Proceedings of The National Conference
On Undergraduate Research (NCUR) 2014
University of Kentucky, Lexington, KY
April 3-5, 2014

Image Processing For Undulatory Locomotion

Sean Waters
Mechanical Engineering
Lafayette College
111 Quad Drive
Easton, Pennsylvania 18048 USA

Faculty Advisor: Robert Root

Abstract

The swimming patterns of undulatory swimmers can be modeled by tracking changes in their medial axis. Previously, the process for extracting this axis from video consisted of time consuming manual analysis. This research proposes to optimize collection of medial axis data via a mathematical model of the image. By identifying the outline of the swimmer in a video frame, it is possible to identify the medial axis through a series of computations based on the concept of a medial axis as first described by Blum. Furthermore, this project will investigate the problems of pixilation and resolution effects and propose methods to mitigate these effects. The proposed algorithm is dependent upon two parameters, the rate at which outline points are chosen for inclusion in the computation, and the size of the contour ratio associated with each candidate medial axis point. Through selective alterations of these parameters the algorithm can be applied flexibly in a wide array of video.

Keywords: Image Processing, Undulatory Swimmers, Blum's Medial Axis

1. Introduction

The medial axis is a valuable tool in mathematical shape analysis. It has gained popularity in the field of shape modeling primarily for its ability to efficiently represent elements in coding.[1] The capacity of Blum's medial axis to model shapes in this compact form streamlines computations and increases the computing speed of modeling programs. However, this was not the main factor in our decision to use the medial axis for this application. The medial axis proved to be a particularly powerful tool in our work as it functions as a good approximation for the vertebral column in the swimmers that we are testing.[2] Thus, tracking changes in the medial axis of undulatory swimmers serves as a good model of their swimming patterns.[3] However, the common method for computing the medial axis is timely and labor intensive. With the conventional method, it can take upwards of an hour to analyze single video frames and weeks to capture the data to analyze a simple swimming maneuver such as a turn.

This paper proposes that automating the image processing and data acquisition facets will significantly expedite the analysis and facilitate researchers abilities to develop meaningful propulsive models. Furthermore, this paper investigates the difficulties introduced by image pixilation and resolution to develop and recommend methods that mitigate these effects. The ultimate goal of this research was to evaluate various image-processing methods to identify those that best address the difficulties encountered in the automation of medial axis acquisition when applied to undulatory swimmers. The methods developed fit the criteria of simultaneously reducing the program's runtime while being sufficiently robust to apply to a variety of different swimmers.

A thorough explanation of the medial axis and the core concepts that are used in its creation are provided in the background section bellow. This report proceeds to describe the operation of the suggested automation function. This section will cover preprocessing, image processing, and the extraction of data from the medial axis. The methods of data acquisition outlined in this section were found to be the most efficient and robust of the various

image processing methods evaluated. The effects of these steps can be observed in the figures that have been extracted from various stages of the processing of a still frame of a swimming shark. Images of shark swimming maneuvers are displayed but the methodology is readily applicable to model any undulatory swimming species.

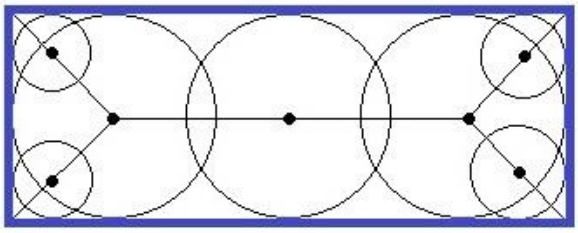
2. Background

2.1. Medial Axis

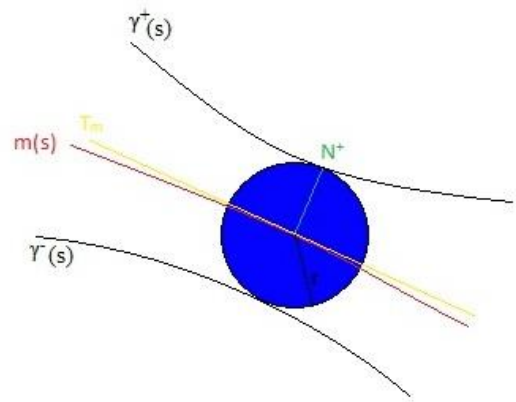
Definition 1: The medial axis of an object is the set of all points having two or more equidistant points on the objects boundary as its nearest boundary points.[4]

