**Functional Design**

**Nematode Feature Extraction System**

**8/8/2014**

1. **Main Processes**

The software system is currently designed as seven separate processes that can be executed independently, allowing for human intervention to examine output and fix errors, if necessary. The high level processes are listed in Table 1.

Table 1: High Level Process Descriptions

|  |  |
| --- | --- |
| **Process Name and**  **Configuration Function** | **Process Description** |
| extractContourAndSkel.m  getEnv\_extractContourAndSkel.m | Creates a Matlab file (structure array) with lists of pixels representing the contour and skeleton of the nematode in each frame. It also calculates shape and size features, head and tail locations, and curvature estimates at the head and tail, all of which can be used to validate the contours and skeletons. |
| createLoopVideo.m getEnv\_createLoopVideo.m | Creates a gray-scale video by extracting frames from a video that consist of only looping sequences and a specified number of frames before and after the sequence. |
| aggregateLoops.m getEnv\_aggregateLoops.m | Creates a CSV file summarizes loops in a video sequences. Each row represents a series of frames representing a loop. A loop sequence number is assigned, the beginning frame, the ending frame, and the series of postures for each frame in the loop are reported. |
| createSegVideo.m getEnv\_createSegVideo.m | Creates an annotated gray-scale video from the data structure of contours and skeleton points. Annotates the head, tail, skeleton, and contour on each gryscale video frame. lists. |
| extractFeatures.m getEnv\_extractFeatures.m | Adds additional features to the data structure of contours and skeletons, and saves it to a new name.. |
| fixSktps.m  getEnv\_fixSktps.m | Fixes skeletons and contours from an existing file by deleting specified frames and verifying that the head/tail designation is correct in specified frames. Saves the results to a data structure with a new name. |

1. Process Standard Behavior

Each process begins by calling an associated configuration function to obtain constants and configuration parameters specific to the process. One of these parameters is the study instance name, which is the name of the directory where all fields related to the analysis of a single video are kept.

The process creates a study instance log in the study instance folder if it has not already been created. The process then writes its name and execution timestamp to the log.

Next the process creates a log file in the study instance folder specific to its current execution. This log file will include information about the execution, including all configuration parameters, execution time, and errors.

Each process then reads and writes to a Matlab variable in the study instance folder that contains all the data currently associated with each frame of the video. This variable is a structure array with a field for each variable. The data items that are written to the data list are described in Appendix A. After the process adds data or modifies data, this data structure is saved to memory at the end of the process, so it can be recalled from memory to be operated on by another process. In addition, any data in the data list that can be represented as a single value is also saved to a comma-separated text file at the end of the process. All files created (log files and output files) are time stamped so that they can be associated together and saved to the same study instance folder.

1. Standard Analysis Workflow

A typical order of execution is depicted in Figure 1.

1. **Timing and Spatial References**

Spatial and timing reference information is extracted from the video and the video tracking log files. The software is designed to handle three coordinate systems or spatial references: a global reference frame, a field of view (FOV) reference frame, which defines the entire field of view of the camera at its current location, and a crop reference frame, defined by a crop window within the camera field of view. Given tracking data and supplied constants, the system calculates a simple row and column offset to easily translate between the global (highest level) reference and local (lowest level) references when necessary for particular feature calculations.

The inputs to the system include an initial starting location for the camera, which is arbitrarily defined, camera movement steps at each frame, and crop box location and size at each frame, if it is defined. If no camera steps are provided, the step for each frame is set to zero. If no crop box location and size is provided, the locations are set to the origin, i.e., (1,1), and the size is set to the size of the video frame. Since the coordinate system within each frame is the image pixel coordinate system with the origin at the upper left and having the value of (1,1), a total offset can be calculated as :

{Total Offset} = {Camera Starting Position} + {Total Camera Offset} + {crop location} – 2

And then the transformation between local and global is simply:

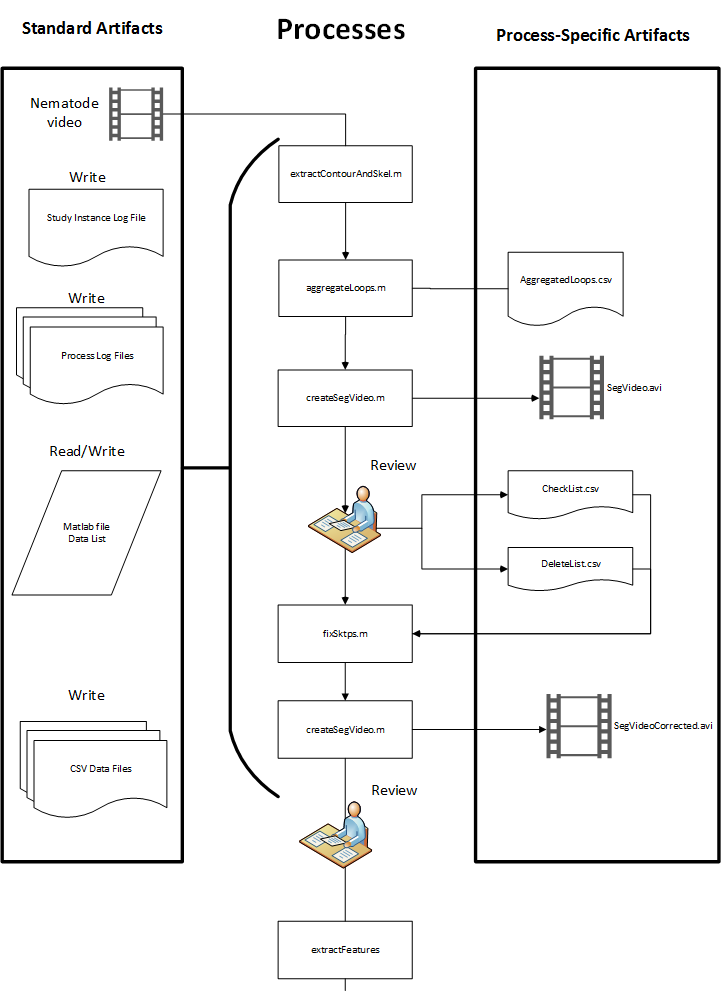
{Global Location} = {Local Location} + {Total Offset};

or : {Local Location} = {Global Location} - {Total Offset};

1. **Output Data**

Data saved to the data structure includes those items listed in Appendix B.

