

For help regarding what a particular button does, **right-click** on it. This is an important information source that is always easily and immediately available.

For general information about the program (with usage tips), click on the **Documentation** button at the bottom of the left-hand panel. A pdf will be launched.

Note: there is a series of YouTube tutorials available. Go to <http://occultations.org/observing/software/pymovie> for easy access to these videos.

PyMovie version history

Version 2.7.6

- Fixed a bug introduced by the addition of size 71 and 91 apertures. The bug appears only when a saved aperture group is restored --- it gets the aperture size wrong. Anyway, it's fixed.

Version 2.7.5

- Added the option of selecting extra large apertures of 71 or 91. These values are NOT included in the 'preferences' list (too little usage to justify the work required on my part) so those sizes will always have to be selected deliberately from the drop-down aperture size list.

Version 2.7.4

- Fixed a bug that was exposed by an image with a preponderance (65%) of zero value pixels in the background (produced by a camera setting that 'clipped' the background).

Version 2.7.2

- Added selection of 1, 10, or 100 as step size for threshold spinner to better accommodate 12 and 16 bit videos --- reduce 'clicking'

Version 2.7.1

- The algorithm introduced in version 2.6.8 fails to perform adequately when the pixel histogram is insufficiently contiguous with too many missing values. A new algorithm has been introduced which seems to perform well no matter what the pixel histogram shape may be.
- Changed the default threshold for static aperture from 9999 to 99999 to better accommodate 12 and 16 bit videos.

Version 2.7.0

- A small change to the Robust Mean algorithm so that it can be used on "finder" images (which have floating point values) as well as single-frame images (which have only integer values).

Version 2.6.9

- Changed the 'help/info' for the Plot Robust Mean button to better explain the algorithm.

Version 2.6.8

- Changed the algorithm for calculating the average value of background pixels. This new algorithm replaced one that was excellent when background noise had a gaussian distribution, but had about a 1% error (too small by 1%) when the noise distribution was skewed. The histogram of a skewed distribution has a longer tail on one side than the other. It is common to find recordings that have background noise that is skewed, usually with a longer tail on the right (toward higher values). Sometime, perhaps due to a camera setting, background levels are 'clipped' and so appear skewed, sometimes without a left tail at all. One other effect is frequently seen in 8 bit avi from a composite video camera --- missing values. These 'missing' values appear when a video with a range of pixel values from 10 to 235 is remapped (stretched) to a 0 to 255 range. When that process is present, every eighth value is missing and never appears as a pixel value.

This new algorithm is insensitive to skew, clipping, and missing values.

Version 2.6.7

- Very minor change – set the default state of the checkbox that controls the display of frame metadata for ADV/AAV frames.

One reason for this change is that this information is rarely needed.

Another reason is that for AAV files, the first frame tag is named Error: and is usually empty, but as it heads a list of other tags, it is easy to think that an error has actually occurred. But if the user had to check a box to see this data display, it will be clearer that no error has occurred.

Version 2.6.6

- Changed setup.py to force loading of Adv2 1.2.0 (needed to make version 2.6.5 work)

Version 2.6.5

- This version allows AAV Version 2 files to be read and processed.

Version 2.6.4

- With this version, the case of a non-version 2 ADV file is treated more gracefully --- just printing a message in the text box rather than a session-ending exception.

Version 2.6.2

- Added requirement that top/bottom redact values be entered (in the “finder” tab panel) before hot-pixel suppression via median filter will take place. This is needed for the case where a timestamp overlay is present that requires OCR. The median filter process has to bypass the region where such timestamps are located, otherwise poor OCR will be a result.

Version 2.6.1

- Working with Hristo Pavlov, I have added support for reading Version 2 ADV files (Astro Digital Video files) to PyMovie.

A side effect of our collaborative effort is that there is now a publicly available Python package on PyPI.org for reading .adv files (Version 2) --- the package is called Adv2 and is available to any Python programmer by the usual `pip install Adv2`. The project and its documentation can be viewed at <https://pypi.org/project/Adv2/>

While being able to read .adv files is a pre-requisite to reading AAV files, it is likely that some additional capabilities will need to be added to PyMovie to deal smoothly with such files. If the demand is there for such support and is made known to me, I will undertake adding the missing pieces.

- Added an option to apply either a 3x3 or a 5x5 median filter to frames as they are being read --- this has the effect of removing hot-pixels automatically, provided the 'hot-pixels' are isolated pixels or in very small groups. This is a much easier option than placing apertures on all the hot-pixel groups and asking PyMovie to record the coordinates of the enclosed hot-pixels for later 'erasure'. This simple method of removing hot-pixels has allowed me to generate very well defined "finder" images from a video that had too many hot-pixels to deal with by placing apertures on them individually. Fortunately, these hot-pixels were all single isolated pixels, so the 3x3 median filter easily suppressed them during the "finder" image generation.

Version 2.6.0

- Added the change of 2.5.9 to .avi files. (Version 2.5.9 only applied to .ser files.)

Version 2.5.9

- Added the 100 nanosecond digit to the SER timestamps so that there now is no difference between the SharpCap timestamp and that of PyMovie. Timestamps to a resolution of 100 nanoseconds have little use, but this eliminates a discrepancy that might be unsettling to some.

Version 2.5.8

- Corrects a math error in the calculation of SER timestamps that was causing a discrepancy of several microseconds between the timestamp displayed (by SharpCap) and the timestamp calculated by PyMovie. Thanks to Hiroyuki Watanabe for pressing the issue.

Version 2.5.7

- In the "finder" image generation panel, we have removed the option to use the Fourier=transform-based image correlation computations as a stacking/image aligning mechanism. It was finicky, had no benefits over the simple to use and understand stacking using a single tracking star, and fails on images without dominant features.
- Also made the name of the aperture to be used for stacking case-insensitive. It must still start with the character string 'stack', but 'StAck' will work just as well.

Version 2.5.6

- In version 2.5.4 we switched to using a filename dialog that provided a list of already-used .csv file names, a useful convenience on occasion. Unfortunately, another characteristic of that file dialog is that it allowed the user to specify a filename that did not have a .csv extension. This version adds code to append a .csv extension to any filename the user provides that does not already have such an extension.

Version 2.5.5

- Squashed bug that kept avi files recorded with a dvsd codec from being read.

Version 2.5.4

- When writing a csv file or saving an aperture group, the dialogs now let you see what names you have used in the past.
- **CAUTION: the use of a computed mask (i.e., a snap-to-blob aperture) for the sole aperture that contains the target star, particularly if the target intensity drops significantly during the occultation, is not recommended! If you use a single aperture on the target star, it should be a static aperture with a fixed radius circular mask!**

The problem is that, as the target intensity drops, there may be a point, just before the default mask comes into use, that a very small computed mask gets used and that mask fails to include all the pixels from the star image, thus creating an artificial drop that can/will affect where the D or R edge appears to occur. This is exacerbated by the use of the 2 sigma level for mask computation, a practice that, while good for tracking and reference stars, is hazardous for target stars.

If you want to use a computed mask on such a target in hopes of achieving an improved snr, **ALWAYS** overlay that aperture with another aperture that uses a fixed circular mask. The composite lightcurve for the target will then make it clear whether or not the computed mask results should be used for timing as it will show two overlaid lightcurves: one with a computed mask and one with the standard (and reliable) fixed circular mask.

The computed mask has proven to be most helpful in providing robust tracking. If you have a 12 or 16 bit camera and windshake, you will also get cleaner lightcurves by using computed masks. But for 8 bit videos with little windshake, the computed mask is best relegated to tracking and reference stars.

Version 2.5.3

- Fixed: during preparation of image for submission to nova.astrometry.net, I was forcing conversion of the image to uint16 for display purposes --- normally this would be ok, but FITS files can be float64 (signed) and cannot always be converted to uint16. I doubt any users will have noticed this, but one of my test data sets is a video of the Pluto occultation that was greatly 'massaged' and as a result came to me as a signed float64 file.
- Fixed: memory leak associated with plots --- some users (and me during testing) found

that pymovie memory usage increased with time, eventually crashing --- I have traced that behavior down to the plot lightcurves function and eliminated the problem. A side effect of the 'fix' is that plots from a previous will no longer persist but rather will be automatically deleted, closed, and replaced by new plots. For some users, this may affect your workflow if you were in the habit of running an analysis, plotting the results, and running a changed analysis expecting to be able to generate another set of plots without losing the first set. If this was your practice, now you will be required to explicitly save plots that you want to use for comparison studies.

Version 2.5.1

- Fixed: SER file folder selection was not enabling 'save aperture group' button.
- Fixed: Initial 'save' of aperture group was not enabling 'restore aperture group' button
- Changed: default aperture name from app00 to ap00

Version 2.5.0

- Removed asinh scaling and normalization to 0...255 range in "finder" images. This will preserve star intensities and (hopefully) make "finder" images look more like the star chart. In addition, with this change, the contrast settings that are useful for an individual frame are also applicable to the "finder" image, so I make that setting 'sticky' during changes to and from a "finder" image.