The exact medial axis of an object can be calculated using the following formula providing that the object's boundary is a differentiable curve.[5]

$$m(s) = \gamma^{+}(s) + r(s)N^{+}(s) = \gamma^{-}(s) + r(s)N^{-}(s) \quad (1)$$



(a) Illustration of medial axis by maximum circles



(b) Medial axis and bounding curve with proper notation

Figure 1: Depictions of the medial axis

Figure 1b shows the medial point m(s) and the two corresponding points on the boundary curve $\gamma^{+}(s)$ and $\gamma^{-}(s)$. The medial and boundary points are associated by adding the vector created by multiplying the unit vector that is normal to the boundary point of interest by radius of the circumscribed circle that passes through that point. Since $\gamma^{+}(s)$ and $\gamma^{-}(s)$ are both radius, r(s), away from the medial point m(s), the medial axis has property of being equidistant from its nearest neighbors at two or more points.[6] That is to say, the medial axis can be constructed by the method of maximum circles in which, the medial axis is comprised of center points of a set of all the inscribed circles that are tangential to the boundary curve at more than one point, as shown in Figure 1a.

Definition 2. As previously described by Strawbridge, (2012), the medial axis of a shape is composed of the collection of centers of all the maximum circles that are contained within the shape and are tangent to the boundary of the shape in at least two places.[7]

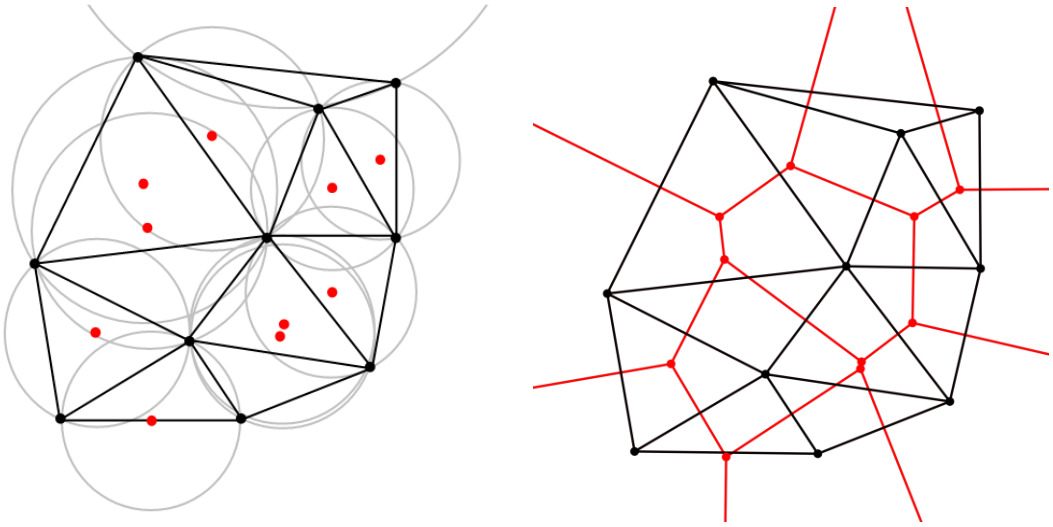
When the boundary curve of an object is not continuous, its medial axis can still be approximated using a variety of techniques. Whether an approximation or an exact computation, the medial axis gives more

information about a shape than the border does alone. Thus, it is helpful to use when analyzing properties of a shape. The conventional method of computing the medial axis is based on this definition of maximum circles. As such, researchers must perform many tedious measurements and calculations to generate a single medial curve. Furthermore, this method does not translate well to programmed automation. Utilization of the Voronoi approximation of the medial axis will prove to be a valuable tool for the transition to medial axis automation.