Appendix A – Standard Analysis Workflow



**Appendix B**

Variables in the Data List

Notes

* 1. Can be modified by the fixSktp process
* 2. Not included in the Csv file
* 3. The SkewerSktp with Centroid Origin - the skeleton points converted to Cartesian coordinates and rotated so that the skewer line (connecting the head to the tail is parallel to the x axis and centroid of the curve is the (0,0) origin.
* 4. The SkewerSktp with Midpoint Origin - the skeleton points converted to Cartesian coordinates and rotated so that the skewer line (connecting the head to the tail is parallel to the x axis and the midpoint between the maximum and minimum x and y values is the (0,0) origin.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable** | **Long Name** | **Category** | **Derived From** | **Description** | **Process**  **and Function Call** |
| SeqNum | Sequence Number | Reference | Video | The number of the data row | extractContourAndSkel( ) |
| FrameNum | Frame Number | Reference | Video | The frame number of the original video sequence. | extractContourAndSkel( ) |
| NumRows | Number of Rows | Reference | Video | The number of pixel rows in the image. This should be the same for all data rows. | extractContourAndSkel( ) |
| NumCols | Number of Columns | Reference | Video | The number of pixel columns in the image. This should be the same for all data rows. | extractContourAndSkel( ) |
| Posture 1 | Posture Code | Posture | Sktp, BwImage | A number representing the posture of the nematode. | extractContourAndSkel( ) |
| SkewerAngle 1 | Skewer Angle |  |  | The angle from head to tail of the | extractContourAndSkel( ) |
| IsLoop | Is Loop | Posture | Binary image | A binary variable indicating whether the nematode binary image indicatesthat it overlaps or abuts itself. | extractContourAndSkel( ) |
| Length | Length | Size and Shape |  | The length of the nematode in pixels. | extractContourAndSkel( ) |
| HeadRow 1 | Head Row | Posture | sktp | The row pixel subscript of the head location in the image (global reference). | extractContourAndSkel( ) |
| HeadCol 1 | Head Column | Posture | sktp | The column pixel subscript of the head location in the image (global reference). | extractContourAndSkel( ) |
| TailRow 1 | Tail Row | Posture | sktp | The row pixel subscript of the tail location in the image (global reference). | extractContourAndSkel( ) |
| TailCol 1 | Tail Column | Posture | sktp | The column pixel subscript of the tail location in the image (global reference). | extractContourAndSkel( ) |
| IntH 1 | Head ntensity | Head And Tail ID |  | The mean intensity of the area in the vicinity of the head | extractContourAndSkel( ) |
| IntT 1 | Tail Intensity | Head And Tail ID |  | The mean intensity of the area in the vicinity of the tail. | extractContourAndSkel( ) |
| Contour 1,2 | Segmented Region Contour | Intermediate Representation |  | The pixel subscripts as an nx2 array of n ordered pairs comprising the contour of the nematode (global reference). | extractContourAndSkel( ) |
| Sktp 1,2 | Skeleton Points (longitudinal axis) | Intermediate Representation |  | The pixel subscripts as an nx2 array of n ordered pairs from head to tail comprising the centerline of the nematode (global reference). | extractContourAndSkel( ) |
| ElapsedTime | Elapsed Time | Trajectory | Tracker Log | The time since the start of the video. | extractContourAndSkel( ) |
| CameraStartRow | Camera Start Row | Reference | Config file | An arbitrarily-defined row subscript indicating the starting pixel location of the camera in the global reference. It provides the user the option of specifying a row that is large to ensure that all global coordinates are positive. | extractContourAndSkel( )->  loadCameraInfo( ) |
| CameraStartCol | Camera Start Column | Reference | Config file | An arbitrarily-defined column subscript indicating the starting pixel location of the camera in the global reference. It provides the user the option of specifying a column that is large to ensure that all global coordinates are positive. | extractContourAndSkel( )->  loadCameraInfo( ) |
| CameraStepRows | Camera Step Rows | Reference | Tracker Log | The approximate number of rows the camera stepped between the prior frame and the current frame. Derived from the number of steps, the step interval in mm, and the spatial resolution (in pixels/mm). | extractContourAndSkel( )->  loadCameraInfo( ) |
