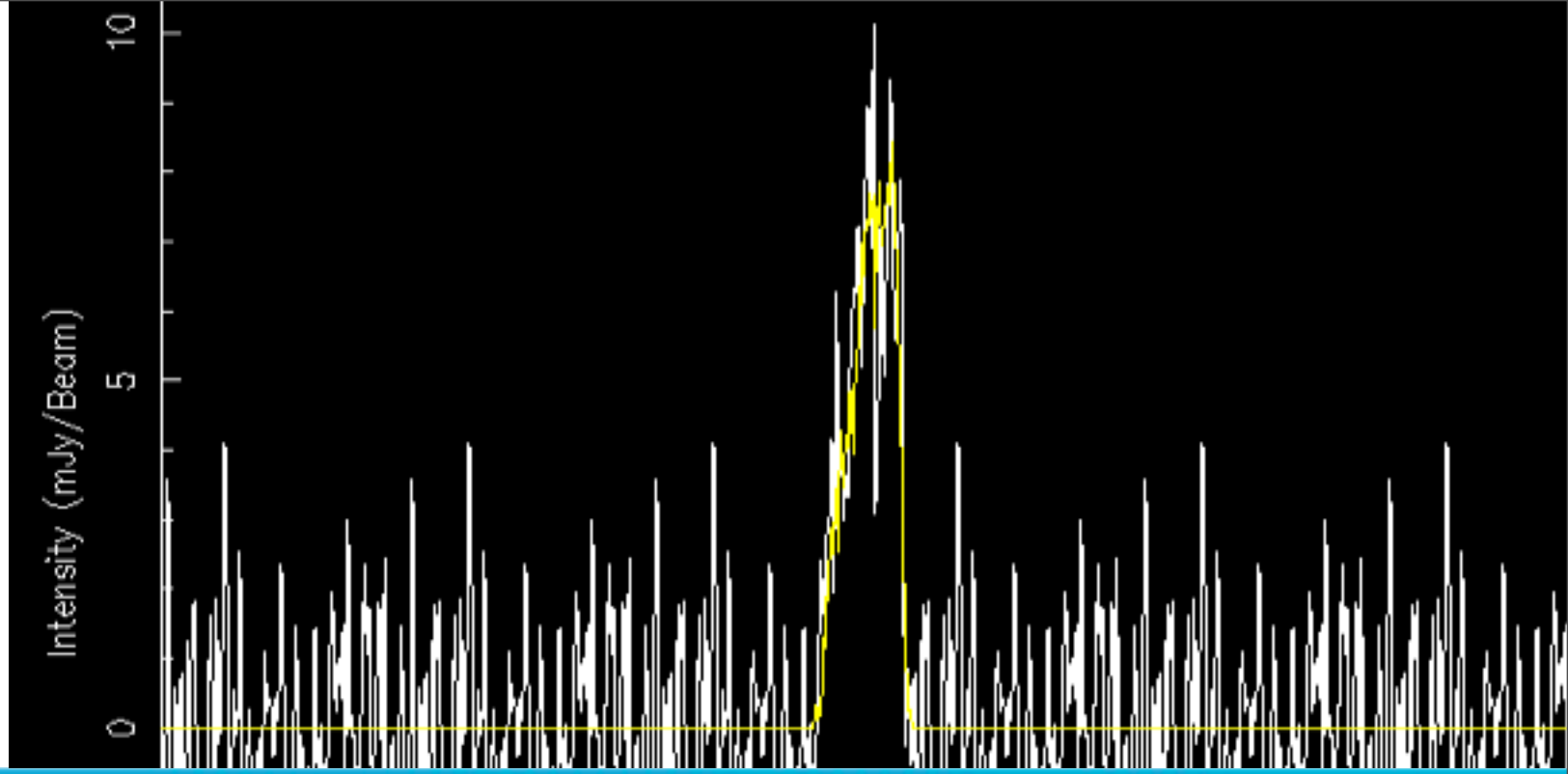


www.csiro.au

HI source finding algorithms

Comparing the general purpose Duchamp algorithm to a purpose built HI source finding algorithm

