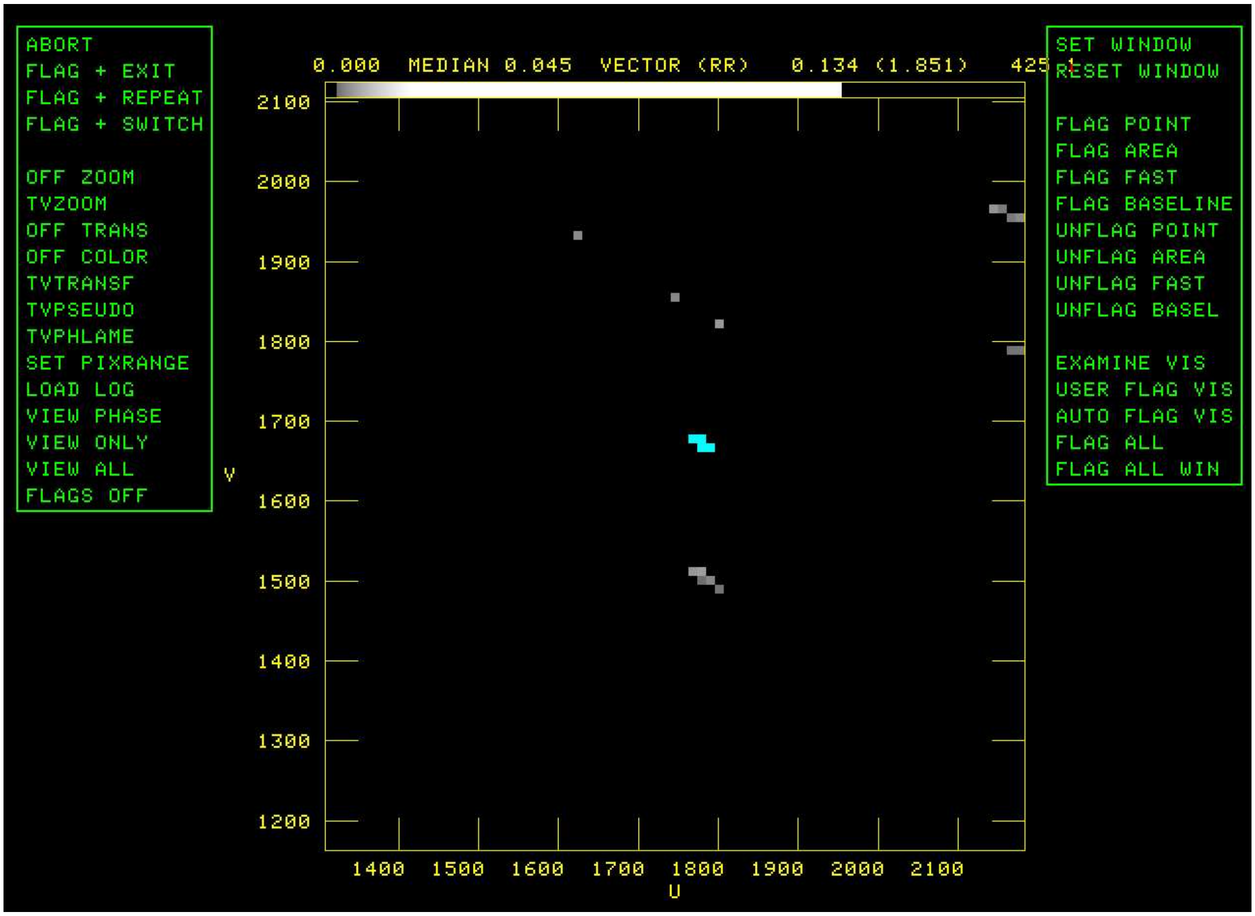

Figure 4: `UFLAG` display after `VIEW ONLY` using a sub-image of the data set shown in Figure 1. Five cells have already been flagged but only 4 are in the view-only range. Note the addition of `FLAG ALL` and `FLAG ALL WIN` menu options. The `31DEC25` option `OFMCOLOR` is not shown here.

whenever the `FLAG ALL` or `FLAG ALL WIN` option is selected or when a new image type is selected. If a `FLAG ALL` operation has been performed for the currently selected image type, menu items `UNDO FLAG ALL`, and, if you are viewing a sub-image, `UNDO FLAG WIN` will appear. The `UNDO FLAG` operations use the list created in either of the `FLAG ALL` operations to allow you to undo one or more of the operations over the entire image or only in the sub-image. If there have been more than one `FLAG ALL` operations on the current image type, the program will show you the list of the operations and ask you which you wish to undo. Enter a number in the terminal window. If you `UNDO FLAG ALL`, then the selected operation will be removed from the list, but, if you only undo it in the sub-image, then it will remain in the list.