| CameraStepCols | Camera Step Columns | Reference | Tracker Log | The approximate number of columns the camera stepped between the prior frame and the current frame. Derived from the number of steps, the step interval in mm, and the spatial resolution (in pixels/mm). | extractContourAndSkel( ) ->  loadCameraInfo() |
| CameraOffsetRows | Camera Offset Rows | Refererence | Tracker Log | The estimated number of rows the current camera position is offset from its origin. | extractContourAndSkel( ) ->  loadCameraInfo() |
| CameraOffsetCols | Camera Offset Columns | Reference | Tracker Log | The estimated number of columns the current camera position is offset from its origin. | extractContourAndSkel( ) ->  loadCameraInfo() |
| CropOffsetRows | Crop Offset Rows | Reference | Tracker Log | The row position of the upper left corner pixel of the crop box in the FOV. | extractContourAndSkel( ) ->  loadCameraInfo() |
| CropOffsetCols | Crop Offset Columns | Reference | Tracker Log | The column position of the upper left corner pixel of the crop box in the FOV. | extractContourAndSkel( ) ->  loadCameraInfo() |
| TotalOffsetRows | Total Offset Rows | Reference | Tracker Log | The calculated row offset between global and local reference frames. | extractContourAndSkel( ) ->  loadCameraInfo() |
| TotalOffsetCols | Total Offset Columns | Reference | Tracker Log | The calculated column offset between global and local reference frames. | extractContourAndSkel( ) ->  loadCameraInfo() |
| DeltaTime | Delta Time | Trajectory | Tracker Log | The difference between the elapsed time of the current frame and the elapsed time of the previous frame. | extractFeatures ->  loadTrajectoryInfo() |
| DeltaX | Delta X | Trajectory | Centroid | The change in trajectory in x direction in pixels. | extractFeatures ->  loadTrajectoryInfo() |
| DeltaY | Delta Y | Trajectory | Centroid | The change in trajectory in y direction in pixels.. | extractFeatures ->  loadTrajectoryInfo() |
| DeltaDist | Delta Distance | Trajectory | Centroid | The total distance moved by the centroid between fame i-1 and frame i. | extractFeatures ->  loadTrajectoryInfo() |
| VectorAngle | Vector Angle | Trajectory | Centroid | The angle of the vector connecting the centroid in the previous frame to the centroid in the current frame. | extractFeatures ->  loadTrajectoryInfo() |
| InsantVelocity | Instantaneous Velocity | Trajectory | Centroid | The Euclidean distance between the location of the centroid in the previsou frame and the location of the centroid in the current frame divided by the delta time. | extractFeatures ->  loadTrajectoryInfo() |
| InstantAccel | Instantaneous Acceleration | Trajectory | Centroid | The difference between the instantaneous velocity in the prior frame and the instantaneous velocity in the current frame divided by the delta time. | extractFeatures ->  loadTrajectoryInfo() |
| CumDist | Cumulative Distance | Trajectory | Centroid | Total Distance travelled along its path. (The sum of all delta distances.) | extractFeatures() ->  loadTrajectoryInfo() |
| Range | Range | Trajectory | Centroid | The straight line distance between the starting point and the current location | extractFeatures() ->  loadTrajectoryInfo() |
| DirectionCode | Direction code | Trajectory | skeleton | 1= forward, 2 = Reverse | ExtractFeatures() ->  getDirection() |
| LclCentroidRow 1 | Local Centroid Row | Posture | Binary image | The centroid row of the binary image in local reference frame. | ExtractFeatures() ->  loadBwShapeAndSize() |
| LclCentroidCol 1 | Local Centroid Column | Posture | Binary image | The centroid column of the binary image in local reference frame. | ExtractFeatures() ->  loadBwShapeAndSize() |
| GblCentroidRow 1 | Global Centroid Row | Trajectory | Binary Image | The centroid row of the binary image in global reference frame. | ExtractFeatures() ->  loadBwShapeAndSize() |
| GblCentroidCol 1 | Global Centroid Column | Trajectory | Binary Image | The centroid column of the binary image in global reference frame. | ExtractFeatures() ->  loadBwShapeAndSize() |
| Area |  | Size and Shape | Binary Image | Number of pixels representing the animal in the binary image. | ExtractFeatures() ->  loadBwShapeAndSize() |
| MajorAxisLength | Major Axis Length | Posture | Binary Image | The major axis of the ellipse with the same second moment as the segmented region. | ExtractFeatures() ->  loadBwShapeAndSize() |
| MinorAxisLength | Minor Axis Length | Posture | Binary Imag | The minor axis of the ellipse with the same second moment as the segmented region. | ExtractFeatures() ->  loadBwShapeAndSize() |