Version 2.4.9

- This version provides a solution to the following problem: if snap-to-blob apertures are placed on a "finder" image, the thresh value (used for calculating sampling masks) will be correct for the "finder" image, but not for the frames of the video --- the average background will different and the noise will lower. The solution incorporated in this version is to detect when the display changes from a "finder" image back to the normal frames. When this change is detected, we now automatically recalculate thresh for all snap-to-blob apertures that are present.

Version 2.4.8

- Removed the requirement when reading a "finder" file that it reside in an avi/ser or fits folder to make it easier to use "finder" images from an external source (like RegiStax)

Version 2.4.7

- Dragging tabs to new positions to reflect a preferred work flow is now 'sticky' between session.

Version 2.4.5

- Changed the extraction of timestamps in SER files to use my SharpCap timestamp extraction routine. Previously, in a SER.py module that was available on the Internet and used by me for reading SER files, timestamp conversion involved first converting to Julian date and then to an ISO timestamp. For some reason, the Julian date version was always exactly one day earlier than the value from my SharpCap routine. Since the SharpCap interpretation matches what is shown visually, I'm electing to use that method of timestamp calculation.

Version 2.4.4

- Solves a problem extracting SharpCap embedded timestamps on some Win10 systems, which can have a default 32 bit integer rather than the 64 bit integer on Mac computers.

Version 2.4.3

- Solves another problem reading large SER files --- again unique to Win10 environment.

Version 2.4.2

- Solves problem reading large SER files that was unique to Win10 --- reading large SER files on Mac was no problem.

Version 2.4.1

- Adds timestamp extraction from SharpCap 8 bit avi captures. SharpCap embeds a 64 bit timestamp (from Windows computer clock --- number of 100ns ticks since midnight on 1 Jan 0001) in the first 8 bytes of each frame (upper left corner). With this version, there is an addition to the VTI drop-down list that allows this form of avi to enjoy automatic timestamp extraction.

Version 2.3.9

- Fixed font issue on Preferences tab and radio button behavior

Version 2.3.6

- Adds initial support for GHS VTI. I have only a single sample to test with. It was a 720 x 480 avi running at 30 fps. The sample was provided by Hiroyuki Watanabe with the timestamp positioning he preferred. If the template extracted from that sample doesn't fit your GHS setup very well, send me a sample. Sometimes a 640 x 480 (or other shape) needs its own template.

Version 2.3.5

- GUI changes: moved Plot Robust Mean back to main page --- because I use this a lot and don't want the extra click on the Misc. tab. Moved Show contrast control to more readily accessible position.
- Added a text box at top of form that displays the name of the file/folder currently being displayed.
- Added Default mask radius to Preferences tab AND made the sigma level and default mask radius 'live' in that they will be in effect at next placement of a snap-to-blob aperture. And (of course) the settings will be sticky between sessions.

Version 2.3.4

- Implemented the Preferences tab. An aperture size and a threshold sigma level (1, 2, or 3) can be specified. They will be 'saved' when PyMovie is closed and applied when next PyMovie is started up.

Version 2.3.3

- Major rearranging of GUI elements to reduce clutter. Uses Tabs.

- Note: the tabs can be dragged to rearrange (but won't be sticky --- I may do that when I implement 'preferences')
- Changed to use of 2 sigma(std) in automatic setting of threshold in snap-to-blob apertures.

Version 2.3.1

- Added test for astrometry.net failing to accept upload of image --- a failed upload happened once when astrometry.net was having some trouble --- so that a clear message to the user is produced.
- Changed default mask radius from 5.3 (nearly always too big to be useful) to 3.2 (which makes a smallish and roundish default mask of more useful size).
- Tracked down and solved some mysterious behavior associated with FITS files. After a deep dive, I discovered that all FITS files are written with data bytes ordered for a big-endian computer. We run on Intel, which is little-endian. Nearly all of my code did operations on image data using numpy, which detects and properly operates on either big or little-endian data. BUT, in my mask calculation, a routine is used that was ported from c++ code which did NOT adapt gracefully to big-endian image data. A test for big-endian data has been put in that routine and does an endian swap when necessary.

Version 2.3.0

- Fixed a bug involving saved aperture groups that did not contain a tracking path --- the code had been requiring the presence of a tracking path definition in a saved aperture group --- clearly that needs to be optional.
- Added some additional 'help' info
- Moved Version info button back to its original position, never to be moved again.

Version 2.2.9

- Added more information to 'help' message obtained by right-click on Generate "finder" button.

Version 2.2.8

- Adds the ability to align frames for stacking into a "finder" frame by using any star that can be tracked by a snap-to-blob aperture to provide the needed frame shift information.

To activate this type of stacking, place a snap-to-blob aperture with a name that starts with **stack**. That's the only change needed. This is the easiest and most reliable stack alignment algorithm and will follow windshake (which 2 point tracking does not).

The priority of frame stacking algorithms is:

1. 'stack' aperture
2. 2 point tracking path
3. Fourier Correlation

Version 2.2.7

- Adds the ability to save multiple sets of apertures (and any associated tracking path). When the **Save aperture group** button is clicked, a dialog will appear requesting that a 'tag' be supplied to give the 2 or 3 files that are used to record a saved aperture group a unique name. The tag is appended to a root name and when the **Restore aperture group** button is clicked, a list of available aperture groups will be presented for selection.

Version 2.2.6

- Many significant changes to finder image usage:

It is now possible to generate and save multiple finder images. This is particularly helpful in the context of setting up for 2 point tracking. The procedure would be to generate a finder image at frame 10 (say) and another at frame 2500 (say). These will be saved as enhanced-image-10.fit and enhanced-image-2500.fit and can be restored by clicking the load finder image button --- there you can select the appropriate enhanced-image-nnn.fit file (or .bmp --- that's still available so that RegiStax can be used as well). The appropriate frame number to position the video to will be extracted from the image name. These two finder images will be of assistance in positioning a yellow aperture and recording the 2 points needed for determining the tracking equation.

It is now possible to place multiple apertures on a finder image. This can be useful to 'find' which of several stars is the occulted one when a sky chart is not available or does not resolve the uncertainty. Pick the likely candidates with apertures --- run an analysis --- look for the occultation.

Note: because of the change above, we no longer treat any particular aperture as a target aperture and do not record a 'target xy' as we had been in previous versions. Previous versions would only allow a single aperture to be placed on a finder image and assumed that this was the 'target' and recorded its coordinates in a special txt file. That feature has been removed in favor of this more general solution.

From a suggestion by Tony George: If there is a tracking path defined and a finder image generation is requested, the stacking algorithm will use that tracking path for aligning frames rather than the Fourier Correlation method. This works very well for drift-through videos that don't have significant windshake even without hot pixels being defined (they now show up as streaks on the finder image, so you may still want to create a hot pixel list).

Version 2.2.5

- When a new aperture is added, the new default state is that jogging with the arrow keys is enabled. At the same time, jogging is disabled on all apertures. This change was made to make the process of positioning apertures on hot-pixels easier.

It is still possible to individually enable/disable jogging on more than one aperture. The only change is when an aperture is first created.

- Added hot-pixel removal from finder images. This is a user guided process --- the user must place small apertures on hot-pixel groups, set a threshold for identifying hot-pixels, and clicking on a button that invokes a program looks through all of the apertures, prepares a list of the coordinates of all hot-pixels, and substitutes the average background for all hot-pixels.

Version 2.2.4

- On lightcurve plots: the composite light curve is back on top. I didn't like it on the bottom so it is back to the top where it will stay. I did ensure that the order of adding curves to the composite lightcurve plot is based on the column order. In this way, if the target lightcurve is the first column in the csv plot, it will be the last added to the composite plot so its 'dots' will be on top.

Version 2.2.3

- When the lightcurves are plotted, the z-order of the composite light curve plot has been changed so that it is now the bottom-most plot rather than the topmost plot. In addition, the individual light curves will appear in the order specified by the order number of the aperture (this will match the column order in the csv file). So an aperture with order number 4 will be plotted on top of an aperture with order number 7. So the user now has complete control of the plot order for the individual lightcurves through judicious use of the aperture edit table invoked by clicking on the Examine/change aperture settings button.

Version 2.2.2

- Made change to 2.2.1 so that my name change is no longer a breaking change. This version accepts either form upon read, looking first for the new form, and then for the old form. It always writes the new form.

Version 2.2.1

- NOTE: this is a breaking change. When an aperture group is saved, the filenames that appear in the folder have been changed as shown below:

markedApertures.p → savedApertures.p
markedFrameNumer.p → savedFrameNumber.p

This means that if you have folders that include 'saved aperture group' information, to be able to access that data, you will need to manually rename those files. Sorry. But this avoids possible confusion between 'saved aperture group' information and that saved (for the session only) when the Mark button is clicked.

