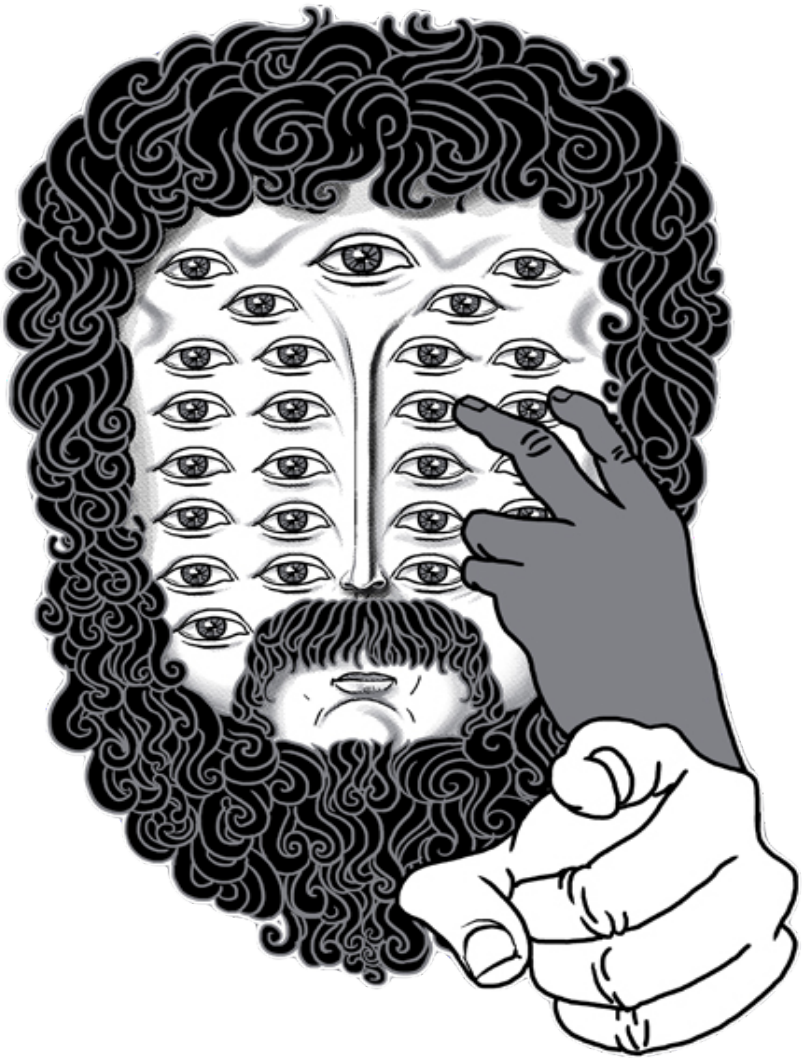


Argus

3D Calibration for the People.
User guide and Documentation

Author: Dylan Ray
Version: 1.1 (2019-01-24)
University of North Carolina at Chapel Hill

Table of Contents

DWarp
A Quick Tutorial

Steps:

1. Open Argus.
2. Open DWarp by pressing the leftmost button labeled '**DWarp**'.
3. Click the '**Open**' button directly under the title towards the top of the window.
4. Browse to find 'dwarp-tutorial.mp4' included in the test files directory and select it.
5. Change the '**Camera Model**' drop down in the '**Lens Parameters**' section to '**GoPro Hero4 Black**'.
6. Change the neighboring '**Shooting Mode**' drop down to '**1080p-30fps-wide (Fisheye)**'.
7. Observe the entry boxes in the '**Lens Parameters**' section becoming un-editable. Fisheye models have coefficients that cannot be changed in DWarp.
8. Click the '**Specify**' button at the bottom of the window and browse for the directory you'd like to write the undistorted test video to. Type the name in as 'dwarp_test_output.mp4' or any other file name ending in '.mp4'.
9. Watch as your video is undistorted and written to the specified location.

Sync
A Quick Tutorial

Steps:

1. Open Argus.
2. Open Sync by pressing the button labeled '**Sync**'.
3. Click the '**+**' button and browse for 'alpha.mp4' included in the Argus test files directory. Select this file. Notice it has been added to the list toward the top of the window.
4. Do the above step for 'beta.mp4' and 'sigma.mp4'.
5. Click the '**Show waves**' button directly below the list, and wait for the audio to be ripped from the three videos.
6. Viewing the three wave files in the window which appears, we can see that synchronization beeps appear throughout the audio, so we can use the whole wave file to cross correlate.
7. Go back to the main window.
8. Uncheck the box '**Specify Time Range**' to use the whole thing.
9. Click '**Specify**' at the bottom and type the name of a csv file to save to such as 'tutorial_offsets.csv', then click '**Go**' and wait for the offsets to be printed and saved to csv.

Patterns
A Quick Tutorial

Steps:

1. Open Argus.
2. Open Patterns by clicking the button labeled '**Patterns**'
3. Click the '**Open**' button directly below the title.
4. Browse for 'patterns_tutorial.mp4' included in the Argus test directory and select it.
5. The defaults in the entry boxes are those for the test video and the printout we include with this suite i.e. 12 row, 9 columns, and a spacing of 0.2 between grid points.
6. Click the '**Specify**' button towards the bottom of the window.
7. Browse for the directory you'd like to write the output Pickle file to. Enter the name 'points.pkl' as the file name or any other file name ending in '.pkl'.
8. Click the '**Go**' button and watch as the patterns are found frame by frame. Some patterns will fail to be found even in our test video. This is normal.

Calibrate
A Quick Tutorial

Steps:

1. Open Argus.
2. Open Calibrate by choosing the right-most button labeled '**Calibrate**'.
3. Click the '**Open**' button towards the top of the window.
4. Browse for the pickle file you created in the Patterns tutorial, 'patterns_tuorial.pkl'. If you didn't complete the Patterns tutorial, this file is included tutorial folder.
5. In the '**Options**' section, change the entry box labeled '**Number of replications**' from 100 to 10.
6. Check the box labeled '**Invert grid coordinates**'.
7. Now click the '**Specify**' button and choose a directory where you'd like to save your calibrations csv file. Name the file 'test.csv' or any other name ending with '.csv'.
8. Click '**Go**' and wait as the 10 replications are finished.
9. After the process, you can take the distortion parameters you obtained and plug them into DWarp to test them. However, the parameters may not be very accurate after just 10 replications.

Clicker

A Quick Tutorial

Steps:

1. Open Argus.
2. Click the second most right button labeled Clicker.
3. Click the plus button on the right and navigate to the video 'alpha.mp4' in the Argus tutorial folder.
4. Repeat step 3 for 'beta.mp4' and 'sigma.mp4'. The offsets for these videos are found the in Sync tutorial. They are -7 and -5 frames respectively.
5. Leave the resolution as the default and click 'Go'.
6. Go to frame ten by hitting G on the keyboard and typing '10' into the popup window.
7. Press X to sync the windows to the same frame. Press it again to turn off sync.
8. Track one side of the visible wand for at least a hundred frames in all three videos. This can be accomplished by clicking on the center of the wand for all the frames or clicking on the center of the wand in the first frame and hitting A to use the auto-tracker. Auto-tracking can be improved by first growing your view finder at the bottom right by using 7, Y, U, and I keys (in arrow configuration). You can view the accuracy of the track in the view finder window. In general it is best to choose a diverse set of frames so that the tracked wand covers a large area in the camera views. Tracking only 100 sequential frames of wand in this video will not result in a usable calibration, but will give you a good idea of the workflow.
9. Hit 'O' to bring up the options dialog.
10. Click the button labeled 'Add track'. This track will be for the other side of the wand.
11. Repeat step eight for the other side of the wand.
12. Press S to save your points as a CSV file. And you're done!