2.5 Flagging input visibilities

All flagging operations considered so far cause every visibility sample contributing to a flagged cell to be marked as bad in the output flag table. This may be a serious over-kill. The gridding operation has also recorded the visibility number of every visibility that contributes to every cell in the grid. This enables `UFLAG` to review the visibilities at selected cells and potentially mark them as flagged individually rather than as a whole. The three menu operations described below do this in different ways, but all three end up writing records in a "flag command" (`FC`) table attached to the input uv data set. Visibilities flagged in this way remain flagged. They are removed from the linked lists. The only way to restore them is to exit with

ABORT which destroys the FC table without applying it.

The first of these operations, EXAMINE VIS, is a "manual" editing function. It brings up an interactive display much like FLAG PIXEL, complete with display of the cell coordinates and value. Hit button A, B, or C to examine a cell. The task then re-reads the input data set at all samples contributing to the selected cell and applying all flags and calibration used when making the original grid. It lists the samples including a sample number, time, baseline, IF, channel range, amplitude, and phase. Then UFLAG asks which one to flag. Enter a number on the terminal to delete that visibility. The question is repeated until you enter a zero. Then, if you asked for some visibilities to be deleted, the remaining ones (if any) are averaged and the image grid values and TV display changed to the new values. Suitable FC table records are written for later use. If no samples remain, the image cell is marked as unsampled. The interactive selection of cells to examine continues until button D is hit.

The second menu choice to examine the input visibilities individually is called USER FLAG VIS. It begins by asking several questions to which you need to enter values on the terminal. The first question is "Examine pixels in range (2 values)" which wants the lower limit and upper limit of values in the current image type that are to be examined. Then it asks "Flag amplitudes below and above (2 values)" which wants the lower and upper limits for visibility amplitudes in Jy which are considered good. All those below and above these limits will be flagged. Note that 0 or -1 is perfectly acceptable as the lower limit if you do not want to flag low visibilities (*e.g.*, target sources). The third question is "'Flag phases below and above (2 values)" which wants similarly the lower and upper limits for good phases. Again, note that an answer such as -200 200 is probably the correct answer for non-calibration sources. The final question is "'Examine flagged (-1), unflagged (1), or both (0)" which determines which cells will be examined. UFLAG then examines every cell *in the current sub-image* and, for those cells selected by the examination range and flag state parameter, performs a detailed examination. It reads every visibility record contributing to the chosen cell and, for every visibility outside the allowed amplitude and phase ranges, causes the visibility to be recorded in the FC table and removed from the linked lists. The remaining visibilities (if any) are averaged and the images updated. If no visibilities remain, the cell in the images is marked as unsampled and, briefly, the cell is also marked in cyan as flagged. When the operation completes, the cyan marks disappear since the cell is now viewed as unsampled rather than flagged.

The third menu choice, AUTO FLAG VIS, performs an operation similar to USER FLAG VIS except that it uses the image statistics (displayed above the step wedge) to set the cutoff limits. These statistics are not adequate to set limits when the phase and scalar minus vector image types are displayed, so AUTO FLAG VIS does nothing when these types are selected. For non-calibrator sources (adverb ISCALIB ≤ 0), all unflagged cells above the mean plus APARM(1) times the rms are examined. Any visibility amplitude greater than the median plus APARM(2) times the rms is eliminated. For calibrator sources (ISCALIB > 0), all unflagged cells below the mean minus APARM(1) times the rms and above the mean plus APARM(1) times the rms are examined. Visibilities having amplitudes more than APARM(2) times the rms from the median are eliminated as are those with phases more than APARM(3) times the rms divided by the median (in radians) away from zero. The default values for APARM are 3, 3, and 4 and are used whenever the input value of the particular APARM(i) is less than 1.0 The operation is performed only in the current sub-image. As with the previous two operations, any remaining visibilities in the cell are averaged and the images updated. If none remain, the cell is temporarily marked as flagged and also as unsampled.