CSIRO

Talk Outline

- Common elements of source finding algorithms
- The Duchamp algorithm
  - Algorithm
  - Strengths
  - Draw-backs
  - Improvements
- An alternative HI source finder algorithm
  - Key differences
  - Algorithm
- Preliminary work
- Conclusion

Common elements of general source finding algorithms

- Define/calculate detection and growing criteria
  - Thresholds or False Detection Rate
- Pre-condition data
- Scan through data and apply detection criterion
- Grow detections using growing criterion
- Merge detections
- Apply size criterion

The Duchamp Algorithm

Duchamp: Algorithm

- Pre-condition data (optional)
  - Blank pixel removal
  - Baseline removal using wavelet reconstruction
  - Define channels to ignore

  - Wavelet reconstruction using a' trous wavelet procedure (priority)

    OR
  - Smooth in frequency space

    OR
  - Smooth spatially

- Set detection and growing criteria
  - User specified (priority)

    OR
  - FDR or calculated from globally determined mean and rms values

Duchamp: Algorithm

- Raster scan data
  - Travel along planes or channels and apply detection criterion
  - If a voxel satisfies the detection criterion
    - Flag it
    - Check it's proximity to all previous detections and merge accordingly
      - Can be turned off for efficiency, but default is ON.
- Merge detections
  - Apply proximity test (again) to all detections
- Grow detections
- Merge detections again
- Apply size criterion
  - Can be done prior to first round of merging

