



Title: Robust Noise Estimators for CASA – Requirements Capture	Authors: Selina, Mehringer, Tueben, Kern, Ott	Date: 07/30/2014
Document No. (N/A)		Revision: 0.5

## **Requirements Capture: Robust Noise Estimators for CASA**

### **I Change Log**

Revision	Date	Sections	Changes
0.5	7/30/14	I, 4.2, 4.3	Added Change Log. Revised Section 4.2 to clarify that sorting is optional. Also noted that statistics are computed on the truncated and symmetrized data set. Corrected spelling of Chauvenet in Section 4.3. Removed goal of reporting progress at the task level (per Jeff). Clarified pending decision on statistics framework in second bullet of Section 5.

### **2 Introduction**

This request originates from Peter Tueben of the ADMIT project. The proposal is to add the following features to `imstat()` within the Image Analysis module of CASA:

- (1) Modify the `imstat` task interface to allow the selection of filters to improve noise estimating statistics.
- (2) Add three new filter methods for noise estimation.

Peter Tueben's initial request is included in Appendix A for reference and is also captured in CAS-6716. An elaborated set of requirements pertinent to implementing these new features are captured below.

There are many methods to get a better signal-free noise statistic. We propose three here, with the intent that `imstat` be adaptable for new methods and/or new parameters. Each method is effectively a filter on the image and `imstat` will provide already-implemented statistics on the filtered data set.

The premise underlying this implementation is that the signals of interest are contained within the outlier values in the data set while the noise has a Gaussian distribution. Removing the outlier values suppresses the signal while preserving the noise. Any analysis on the remaining values should better represent signal-free noise statistics.

Note: the MAD (which is available through `imstat`) is a competent RMS estimator. Formally  $\sigma = 1.4826 * \text{MAD}$  for a normal distribution.

There is a tradeoff between the use MAD on the original distribution, or finding outliers and recomputing the statistics. The addition of these three methods will give users choices. Depending on the use case, a particular method may be preferred.

Note that it is anticipated that the user who wishes to estimate the noise in the image using these new methods would run `imstat` twice. First to produce statistics on the full, unfiltered data set and the second time to produce statistics on the filtered sub set of the data that is believed to better represent the noise within the image.

### **3 Imstat() Interface**

`Imstat`'s interface will be adapted through new keywords describing the filter method (a string) and associated parameters (numerical values) which influence the behavior of the selected method.



Title: Robust Noise Estimators for CASA – Requirements Capture	Authors: Selina, Mehringer, Tueben, Kern, Ott	Date: 07/30/2014
Document No. (N/A)		Revision: 0.5

While the focus is on the addition of three specific new methods, the interface change shall permit additional methods to be added using new string values and keywords, if necessary.

New parameters names, consistent with the existing convention for CASA keywords, will be added to select a filter method and pass associated input parameters. Example:

```
> imstat(..., filter="hinges-fences"1, [filterparams]=-1.0)
```

filter<sup>2</sup> – An input parameter (string) to define the filter method/algorithm employed.

[filterparams] – Filter specific keyword(s) and input parameter(s) providing additional input variable(s) to control the behavior of the method/algorithm. The unique keywords for each new robust RMS filter method/algorithm are described below.

The default values of all new keywords shall maintain the current behavior of imstat and ia.statistics().

## 4 Noise Estimating Filter Algorithms

### 4.1 Hinges & Fences

The Hinges and Fences algorithm is defined by Tukey in Exploratory Data Analysis<sup>3</sup>. It is a robust RMS, which uses the quartiles of a distribution and removes a fraction of the two tails. Imstat then recomputes the standard statistics (mean, median, rms, etc.) for the filtered data set.

The inputs, transformations, and outputs associated with the method are described below.

#### 4.1.1 Inputs

```
> imstat(..., filter="hinges-fences", fence=1.0)
```

filter – An input parameter (string) to select the hinges & fences algorithm. “hinges-fences” is the proposed string.

fence – A single parameter (float) which will define the robustness of the filter by determining the locations of the “fences”. See parameter *f* below.

---

<sup>1</sup> Like most parameters in ia tasks that take identifying strings, filter (or whatever it ends up being called) will accept minimum unambiguous matches. E.g., if no other supported strings start with "h", filter="h" will be the same as filter="hinges-fences".