| Elongation | Elongation | Bending | Binary Image | The major axis divided by the mino raxis of the ellipse with the same second moment as the segmented region. | ExtractFeatures() ->  loadBwShapeAndSize() |
| ComptFactor | Compactness Factor | Bending | Binary Image | The ratio of the area of the region to the area of the minimum enclosing rectangle. | ExtractFeatures() ->  loadBwShapeAndSize() |
| Heywood | Heywood Circularity Factor | Posture | Binary Image | .Perimeter/( 2\* sqrt( pi \*.Area) | ExtractFeatures() ->  loadBwShapeAndSize() |
| Hydraulic | Hydraulic Radius | Posture | Binary Image | Area/Perimeter of the segmented region | ExtractFeatures() ->  loadBwShapeAndSize() |
| Waddel | Waddell disk Diameter | Posture | Binary Image | 2\* sqrt(area/pi) (Note: The diameter of the disk with the same area as the segmented region) | ExtractFeatures() ->  loadBwShapeAndSize() |
| RectBigSdie | Rectangle Big Side | Posture | Binary Image | The length of the longest side of the rectangle that has the same area and perimeter as the region | ExtractFeatures() ->  loadBwShapeAndSize() |
| RectRatio | Rectangle Ratio | Posture | Binary Image | The ratio of the longest side to the shortest side of a rectangle that has the same area and perimeter as the region | ExtractFeatures() ->  loadBwShapeAndSize() |
| Perimeter | Perimeter | Size | Binary Image | The perimeter of the segmented region. | ExtractFeatures() ->  loadBwShapeAndSize() |
| Ixx | Moment of Inertia in the x direction | ? | Binary Image | Moment of inertia of the BW image in reference to the x axis | ExtractFeatures() ->  loadBwShapeAndSize() |
| Iyy | Moment of Inertia in the Y direction | ? | Binary Image | Moment of inertia of the BW image in reference to the y axis | ExtractFeatures() ->  loadBwShapeAndSize() |
| Ixy | Moment of inertia | ? | Binary Image | Product of the x and y moments of inertia | ExtractFeatures() ->  loadBwShapeAndSize() |
| MaxWidth | Maximum Width of the Segmented Region | Size and Shape | Binary Image | Width (in pixels) of the worm at its midpoint. Measured as the value of the distance transform at the midpoint of the skeleton. | ExtractFeatures() ->  loadBwShapeAndSize() |
| LAIIHead | Head Curvature | Shape | Binary Image | The Local Area Integral Invariant Signature at the tip of the head. | ExtractFeatures() ->  loadCurvAtEndpoints() |
| LAIITail | Tail Curvature | Shape | Binary Image | The Local Area Integral Invariant Signature at the tip of the tail. | ExtractFeatures() ->  loadCurvAtEndpoints() |
| Curv\_EndARow | End A row subscript determined by Curvature | Shape | Binary Image | The row subscript of the endpoint of the nematode as determined by an analysis of curvature of the contour. If only one end is visible, the other end is assigned a value of 1. | ExtractFeatures() ->  loadEndpointLocsFromCurv() |
| Curv\_EndACol | End A row subscript determined by Curvature | Shape | Binary Image | The column subscript of the endpoint of the nematode as determined by an analysis of curvature of the contour. If only one end is visible, the other end is assigned a value of 1. | ExtractFeatures() ->  loadEndpointLocsFromCurv() |
| Curv\_EndBRow | End A row subscript determined by Curvature | Shape | Binary Image | The row subscript of the endpoint of the nematode as determined by an analysis of curvature of the contour. If only one end is visible, the other end is assigned a value of 1. | ExtractFeatures() ->  loadEndpointLocsFromCurv() |
| Curv\_EndBCol | End A row subscript determined by Curvature | Shape | Binary Image | The column subscript of the endpoint of the nematode as determined by an analysis of curvature of the contour. If only one end is visible, the other end is assigned a value of 1. | ExtractFeatures() -> |
| SkelNumPixels | Number of pixels in the Skeleton | Posture | Skeleton | Number of pixels in the skeleton calculated as the area of the skeleton image. | ExtractFeatures() ->  loadSkelShapeAndSize() |
| Length | Skeleton Length | Posture | Skeleton | The quasi-Euclidean length of the skeleton | ExtractFeatures() ->  loadSkelShapeAndSize() |
| LengthToPixels | Length to Pixels Ratio | Posture | Skeleton | Ratio of the length of the quais-Euclidean length of the skeleton and the number of pixels that comprise the skeleton. | ExtractFeatures() ->  loadSkelShapeAndSize() |