Depiction of raster scanning

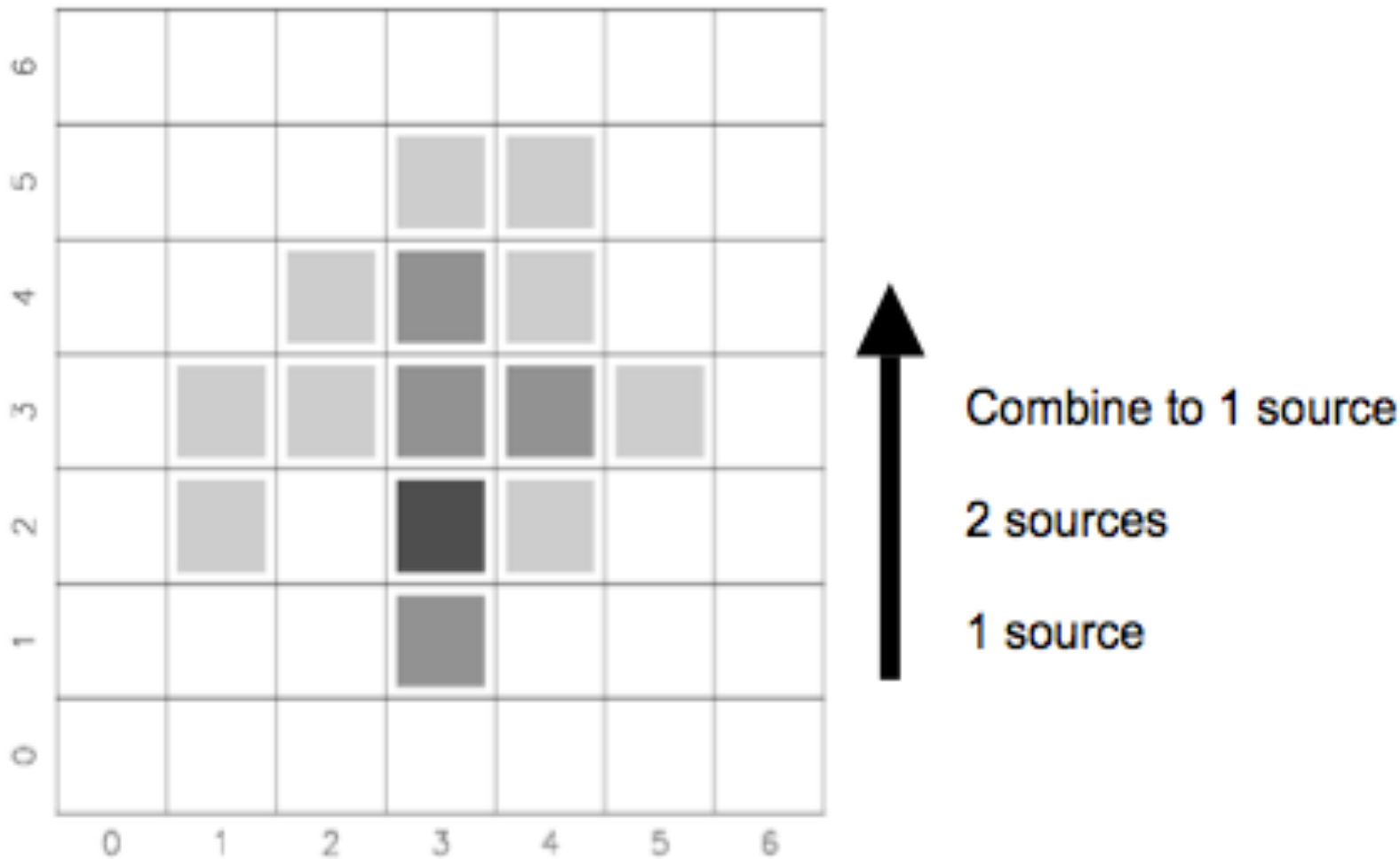


Image credit: Matt Whiting

Depiction of threshold usage

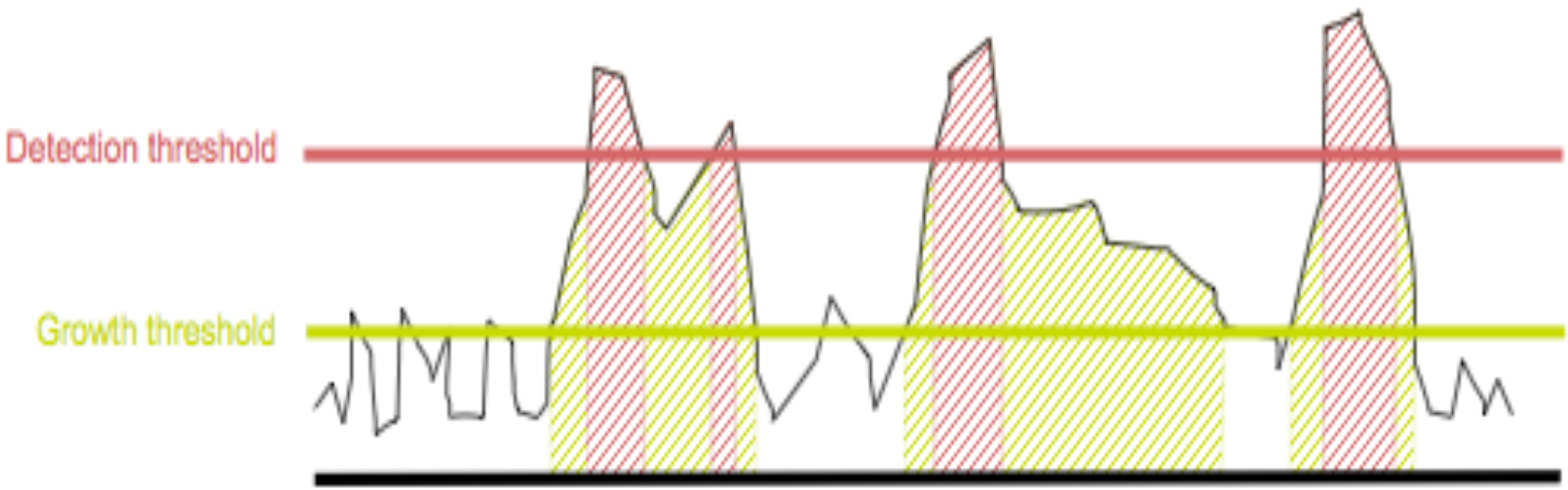


Image credit: Matt Whiting

Duchamp: Strengths

- A truly general source finding algorithm
- Makes minimal assumptions
- Extremely flexible source detection
- IT EXISTS! and IT WORKS!
- Output is feature rich