2.2. Voronoi

Voronoi Diagrams are composed of points called generators, in this case the points that create the border of the object, and corresponding regions called Voronoi cells that consist of all points that are closer to a particular generator than to any other. These cells form convex polygons that are bound by segment lines. All segment points of a Voronoi Diagrams are equidistant from the two nearest generators and intersect each other at the diagrams Voronoi vertices.[8] The Voronoi Diagram is notable because it offers another way to approximate the medial axis. This is useful, because a digital image is comprised of pixels that create a non-continuous border. The Voronoi approximation is clearly stated in this Theorem:

Theorem 1. The Voronoi vertices created in the Voronoi diagram approximate the medial axis of a curve in 2D. Furthermore, if the sample density approaches infinity, the Voronoi vertices in this case converge to the medial axis.[9]



(a) The Delaunay triangulation with all the circumcircles and their centers (in red)

(b) Connecting the centers of the circumcircles produces the Voronoi diagram (in red).

Figure 2: Voronoi diagram and Delaunay triangulation dual graphs

Observing the Voronoi diagram's dual graph, the Delaunay triangulation, illustrates the way the Voronoi approximation works. Delaunay triangulations consist of a set of triangles that are generated by using the original points on the same plane as their vertices. These triangles are constrained by the sole Delaunay condition, which requires that the circumcircles of all triangles have empty interiors (Figure 2a). To maintain this condition, the

boundary points that generate the third vertex of the triangles must become infinitely close to the limiting vertex as the sample size of points on the boundary curve approaches infinity. This phenomenon allows the Delaunay triangulation to approximate the medial axis by using the maximum circles definition of the medial axis. This finding can be extended to the Voronoi diagram by the dual nature of the two graphs. As defined by graph theory, a dual of a planar graph has a vertex that corresponds to each face in the original graph.[10] Accordingly, the centers of the Delaunay triangulations' circumcircles are equivalent to the Voronoi vertices in this dual set (Figure 2b). By extension of this definition, the Voronoi vertices must converge on the medial axis as sample boundary density approaches infinity. Furthermore, the ease at which the Delaunay triangulation can be programmed and transformed into a Voronoi diagram makes this approximation a valuable tool for automation.

2.3. Pixilation Effects And Boundary Smoothing

A fundamental problem encountered in medial axis automatization is the pixilation effect of digital photography. Natural pixilation of the original image creates a rough boundary. As a result, this pixilated border creates numerous false medial points due to boundary fluctuations. It is beneficial to initially smooth the border using median filtering technique. Median filtering takes the original boundary and produces a revised boundary with new pixels with xy-coordinates given by the median of their neighbors' values. Thus, it takes the discrete information returned by the erosion and differencing process, defined in the image processing section below, and makes it more smoothly varying. This method was shown to be reliable because median filtering is better suited to reducing noise while preserving edges than are other prevalent boundary smoothing methods.

Despite the boundary smoothing done by the median filtering technique, the border will still produce extraneous medial points within the boundary of the subject. The best way to remove these extraneous points is to use the concept of the contour ratio to discard the medial points that are less central to the main structure of the medial axis.[7] To understand this technique, one must recall the definition of the medial axis by maximum circles that illustrates each point on the medial axis corresponds to two or more boundary points at the these points of tangency. Accordingly, each medial point has two boundary points associated with it; there are its maximum circle's points of tangency. These boundary points are used to segment the objects border into two or more sections. The second largest of these segments is selected and its arc length is assigned the value l(P). This segment is then compared to the total length of the boundary curve, L (Figure 3). The ratio, R(P) = l(P) / L, is then compared to a predetermined threshold value. Values of R(P) that exceed this threshold prove to be more essential to the main structure of the medial axis, therefore, their boundary and medial points are preserved while all others are discarded. The result of this process is the final pruned boundary and medial data. Higher threshold values result in more simplified sets of boundary and medial points. Accordingly, the threshold is chosen depending on how much of the information about the boundary needs to be preserved for the purpose of the particular application.