<sup>2</sup> The string ‘robust’ will not be an allowable entry for any new keywords. Although imstat does not have a parameter called robust, ia.statistics does (imstat sets this to True when it calls ia.statistics()). Robust=True in ia.statistics() means calculate things like the median, quartiles, etc. Any new input parameter added to imstat must also be added to ia.statistics. So the use of ‘robust’ to define a new keyword or method would be confusing in this context. Robust in ia.statistics() to remain a bool.

<sup>3</sup> [http://en.wikipedia.org/wiki/Exploratory\\_data\\_analysis](http://en.wikipedia.org/wiki/Exploratory_data_analysis)



Title: Robust Noise Estimators for CASA – Requirements Capture	Authors: Selina, Mehringer, Tueben, Kern, Ott	Date: 07/30/2014
Document No. (N/A)		Revision: 0.5

The fence factor controls how much of the tails are cut out. It requires that the data to be sorted, and since this comes at a cost, this should be an option.

A negative value shall indicate that the full dataset be used. This shall be the default in order to preserve the current behavior of `imstat()` and `ia.statistics()`.

Positive values of  $f$  will be used as described in section 3.1.2.

#### 4.1.2 Transformations

$Q1$ ,  $Q2$  and  $Q3$  are the boundaries between the 4 quartiles of the data set, such that  $Q2$  is the median.  $D=Q3-Q1$  is the width of the central portion of the PDF where half the raw measurements are.

For a robust measurement one now only considers values between:

$$Q1 - f*D \text{ and } Q3 + f*D$$

As defined in Tukey's Exploratory Data Analysis;  $Q1$  and  $Q3$  are the *hinges*,  $Q1 - fD$  and  $Q3 + fD$  are the *fences*. Normally  $f=1.5$ , and for large values of  $f$  the full distribution with all its biases will be recovered. If  $f=0$  only half the data are considered.

#### 4.1.3 Outputs

`Imstat` will output any of the keys currently available in the task, based on an analysis of the values the filtered data set.

Two new keys,  $q1$  and  $q3$ , shall be added to output the values of  $Q1$  and  $Q3$  for user validation.

### 4.2 Fit to Half

This method is only applicable to a normal distribution. Values to the right of the mean or median are discarded and statistics are then recomputed on the truncated and symmetrized data set.

#### 4.2.1 Inputs

`> imstat(..., filter="fit-half", center="mean" lside=true)`

**filter** – An input parameter (string) to select the Fit to Half method. “fit-half” is the proposed string.

**center** – An input parameter (string) to select the best estimate of the center of the distribution. “mean”, “median” or “zero” are allowable values.

**lside** – An input parameter (bool) indicating if the left side of the selected center should be used. True indicates to the left of the selected center, while false indicates to the right of the selected center. Left will be the default behavior.

#### 4.2.2 Transformations

The mean and median are computed and the data set is optionally sorted. Values to the selected side of the selected center are retained, while the remaining pixel values are discarded.



Title: Robust Noise Estimators for CASA – Requirements Capture	Authors: Selina, Mehringer, Tueben, Kern, Ott	Date: 07/30/2014
Document No. (N/A)		Revision: 0.5

Statistics are recomputed on the reduced data set, and adjusted to reflect that only half the distribution was analyzed. The distribution is assumed to be symmetrical about the selected center for this purpose. E.g., the mean and median would by definition be the selected center.  $\sigma$  would be relative to this adjusted mean.

### 4.2.3 Outputs

Imstat will output any of the keys currently available in the task, based on an analysis of the values the filtered data set.

## 4.3 Chauvenet's Criterion

Chauvenet's Criterion<sup>4</sup> is another statistical method for determining outliers in a data set. The method iteratively removes a fraction of points that fall outside of  $z * \sigma$  until either a max number of iterations have been reached or no more points are removed.

### 4.3.1 Inputs

```
> imstat(...., filter="chauvenet", zscore=-1)
```

filter – The input parameter (string) is used to select the Chauvenet method. "chauvenet" is the proposed string.

zscore – The input parameter (float) used to define z, the number of standard deviations a data point may be from the mean and be retained. Any negative value will direct the method to use Chauvenet's criterion to compute z. Any positive values will be used directly for z. The default will be to use Chauvenet's criterion.