Feature rich output

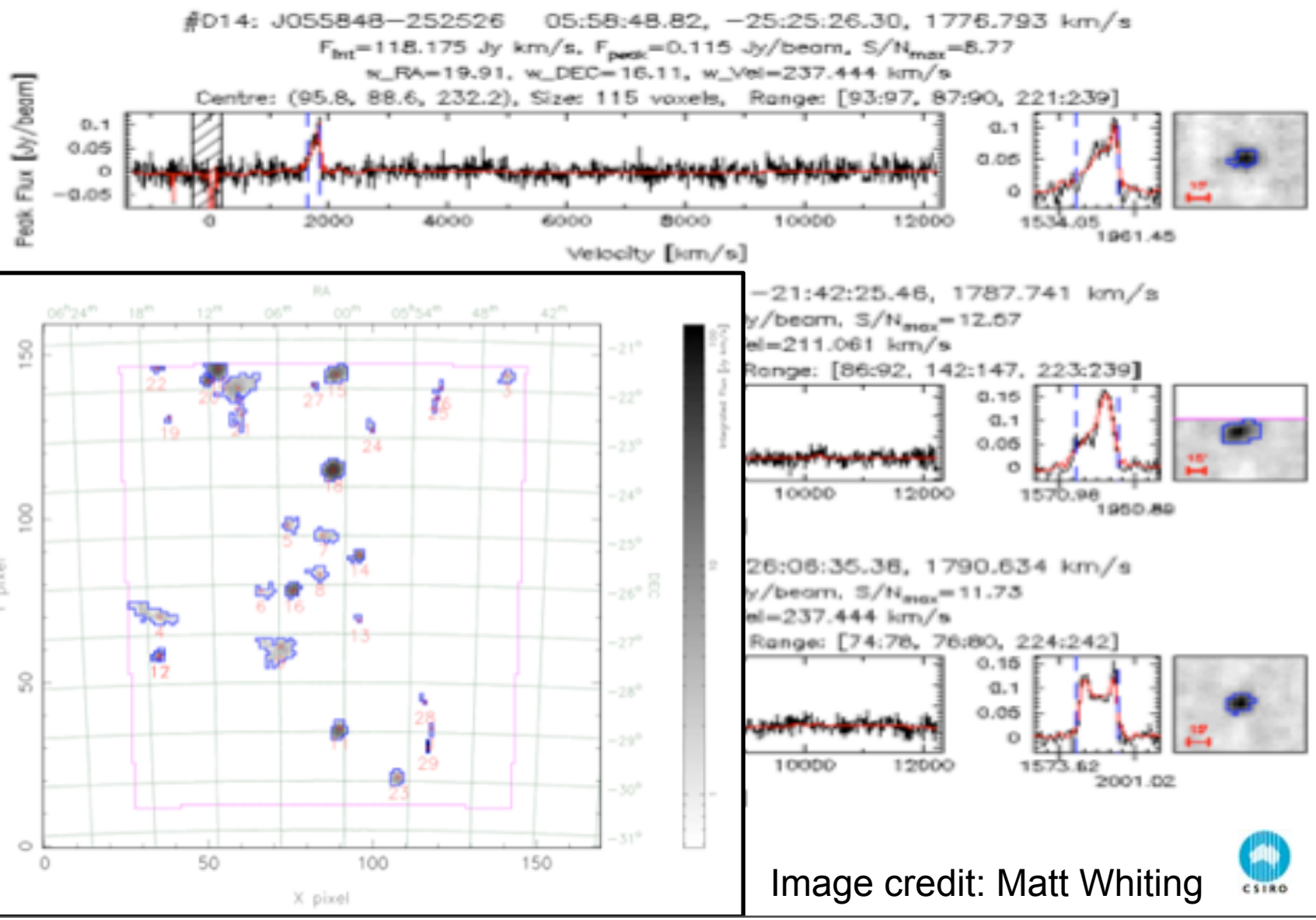


Image credit: Matt Whiting

Duchamp: Draw-backs

- Efficiency decreases with the number of detections
  - Searching for faint sources is very inefficient
- Default is to run a merging routine every time a detection is made
  - Compared to every! previous detection
- Merging is carried out multiple times
- Size criterion is applied at the very end
  - Inefficient but necessary
- Global detection and growing criteria are used
  - Noise varies throughout the cube
  - Detect 'crud' in some regions, miss detections in others
- Multiple detections of single source
- Detection threshold doesn't correspond to source S/N level
  - $S/N_{voxel} = 2\text{-}5 \times S/N_{source} / \sqrt{m}$, where m is the channels covered by source