Wand

A Quick Tutorial

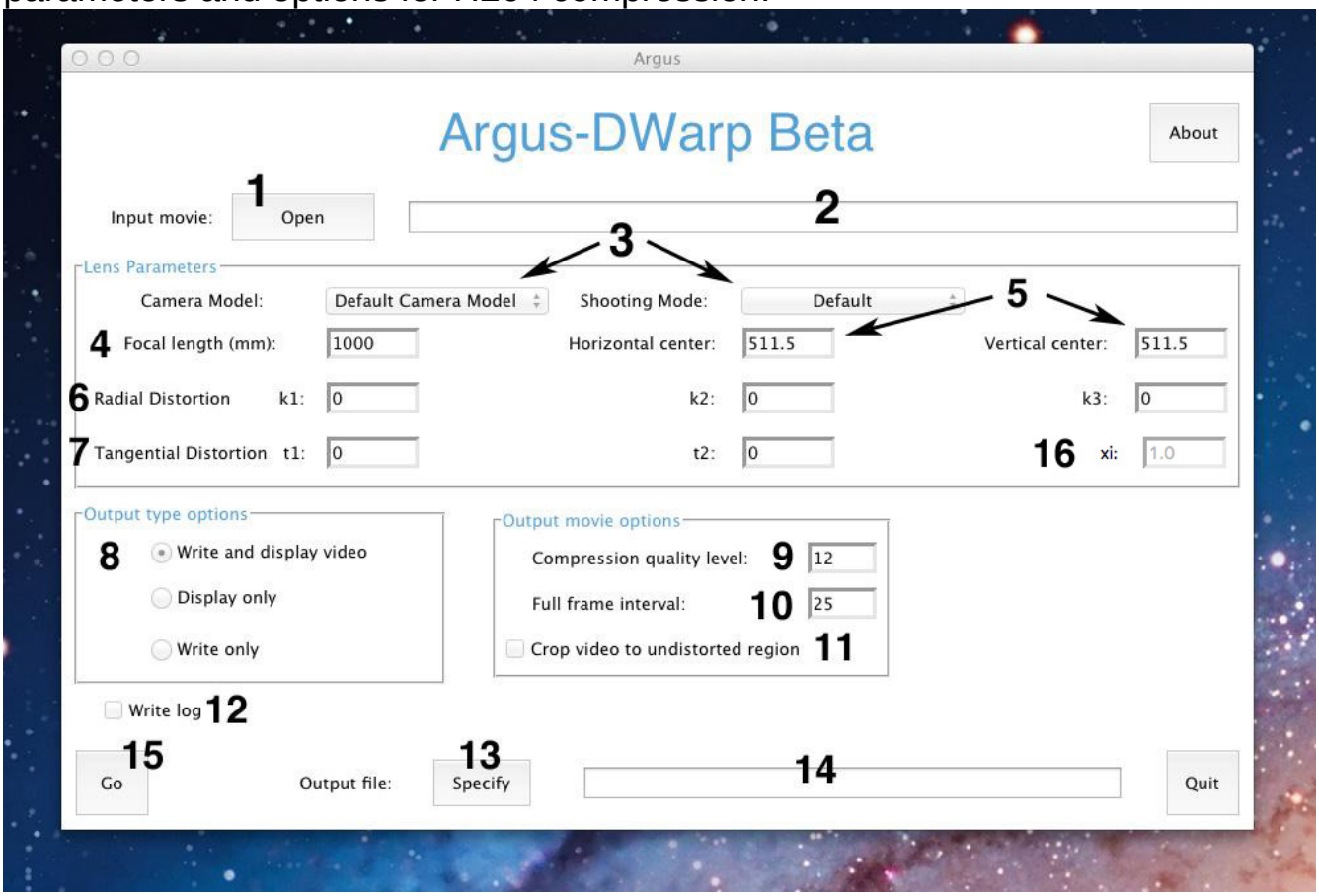
Steps:

1. Open Argus.
2. Click on the right-most button labeled 'Wand'.
3. Click the first button labeled '**Open**' beside 'Open cameras: '. This button allows you to navigate for a Camera Profile .TXT file explained in detail in the run-through section. Select 'wand_tutorial_cam.txt' in the Argus-Wand tutorial folder.
4. Clicker second button labeled '**Open**' beside 'Open Paired points: '. This allows you to browse for a paired points .CSV file. Browse for the CSV file you created in the Clicker tutorial, or the 'tutorial-wand-xypts.csv' included in the package.
5. Set the scale as '0.2' meters or 20 cm.
6. Leave the optimization drop-down menus as their defaults of 'Optimize none'.
7. Hit 'Specify' at the bottom and type 'tutorial' and pick a place on your hard drive to save the results.
8. Click 'Go' to see Wand reconstruct a 3D scene out of the pixel-coordinates and camera profile you provided.
9. The pixel errors for this calibration are very undesirable (> 100). In practice you want pixel errors that are far less than the radius of the object you're tracking. You can improve them by adding more common pixel coordinates in Clicker.

DWarp

An in-depth run through

Often, one wishes to track objects in two or three dimensions using one or more cameras, but having no perfect pinhole camera to record with, the coordinate systems from their videos are warped. DWarp takes distorted video and undistorts using various settings specified by the user including the lens parameters and options for H264 compression.



1. Brings up a file viewer which allows the user to browse for the video he or she wishes to undistort. Currently, DWarp supports '.mp4', '.mov', and '.avi' formats, and the file viewer is set to only display these types of files.
2. Displays the path of the video selected. Can be altered manually if you know the path or if the path is hidden.

3. Drop down menus for selecting the model of camera and the shooting mode you shot your video in if it is supported. DWarp currently supports the following GoPro camera models:

4. The focal length of the lens of the camera you shot your video in measured in millimeters.
5. The horizontal and vertical optical midpoints measured in pixel coordinates.
6. The radial distortion coefficients which describe how pixel coordinates are distorted in a kind of 'barrel' or 'fish eye' effect. They are used by OpenCV to accomplish the following projection:

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

7. The tangential distortion coefficients which describes how antiparallel the lens is to the image plane. For modern cameras, these are very nearly close to zero. However, they are used by OpenCV to accomplish the following projection. Here they are represented as p_1 and p_2.

$$x_{corrected} = x + [2p_1 xy + p_2(r^2 + 2x^2)]$$
$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]$$

8. Option whether to write and display the undistorted video, simply display it, or simply write it. If you choose to write, the video will be written frame by frame as pngs to a temporary directory. Afterwards, it will be re-compressed into an .mp4 file with FFMPEG maintaing the frame rate, audio track, and resolution.
9. An FFMPEG parameter also called quantizer scale (any integer between 0 and 51). Lower is better. 0 will result in lossless video, 23 is default for FFMPEG, and 51 is the worst setting.
10. The maximum number of frames in between a key frame or full frame.
11. Check this option if you would only like to see the undistorted region in your outputted video.
12. Check this option if you would like to record everything that appears in the log window when you click '**Go**'.
13. Specify the directory and name of the video to be outputted. Cannot be a

file that already exists. Name must end with '.mp4'

14. Displays the future path of the video you will be writing.
15. Click to begin the undistortion process with the specified settings.
16. Lens shape parameter for new Omnidirectional model included in OpenCV 3.0.

