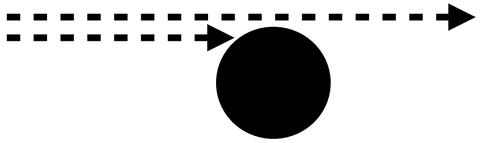
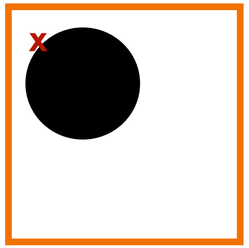
Notes on updates to centroid code

Spot Detection

Previous method



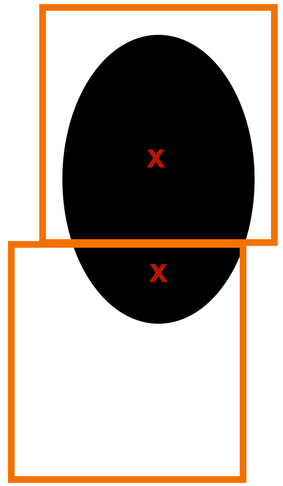
Search image to find a point above the threshold.



- Search in a box, marking all points above the threshold as part of the spot.
- Does not check for contiguity.
- Calculate a running first and second moment
- Moments are input to the windowed centroid routine

Problem:

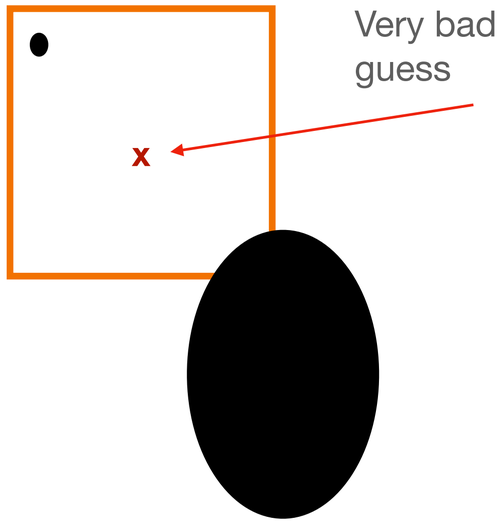
- If the spot is larger than the box, two initial spots will be detected.
- Centroids may converge to almost the same position, but they may be several pixels different, depending on the shape of the PSF



Two initial first guesses

Make the Box larger?

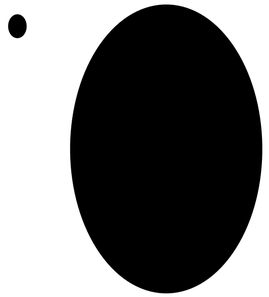
- A very large box can run into problems with hot pixels which give wrong centroids, based on a very bad choice of initial position.
- This is particularly a problem when setting the threshold fairly low, when you may pick up noise.
- If the box is too large, you may pick up more than one spot, for collided cobras.



Very bad first guess

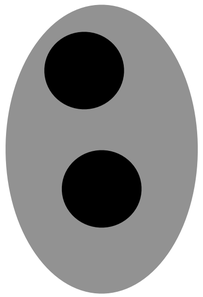
Solution:

- Use routine that finds contiguous spots above the threshold
- Not limited by spot size
- Standard “flood-fill” algorithm



Potential failure

- Too high threshold and double peaked PSFs can find two spots
- Too low a threshold / high background will detect a giant spot (eg, dome lights are on)
- In rare cases, with the current PSF, two touching cobras can have blended spots if the threshold is too low.

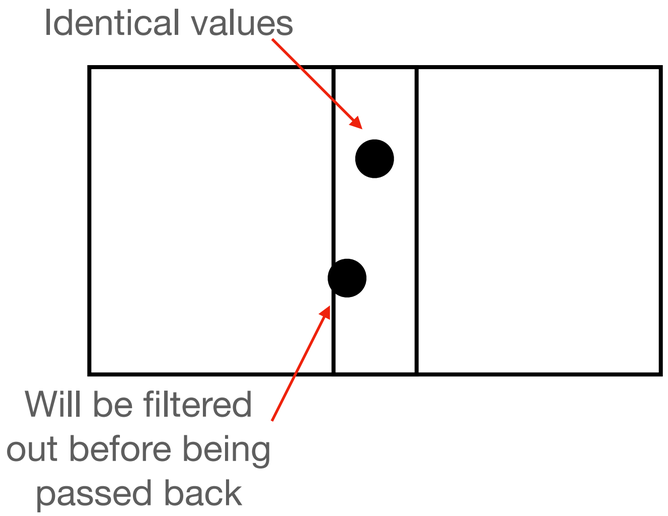


- Detection now depends on threshold only (no box size)
- Filter by minimum number of pixels to filter out noise / hot pixels before sending to windowed threshold
- Will not return duplicate spots