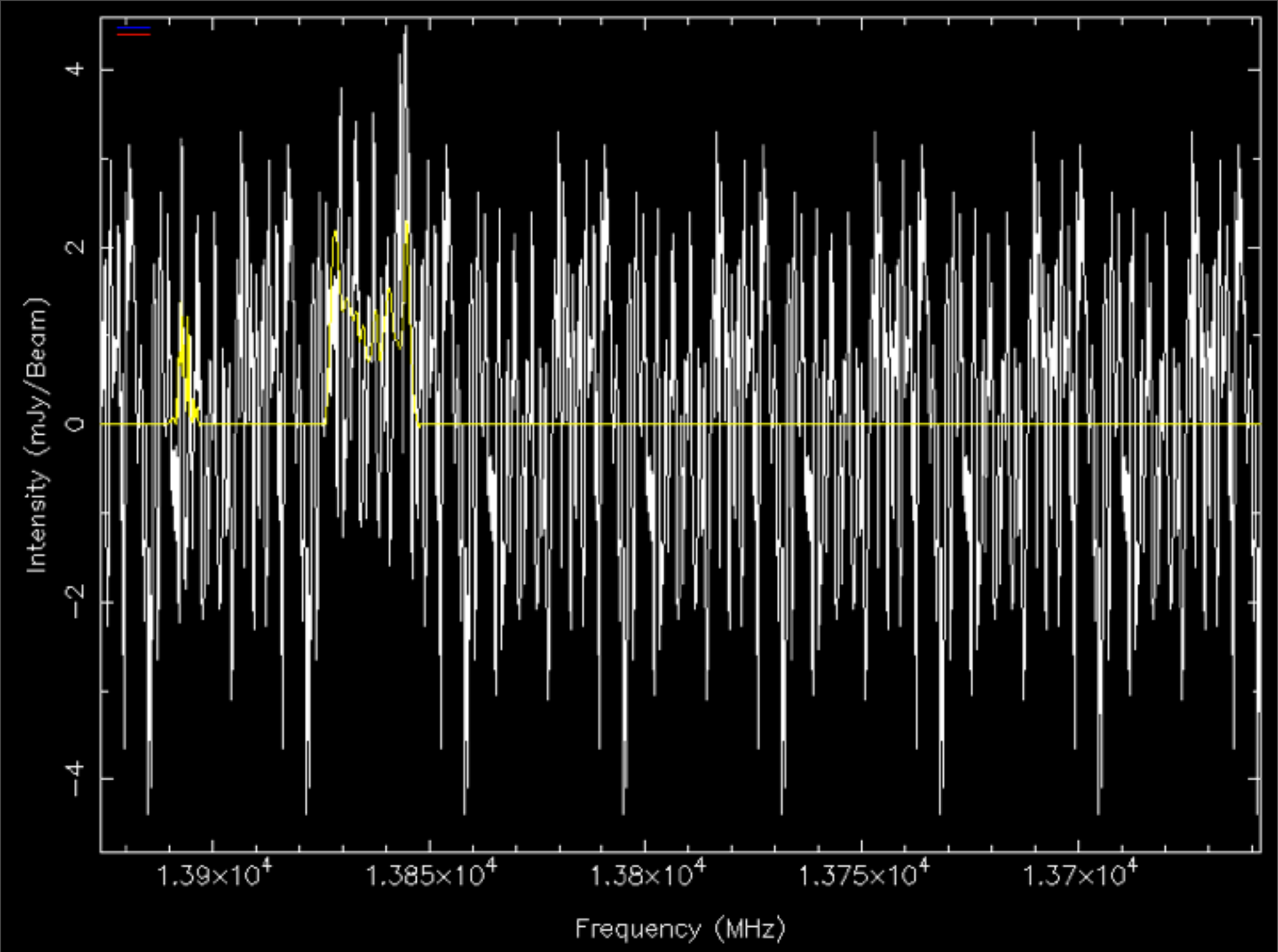
Duchamp: Improvements

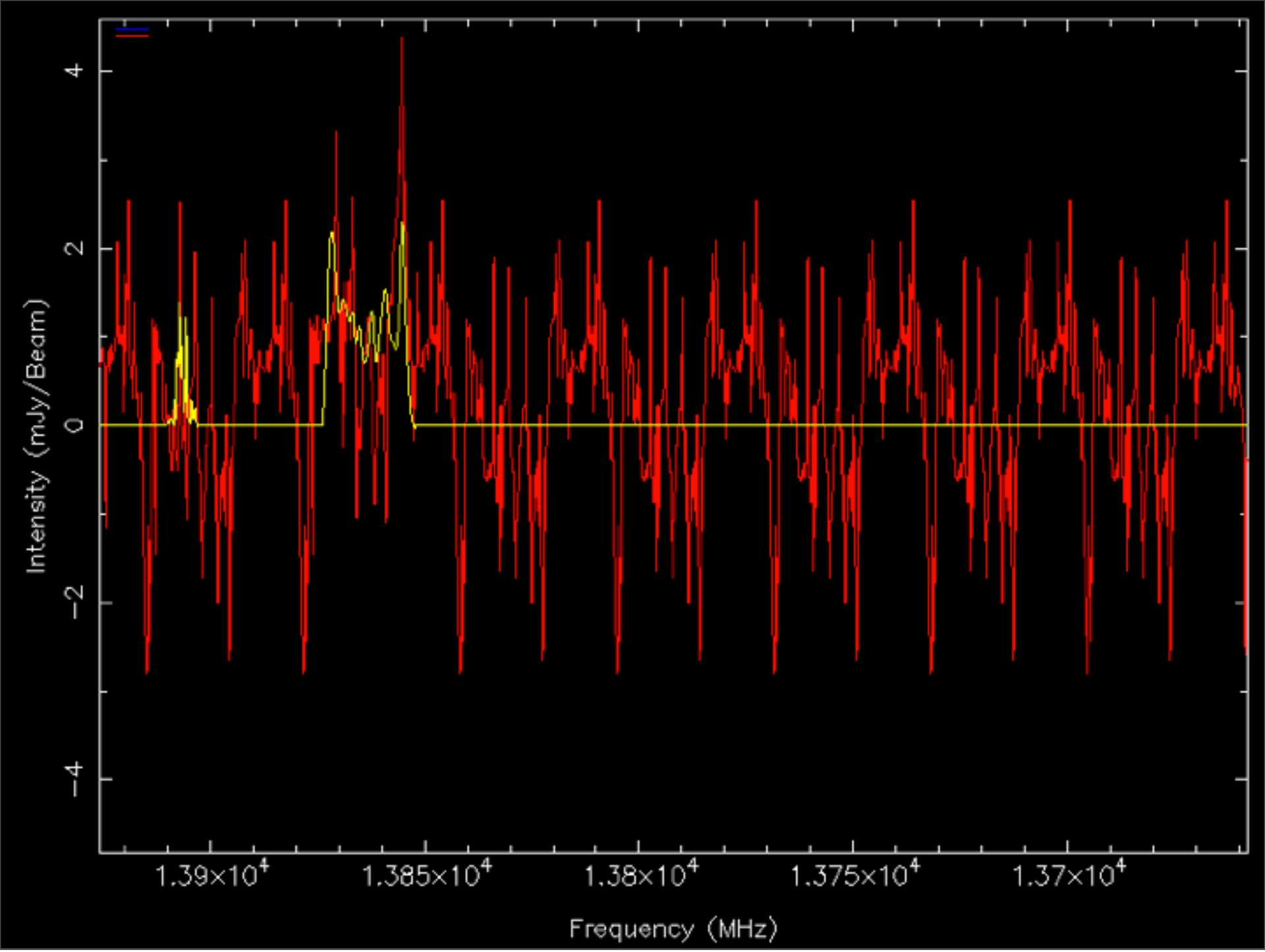
- Sub-sample channels when raster scanning
  - Sampling set to size criterion
  - Minimise detections that eventually would fail size criterion
- Define a data volume to check for previous detections
  - To be used when initial merging not turned off
- Grow detections, merge (just the once), apply size and detection threshold criteria
  - Apply growth criterion out to merging distance to fold in initial merging pass
- Use a local measure of noise