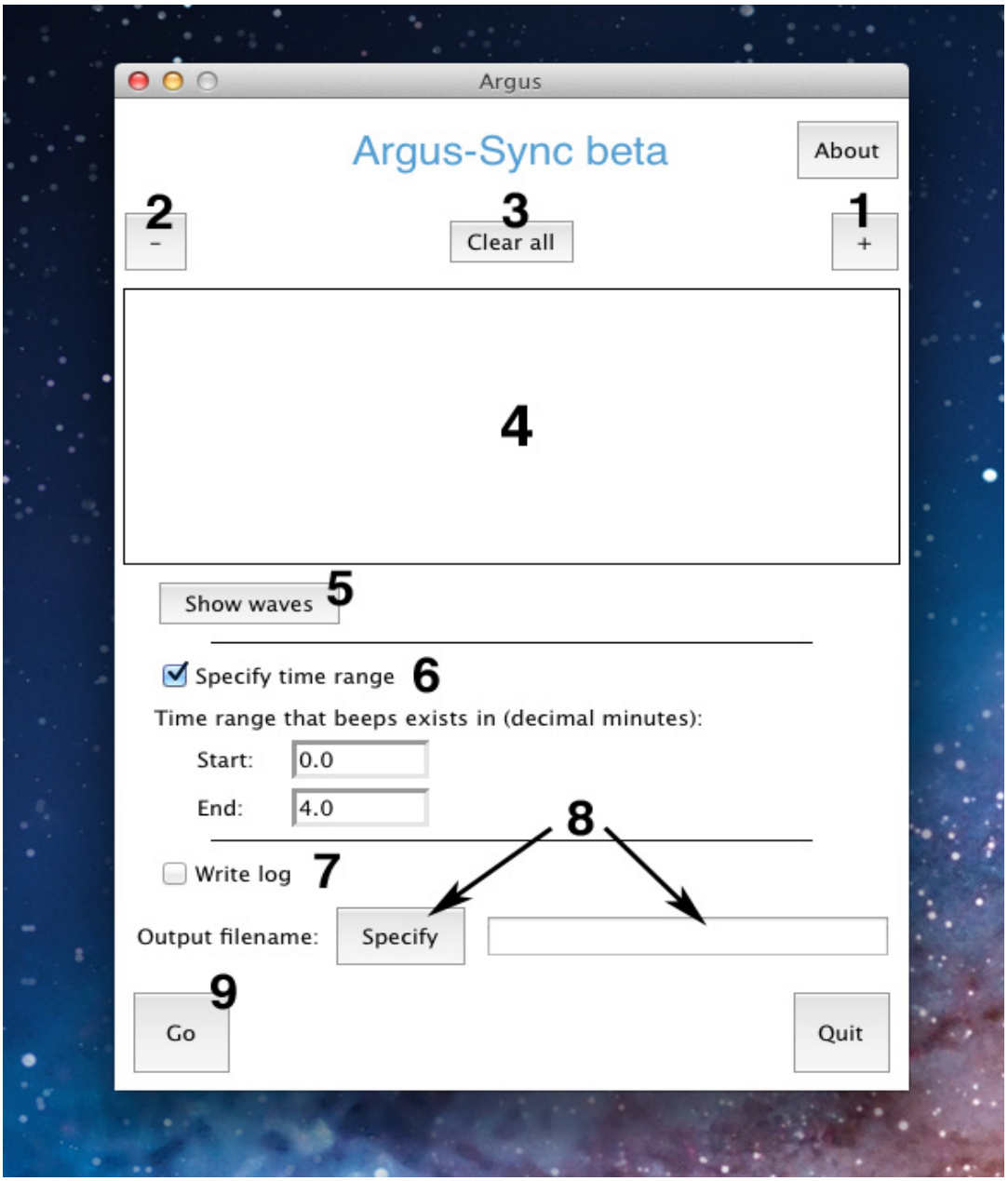
Sync

An in-depth run through

Most modern cameras have devices sold separately which allow you to fire some number of them simultaneously. However, with many of the cameras we've used, including GoPros, the streams do not always start at the exact same time. Sync allows you to find the offset between any number of videos which are meant to film the same scene synchronously. It uses the audio tracks of the videos to accomplish this.

Proper field procedures:

Sync banks on the fact that for all the cameras you are trying to find the offset for, they've recorded multiple high frequency, high intensity sounds at the same time. To ensure this, make sure the sounds are produced equidistant from all of the cameras to reduce any phase shifts and ensure there is as little background noise as possible when this happens. Our procedure was to hang walkie talkies on the tripods of our cameras and then send multiple beeps to all of them somewhere in the recording process, usually towards the beginning or end of our videos.



1. Click this button to browse for a video to add to the list of videos to be synced.
2. Click this button to removed the last video added or a video selected from the list. If any wave file has been cached for this video, it is deleted.
3. Click this button to clear all videos from the list and any cached wave files. If Sync is returning strange errors in the log, try clicking this button and starting over.
4. A list of videos to find the offset between. Sync find the offset between the

first video in the list and all the others.