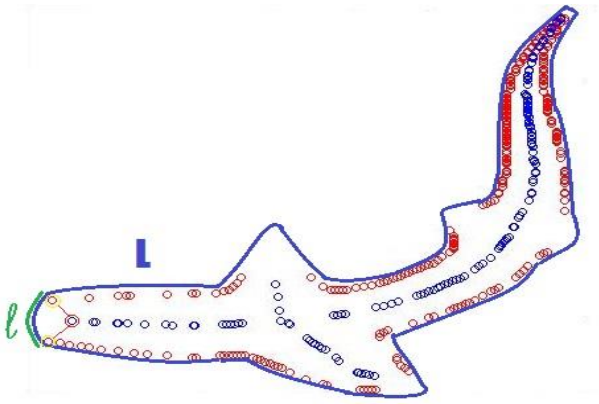


Figure 3: Illustration of the contour ratio thresholding technique

3. Methods

3.1. Preprocessing

Before video frames can be analyzed, they must be converted to grayscale, and both the grayscale and contour ratio thresholds must be specified. These values depend on factors that are specific to that particular set of images. For example, RGB to grayscale weightings must be selected based on the prominent colors of the subject and the background. It is important that the colors that are unique to the subject are weighted heavier than the rest so the result is a subject that is darker than most of the background. To make the selection of all of these values more precise we have included a simple command line interface that guides the user through the selection of these parameters. In this interface the user can simply adjust the indicated values and get immediate graphical feedback. By observing the output figures of this code, selection of these initial variables is fairly straightforward. This simple preprocessing user interface significantly reduces human error and the amount of effort the user must expend to obtain the parameters required for image processing.

3.2. Image Processing

The code for final image processing begins with a procedure called cluster analysis. While there are many clustering techniques, it implements grayscale threshold clustering to convert our complex image into binary. This algorithm works by first converting the image from color to grayscale using the RGB conversion ratios determined in the preprocessing steps. Next, pixels in the grayscale images are compared against a black/white threshold that is unique to each cache of images. Pixels that equal or exceed this threshold are assigned the value one, or black, while the rest of the pixels are assigned zero, or white. After this conversion, our program searches for the largest cluster of pixels with the value of 1. This cluster is the swimmer. The final step in formatting the image is to identify the object's border. This is accomplished by taking the largest cluster and shrinking it by one pixel, by a process called erosion. Binary erosion is a fundamental morphological process that has many applications in image processing. The purpose of binary erosion is to probe the image with a specified shape, called a structuring element. For our purposes, a diamond was selected as the structuring element. These diamonds are centered at every pixel represented by the binary digit zero in the image. Running this process with a diamond structuring element returns an image that is similar to the original image except the outermost pixel in the a neighborhood of pixel is assigned the value of 0. This effectively shrinks the image by one pixel in all directions. The new, shrunken image is superimposed over and subtracted from the original image leaving a border that is one pixel in width. Figure 4a shows the result of this process.

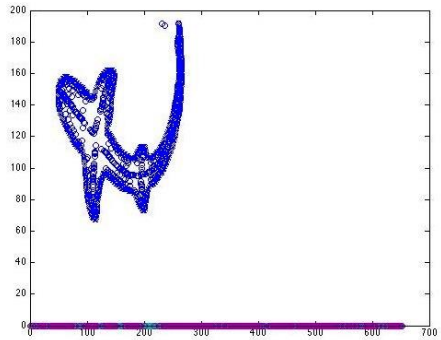
The method of threshold clustering followed by erosion provides a much more precise boundary than the standard boundary tracing techniques. In addition, this methodology makes the program far more robust since extraneous objects and shadows in the background of the image are not mistaken for objects and extraneous medial points are not introduced. Furthermore, clustering and erosion are fairly simple operations that take very little time for the program to process.