- In addition, now when 'Restore aperture group' is clicked, any apertures saved by a 'Mark' will be erased before the saved aperture group is loaded. This is a cleaner 'start over' and, without this change, if you click Restore aperture group, then Mark, then Restore aperture group again you got a duplicate set of apertures.

Version 2.2.0

- Made the 'help' text come up whenever the lunar box is checked. It serves as a warning.

Version 2.1.9

- Added a 'lunar' checkbox which should ONLY be used when an aperture is placed on a sunlit lunar limb for a lunar occultation.

This is experimental: until this point, PyMovie has focussed on asteroid occultations only --- this is a deviation from that focus.

When this box is checked, it signals the Robust Mean extraction routine to statistically separate sky pixels from lunar limb and star pixels for use in background calculations. This emulates what Limovie does when a properly oriented 'Avoid Sunlit Face' aperture is utilized and so should produce comparable lightcurves. Time will tell.

Version 2.1.8

- I now save tracking path data when an aperture group is saved.

Version 2.1.7

- Added the ability to set a 'tracking path' by specifying two points, one early in the video and one late in the video. This feature is primarily useful in processing difficult drift-through videos where no other good options for tracking are present.

To learn a bit more, right-click on a non-yellow aperture and then click one of the menu items involving 'track path'. That will cause a help panel to appear.

Version 2.1.6

- Robust Mean extractor changed so that when processing a 'clipped' background, star pixels will be excluded from the calculation by not including pixels with values that are more than 3 sigma from the mean.

Version 2.1.5

- Modified Robust Mean extractor to detect 'clipped' backgrounds and employ a different algorithm for the determination of mean and standard deviation.

Version 2.1.4

- ! Fixes a problem with saved aperture groups that used an aperture size that did not match the currently selected aperture size in the dropdown list. What should have happened is that the aperture size selection should have been changed to match the aperture size used in the saved group when it was restored. That is now the case.

version 2.1.3

- To simply setting up an OCR profile: added ability to retrain/resample any digit(s) after all digits have been initially sampled/trained. When there are missing sample digits, the menu of characters to record will show those that remain to be sampled, as it was before this version --- that feature speeds up the initial sampling of the digits. The

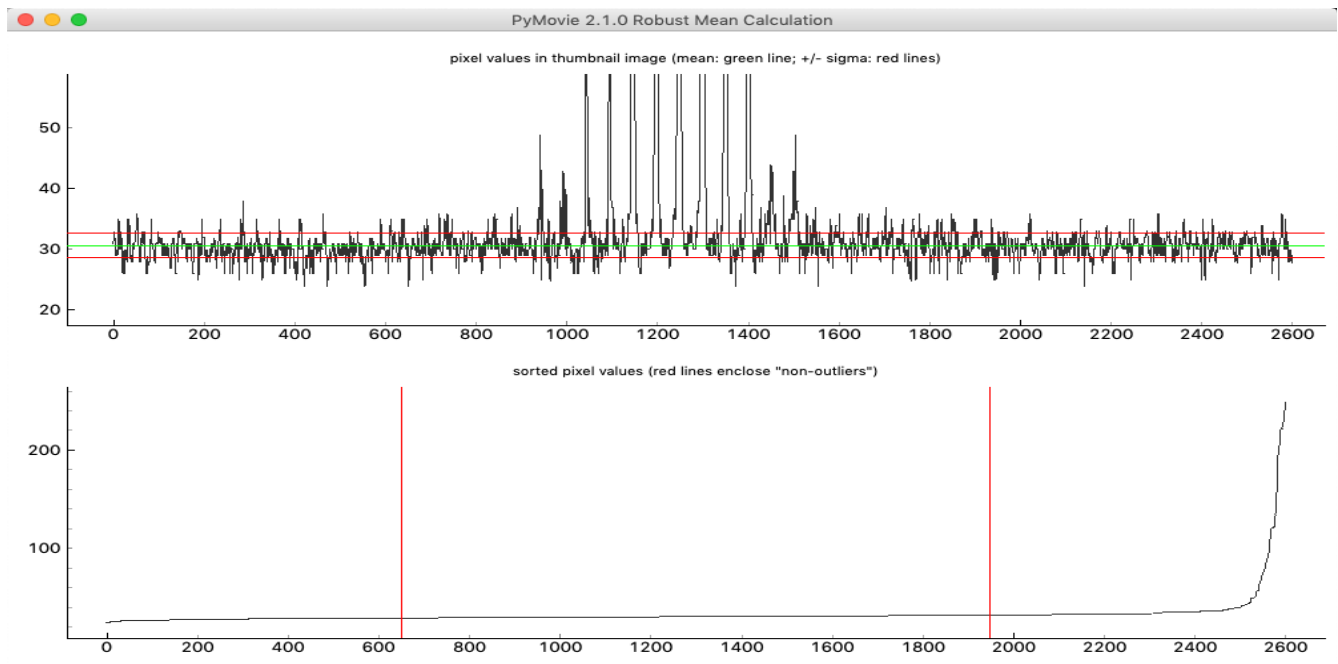
change made in this version is that, if all digits have been sampled, a complete menu of digits to record will be produced when a digit is right-clicked so that any poorly sampled digit can be re-recorded without the need to start over from scratch.

version 2.1.2

- Added ability to correct pixel response curve for non-unity gamma setting of camera

version 2.1.1

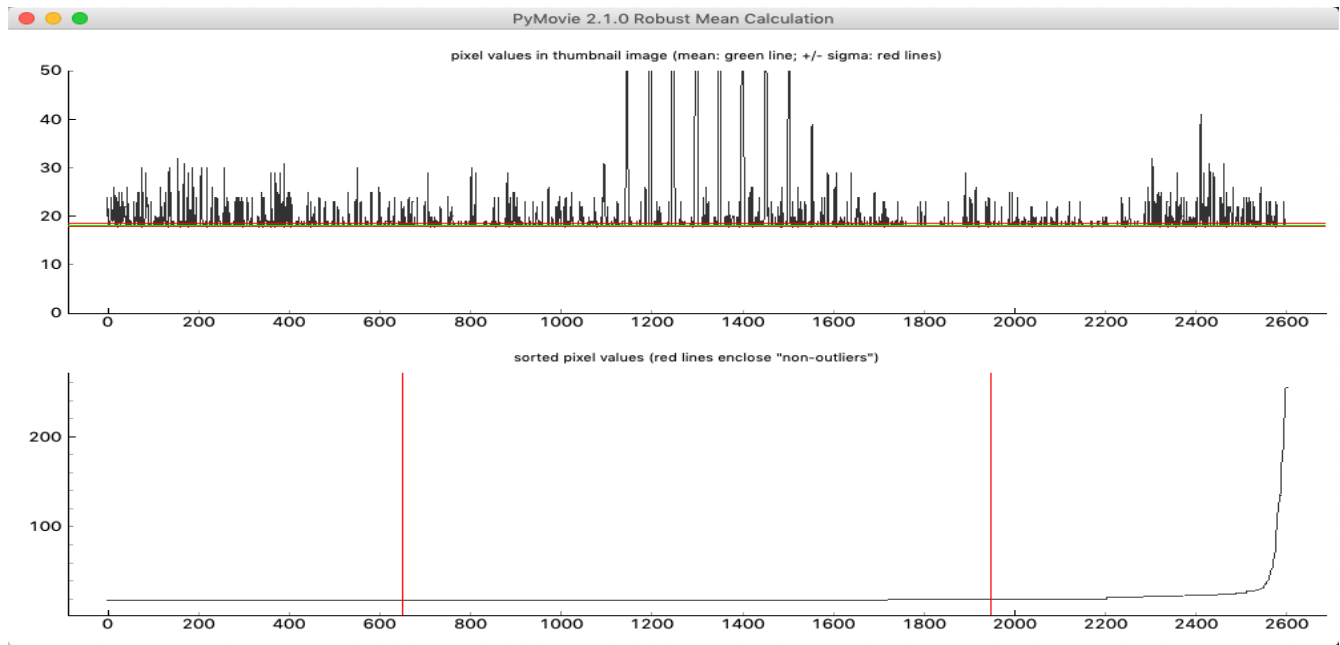
- Modified the robust mean estimator to deal with videos that have 'clipped' backgrounds, a situation that is surprisingly common.