| Thickness | Midpoint Thickness | Size and Shape | BwImage  Skeleton | The thickness at the midpoint of the skeleton as determined from overlaying the distance transform on the skeleton. | ExtractFeatures() ->  loadSkelShapeAndSize() |
| Fatness |  | Size and Shape | BW Image | Area/Length |  |
| SktvAglAve | Mean Skeleton Vector Angle | Posture | Sampled Sktp | Mean of the absolute values of the angles of the vectors from one sampled skeleton point to the next | l ExtractFeatures() ->  loadBendingStats() |
| SktvAglMax | Maximum Skeleton Vector Angle | Posture | Sampled Sktp | Max of the absolute values of the angles of the vectors from one sampled skeleton point to the next | ExtractFeatures() ->  loadBendingStats() |
| SktvDistAveToLength | Normalized Mean Skeleton Point Separation | Posture | Sampled Sktp | Mean of the distance from one sampled skeleton point to the next divided by the length of the skeleton | ExtractFeatures() ->  loadBendingStats() |
| SktvDisMaxToLength | Normalized Max Skeleton Point Separation | Posture | Sampled Sktp | Maximum of the distance from one sampled skeleton point to the next divided by the length of the Skeleton | ExtractFeatures() ->  loadBendingStats() |
| SktvDisMinToLength | Normalized Minimum Skeleton Point Separation | Posture | Sampled Sktp | Minimum of the distance from one sampled skeleton point to the next divided by the length of the skeleton | ExtractFeatures() ->  loadBendingStats() |
| SkewerLength | Skewer Length | Posture | Skewer Sktp with Centroid Origin | The distance between the transformed maximum and minimum x-values of the skeleton. | ExtractFeatures() ->  loadSkewerStats() |
| SkewerAngle | Skewer Angle | Posture | sktp | The angle between the horizontal line and the skewer line connecting the head to the tail. | ExtractFeatures() ->  loadSkewerStats() |
| SktAmpRatio | Skeleton Amplitude Ratio | Posture | Skewer Sktp with Centroid Origin | The ratio of the maximum distance below the centroid and the maximum distance above the centroid. | ExtractFeatures() ->  loadSkewerStats() |
| SktCmptFactor | Skeleton Compactness Factor | Posture | Skewer Sktp with Centroid Origin | The length divided by the product of the track amplitude and the skewerlength. (The area length of the skeleton curve divided by the enclosed area of its bounding box.) | ExtractFeatures() ->  loadSkewerStats() |
| SktElgFactor | Skeleton Elongation Factor | Posture | Skewer Sktp with Centroid Origin | The ratio of the track amplitude to the skewer length. | ExtractFeatures() ->  loadSkewerStats() |
| SktIxx | X Moment of Inertia | Posture | Skewer Sktp with Centroid Origin | The sum of the x values squared divided by the number of values. | ExtractFeatures() ->  loadSkewerStats() |
| SktIyy | Y Moment of Inertia | Posture | Skewer Sktp with Centroid Origin | The sum of the y values squared dived by the number of values. | ExtractFeatures() ->  loadSkewerStats() |
| SktAglAve | Average Skeleton Angle | Posture | Skewer Sktp with Centroid Origin | Mean radial angle from each skeleton point to the centroid. | ExtractFeatures() ->  loadSkewerStats() |
| XSym | X Symmetry | Posture | Skewer Sktp with Midpoint origin | The sum of the x-values | ExtractFeatures() ->  loadSkewerStats() |
| YSym | Y Symmetry | Posture | Skewer Sktp with Midpoint origin | The sum of the y values | ExtractFeatures() ->  loadSkewerStats() |
| XYSym | XY Symmetry | Posture | Skewer Sktp with Midpoint origin | The sum of the product of each x and y value | ExtractFeatures() ->  loadSkewerStats() |
| TrackAmplitude | Track Amplitude | Posture | Skewer Sktp with Centroid Origin | The distance between the transformed maximum and minimum y-values of the skeleton. | ExtractFeatures() ->  loadSkewerStats() |
| TrackPeriod | Track Period | Posture | Skewer Sktp with Midpoint origin | The period of the 1st frequency determined from the Fourier transform. | ExtractFeatures() ->  loadSkewerStats() |
| WidthProfileA | Width Profile A | Shape | Binary Image | The sum of all widths produced by superimposing the skeleton on the distance transform. | ExtractFeatures() ->  loadWidthProfiles() |
| WidthProfileB | Width Profile B | Shape | Binary Image | The sum of all widths produced by superimposing the skeleton on the distance transform. | ExtractFeatures() ->  loadWdithProfiles() |
| SktpMovement | Skeleton Point Movement | Local Movement | Sktp | Mean distance that skeleton points move from the prior frame to the current one. | ExtractFeatures() ->  loadSktpMovement() |