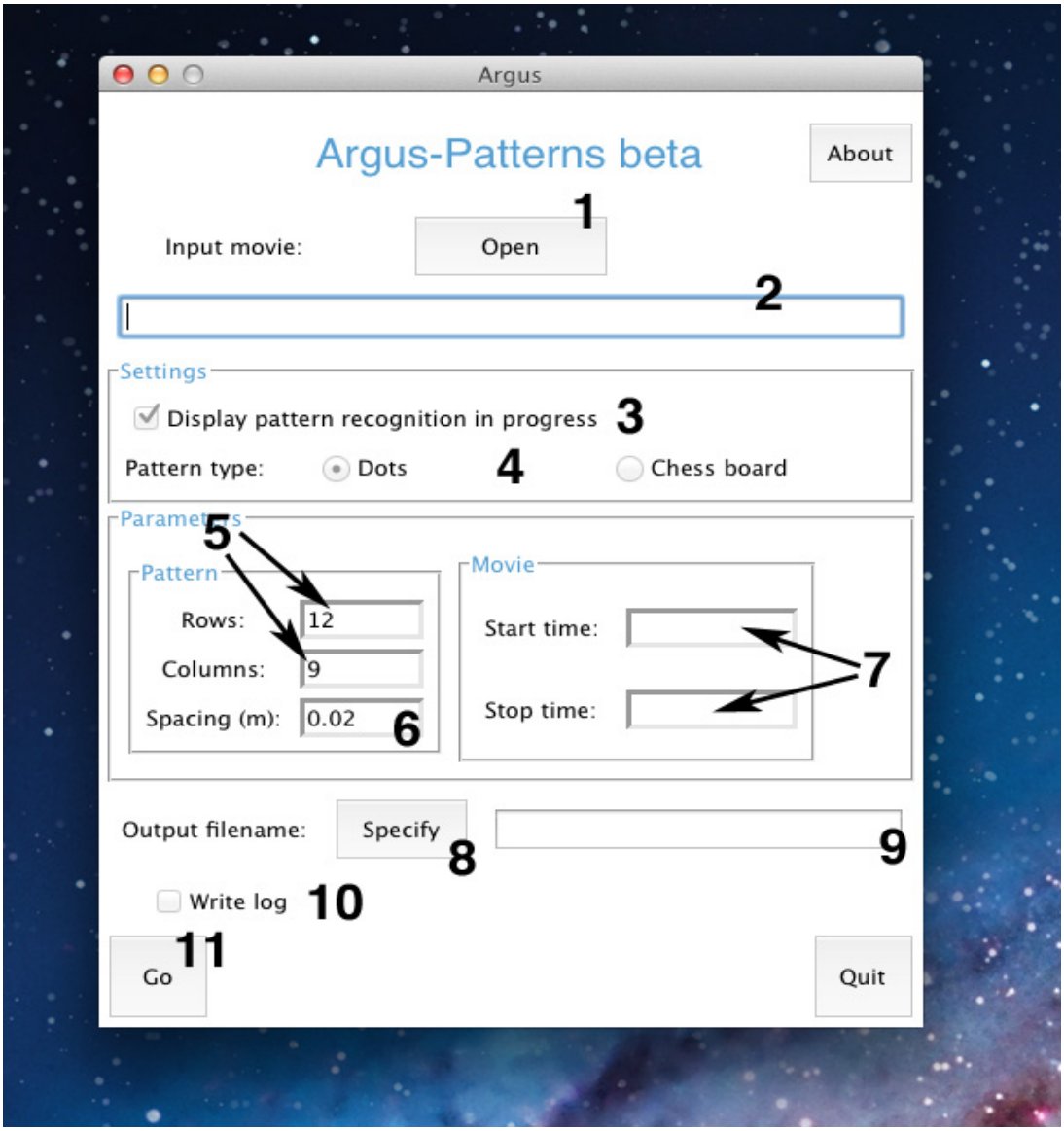
5. Click this button to view the audio waves of the videos in the list. This is a recommended step as it allows one to determine the time range in decimal minutes when the high frequency sounds occurred. Also, if you keep the same files in the list, Sync will not have to rip the audio again to find the offsets.
6. Check this box and fill in the entries below to specify a time range where the beeps exist for all the videos. This is highly recommended especially for long videos. The operation which finds the offset finds the cross correlation at all possible lags for the wave files, so the shorter the interval you specify, the faster the operation will execute and the less RAM it will use.
7. Check this box if you'd like everything written in the log to be saved to a .txt file.
8. Specifies and displays the path of the to-be-written CSV file with the offsets.
9. Click this button to begin finding offsets. After the offsets are found, they are displayed in a table in the log window.

Patterns

An in-depth run through

Patterns is one of two programs in the Argus suite which allows one to determine the lens parameters (the very same that are used in DWarp) for any given camera. It uses OpenCV and user specified input to find the 'image points' (what you see) and the 'object points' (what you ought to see) of a grid filmed at various angles. It writes these two coordinate sets for every frame which its algorithm can determine them for to a '.pkl' or Pickle file which can then be used by Calibrate to determine the lens parameters.

Procedures for filming grid:
Film the provided grid on a dark background from various different angles. The grid should take up a large portion of the frame. Do not move the camera too fast when changing angles, as the motion blur will confuse OpenCV's algorithm and result in unfound patterns.



1. Click to browse for a video to use. Like DWarp, Patterns currently only supports '.mp4', '.mov', and '.avi' files.
2. Displays the path of the video chosen. Can be edited manually if you know the path or if the path is hidden.
3. Check this box to display the video as patterns are found. Patterns are superimposed over the grid by OpenCV. If the pattern is not found for a certain frame, you simply see a small grid off to the corner.
4. Use this radio button to choose whether the video is of a grid or of a chess board.
5. Specify the number of rows and columns in the grid. It's not always intuitive which one is which, and so if no patterns are being found, try

switching these parameters.

6. Spacing between grid points.
7. Start and stop times if specifying a range. Times are in decimal minutes.
8. Click this button to browse for a directory to write your Pickle file to and choose a name for it. Name must end in '.pkl'.
9. Displays the future path of the Pickle file to be written.
10. Check this box to write all that appears in the log window to a text file.
11. Click this button to begin finding patterns.