Here is a 'normal' RobustMean plot:

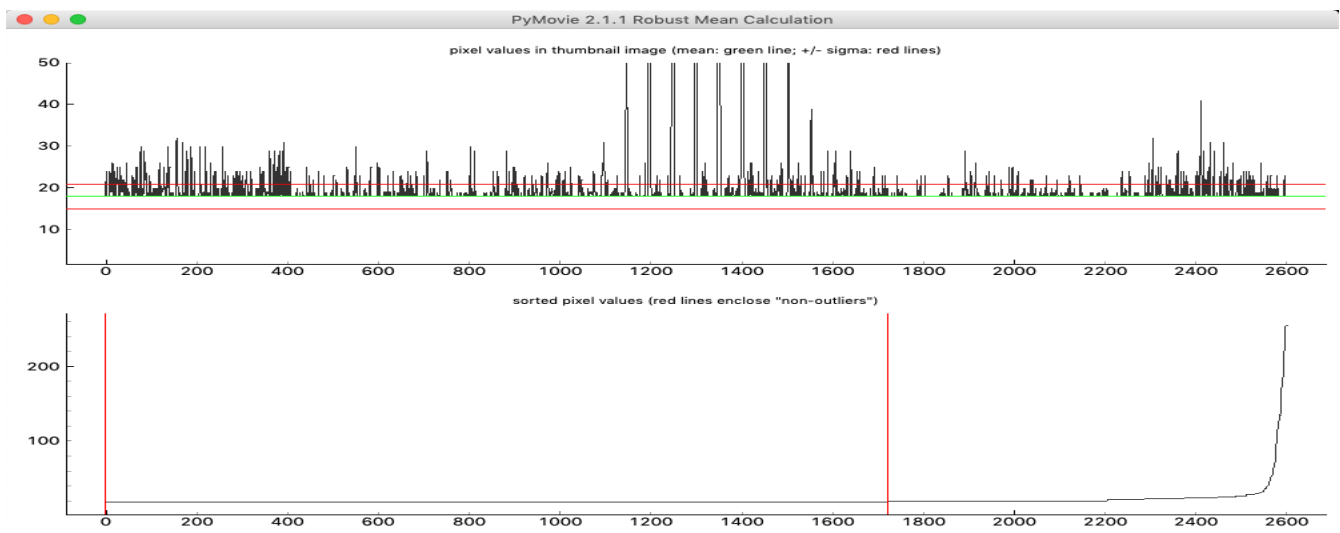
It shows the usual gaussian background noise in a 51x51 pixel aperture with some stars in the aperture as well.

But sometimes we see that somewhere in the processing chain (could be a camera setting --- could be in the recording software) that some sort of 'clipping' has occurred. It exhibits as a 'one-sided' noise distribution with many of the background pixel values equal to a fixed value. This is illustrated in the plot below:



The problem now is how to calculate something that approximates a standard deviation for the background noise. This is an important value because PyMovie uses that number to set an initial threshold value which is used in the calculation of the sampling mask. But the presence of very many constant values causes the current computation to produce a very small (often 0.0) value for the estimated std and that causes mask threshold values to be set unreasonably low.

The changed robust mean estimator, applied to the same 51x51 aperture produces:



and the resulting standard deviation is much more realistic and produces a useful initial value for the mask threshold.

Note: such clipping has an effect on photometry. It appears small, so can often be ignored,

but if you can set your processing chain to avoid such clipping, that would be the recommended way to go.

version 2.1.0

- Fixed math problem in robust mean estimator when 16 bit uints are involved (QHY FITS)

version 2.0.9

- Made a change to the 'robust mean estimator' to better deal with videos that have been (artificially) clipped so that the background has many 0 values. This shouldn't have any affect on properly recorded backgrounds, but stay alert!

version 2.0.8

- Added ability to redact images from top and bottom simultaneously.
- Modified IOTA timestamp ocr parser to allow for a 3 digit milliseconds field (fourth entry blank). This makes ocr timestamp reading possible for the smopiVTI by treating it as an IOTA VTI with repositioned character boxes.

version 2.0.7

- Fixed 3D Thumbnail display (it got broken in version 2.0.2 when left to right image flipping was added)
- Changed Open AVI file button label to Open AVI/SER file.

version 2.0.6

- Added PAL mode Kiwi OCR timestamp recognition. In PAL mode, Kiwi timestamp characters are vertical. In NTSC mode, the characters are slanted.

version 2.0.5

- Adds the ability to read 8 and 16 bit gray-scale (not color) SER files

NOTE !!!! Quite a few changes were made to the code to add this capability.

NOTE !!!! Although changes were carefully isolated and made specific to SER files,

NOTE !!!! the possibility that a bug was introduced is high.

PLEASE: email me immediately if you find such a breakage
(bob.anderson.ok@gmail.com)

version 2.0.4

- Added a button on the main panel to delete ocr data files --- to be used as a clean 'start-over' when an OCR setup/training has gone seriously pear-shaped.

version 2.0.3

- When in field mode and ocr selection boxes are present, moving the cursor into a character box automatically displays that character in the Thumbnails. This probably eliminates the need for the Show property menu item in the context menu, but we'll leave that in place until we're certain there is no longer a need for it.

version 2.0.2

- Added ability to use “finder” image as submission to nova.astrometry.net. This has helped in at least one case, but use sparingly.
- Added check box to allow flip of images left to right (around y axis)

version 2.0.1

- Negative values for line redaction was not working when used for an astrometry.net calibration. This oversight has been corrected.

version 2.0.0

- Fixed reading of FITS files --- had been reading as int16 --- should have been uint16
- For “finder” image generation: added setting of background to zero (controlled by a threshold value that can be set by the user or automatically by the program) for creating ‘finder’ images. This is necessary so that the stacker aligns to star features rather than background features/texture.

version 1.9.9

- Major rewrite of manual WCS calibration code. Biggest change was to solve for the rotation angle between the RA/Dec and x/y coordinate systems numerically rather than with trig. This numerical solution tolerates all possible orientations. The change was necessitated when an alt-az example showed that the trig solution was too temperamental --- and there were problems earlier with inverted or flipped images that were dealt with as special cases and now can be treated without special code. The other change was to use a 2 x 2 rotation matrix to calculate the target x/y coordinates. The rotation matrix values depend only on the rotation angle between the RA/Dec and x/y coordinate systems, determined robustly by the numerical ‘search’ referred to earlier.

version 1.9.8

- IF you have enabled a SINGLE ocr selection box for jogging, jogging that box will automatically show you the expanded view in the Thumbnails. This feature is automatically suppressed if more than one box is being jogged (which box should be displayed??). This change will not affect anyone that has gotten used to jogging groups of boxes, but for people electing to jog each box individually into place, this will be more intuitive and save the extra right-click to ‘show properties’.

version 1.9.7

- Adds a ‘secret’ key sequence to enable/disable diagnostic printing of keystroke data.

2 uppercase K in a row turns keystroke diagnostic print ON
 1 lowercase k turns keystroke diagnostic print OFF

This capability was added when one user encountered difficulty getting their arrow keys to be recognized. But if one user has a problem, others may follow.

version 1.9.4

- Made some GUI changes specific to Windows operation to deal with font differences between Mac and Windows platforms.

version 1.9.3

- Thanks to Christopher Bennett: he found and fixed some issues related to using PyMovie on Linux and posted his proposed changes as a pull-request in my GitHub pymovie repository. I have accepted and merged his changes --- AVI-WCS folders should now work properly on Linux systems.

version 1.9.2

- Fixed bug where retraining digits deleted template digits but was failing to allow retraining menu list to appear until the folder was reopened.

version 1.9.1

- When, in version 1.8.9, we added the option for an aperture stack, we decided to suppress the request for names --- went too far and, as a result, we were no longer prompting (rudely) for an aperture name when a static aperture was being placed. Fixed it --- we're back to prompting for an aperture name when a single static aperture is placed, but don't prompt for names when a stack of 5 is placed.

version 1.9.0

- Updated the right-click help message for the Examine/change aperture settings button to reflect the new functionality of using the thumbnails to show aperture contents and either computed mask or default mask.
- It is expected that this will be the version released for public use --- end of Beta testing --- start of production testing.

version 1.8.9

- Added a menu item to add a stack of 5 static apertures.
- Also allow red apertures to 'snap' as their threshold is changed from the Aperture Edit list. Previously, on a 'green' aperture would snap.

version 1.8.8

- Made a 'live' connection between the Aperture List Dialog and the thumbnails so that is easy to set default mask sizes and thresholds with a 'live' view of the effects. Also eliminated the need for a 'write' button by making any change take effect immediately when enter key pressed, cell focus changed (by tab or arrow key or mouse click), or dialog closed.
- Fixed csv write routine so that it uses the correct folder location (AVI-WCS or FITS) as the default storage location.

version 1.8.7

- Generate Finder image now works with FITS files.

version 1.8.6

- Added message in textOut panel that PyOTE is starting up --- that process takes a few seconds and the user needs to know to be patient.
- Updated the main documentation file to add a table of contents, hyperlinks, and an acronym list.

version 1.8.5

- Added automatic startup of PyOTE when a csv file is written. PyOTE will open and load that csv file.

version 1.8.4

- Defaulted the lightcurve plot 'zoom' (drag with right button) to affect x axis only rather than both x and y at the same. There is a right-click context menu for each plot that let's you change how the right button drag is applied to x and y axis --- it is called Mouse Enabled and is individually settable for each axis.
- Added a bit more text to the lower row of transport buttons to make them a little less cryptic at the expense of a little more real estate.

version 1.8.3

- GUI changes to better group buttons and controls by 'function' and in approximate workflow-order.