### 4.3.2 Transformations

Remove the points that fall outside of  $z * \sigma$ .

If z was user specified as an input, use this value. If a value of -1 was passed, then calculate z using Chauvenet's criterion. In this case, the z-score is a function of the number of points in the sample, where:

$$P_z = 1 / (2n)$$

The corresponding z value is then most easily determined through the use of a look-up table.

Then  $z_x$  for a given point x in the data set is given by:

$$z_x = (x - \mu) / \sigma$$

Points are discarded if  $|z_x| > z$ .

---

<sup>4</sup> [http://en.wikipedia.org/wiki/Chauvenet%27s\\_criterion](http://en.wikipedia.org/wiki/Chauvenet%27s_criterion)



Title: Robust Noise Estimators for CASA – Requirements Capture	Authors: Selina, Mehringer, Tueben, Kern, Ott	Date: 07/30/2014
Document No. (N/A)		Revision: 0.5

The method shall be iterative, with new values of  $n$ ,  $\mu$  and  $\sigma$  computed each iteration.  $z$  would also be computed if the user elected to apply Chauvenet's criterion. The filter shall iterate through the data **TBD** times by default.

If no points are removed in the last iteration, the loop shall exit and report the requested statistics. It shall also report early termination of the loop to the CASA logger.

#### 4.3.3 Outputs

Imstat will output any of the keys currently available in the task, based on an analysis of the values the filtered data set.

### 5 Other Requirements or Considerations

- Any new keyword arguments or keys (parameter names) will be consistent with current CASA naming conventions. Underscores and other special characters are not used, and abbreviated names are preferred.
- It is desirable that new methods, described in Section 3, be implemented in such a way that the algorithm may be called from other tasks (in addition to imstat) in future iterations of CASA. E.g., it may be desirable to filter data elements other than image cubes, such as a set of visibilities. **(TBC** - This will be dependent on a decision to advance a new CASA statistics framework)
- New filter methods shall not support complex-valued images. The filters rely on parameters such as quartile values which may not be well defined for a set of complex numbers. `ia.statistics()` and `imstat` currently only support float-valued images and this limitation shall remain.
- If a new filter method is invoked, all keys will report values for the filtered population. New keys may be added, but no keys will be renamed or redefined as part of this proposal in an effort to maintain backward compatibility.
- The logger shall provide a clear indication that the selected filter was applied when calculating the reported values.
- There is no need to preserve a record of the filtered data set after `imstat` exits.
- There is no need to preserve a record of the outlier values removed with the filter. If the outlier values are of interest, generating a mask is a preferable approach of identification (and is out of scope).
- If filter parameters are included in the call, but are either not applicable to the selected filter or a filter is not selected, `imstat` will silently ignore the irrelevant parameters.



Title: Robust Noise Estimators for CASA – Requirements Capture	Authors: Selina, Mehringer, Tueben, Kern, Ott	Date: 07/30/2014
Document No. (N/A)		Revision: 0.5

## Appendix A: Initial Requirements Capture for Feature, based on Peter Teuben Note

(11 June 2014):

There are many methods to get a better method of signal-free noise statistic. We propose a few here, but with the intent that `imstat` can easily be adapted (by developers) for new methods and/or new parameters. Ideally something through two keywords describing the method (presumably a string) and associated parameters (presumably a mix of numbers)

1) The first one that comes to mind is a robust RMS, which uses the quartiles of a distribution and removes a fraction of the two tails and recomputes the standard statistics (mean, median, rms etc.). Tukey coined these hinges and fences (see below) in his Exploratory Data Analysis (EDA) book. Note that "robust" is a very generic term, in fact used in `ia.statistics()`, but this is a specific implementation.

There is one parameter governing this one, the robustness factor, which controls how much of the tails you cut out. It requires the data to be sorted, and since this comes at a cost, this should be an option. However, it should also be noted the default in `imstat()` computes a median already, so it has probably already sorted the data.

Let's assign  $Q1$ ,  $Q2$  and  $Q3$  the boundaries between the 4 quartiles, such that  $Q2$  is the median. Call  $D=Q3-Q1$  the width of the central portion of the PDF where half the raw measurements are.