A sample figure for these operations is not really possible — a movie would be more appropriate as one watches the bright pixels be corrected in sequence. These operations are painstaking and so are rather slow.

2.6 Exiting and re-starting UFLAG

There are four menu operations to exit or re-start UFLAG. The simplest one is ABORT which exits the task deleting the FC table (if any) and writing no output flag table. All three of the others begin by writing the output flag table if any flags were generated. The input flag table (FLAGVER ≥ 0) is copied to the output flag table if it is a new flag table. Then the data set is re-read, the location in the grid of each re-read sample is determined, the counts at each cell are decremented, and, if appropriate, a record written to the output flag

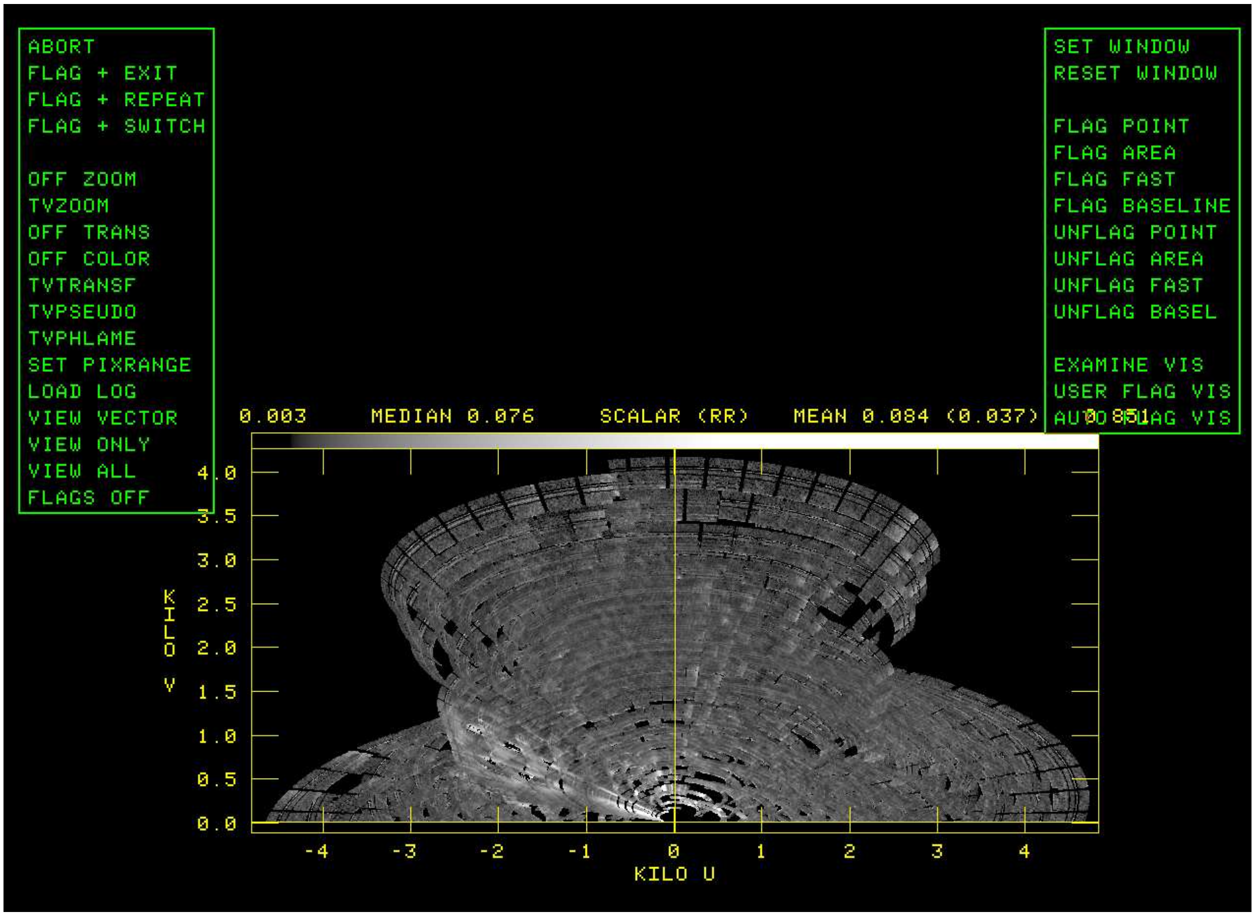

Figure 5: `UFLAG` display using an input flag table generated by a sequence of *AIPS* flagging tasks. Scalar average is shown since it reveals a region of higher than normal amplitudes while the vector average looks quite okay because of very noisy phases. The `31DEC25` option `OFMCOLOR` is not shown here.