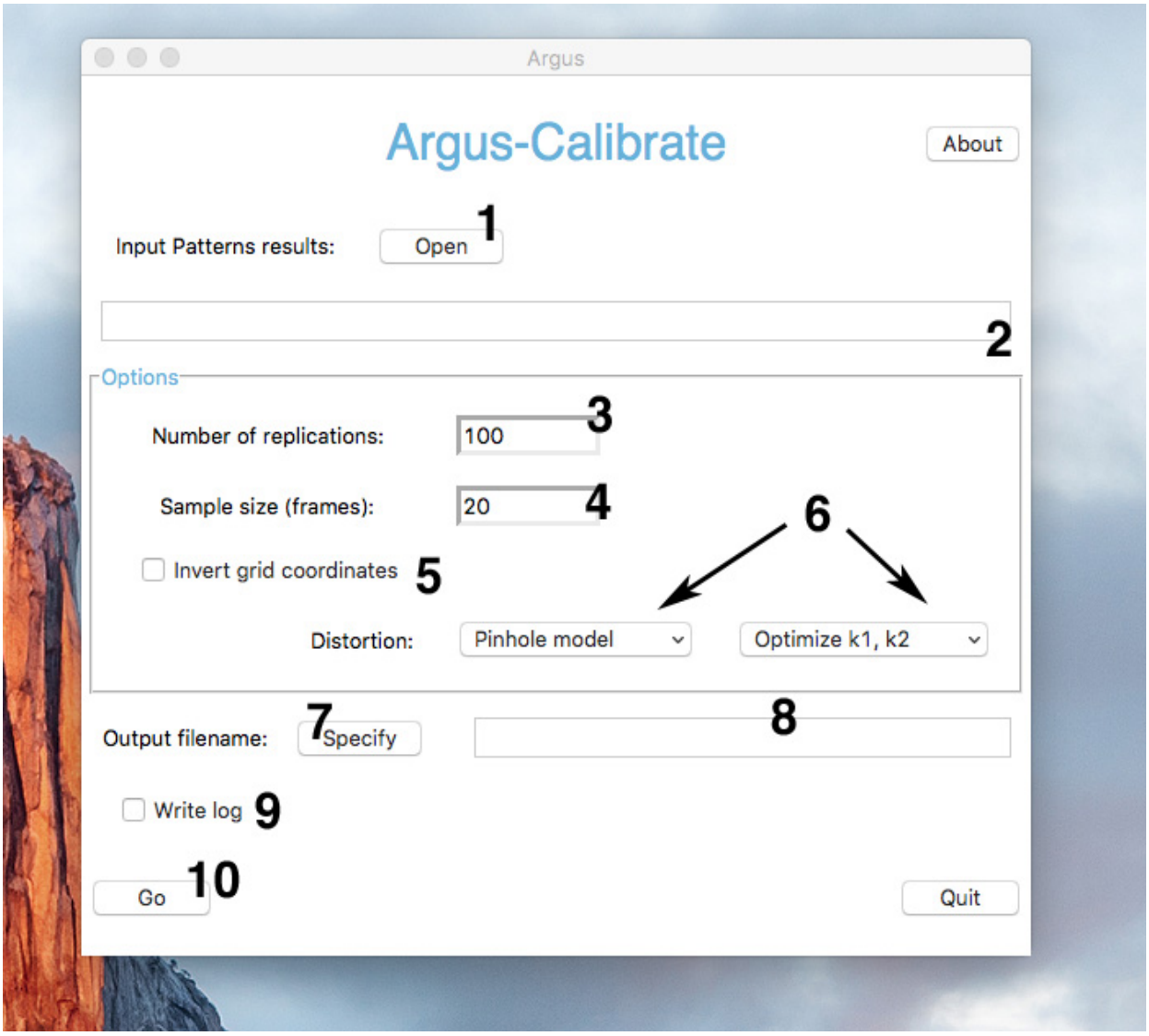
Calibrate

An in-depth run through

Calibrate takes Pickle (i.e. *.pkl) files made by Patterns and uses OpenCV to solve the distortion equations for the lens parameters. The distortion equations can be written as:

$$x_{corrected} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_{corrected} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$x_{corrected} = x + [2p_1 xy + p_2(r^2 + 2x^2)]$$
$$y_{corrected} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]$$
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Where the first two sets of equations are for radial and tangential distortion, and the last expression describes the transformation between object and image points. Calibrate solves these equations by taking a sample of the frames found in Patterns multiple times and afterwards taking the coefficients with lowest root mean squared projection error. We recommend using the defaults for sample size and number of replications to get good results.



1. Click this button to browse for a Pickle file created by Patterns.
2. Displays the path to the Pickle file specified. Can be edited manually if the path is known or hidden.
3. Number of trials where a sample of frames from the Pickle is taken each time and the distortion equations are solved numerically.
4. Number of frames sampled for each replication.
5. Makes the object points negative, effectively flipping them over the vertical axis. This has to do with the orientation of grid relative to the camera. We don't really know why it's important to OpenCV when you have a symmetric grid, but if you're getting warnings about high RMSEs, try checking this option and starting again.
6. Use these drop-down menus to decide what model to use and what

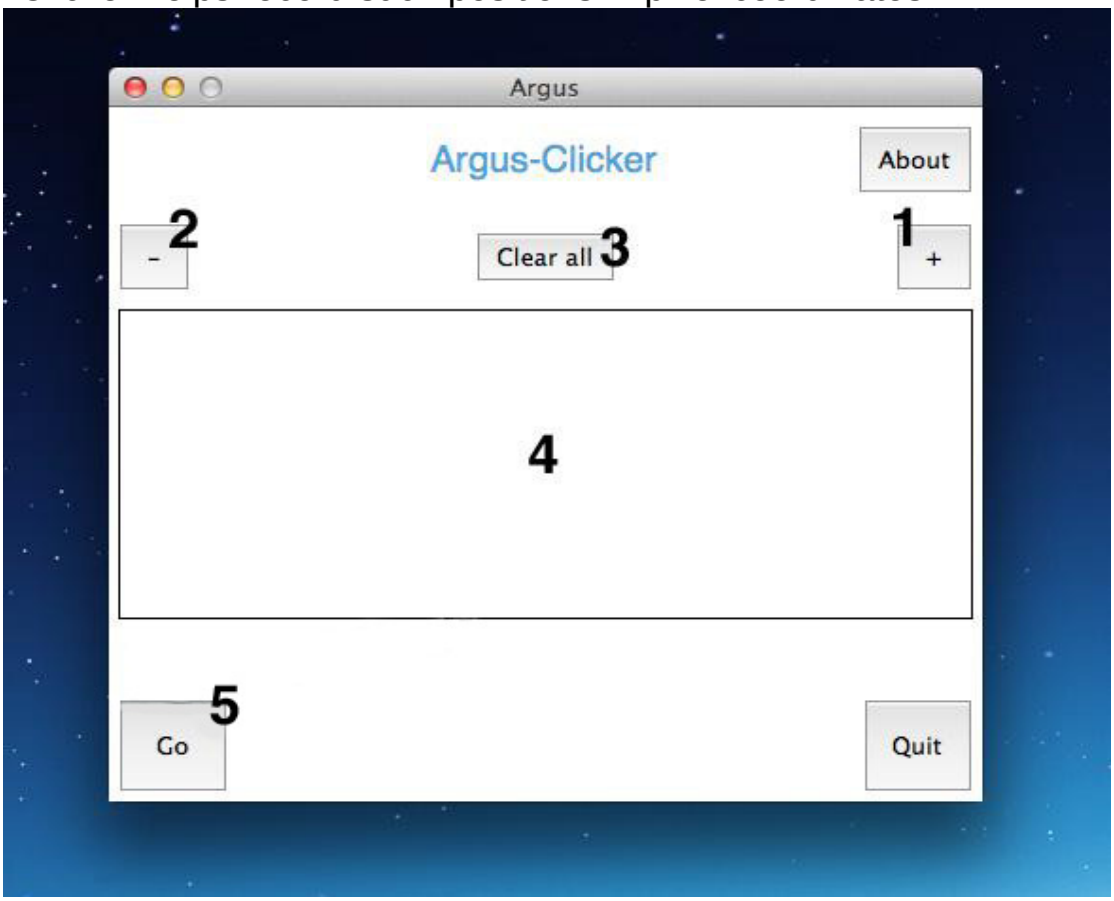
distortion coefficients you want to solve for. If using the standard Pinhole model, we recommend using the default setting as k3, t1, and t2 vary quite a lot from trial to trial. When using C. Mei's Omnidirectional model, all distortion coefficients are solved for due to weirdness in OpenCV routines. Use the omnidirectional model for high-FOV, highly warped cameras.