Centroiding and Filtering

Windowed Centroid

- pass kernel size as an external input, based on typical spot size/shape. Value is set to 4,9 in the x/y direction (in pixel units), based on testing the reduction with the current PSF and different input kernels.
- we can't calculate on the fly for parallel mode, as sub-sections of the image have slightly different characteristics.
- no actor level change needed for this implementation
- two input parameters: size of box over which centroid calculated, kernel sizes
  - box size set to the point where increasing it does not increase accuracy
  - kernel size based on typical spot size/shape



Filtering Duplicated Spots

- updates to the detection method means that duplicate spots are no longer returned for a single image
- for parallel mode, duplicate spots in the overlap region are now identical

Tests

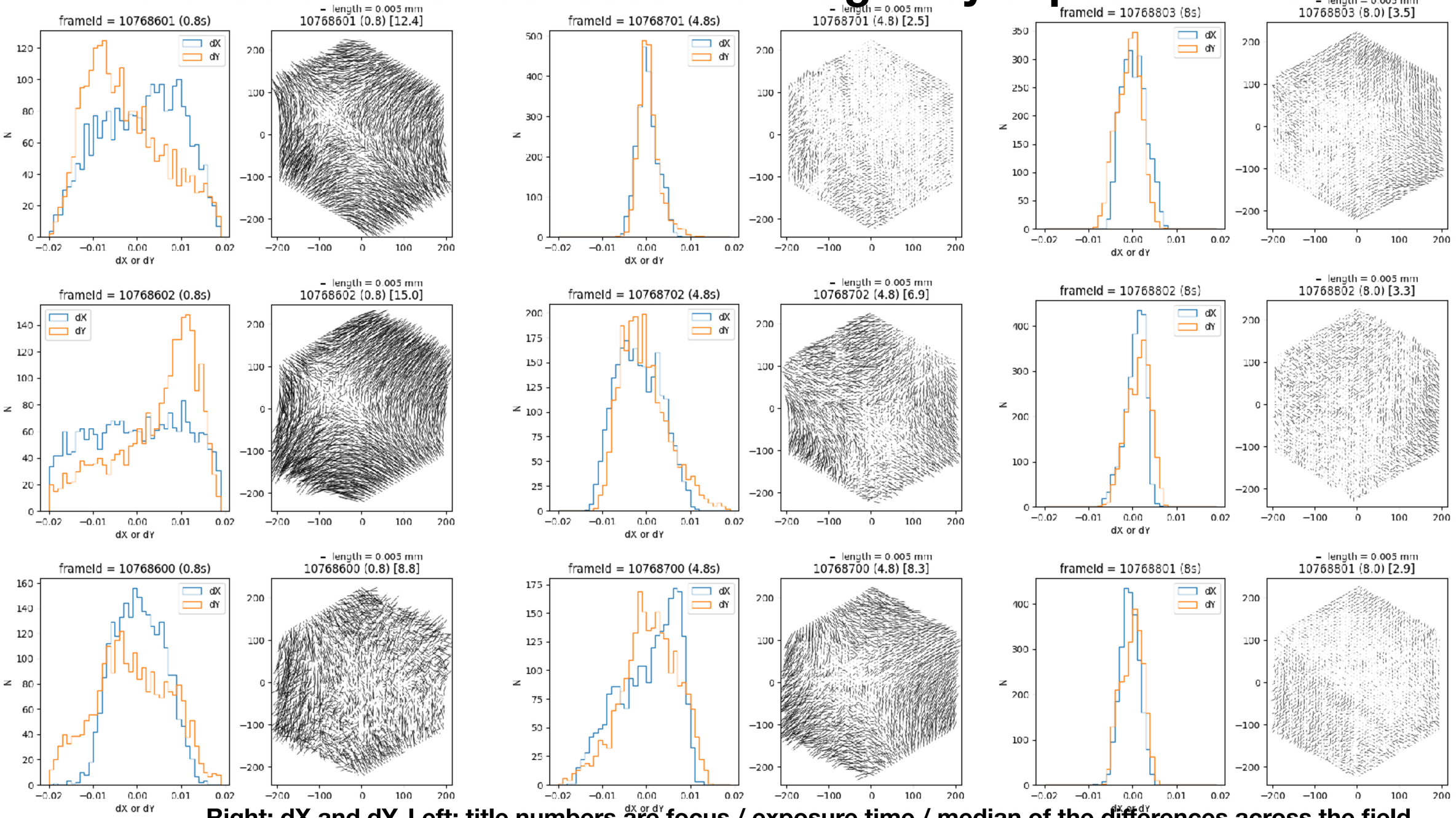
- checked for memory leaks via the leaks command.
- tested on real system with file input and test database
- actor runs smoothly
- tested on sets of consecutive images taken with a variety of parameters
  (Plots on following pages)
  - different exposure times
  - focus sweep data (different PSF sizes)
  - pre-dropping images have some issues with the current transformation code, as the geometry files change.