These points must be ordered and indexed so that the program can accurately perform multiple steps in the data extraction process. In theory, establishment this index is a simple process of selecting an arbitrary boundary point and searching for its nearest neighboring. These two points initiate the creation of an indexed 2XN matrix, where N is the number of boundary points. The first column of this matrix is simply an index containing values from 1 to N while the latter contains the complex coordinate of the corresponding boundary point. This process of searching for nearest boundary to the latest point added to the index matrix continues until all of the points have be added, ensuring an ordered set of boundary points. However, this process is more complicated in its application. In the case where the boundary narrows and the image resolution is low, it is not uncommon for the image to be less than three pixels wide in narrow regions.

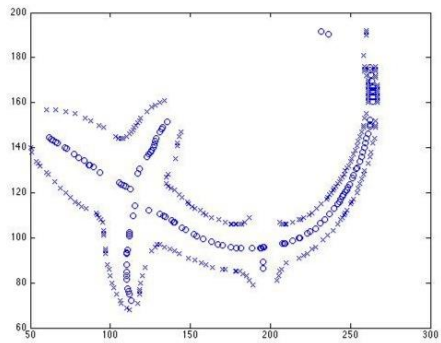
This creates locations on the boundary where the correct point must be selected from two equidistant boundary points. Ensuring that the correct point is added to the indexing matrix was a challenge that was solved by using a second order fit on the previously indexed points to generate an estimate for the next point to be indexed. In theory, the correct boundary point should follow the border's local curvature. Accordingly, the point that most followed this local curvature was selected as the next point to be indexed in such cases. This method was automatically applied at each instance of this phenomenon and the correct boundary point was selected in all of the cases.

3.3. Data Extraction

Once the border is defined and the median filtering has been applied to this set of points the medial axis can be defined. A Voronoi diagram is constructed using new set of smoothed boundary points as generators. The Voronoi vertices are then selected, as they represent the medial points in accordance to the aforementioned approximation method. This technique produces extraneous Voronoi points that lie outside the boundary of the shape. These points are dismissed, leaving only interior points. By studying the plot of the medial axis it becomes evident that there is an excess of medial points (Figure 4a). Many of the points must be rejected since we are only interested in the medial points that converge on the swimmers vertebral column. These desirable medial points are more central to the main structure of the medial axis. Contour ratio is applied with the pre-determined custom values to ensure only the essential points are retained. It is through this method that we prune the medial axis to a manageable state (Figure 4b).



(a) Image before medial filtering and contour ratio



(b) Image after medial filtering and contour ratio

Figure 4: Boundary of shark with medial data before and after smoothing techniques

Once the medial axis has been pruned, we can begin to extract the vertebral data of the swimmer. To run the propulsion analysis, we must eliminate the medial branches that correspond to the appendages (fins) of the swimmer and the remaining medial axis must be partioned into thirty sections of equal length. A method of indexing was utilized to accomplish this. This process begins by searching for the maximum curvature of the boundary points, because this occurs at the tip of the tail in all of the swimmers that were analyzed. Once the tail point has been identified, the code indexes the medial points as it did the boundary points. The code searches for the nearest medial point, adds it to a new indexed set of medial points, and stores distance between the two points. The indexed point is then discarded from the original set of medial points. Next, the program searches the original medial set for the closest point to the newest indexed point. To ensure that the fin branches are not included in this indexed set, all points must satisfy the condition of relative co-linearity in order to be included. Points that do not fulfill this condition are simply discarded from the original medial set. This process is repeated until the original set of medial points is empty. The result is an ordered set of vertebral points and an array containing the distances between adjacent points. With this information vertebral points can automatically be selected at the appropriate distance intervals (figure 5).