version 1.8.2

- Added the ability to save an aperture configuration between 'runs'. Preserved is the frame showing when the **Save aperture group** button was clicked and the apertures showing at that time, including all of their properties. The configuration can be retrieved by clicking the **Restore aperture group** button.

version 1.8.1

- Added +/- 1 frame buttons and removed the the up/down buttons on the current frame and stop frame spinners.
- Made it easy to change pixel dimensions during WCS operations. This is normally a set-and-forget operation --- please do not experiment with dimensions to 'guess' at pixel dimensions --- that would be an abuse of nova.astrometry.net. A single test to prove that it **does** matter is ok. Figure out what your pixel dimensions are from camera specs/sensor chip manufacturer, not from inducing nova.astrometry.net to 'fail', which slows down the service for everyone world-wide.

version 1.8.0

- 'play' was not initializing tracking --- added the needed initialization so that 'play' will show exactly what would happen during an analysis.

version 1.7.9

- Fixed indexing problem with lightcurve plots in field mode that was causing incorrect frame numbers to be displayed as hairline was moved.

version 1.7.8

- Fixed lightcurve plots: they didn't work right in field mode; now they do.

version 1.7.7

- Added timestamp to data displayed as hairline is dragged across a lightcurve plot.

version 1.7.6

- During manual WCS calibration, adjusted assignment of RA DEC values to reflect the center of the pixel, rather than the upper left corner.

version 1.7.5

- Corrected the display of WCS coordinates that appear in the status bar (when astrometry.net solution is active in a frame) when the cursor is inside the image --- these values needed to take into account pixel aspect ratio.

version 1.7.4

- Added non-square camera pixel compensation to manual WCS calibration. That compensation is now in place for both WCS calibration methods: nova.astrometry.net and 2-star manual

version 1.7.3

- Added compensation for pixel aspect ratio (W/H) for submissions to astrometry.net. This deals properly with non-square camera pixels by resizing the image submitted to astrometry.net to remove the distortion caused by rectangular pixels.
- This compensation is NOT YET incorporated in manual WCS calibration.

version 1.7.2

- Moved all the major operational buttons (clear data, write csv, plot curves) to the transport button complex.

version 1.7.1

- Rearranged transport buttons to allow more flexibility in splitter placement and also to bring the current frame spinner to a more accessible place.
- Begun adding the elements needed to deal with non-square pixels during WCS calibrations --- GUI elements only at the moment.

version 1.7.0

- Implements the save/restore state feature. Only the number of apertures and their positions are 'marked'. The frame number is 'marked' as well. This 'mark' is placed automatically when the first analysis is initiated on a newly loaded video. However, there is a button that can be utilized to override this 'mark' if the automatic 'mark' was somehow incorrect (or you just changed your mind about where to start an analysis or how many apertures to use, etc).

version 1.6.9

- Transport buttons fully functional.
- Save and restore state buttons added to GUI, but not yet implemented.

version 1.6.8

- Added additional entries to the OCR profile dictionary to accommodate the need for four sets of character selection boxes for Kiwi. That need became apparent when it was determined that the left-right shift of Kiwi characters was not a constant but instead there was a need to customize even that character shift.

A consequence of this change is that existing custom profiles that you have saved in your list must be deleted. Just open the list, delete each saved profile, then restore

them from relevant AVI-WCS folder(s) as required. It's not often that I make a breaking change, but this was unavoidable.

- Do not get too excited about the transport buttons. They are conceptual/visual only --- no functionality yet.

version 1.6.7

- Fixed breakage caused by incomplete removal of default mask radius spinner.

version 1.6.6

- When an AVI-WCS folder is created by clicking the **Create AVI-WCS folder from AVI file** button, a help panel is popped showing the location and name of the newly created folder. (I actually had been printing that info to the textOut panel, but there was an automatic opening of the new folder that was erasing that panel.)
- An experimental GUI change has relocated the **Plot symbol size** spinner to a place where it won't be the accidentally invoked by a user intending to click the mskth spinner.

version 1.6.5

- Made some small changes to documentation pdf suggested by Tony
- Relabeled the angle printed at completion of manual WCS calibration from **field-rotation** (incorrect) to **ref1-to-target**. This angle is a diagnostic printout, useful to me, but not to the user. To interpret it, you need to know that the coordinate systems in use are: x increases to right, y increases down, RA has East to the right, DEC has North down, and angles in the x-y plane (the image) are always positive and are the number of degrees in the ccw direction needed to rotate a vector until it lines up with the positive (i.e., down) y axis. So ref1-to-target means the origin is ref1, and we are rotating the vector from ref1 to target around ref1 ccw until it is aligned with the positive y axis.

version 1.6.4

- Updated documentation pdf

version 1.6.3

- Added units to the plate scale that is printed at completion of manual WCS calibration.
- Added field rotation angle printout at completion of manual WCS calibration.

version 1.6.2

- For manual WCS calibration: automatically adjusts to star fields that are flipped left-to-right and/or top-to-bottom.

version 1.6.1

- The version is same as 1.6.0 and serves only to test whether 1.6.0 will detect 1.6.1 and install it upon request.

Version 1.6.0

- Added the ability for PyMovie to check for more recent versions on startup and allow the user to choose whether to accept or not.

version 1.5.7

- select VTI dropdown list is disabled unless there are missing model digits.
- Suppressed the star location dialog --- now only appears when an astrometry.net calibration is requested.

version 1.5.6

- Adds full edit capability for saved ocr profiles. A single dialog box allows for: the naming and saving of the current ocr profile; the deletion of a saved profile; the renaming of a saved profile; the loading of a saved profile.
- Item 3 in version 1.5.4 notes refer to the too tortuous logic to implement. Well, it has been implemented anyway so training can be done from any position in the avi where the upper and lower box positions are the same.

version 1.5.5

- When Kiwi is used, a vote of the first three frames is used to determine the position of t2 on a timestamp line --- either left or right. After that, the other digits (t1) will not be used. This will reduce the number of times an ocr error occurs because of a mis-recognition of a digit in a part of the timestamp that should not have been used.

version 1.5.4

- Changed blank detection for Kiwi to reflect the new 'normalized' model characters.
- Added (Kiwi) 8 versus 9 secondary test to resolve confusions (same thing as is being done for 6 versus 8)
- The logic to allow a Kiwi profile to be saved from other than a frame where the characters are in their training position was too tortuous to implement. Now we just reposition to frame 1 before doing the save. This means that all training needs to be accomplished while the characters are in the position of those in frame 1.

version 1.5.2

- When starting from empty folder to train Kiwi, the selection boxes initially were being displayed as red rectangles instead of yellow slanted selection boxes --- that's fixed
- At completion of choosing model digits, we now start timestamp reading.
- If a Kiwi training is done and a trial run is made that ends with boxes in the alternate position, we now detect the new position so that if the ocr profile is saved from this position, it will be saved with the current box positions as well as the model digits.

version 1.5.1

- Major change to Kiwi digit processing: trapezoidal boxes for selection (matching the 'lean' of the font); removal of blank lines in font; removing 'lean' from the characters; no longer use Gaussian blur.

Initial tests show that it is much easier to place ocr selection boxes and recognition error rate is no longer so sensitive to the precise box placement.

version 1.5.0

- Fixed a bug that has been present forever but remained undetected until now: whenever an analysis (data recording) was started and then paused, the frame where the pause took effect was being recorded again if the analysis was restarted from that

point. That caused duplicated frames to appear in the csv that were not present in the avi whenever multiple start/pause/look-around/start-again sequences were performed by the user (a common practice).

The 'fix' was to set a flag when an analysis is started for the first time after either a program start or aperture data has been cleared. The start-at frame is recorded if this flag is not set, and then the flag is set to inhibit the recording of the start-at frame should a pause/restart occur.