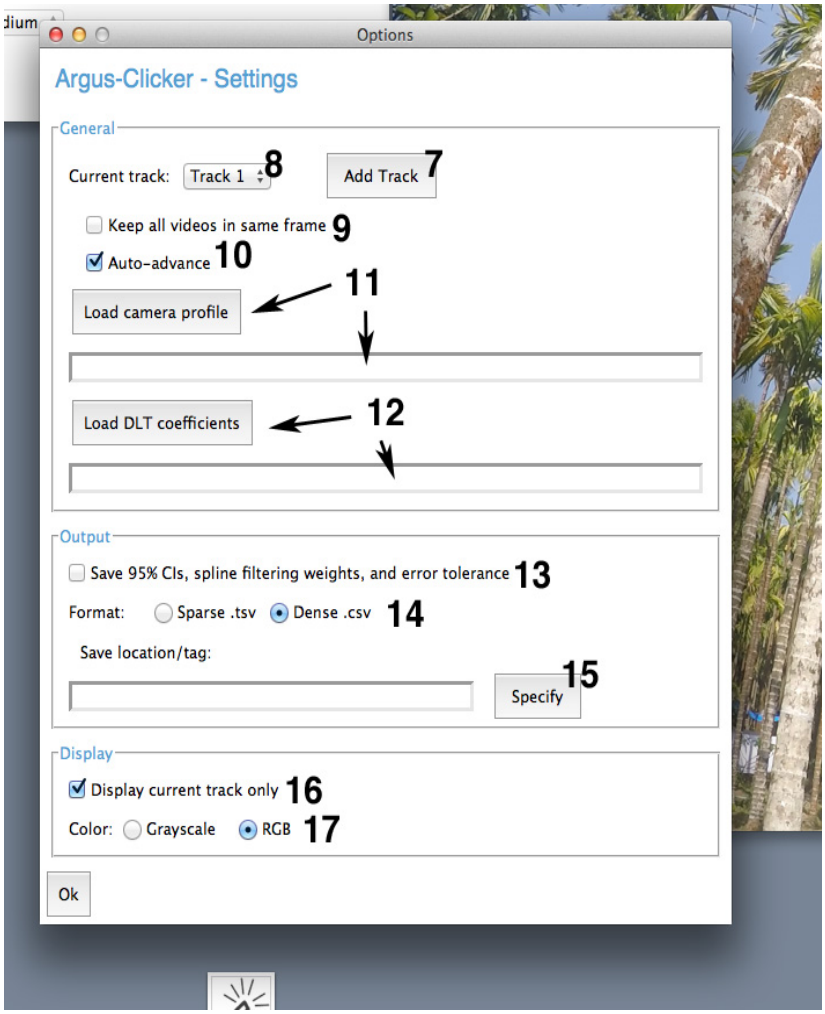
7. Click this button to browse for a directory to write your calibrations csv file. You must specify a name that ends in '.csv'.
8. Shows the path of the .csv file to be written.
9. Check this box to write everything appearing in the log to a text file.
10. Click this button to begin getting replications.

Clicker

An in-depth run through

In order to track objects in 3D or 2D by viewing said objects with multiple cameras, one must know *where* in those views the object showed up frame-by-frame. Clicker helps record such positions in pixel coordinates.





1. Add a movie to the list of movies you'd like to 'click' through.
2. Delete a movie from the list.
3. Clear all movies from the list.
4. Displays the list of movies.
5. Press to begin clicking.
6. N/A (empty space)
7. Add a track. A track signifies a series of pixel coordinates that corresponds to an object seen in the camera view.
8. Drop-down menu that contains the tracks you're working with. The selected track is the current one you're working with.
9. Specifies whether or not to keep all videos synced or on the same frame. In other words, when you skip forward in one video you skip forward in all

the rest.

10. Specifies whether or not to go to the next frame after marking a point.
11. Load a camera profile, containing intrinsics and distortion information
12. Load DLT coefficients made in Wand or elsewhere.
13. Specifies whether or not to save 95% confidence intervals for 3D points as well as spline filtering weights and error tolerances. Argus does this through bootstrapping which can take a long time upon saving.
14. Specifies the file format you wish to save in. Sparse TSVs (Tab separated) only include non-NAN values marked with row and columns and can save disk space and read/write times. CSVs contain the dense matrix of points, NAN values included.
15. Specify the output file location and tag. This tag is appended with things like '-xypts.csv' or '-xypts.tsv'. Save often!
16. Specifies whether or not you wish to display all marked points at once, rather than the current, working track.
17. Display frames in RGB (color) or in Grayscale.

Non-keyboard commands:

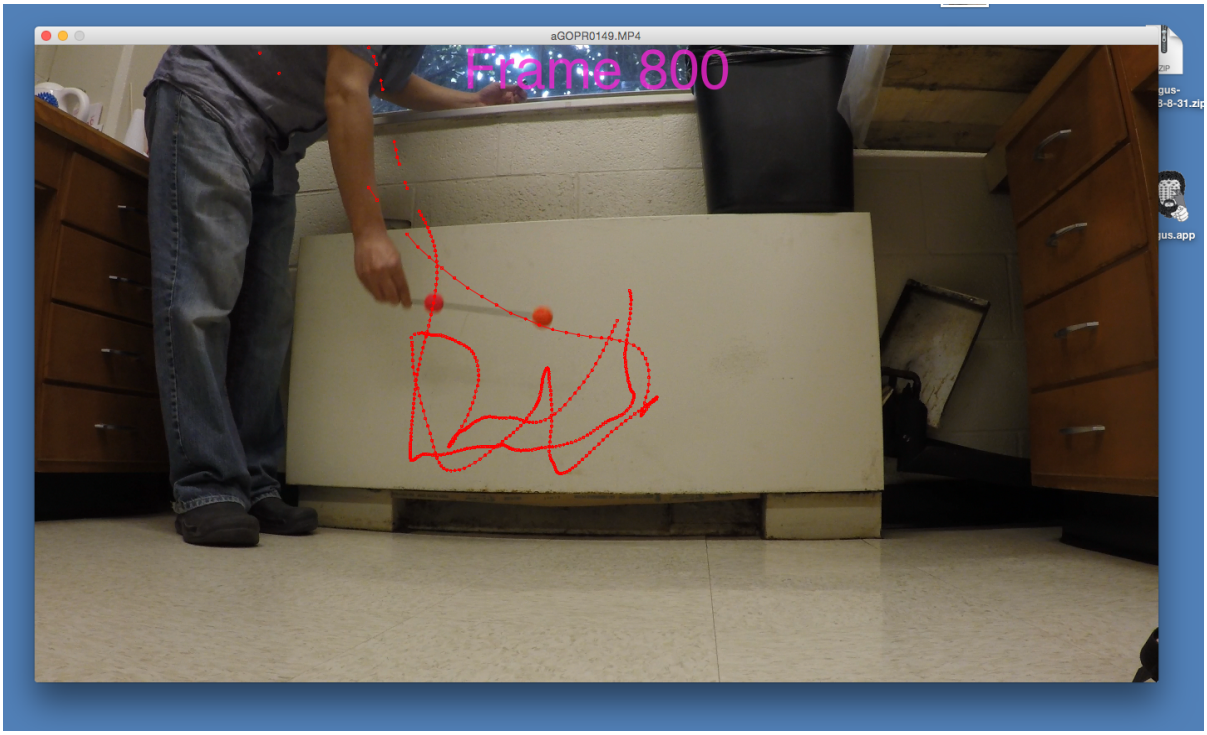
- Click: Marks a point in the current working frame and track
- Ctrl-click: Samples a color for auto-tracking
- Shift-click-and-drag: Pan in the image.
- Scroll: Zoom in or out.