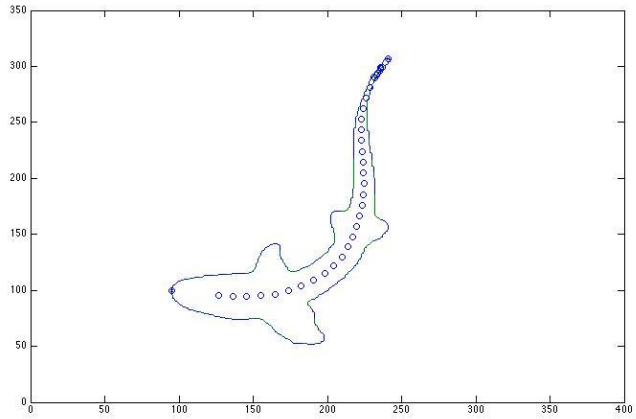


Figure 5: Final image with complete boundary and output vertebral points

4. Conclusion

Our improved computational approach builds on the conventional usage of Blum's medial axis for shape modeling. Image pixilation and resolution are the fundamental challenges to this computational approach. The pixilation effects were addressed through implementation of various boundary smoothing techniques; namely, medial filtering and contour ratio thresholding. Furthermore, utilization of a second order curve fit ensured that the boundary was appropriately indexed in areas of low resolution. With these major complexities accounted for, the program can be easily applied to suit a wide array of undulatory swimmers by varying a few simple parameters. The correct values of these parameters can be found quickly and reliably via use of our optimized preprocessing procedure. Processing efficiency was improved with our computational approach by reducing the time required to collect data from a single still frame from one hour to a mere fifteen seconds. As a result, data from 240 images can be collected in the time that the conventional method required to analyze one image. Given that hundreds of still frames must be analyzed to model simple swimming maneuvers, this advancement in computational analysis makes the study of complex swimming maneuvers possible. This improved computational approach will facilitate data analyses and aid in the development of undulatory propulsion models.

5. References

1. K. Leonard, "Efficient shape modeling: ε-entropy, adaptive coding, and Blum's medial axis versus the boundary curve," *Int. J. Comp. Vis.* 74 (2007): 183–199.

2. Robert G. Root et al., "Flapping Flexible Fish: Periodic and Secular Body Reconfigurations in Swimming Lamprey, Petromyzon marinus," *Experiments in Fluids* 43, no. 5 (2007): 779–797.

3. Jr. John H. Long et al., "Go Reconfigure: How Fish Change Shape as They Swim and Evolve," *Integrative and Comparative Biology 50*, no. 6 (2010): 1120–39.

4. Harry Blum, "Biological Shape and Visual Science," J. Theor. Biol. 38 (1973): 205–287.

5. Christine Caples and Carlos Hernandez, *Geometric 2D Shape Modeling*, technical report (California State University, Channel Islands, 2010).

6. Leonard, "Efficient shape modeling: ε-entropy, adaptive coding, and Blum's medial axis versus the boundary curve."

7. Rebecca Strawbridge, *Strawberry Report*, technical report (California State Uni- versity, Channel Islands, c2012), http://www.csuci.edu/writing-ci/guide/ samples/documents/strawberry-report.pdf.

8. Franz Aurenhammer, "Voronoi Diagrams - A Survey of a Fundamental Geometric Data Structure," *ACM Computing Surveys* 23 (September 1991): 345–405.

9. Tamal K. Dey and Wulue Zhao, "Approximating the Medial Axis from the Voronoi Diagram with a Convergence Guarantee," *Algorithmica* 38 (2004): 179–200.

10. Adrian Bondy and U.S.R. Murty, *Graph Theory*, vol. 244, Graduate Texts in Mathematics (Springer, 2008).