For a robust measurement one now only considers values between

$$Q1 - f \cdot D \text{ and } Q3 + f \cdot D$$

and recomputes the statistics from those points. The current version of `imstat()` does compute the "quartile" (called  $D$  here), but does not export  $Q1$  or  $Q3$ , and thus becomes impossible even run `mask` and recompute this statistic.

Normally  $f=1.5$ , and for large values of ' $f$ ' the full distribution with all it's biases will be recovered. At the other extreme  $f=0$  only uses half the data are considered. In our sample ALMA data, slightly larger values of  $f$  were preferred.

"hinges" and "fences" as defined in Tukey's Exploratory Data Analysis;  $Q1$  and  $Q3$  are the hinges,  $Q1-fD$  and  $Q3+fD$  are the fences.  $f=1.5$  and  $f=3$  are often used for the inner and outer fences. See e.g. the `rstat` and `robust_rms` code in IDL's `astrolib`)

Example?

```
imstat(....,rms_method='robust',rms_pars=1.5)
```



Title: Robust Noise Estimators for CASA – Requirements Capture	Authors: Selina, Mehringer, Tueben, Kern, Ott	Date: 07/30/2014
Document No. (N/A)		Revision: 0.5

Note: the MAD (which is available through imstat) is actually a pretty good RMS estimator. Formally  $RMS = 1.4826 * MAD$ .

In the end, you can ask, what's better, use MAD on the original distribution, or find outliers and recompute the statistics. I believe we should give users a choice, depending on the situation, a particular method may be preferred.

Other methods come to mind as well, all work well in a normal distribution. When the noise is not gaussian (e.g. in a cleaned map from badly sample UV data), all bets are off.

2) fit to half (the negative side) the signal. Basically you can replicate each negative point to be positive and do the stats on that?

3a) iteratively removing a fraction of points that fall outside of  $k * \sigma$  until either a max number of iterations have been reached or no more points are removed. K is the parameter, could be 3 or 4 or 5, the user chooses. This is very ad hoc and not statistically sound. See 3b) for better alternatives.

3b) Proper Chauvinet criterion; this is like (3a) except the k factor should depend on the number of points in the sample. After all, for small number of points you cannot pick K too low, or it will reject noise. Variations on this theme are plenty, Grubbs outlier test, Peirce criterion, Dixon's Q test, etc.

It would also be good if imstat could then give a probability that the distribution is normal.

Each of these have their respective strong and weak points.

#### NOTES:

- The cookbook doesn't list the stretch= and append= keywords that the inline help shows
- The cookbook lists the async= keyword, which is not in the inline help
- help(imstat) claims the quartile is half the difference between the Q3 and Q1 (75% and 25%), but it's actually the full difference. The text says

quartile - the inter-quartile range. Find the points which are 25% largest and 75% largest (the median is 50% largest), find their difference and divide that difference by 2.

can be easily fixed by leaving out the last "and divide that difference by 2."  
The same fix needs to be applied to help(ia.statistics)



Title: Robust Noise Estimators for CASA – Requirements Capture	Authors: Selina, Mehringer, Tueben, Kern, Ott	Date: 07/30/2014
Document No. (N/A)		Revision: 0.5

## Appendix B: Probability of Gaussian Distribution

A number of the proposed noise reduction methods rely on normal, Gaussian distribution of the signal within the data set. As such, it would also be advantageous if a task could give a probability that the distribution within a data set is normal.

This functionality is has been removed from the scope of this development target, given poor fit within the scope and architecture of imstat (Fitting should not be done within imstat). However, it is retained here for future consideration.

Should this be implemented, the project will have to determine an algorithm to calculate the probability that the data set has a Gaussian distribution. One option:

- Calculate the median and standard deviation of the data set.
- Fit to a Gaussian of same median and standard deviation.
- Calculate the residuals.
- Sum the square of the residuals and perform Pearson's chi-squared test, computing the probability of observing this difference.<sup>5</sup>

---

<sup>5</sup> [http://en.wikipedia.org/wiki/Goodness\\_of\\_fit](http://en.wikipedia.org/wiki/Goodness_of_fit)