Keyboard commands:

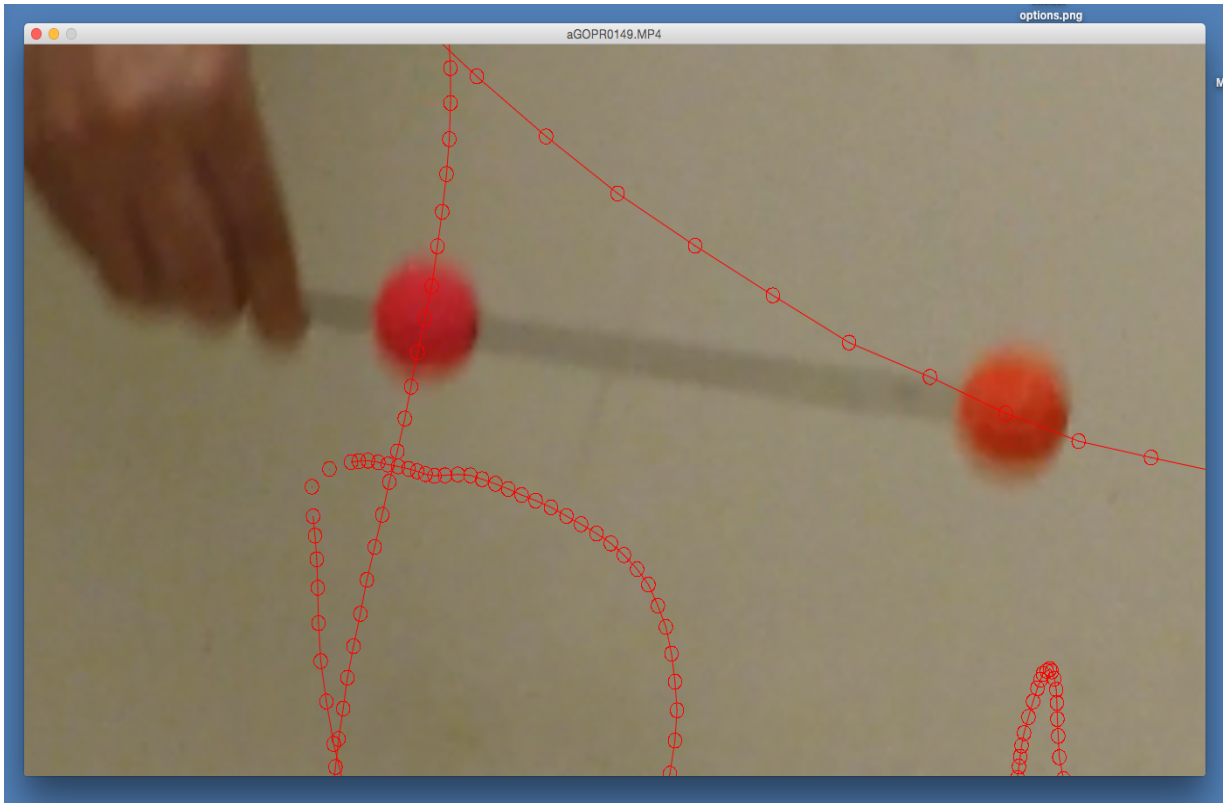
- A: Starts Kalman-filtered, 2D cross-correlation based auto-tracker (must have a point marked in the previous frame)
- B: Go back a frame.
- C: Clears all tracks for the current working video
- Shift-C: Set directory for saving frames to
- Alt-C: Capture the current frame to specified directory as JPG (does not capture tracks)
- D: Delete the point in the current frame and current track.
- Shift-D: Delete the current track with confirmation.
- F: Go forward a frame.
- Shift-F: Go forward 50 frames.
- G: Go-to a specified frame
- , : Go to previous track in track list
- . : Go to next track in track list
- R: Reset view to full frame.

- Right-arrow: Go to the frame with the next marked point
- Left-arrow: Go to the frame with the previously marked point
- Up-arrow: Increase offset by one frame for current video
- Down-arrow: Decrease offset by one frame for the current video
- O: Bring up the options window
- S: Save the points
- Ctrl-S: Save as
- V: toggle the display of the view finder in the lower right corner
- X: Toggle the 'sync' option which keeps all videos on the same frame
- 7: Grow the view-finder vertically by a pixel
- U: Shrink the view-finder vertically by a pixel
- Y: Grow the view-finder horizontally by a pixel
- I: Shrink the view-finder horizontally by a pixel
- P: Plot 3D positions (requires DLT coefficients and a camera profile)
- L: Load points (from .csv or .tsv file)
- 1: Background subtraction (starts a background subtractor that saves every frame skipped to; press twice to reset)

Tracking one side of a wand:



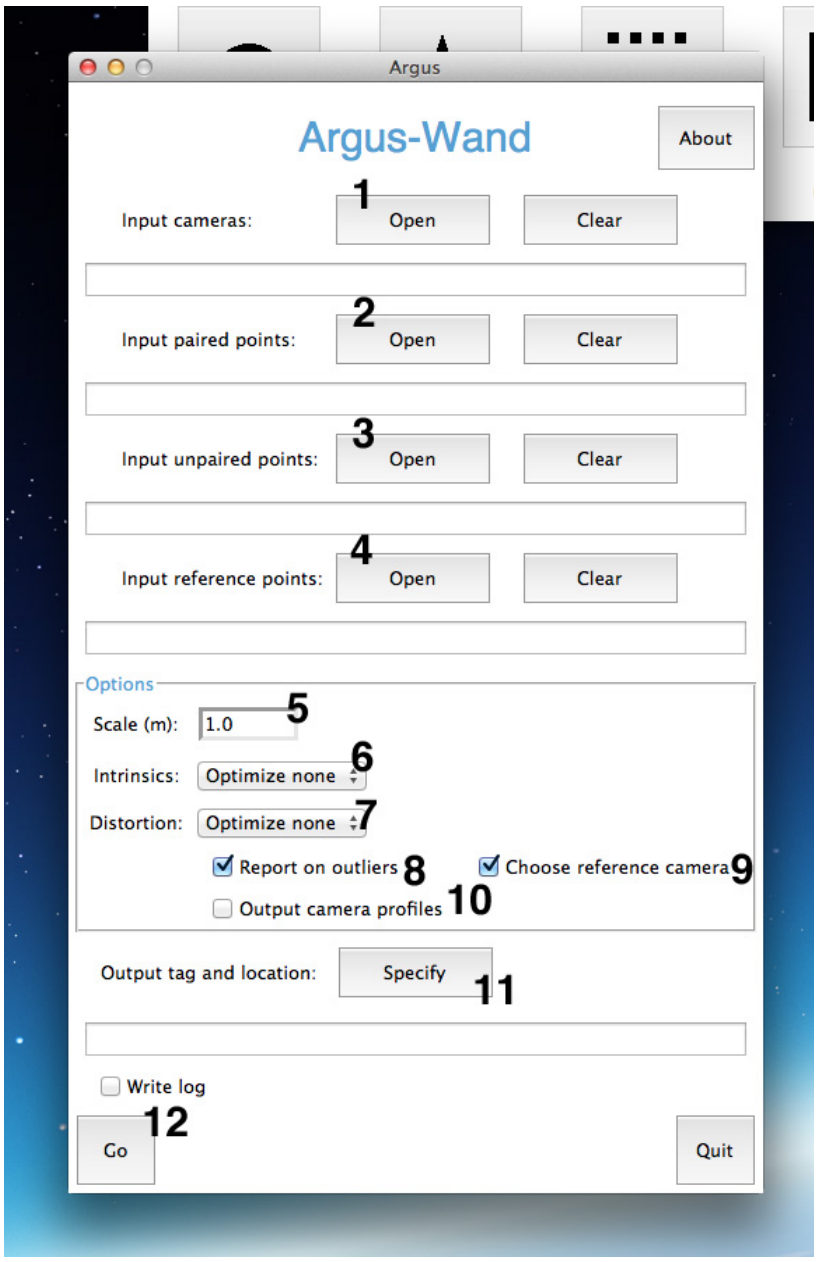
Zooming in on the track (see that the current point is marked in the middle of the red sphere):



Wand

An in-depth run through

Wand marks the end of the 3D tracking workflow as it takes points marked in Clicker and constructs a 3D path with real scale. It also outputs camera extrinsics, i.e. where all the cameras are in a scene, and can optimize camera intrinsics such as focal length and distortion coefficients.