**Results**

**Execution Times for Video of 36037 frames (approx. 1 hour at 10 fps)**

|  |  |  |
| --- | --- | --- |
| **Process** | **Execution time (min)** | **Frames per Second** |
| extractContourAndSkel | 366 | 1.7 |
| createSegVideo | 78.4 | 7.7 |
| aggregateLoops | 3.4 | 176.0 |
| fixSktps | < 1 | 600 |
| extractFeatures | 45 | 13 |

Summary of Errors in Frame Annotations

|  |  |
| --- | --- |
| Number of ErrorFrames | Description |
| 1 | Failure in extractContourAndSkel.m ( no data produced) |
| 3 | Subscripted assignment mismatch in createSegVideo. Corrupted string representation of row, column subscripts. |
| 5 | Holes in the segmentation that were not the result of looping postures. These skeletons were good approximations of the actual skeleton, but the posture was misidentified as a looping posture. |
| 3 | Failure to segment a hole, resulting in a looping posture being identified as a non-loop. In these cases, the skeleton was badly-formed. |
| 11 | Improper identification of a nematode end due to dirt on the lens that resulted in over-segmentation to include the dirt, creating the appearance of false region of sharp curvature. These skeletons were badly-formed. |
| 11 | Head/Tail identification incorrect during delta posture of a loop. Also, errors were propagated to subsequent frames. |
| 33 | Total |

Failure to Segment Hole of Loop

|  |  |  |
| --- | --- | --- |
| Frame 7829 | Frame 7830 | Frame 7831 |
|  |  |  |

Frame Sequence of Prototypical Example of Failed Head/Tail Association

|  |  |  |  |
| --- | --- | --- | --- |
| Frame 1979 | Frame 1981 | Frame 1983 | Frame 1985 |
|  |  |  |  |
| Frame 1987 | Frame 1989 | Frame 1991 | Frame 1993 |
|  |  |  |  |

**Histograms for Features Designed to Distinguish Between the head and tail (includes only frames of non-looping nematodes)**

|  |  |
| --- | --- |
| **Head** | **Tail** |
|  |  |
|  |  |
|  |  |
|  |  |

**Discussion**

* What are the main characteristics of the analysis workflow that the system must be integrated into?
  + What manual validation is anticipated?
  + Where must the data reside?
* What analysis workload must the system support?
* What is the appropriate frame rate? Pixels on the nematode longitudinal axis move about 5 pixel locations between frames at 10 frames per second. What is the purpose of increasing the frame rate? What is the benefits if it increases processing time?
* How accurate must the longitudinal axis estimates be for self-occluding nematodes? The current estimates capture the general posture, but the pixels are assumed to be on the are on the

**Future Work**

* New data/test trajectory features
* Compare results to other available software packages