Processing time: ~ 0.3 seconds per frame (centroid detection only)

Plots on next pages

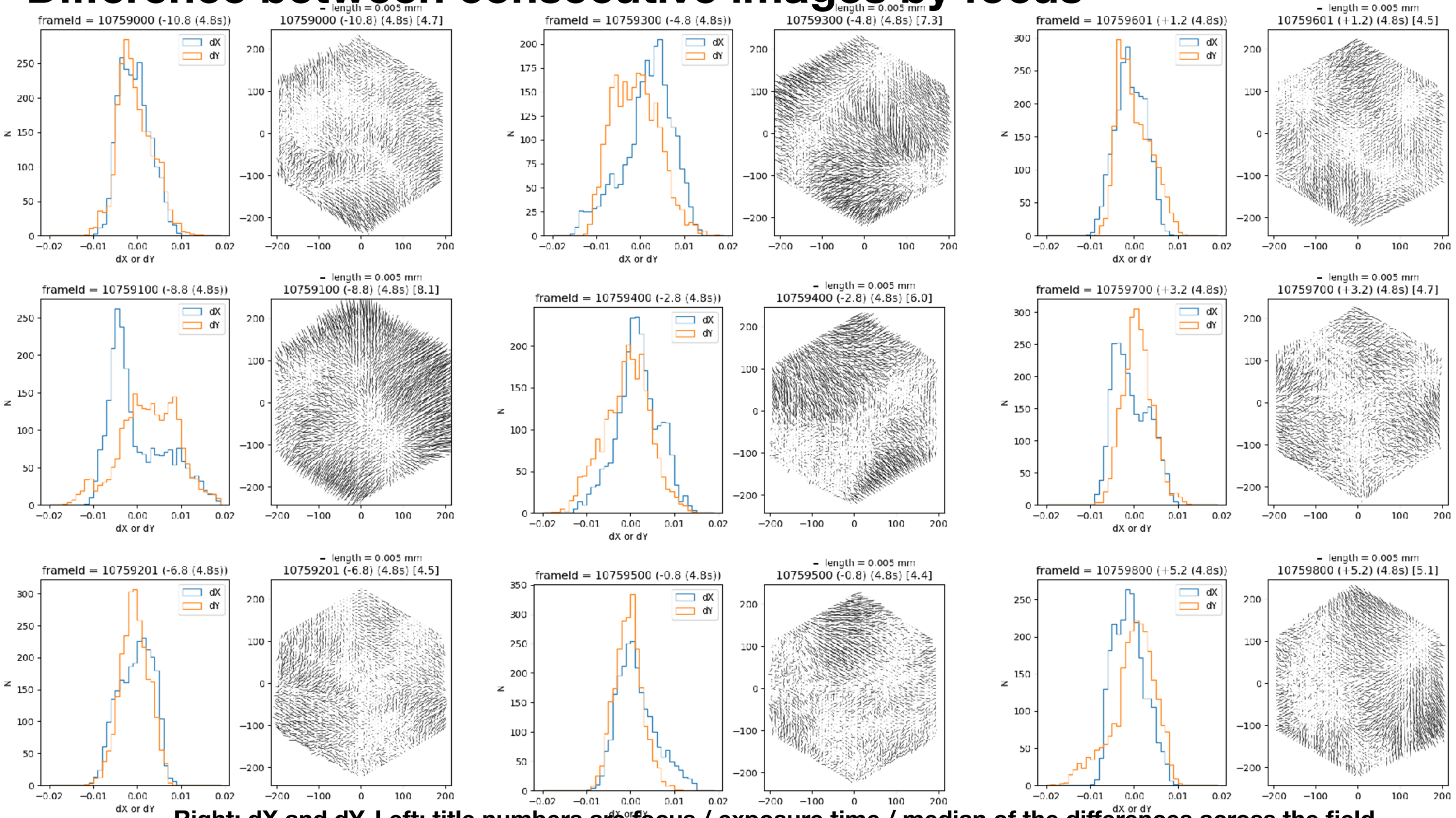
- all spots detected with all PSF sizes
- Each plot shows histogram of dx, dy between frames, and a quiver plot of the same
- sequence of plots with different exposure times (0.8, 4.8, 8) dominated by seeing effects
- sequence of plots with different MCS foci, using input kernel size for the centroiding, all 4.8s exposures; kernel size doesn't have a strong effect on the results (frame-to-frame variations are larger).

Difference between consecutive images by exposure time



Right: dX and dY. Left: title numbers are focus / exposure time / median of the differences across the field

Difference between consecutive images by focus



Right: dX and dY. Left: title numbers are focus / exposure time / median of the differences across the field