1. Input the camera profile text file. Must have as many rows as there are cameras in the scene. Must have ten columns with all values separated by spaces. The ten columns correspond to the value camera intrinsics: focal length (pixels), horizontal center (pixels), vertical center (pixels), aspect ratio, skew, first radial distortion coefficient, second radial distortion coefficient, first and second tangential distortion coefficient, and third radial distortion coefficient.
2. Input the paired points. Paired points are the coordinates of two objects which you've tracked with Clicker that are the same physical distance from

each other in every frame, for example a wand. By filming a wand or some other object which maintains its shape throughout the scene, you give a scale to the 3D coordinate system which wand spits back at you. If you have no paired points, the scale in Wand's output is arbitrary but consistent. The paired points file must be a CSV delineated by commas with no row index. It must have 4 times the number of cameras columns. That's (2 pixel coordinate values (x and y))*(2 objects tracked)*(the number of cameras) = 4*(number of cameras)

3. Input the unpaired points. Unpaired points are simply the objects which you've tracked, for scientific purposes or otherwise, in Clicker. There is no requirement of constant distance between the objects, and no limit to the amount tracked objects Wand will take. Again, it must be a CSV file delineated by commas with no row index. It must have a multiple of 2*(number of cameras) rows.
4. Input reference points as a CSV file. These points are used to define the origin and direction of the XYZ axis. There must be one, two, or three points marked in all N cameras. Wand currently supports only one track for reference files. As such the correct CSV file will have 2*(number of cameras) rows with one, two, or three points marked in different frames.
5. Input the measured distance between the paired points.
6. Specify what intrinsic elements of the camera profile you'd like Wand to optimize.
7. Specify which distortion coefficients you'd like Wand to optimize.
8. Report on outliers at the end of optimization. Gives the user an option to try again with outliers removed
9. Optimize the choice of reference camera such that there are the most 3D triangulatable points, i.e. camera with the most shared information.
10. Option for outputting camera profiles. Outputted camera profiles can be used in Clicker.
11. Specify a tag and location for the outputted files. Wand outputs:
    - 3D coordinates for paired and unpaired points with frames intact
    - SBA profile which includes camera intrinsics and quarternions and translations which describe camera positions and orientations relative to each other (see next section)
    - Camera profiles for all cameras (If option 9 is selected)
12. Start Wand!

File Formats and Protocols

Program	Input filetypes
Dwarp	MP4, AVI, MOV
Sync	MP4, AVI, MOV
Patterns	MP4, AVI, MOV
Calibrate	.pkl files produced by Patterns
Clicker	Movies: - MP4, AVI, MOV Camera Profiles: - .txt files DLT Coefficients: - .csv files produced in the same fashion as Argus-Wand
Wand	Camera Profiles: - .txt files Paired and unpaired points, and reference points: - .csv files (formatting specified in the in-depth run-through of Wand (pg. 24))

Notes on formatting:

Camera Profiles:
Argus Calibrate outputs a camera_profile.txt file with the best camera profile and also a user specified CSV file with all profile replicates. These contain essentially the same information, but in slightly varying format. Clicker uses the Calibrate output structure, but requires a file with one line for each camera. Column orders are as follows:

Camera Index (1, 2, 3, …) [camera_profile.txt only]
Focal Length (pixels)
Image width [camera_profile.txt only, can be 0 for Clicker]
Image height [camera_profile.txt only, can be 0 for Clicker]
Horizontal optical center (pixels)
Vertical optical center (pixels)
Aspect Ratio (Always one for modern cameras)

Skew (Always zero for modern cameras) [replicates *.csv only]
2nd order radial distortion coefficient (k1)
4th order radial distortion coefficient (k2)
1st tangential distortion coefficient (t1)
2nd tangential distortion coefficient (t2)
6th order radial distortion coefficient (k3)
omnidirectional parameter (xi) [OpenCV omnidirectional model only]
rmse [replicates *.csv only]

Clicker also supports loading camera profiles subscribing to two generalized Omnidirectional or Fisheye models: Scaramuzza's (calibrated with Ocam Calib, a MATLAB toolbox) or C. Mei's calibrated internally with OpenCV 3.0. The latter has the same general form as the pinhole model and even has non-zero pinhole coefficients, with the addition of one final parameter (xi).

Scaramuzza's model, however, is quite different than the camera profile above. The camera profiles are still *.txt files with values delineated by spaces, but each line of an Omnidirectional camera profile contains the same information as does the calibration CSV files that come packaged in DWarp in the following order:

FC (a number specifying the height of the projected upon image plane when undistortion is accomplished)
Image width
Image height
C
D
E
Horizontal optical center
Vertical optical center
SS polynomial coefficients (five coefficients in all)
Other polynomial coefficients (polynomial of an arbitrary order)

SBA profile

Wand outputs a file called an SBA profile. This contains information that was optimized by SBA (Sparse Bundle Adjustment) during the calibration, including camera intrinsics and extrinsics. There is a row for each camera in the order that they were passed to Wand. This file is not used directly by any other The columns of that profile are as follows:

Focal Length (pixels)
Image width
Image height
Horizontal optical center (pixels)
Vertical optical center (pixels)
Aspect Ratio (Always one for modern cameras)
Skew (Always zero)
2nd order radial distortion coefficient (k1)

4th order radial distortion coefficient (k2)
1st tangential distortion coefficient (t1)
2nd tangential distortion coefficient (t2)
6th order radial distortion coefficient (k3)
4 components of a quaternion specifying the rotation from the last camera
3 components of translation from the last camera (arbitrary scale)

Focal lengths of Omnidirectionally undistorted wide-angle shooting modes:
GoPro Hero4 Black: (All shooting modes were calibrated with ProTune defaults)

Shooting Mode	Focal Length (pixels)
4k	910.7
2.7k 4:3	470.88
2.7k	702.67
1440p	456.32
1080p	488.01
960p	261.07
720p	320.53
WVGA	215.41

GoPro Hero3 Black: (ProTune off)

Shooting Mode	Focal Length (pixels)
4k	963.88
4k-cinematic	950.7
2.7k	672.26
2.7k-cinematic	555.4
1440p	477.33
1080p	472.95
960p	366.92
720p	318.3
WVGA	212.5