A side effect of this new flag is that if one uses the 'run backward' --- 100 down to 50 (which stops at 51) then 1 to 50 to complete the analysis, frame 1 will not be recorded; only 2...100 will be present. A small price to pay for the ability to freely interrupt an analysis, check some things, and restart as many times as needed.

version 1.4.9

- Improved Kiwi timestamp recognition capability by adding second stage testing to resolve the 6 versus 8 confusion. Added logic to deal with the ms fields that sometimes appear completely blank and sometimes as 1000 (normally only a 3 character field). (Kiwi has 6 different timestamp formats in addition to the 2 possible character positions.)
- Squashed the bug that was causing crash when training an OCR profile from an empty folder and not providing a star location when prompted.
- Very high quality (low error rates) OCR for Kiwi timestamps is available IF and ONLY IF you take the time to jog the character selection boxes into good positions. A good position has the character right-justified in the box. Use 'show property' to get a view of EVERY character box placement and adjust accordingly before saving the model digit. It's fiddly, but I routinely get 100% recognition rates on Kiwi timestamps by following this advice.
- Changed the name of the users' list of custom profiles from pymovie-ocr-profiles-<username>.p to simply pymovie-ocr-profiles.p What this implies is that we no longer support the presence (and automatic merging) of multiple custom profile dictionaries in the users' home directory. This feature was highly unlikely to be useful and greatly complicated the editing (deleting) of profiles from the custom directories. That is much easier to implement when there is but a single dictionary. This will be done in the next version.

version 1.4.8

- Reduced the Kiwi timestamp ocr error rate by more than a factor of 100 by applying a 5 x 5 Gaussian blur to the model digits during training. The model digits now look a little dim (because the blur reduces the average intensity), but they are much more effective during cross-correlation scoring.

version 1.4.7

- Memory usage is back to normal.
- Better detection and correction of Kiwi timestamps at seconds rollover.

version 1.4.6

- Experimental. Probably fixes the garbage collection delay at the end of Kiwi runs. But each analysis run causes PyMovie to claim more and more memory.

version 1.4.2

- Found a way to deal with the odd timestamp that results when there is a minutes roll-over in a Kiwi timestamp. As a result, Kiwi timestamp ocr errors are almost always just the 'confusion' between the character 6 and the character 8 (check the confusion matrix that prints whenever the model digits are viewed to see what I mean), an unfortunate result of the Kiwi font for which (as far as I know) nothing can be done.

version 1.4.1

- Kiwi timestamp extraction is now (maybe) working. The 'maybe' is because the implementation for dealing with the, unique to Kiwi, character position change when minutes change is:

There is a left position and a right position (differing by 11 pixels)

The initial frame of your recording can have the timestamp in either of those two positions

You will therefore need to choose the correct ocr box set when setting up the ocr profile

Subsequently, whenever the minute field changes, the other position is applied and remains

active until the next minute change.

As a consequence of this, you must make sure that any initial 'training' (model digit recording) never crosses a minute boundary. This is not usually hard to do as the complete digit set is usually present in the first 3 or 4 frames.

In addition, take care not to adjust (jog) Kiwi ocr boxes unless the character position is the same as frame 1.

version 1.4.0

- Added automatic opening of the newly created AVI-WCS folder after it has been created by clicking the **Create AVI-WCS folder from avi file button**. So the workflow instructions that came with 1.3.9 are no longer needed.

version 1.3.9

- Removed the message inadvertently left in place that said only partial support for Windows for creation of AVI-WCS folders. It wasn't true.
- When an AVI-WCS folder is created from an open avi file, I make that folder be the default when you click on Select AVI-WCS folder. So work flow is open avi, click create, click Select AVI-WCS folder, click accept (or press return).

version 1.3.8

- Added a button that will create an AVI-WCS folder from an open avi file. The newly created AVI-WCS folder name will always be the file name of the open avi without it's extension. A dialog box will appear to allow the location where the AVI-WCS directory

will be created to be changed by the user. It defaults to the directory that holds the avi.

Work flow:

- open an avi (this will enable the **Create AVI-WCS folder from avi file button**)
- click the **Create AVI-WCS folder from avi file button**
- use the Directory dialog to select (or create a new folder) the directory that is to hold the AVI-WCS folder or just accept the default (directory of the avi)
- the folder will be created (if necessary) and populated with an alias (for mac users)
- or a Shortcut (for Windows users)

This button has been added because of the decision to perform timestamp OCR only on avi files that live in an AVI-WCS folder. That is a good way to collect the (many) files that result from OCR, PyMovie, and PyOTE operations in one logical place, but there are likely to be users that have opted for a flat(ter) directory structure. This button will ease (hopefully) the transition to an alternate file organization.

version 1.3.7

- Fixed bug where adding a custom profile was not removing existing ocr selection boxes.
- Gui change to keep 'cascade' label from disappearing on some displays.
- Started adding ability to create an AVI-WCS folder from an already open avi file. Just the gui and a stub where the code will go.

version 1.3.6

- Fixed bug that required AVI-WCS folder to be reopened in order for a newly selected custom ocr profile to take effect. Now timestamp reading starts up as soon as a custom profile is selected.

version 1.3.5

- Added printing of the 'confusion matrix' (correlation scores of each model digit against all the others) when 'show model digits' is invoked.
- Fixed the spurious exception that was occurring when FITS folder was in use (caused by trying to do timestamp extraction as though a VTI were in use)

version 1.3.4

- Fixed BoxSprite 'start-from-scratch' specifying None instead of boxsprite as timestamp formatter.

version 1.3.3

- Fixed BoxSprite 'start-from-scratch' specifying iota instead of boxsprite as timestamp formatter.

version 1.3.2

- Saving and reusing custom ocr profiles (i.e., ocr boxes and model digits from an AVI-WCS folder), is now functional. The custom profile list is placed in the users home directory and named

pymovie-ocr-profiles-<username>.p For me, <username> is bob, so the file is named

pymovie-ocr-profiles-bob.p

It is possible to share your complete list of custom ocr profiles with another user. For instance, I have another custom profile list in my home directory named

pymovie-ocr-profiles-tony.p

That ocr profile list came from Tony George (and implements profiles that allow the reading of 4 line BoxSprite timestamps). All lists of the form pymovie-ocr-profiles*.p in a users home directory are available as custom ocr profiles.

- Sharing a complete list of profiles is probably not the way to go most times as there would be too many entries from someone like Tony (who has to deal with many recording chains, each requiring slightly different ocr profiles). Instead, you could share the four files (in an AVI-WCS folder) that comprise the ocr profile:

custom-boxes-upper.p

custom-boxes-lower.p

custom-digits.p

formatter.txt

Sharing just those files is all that is required. Put them in your AVI-WCS folder and then click on the Save OCR profile button to save that profile with a name of your choosing in your personal list of custom ocr profiles.

version 1.3.1

- Added to ocr box right-click menu: retrain model digits, enable/disable upper/lower boxes.
- Added automatic switch to frame view if field view was selected and then Start analysis was clicked. It is possible to Pause analysis and then switch to field view without loss of data.
- Removed threshold spinner as that mechanism is no longer used for OCR --- switched to custom model digits per AVI-WCS folder instead.
- Fixed problem that kept ocr box changes and model digits from being properly preserved and restored from an AVI-WCS folder.
- Laid some framework in place for saving and reusing custom profiles (i.e., ocr boxes and model digits from an AVI-WCS folder)

version 1.3.0

- We now require that an avi --- for which timestamp OCR extraction is wanted --- be placed in an AVI-WCS folder. This allows completely custom ocr box placement and modelDigit training. This need only be done once. When that folder is reopened, it will snap to frame 1 with ocr boxes placed and the correct vti selected. The value of training for each file is that the highest OCR accuracy will always be obtained.

version 1.2.9

- Made the cascading of lightcurve plots optional (and sticky)
- Made the ocr box context menu sensitive to which digits are missing from the modelDigit list.
- When a Kiwi or BoxSprite avi is in a folder, we put modelDigits and ocr box table in that directory so that OCR can be customized on a file basis.
- When an IOTA avi is in a folder, we put the ocr box table in that directory, but keep

using the standard model digits from the home directory. So the user can customize ocr box placement on a file by file basis for IOTA VTI timestamped avis without affecting the preset (optimized) model digits.

version 1.2.8

- Timestamp OCR for IOTA VTI models 2 and 3 are functional and useable.
- Timestamp OCR for BoxSprite is useable, but a little tedious as it requires 're-training' at every run in order to deal with the lack of well-formed and reproducible characters.
- Timestamp OCR for Kiwi is not useable (except to demonstrate and play with the challenges of the Kiwi timestamp quirks).

version 1.2.5

- Adds both lefthand and righthand ocr selection boxes to the IOTA VTI to deal with the ambiguity that arises when some cameras emit the odd field first in time while others emit the even field first in time.

version 1.2.4

- Removed the main gui buttons that jogged the ocr selection box constellation up/down/left/right. That function has been taken over by a new right-click context menu item that allows all ocr boxes to become joggable by use of arrow keys.

version 1.2.3

- Fixed the omission: when AVI-WCS folder opened, the navigation buttons were not being set to match the detected fps; that is fixed.
- Timestamp OCR is now functional for the IOTA VTI. Model 3 is directly supported. Model 2 (which interchanges the position of the early millisecond characters (VTI3: early on left; VTI2: early on right) can be decoded as well if one takes the time to reposition the ocr selection boxes. Such a change is 'sticky'. This is useful when there is a single tool chain (camera/VTI/frame-grabber) because, once set for your equipment setup, it will be ready to go next time.
- Recommended work flow: open avi, select VTI, click the appropriate radio button that tells PyMovie which field (top or bottom) is first in time, adjust ocr selection boxes (if needed), leave field display mode and place measurement apertures as usual.

version 1.1.8

- Changed the titling on the frame navigation buttons to use time units for ntsc/pal recordings --- fits still uses frame units as there is no set correlation between frames and time for fits recordings.
- Lightcurve plots now cascade (don't overlap) from the upper left corner of your screen down and to the right (rather than be plotted on top of each other in the center of your screen).

