PyMovie version history

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) wilnot 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 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 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 >= 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(). Their 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 a 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