table. Then all `FLAG BASELINE` flags are written to the output flag table. If there is an `FC` table, its rows are translated into commands written to the output flag table. If `PRTLEV` > 1, an elaborate summary of flagging by correlator is printed. Then the `FC` table is deleted. At this point, menu choice `FLAG + EXIT` simply exits the task.

Menu choice `FLAG + REPEAT` instead resets the input flag table (`FLAGVER`) to the output flag table, sets the output flag table to be a new one, and loops to construct the grid images over again. `FLAG + SWITCH` also does this, but, additionally, it switches the polarization in pairs. Thus RR and LL switch, RL and LR switch, I and V switch, Q and U switch, XX and YY switch, and XY and YX switch. Note that changing the input and output flag tables depends on the output flag table having been a new table, or at least the highest numbered one. Really, you should always set `OUTFGVER` to zero.

3 Conclusions

It is clear that `UFLAG` is a useful addition to the arsenal of *AIPS* data flagging tasks. Just how useful remains to be seen. If an image shows stripes, then `UFLAG` is both easier to use and more powerful than the traditional Fourier-transform approach and should show the bad cells that are causing the stripes. However, flagging every sample contributing to a grid cell is in many cases excessive. It has been asserted that there are situations with noisy input data where stripes appear while imaging. The data are said to be too noisy to flag individually, but with enough observing sessions, the uv grid can reveal bad cells. `UFLAG` is able to address this situation easily, but the author has yet to see an example. The operations that address the visibilities individually are really only slightly better informed clipping operations which might be addressed more simply with $\mathcal{AIPS}$ task `CLIP`.

Of more concern than these points is the fact that each visibility sample falls in exactly one cell. This has been seen to disguise really bad visibilities if there are enough good samples also contributing to the cell or if the bad visibility amplitudes in the cell have widely varying phase. The averaged cell will probably have an incorrect amplitude and phase, but perhaps not so incorrect as to stand out from its neighbors. A slight change in the grid sampling, however, may suddenly make some of the bad samples very visible while disguising others. Thus `UFLAG` should be run more than once using different u and v sampling.

Perhaps a useful sequence to flag data sets would be `FTFLG` to flag spectral channels that are generally bad, one or two passes of `RFLAG` using the cross-hand data, a pass of `RFLAG` using the parallel-hand data, and then a `CLIP`. Since `RFLAG` gets its power from examining data over short time intervals, it is probably best to run `CLIP` afterwards to catch any egregious samples missed by `RFLAG`. If it is run first, the contiguity of the samples may be lost allowing more bad data to slip through. `UFLAG` using the output flag table from this sequence could be a very powerful way to look for any residual problems either before, or after, some initial imaging.

Figure 5 shows an example of a data set that has passed through `FTFLG`, `RFLAG`, and clipping using `WIPER`. The vector average image looks fine, the phase image is a big mess (to human eyes), but the scalar and scalar minus vector images reveal a region of the uv plane that differs significantly from the rest. Often this can occur from phase discrepancies between the samples contributing to the grid cells. In this case however, phases are noisy everywhere in this target source while this region has higher than normal visibility amplitudes. Many of the grid cells have hundreds of samples contributing from only one or two baselines and a few spectral channels. A `VIEW ONLY` followed by a `FLAG ALL` would probably do almost the same as an `AUTO FLAG VIS` and do it very much faster. An `AUTO FLAG VIS` was run on a modest sub-image near the center of this image and, eventually, produced several hundred thousand entries in the `FC` table. Note too that any algorithm that depends on a cell standing out from its immediate neighbors will fail in the present situation.

4 Acknowledgments

The author wishes to acknowledge Emmanuel Momjian for numerous useful conversations and for providing the data set used both to test `UFLAG` and to make the figures for this Memo. It is a CASA tutorial data set initially observed on the EVLA by Sanjay Bhatnagar 23 August 2010. The observations are of a weak, extended supernova remnant G55.7+3.4 and the spectral window appearing in all figures is centered upon 1.32 GHz.