version 1.1.7

- Made the jump forward and backward frame navigation buttons adapt to ntsc frame rate (uses 30 and 300 frame jumps) and pal frame rate (uses 25 and 250 frame jumps). For fits files the jumps remain at 25 and 200

version 1.1.6

- Added a 'Write' button to the Edit Aperture widget so that it is not necessary to close that widget to make changes take effect. Closing the widget still makes changes take effect --- that remains unchanged.
- It is now possible to change the x,y value of an aperture from the Edit Aperture widget. This will make it easy to stack apertures by a simple copy and paste operation of the (x,y) data.

version 1.1.5

- Added spinner to change default mask radius of any 'green' aperture present.
- Added linkage from the aperture edit table to the default mask radius spinner and threshold spinner so that when the color of an aperture is set to 'green' (even if it already was 'green'), the def mask radius and thresh are copied into the spinners on the main gui..

version 1.1.4

- Fixed the aperture position limit issue that occurs when the aperture size is changed AFTER the image has been loaded.

version 1.1.3

- Removed some no longer needed diagnostic messages being printed in the textOut panel.

version 1.1.2

- Implements output of csv aperture data in the order specified by the user rather than the default order based on the order of adding apertures to the image. For now, PyOTE accepts only up to the first four light curves, so when multiple apertures are in use, this re-ordering will make sure that you will be able to use PyOTE on the relevant lightcurves. It is possible (in the future) that PyOTE will be modified so that more than 4 lightcuves can be processed at the same time. Until then, be judicious in selecting/ordering the apertures.

version 1.1.1

- Removed the log scaling checkbox. Log scaling rarely used and the image control is an adequate and flexible tool for image clipping and scaling to better see stars in an image
- Added an Edit Aperture tool that pops up a list of all aperture with their properties and provides a central place to modify same. This will make it easier to deal with stacked (overlapping) apertures.
- Added a default mask radius setting for each aperture rather than the one-size-fits-all of prior versions to better support the use of multiple sampling aperture/masks.
- Removed the spinner previously used for setting a global default mask.
- Changed status bar display when mouse is hovered over an aperture to simply show a list of any apertures that are under the cursor (again in support of stacked/overlapping apertures). Previously all the special properties of the topmost aperture were output to the status bar. That was not helpful when apertures were stacked on top of each other. Now the Edit Aperture list lets one see all the properties of all the apertures, so

it's more important to be able to locate apertures via mouse hover, hence this change.

version 1.1.0

- Added a check on the validity of RA Dec strings on a program path that was missed in 1.0.9.

version 1.0.9

- Tests the validity of RA Dec strings at the point of entry --- shows errors in textOut panel.
- Returns the aperture jogging capability that was inadvertently suppressed in 1.0.8 by the new right-click-for-help system

version 1.0.8

- Made splitter settings 'sticky' so that user choices as to how much gui real estate to give to buttons versus image and textOut versus Thumbnails is preserved between sessions.
- Fixed the tooltip/help info re UCAC4 format to say XXX-XXXXXX (had been XXX-XXXXX)

version 1.0.6

- Polished the right-click help system. Now the help dialog box can be sized and placed by the user and it will be reused (as it was in the original hover-for-help system) for subsequent right-click-for-help events. The user is encouraged to position and size the help dialog box to suit his preferences and to not worry if it gets hidden because a new right-click-for-help will pop it right back up at the previous size and position.

This help system is much easier to use to quickly explore the help available for each gui control.

version 1.0.4

- Hover is now properly disabled so that help only appears with a right-click

version 1.0.3

- Changed the 'help' system from a 'hover to get help' to a right-click to get help.

version 1.0.2

- Fixed conversion of color image to grayscale image for avis: now processes YUV color space correctly.

version 1.0.0

- Added documentation: PyMovie-doc.pdf
- Pushed source code to git-hub for the first time
- Updated Windows PyMovie.bat file to require the user to press the Enter key to close the script. This keeps the prompt window open so that if anything went wrong, the diagnostic messages will remain visible. This file is created in [C:\Anaconda3](#) the first time PyMovie is run. However, we do not over-write an existing PyMovie.bat file, so if you want the updated file, then you must remove PyMovie.bat from [C:\Anaconda3](#) to

force the re-creation of that file from the copy that comes with the package.

version 0.9.9

- made the gui font size 'OS aware' to try to make the Windows PyMovie gui look as good as the MacOS version. If you run on MacOS, you get a font size of 13 pts. That changes to 8 pts if you run on Windows

version 0.9.8

- modified the setup.py file so that scikit-image $\geq 0.15.0$ will be automatically installed by pip if it is not already present in the Anaconda installation when PyMovie is installed.
- Added messages to the user that appear if the redact lines and/or num frames boxes associated with the Generate "finder" image button are left empty.

version 0.9.6

- added the ability (when working in the context of an AVI-WCS folder) to generate a "finder" image formed by registering and summing several hundred frames. An aperture placed on this image remains when the user switches to the avi. The user then adds a 'tracker' aperture, adjusts its threshold, and runs the analysis.
- to facilitate the use of 'finder' images (produced by stacking images), when a static aperture is added, its threshold is set very high so that it will be forced to use a default mask. An aperture that uses a default mask will not move on its own. So switching from the 'finder' image to the avi can be done without concern that your careful placement will change when the avi comes up.
- added crosshairs to all apertures, a nice visual aid, particularly in zoomed mode.

version 0.9.3

- the calculations for manual WCS calibration have definitely been 'fixed' .
- Miscellaneous small changes to GUI: labels on edit boxes for redact lines and plate scale; clear the VizieR response as soon as a change is made in UCAC4 box.

version 0.9.2

- the calculations for manual WCS calibration have been 'fixed' (we hope).
- added the ability to specify a plate scale to use for the manual WCS calibration. Any value entered in the box to the right of the the Manual WCS calibration button will used as the plate scale. But be sure to leave this box empty if you want the maths behind manual WCS calibration to estimate the plate scale from the calibration points provided; this is the normal expected use.

version 0.9.0

- same as 0.8.9 but added astroquery as a required package. This is used to make VizieR calls but is not a standard part of an Anaconda installation.

version 0.8.9

- this version adds manual WCS calibration. It is not yet very accurate likely due to the assumption that the RA Dec coordinate system covers a small enough area that the curvature of RA/Dec grid lines could be ignored --- this may not a good assumption. Currently a target aperture may be placed 2 to 4 pixels from the correct location (per

nova.astrometry.net WCS calibration).

- Added the ability to get UCAC4 star coordinates through Vizier. This is in support of manual WCS calibration. If the star chart has UCAC4 annotations, it makes it much easier to do the manual WCS calibration process.

version 0.8.8

- analysis can now be run in reverse by setting the current-frame higher than the stop-at-frame. When such settings are used, running an analysis will process frames from current-frame down to stop-at-frame + 1. Then, if you set current-frame to 0 and run the analysis, frames 0 up to and including the stop-at-frame will be processed. Both the plotting routines and the csv write routine will reorder the data based on frame number so the section of data that has frame numbers counting down will be restored to its normal ordering. This somewhat unusual capability make it easier to deal with an video recording that has but a single star (no companion that can used for tracking) that is occulted strongly enough that it is only visible (and so can self-track) before D and after R. In this case, the R side can be better handled by running the analysis in reverse.

version 0.8.7

- WCS calibration through nova.astrometry.net now works for both AVI-WCS folders and FITS folders.

version 0.8.5

- adds an option for WCS calibration when AVI-WCS folder is used to hold observation video (or a link to one) through an on-connection to nova.astrometry.net

version 0.8.4

- removed the self.pointed_at_aperture = None statement in removeAperture method. Just trying stuff.

version 0.8.3

- still working on avoiding crashes while deleting apertures under win7. Added manual disconnect of slot/signals for the aperture before deleting it.

version 0.8.2

- another variation on deleting apertures, this time without the aperture.deleteLater() call.

version 0.8.1

- another attempt to squash the win7 crash-on-delete-then-run bug.

version 0.8.0

- added self.pointed_at_aperture = None when an aperture is deleted. This is a possible cause of referring to an aperture that has been deleted and thus causing a memory violation exception. Crossing our fingers on this one that it finally solves the win7 crash-sometimes-when-aperture-is-deleted problem.

version 0.7.9

- in another attempt to find a solution to the occasional crash that a beta tester has experienced when deleting an aperture, I changed from the use of

`removeItem(aperture)` to `aperture.deleteLater()`. There was some chatter on the internet that this is a better/safer procedure.

version 0.7.8

- restored the automatic green property for a newly added aperture.

version 0.7.7

- fixes the 'jumping mask' problem that appears when 'use yellow mask as default' is in use.

version 0.7.4

- added code to ensure that when the yellow aperture mask is being used as the default mask that it is evaluated first whenever there is a frame change.