A purpose built HI source finder algorithm

Key differences

- Treat datacube as a set of spectra rather than a collection of voxels
- Use shape information rather than a detection threshold
  - Can potentially detect faint objects that a detection threshold would miss
  - Recover ‘true’ extent of source compared to using growth threshold
- Implicit is the assumption that every detection has a discernible shape
- Assume that we have a well defined psf


Intensity (mJy/Beam)
4
2
0
−2
−4
1.39×10^4
1.385×10^4
1.38×10^4
1.375×10^4
1.37×10^4
Frequency (MHz)


Intensity (mJy/Beam)
4
2
0
−2
−4
1.39×10^4
1.385×10^4
1.38×10^4
1.375×10^4
1.37×10^4
Frequency (MHz)

Key differences

- Treat datacube as a set of spectra rather than a collection of voxels
- Use shape information rather than a detection threshold
  - Can potentially detect faint objects that a detection threshold would miss
  - Recover ‘true’ extent of source compared to using growth threshold
- Implicit is the assumption that every detection has a discernible shape
- Assume that we have a well defined psf

Specific HI source finding algorithm

- Divide data cube amongst CPUs
- Clean side-lobes from data cube
- Sub-sample the data cube
- For a given spectrum
  - Pre-condition using iterative median smoothing
  - Use wavelet analysis to construct the noise spectrum + baselines and remove
  - Detect objects using shape information
    - Cross-correlation?
    - Wavelet analysis?
    - Gamma test? (Even if just for measure of noise in spectrum)

Specific HI source finding algorithm

- For each detection, scan neighbouring positions in spiral pattern to determine the volume containing the detection
  - Have a frequency range to process for neighbours
  - Well-known (and SOLVED) mouse navigating a maze problem
    - The solution provides a ‘shrink-wrapped’ volume
- Merge detections
- Merge CPU results
- Apply size criterion
  - May have been incorporated earlier

Preliminary work

- Prototyping iterative median smoothing as a pre-conditioner
- Using the WSRT simulated datacube
- Comparing to performance of Hanning filtering
- Results
  - Quantitatively, residuals cf. input spectrum are reduced by ~20-40%
  - Comparable to Hanning filtering, but doesn't add/remove structure in the cases where Hanning filtering does

Conclusion

- Duchamp is a great general purpose source finder
- The efficiency of Duchamp could be improved
- Proposing to treat datacube as a set of spectra and use shape information to find HI sources
- Development underway