version 0.7.3

- returned the option of simply opening an avi file --- use the Open AVI file button.
- Now there is a new button labelled Select AVI/WCS folder for when the user wants to utilize the WCS calibration capability of PyMovie. They will need to create a folder specifically for that purpose. In that folder they will need to place either the avi file to be processed or a shortcut (Windows) or alias (Mac) to that file (so the user does not have to relocate his avi just to satisfy PyMovie). This folder will hold star position and wcs calibration data unique to frame 0 of the avi. As before, this folder will be the default location for csv files and (in the future) png dumps of lightcurve plots.

version 0.6.9

- changed to requiring the use of a folder per observation. In that folder must be exactly one avi and related information such as a little file that gives the target coordinates in Ra and DEC. It also holds the WCS calibration data (if the user has asked/created one) that is used to place a target aperture 'blind' on frame 0. This folder will be the default location for csv files and (in the future) png dumps of lightcurve plots.
- finally found bug that was causing confusing thumbnail updates. Should be better now.

version 0.6.8

- tightened up the spacing between GUI elements to provide more vertical space for text box
- changed labelling of mask threshold spinner

version 0.6.7

- rearrange items on GUI to allow for large Clear plot data button
- added 'hooks' for experimenting with WCS/astrometry.net
- fixed issue where Thumbnail One was not always synchronized with Thumbnail Two (probably)
- made thumbnail display policy the same whether stepping through or running an analysis

version 0.6.6

- new policy: all newly created apertures default to having auto-display enabled.
- to control which of a group of apertures, all with auto-display ON, has its thumbnail

dynamically updated during an analysis run, a property was added to make a specifically chosen aperture be the source for Thumbnail One. There will be either one or no apertures with this property set, enforced by the program. Should there already be an aperture with this property set and you choose a different aperture as the Thumbnail One source, the earlier aperture automatically has this property unset.

- added a warning when a user had been using a RegiStax star locator bitmap and failed to set at least one yellow tracking aperture in the aperture constellation before trying to open the avi or fits file for the observation. Failing to do this will cause the carefully set relative positions of the apertures to be lost as the observation video is read and the apertures all try to 'snap' to 'better' positions, something that is prevented if there is a yellow aperture in the set which locks the apertures together in a rigid constellation.

version 0.6.4

- with the success of 'dynamic' aperture display, we have adopted a policy when creating an aperture of setting all its special properties (jogging and auto-display) OFF and NOT touching the properties of any already-in-place apertures. In the previous version, adding an aperture caused all special properties in existing apertures to be cleared (quietly --- this adds to confusion --- didn't I just set that?) and the new aperture came preset with jogging and auto-display ON. In this version, it is up to the user to set the special properties and they will never be quietly changed.

version 0.6.3

- added 'dynamic' aperture display. If the mouse cursor is placed over/in an aperture, its stats and thumbnail are displayed, regardless of its auto-display setting. This is active during an analysis as well so that any aperture can be examined/viewed during a 'run' simply by pointing at it.

version 0.6.2

- changed the mask for a white aperture to have a black border. This is visually clearer as to what is going and avoids the 'complaints' from within the pyqtgraph histogram widget when all pixels have the same value
- added a 3D display (rotatable and zoomable by left and right mouse drag) for the Thumbnail One image.

version 0.4.8

- now, if you put your cursor on an aperture, the name and major properties of the aperture appear in the status bar in the very bottom left corner of the GUI

version 0.4.6

- added the name of aperture to its context menu --- appears at the top of the right-click menu
- added ability to use arrow keys to jog apertures. Each aperture has a flag to enable it to respond to an arrow key, so apertures could be manually jogged as a group should that be useful. When an aperture is added, it is born 'joggable' and with 'auto-display'
- Eliminated the 'green' mode for an aperture. Now each aperture can request that its stats be printed at each frame change. There is only one thumbnail display however so only the last aperture in a group that gets printed will have its thumbnail displayed.
- Added cross hairs to thumbnail One to aid in manual positioning of apertures.

version 0.4.5

- removed debug printout from Demo Robust
- added auto-naming to no-snap aperture creation (just overlooked it earlier)
- ThumbNailOne no longer uses log scaling. Instead, it uses whatever scaling is in force for the frame image.

version 0.4.4

- added button to read a bmp file, assumed to be a star locator. Any apertures positioned on this image will be preserved when the avi file is opened.

version 0.4.3

- made it unnecessary to close the image range control for level changes to be reflected in subsequent frame changes

version 0.4.2

- the 'no snap' aperture was actually 'snapping' --- that's been fixed
- added a demonstration plot to let the user see visually what the robust mean and std calculations are doing.

version 0.4.1

- made the changes in image display through use of the image range control 'sticky' so that any changes made apply to all images going forward
- changed the number of pixels plot to use lines only (no circles at the data points) and show the negative values (from a default aperture) as positive. It's easier to look at and the data plot above clearly identifies data points that were obtained using a default mask. The redundancy was not needed or useful

version 0.4.0

- added an image range control to give user complete control over image 'stretching'. This is a visual effect only as an aid to identifying stars --- it does not affect any underlying data values.
- Added an 'invert images' checkbox to make it easy to switch between avi and fits files with their differing convention of the image origin (fits: lower left corner; avi: upper left corner)

version 0.3.9

- added fast navigate buttons to speed going forward and backward through images
- now the text box gets cleared when a new image file is successfully opened
- changed the way apertures are deleted in support of chasing a pesky sporadic crash when apertures are deleted
- add a spinbox to allow the user to select the plot symbol size --- it's set by a spinbox that is 'sticky'
- some rearranging of GUI elements
- changed label on threshold box to help user understand better what a value in this box means
- added a legend to the composite lightcurve plot to identify the curves

version 0.3.8

- added a try/except block around aperture delete code to see if we can track down crashes that are occurring sporadically under Win 7 OS (but not Mac or Win 10)

version 0.3.6

- made the log scale image checkbox 'sticky' so that PyMovie can 'adapt' to your preferred image view.
- removed white apertures from the aperture constellation during tracking so that it/they will remain fixed on the image wherever you originally placed it/them.

version 0.3.5

- fixed double yellow tracking that was accidentally broken by a change introduced in 0.3.4

version 0.3.4

- added timestamp printout when running fits files with DATE-OBS in fits metadata
- initialized timestamp correctly

version 0.3.3

- added timestamp extraction from 'fits files under the assumption that they came from a QHY-174M-GPS. This completes PyMovie for users of the QHY-174 that record in int16 fits files.

version 0.3.2

- added the option of a 'white' aperture to be used when flash-tags have been recorded in the video. The special thing about a white aperture is that all the pixels in the aperture are summed (without background subtraction) and output as the 'signal' for that aperture.

version 0.3.1

- field processing of avi files now operational

version 0.3.0

- fixed tracking during wind shake (accidentally broken in 0.2.9)
- now we enable/disable controls that are specific to avi versus fits file processing
- added printout of frames per second read from avi files
- added ability to view both fields of an avi frame
- added update of default mask when aperture size is changed to solve the mask/thumbnail shape mis-match exception from occurring
- added aperture constellation tracking during manual (frame spinbox) changing of frame. It was confusing to have aperture constellation tracking only active during a 'run'

version 0.2.9

- added the printing of the FOURCC codec ID extracted from avi files that could be opened. This may have diagnostic value in some case where an unusual codec was employed. PyMovie handles Lagarith compressed files without issue and without the

- need for the user to separately find and install a Lagarith codec.
- Added the controls for dealing with field level avi stuff --- no code behind the controls yet
- found a way to do random access reading of an avi file so it is no longer necessary to read the entire file into memory!! (which has been a real problem)

version 0.2.8

- generalized the display of FITS metadata to show the header data (metadata) from the current frame rather than just frame 0
- rearranged the columns in the csv file so that all signals appear starting in the column following time stamp data. R-OTE and PyOTE can read this file format.

version 0.2.7

- changed the snap-to-blob function to require user to place cursor on/near star of interest. Previously, snap-to-blob snapped to the brightest star in the entire aperture UNLESS there was already a yellow aperture defined. This seemed overly complicated, hard to explain, and causes the behavior to be sometimes mysterious, particularly if one forgets whether or not there is a yellow aperture. Requiring cursor placement within 6 pixels of the star-of-interest at all times is easier to get used to and allows a dim star that has bright neighbors to be easily singled out.
- The aperture name dialog now pops up whenever an aperture is created. This save mouse clicks and encourages good practices.
- The title on the Number of pixels plot has been expanded to clarify the meaning of negative mask pixel counts.
- Removed Inc threshold and Dec threshold menu items from the aperture specific context menu. The Inc and Dec functions are more clearly handled by the spinbox on the main GUI window.

version 0.2.6

- added PyMovie-info.pdf to the distribution (to make the About button work)
- added opencv-python to the required packages list so that Anaconda installations that lack this package (known as cv2) will have it automatically added
- increased the allowed distance of computed masks from aperture center from 4 pixels to 6 pixels (this only comes into effect when there is a yellow aperture present)
- set thresh = 0 when aperture is created rather than None to deal with cases where an aperture is added where there is no image

version 0.2.5

- Initial Beta release – no timestamp OCR – functional for lightcurve extraction