# Polygon Scaling and Outlining Algorithm

Simon Blasby

December 11, 2023

## 1 Overview

The objective of this algorithm is to generate a modified simple polygon from a given set of points defining a simple polygon. It ensures that the resulting polygon has an area some percentage larger than the original while preserving the angles between its edges.

A notable feature of this algorithm is its efficiency, operating in linear time with respect to the number of points in the input polygon. This efficiency makes it well-suited for handling polygons of varying complexities without incurring a significant increase in computational cost. Particularly, it is adept at processing 3D shapes composed of 2D polygon slices.

## 2 Assumptions

Although few this algorthim still makes a few assumptions about the polygon and the order of the coordinates that are inputted, they are as follows:

- The polygon is simple
- The coordinates must be stored in order (clockwise or counter clockwise)

Although the polygon must be simple, if there is a known region inside of the polygon (making it hollow) then this region can be extracted and the algorithm can be preformed on both the outer and inner region separately, before recombining them to form the resulting polygon. All calculations in this document will be preformed assuming the coordinates are stored in clockwise order, however the math is easily adaptable (switching a plus to minus) for coordinates stored in the counter clockwise direction.

## 3 Algorithm Overview

The primary steps of the algorithm are outlined as follows:

1. Create vectors between consecutive points in the input polygon.
2. Translate each vector in the direction orthogonal to it by a specified length $l$.
3. Determine the points of intersection between the adjacent translated vectors.

## 4 Mathematics for Determining the Translation Length $l$

In determining the translation length $l$, we employ a calculated approach to avoid the inefficiencies associated with trial and error. The use of guessing and checking could significantly impact the runtime and introduce inconsistencies.

To achieve a consistent runtime, we mathematically derive the optimal value for $l$, enhancing the algorithm's efficiency and ensuring a reliable and predictable outcome.

The process for calculating $l$ involves creating an equation that describes the area of the resulting polygon. Importantly, this equation must be formulated solely in terms of parameters from the original polygon. This restriction is necessary for solving and obtaining the required value for $l$. By imposing this restriction, we formulate an equation that is quadratic in $l$, describing the area of the resulting polygon.

This equation is derived from a series of geometric proofs and realizations. The main idea is as follows: The resulting area can be calculated as the sum of the original area and the area of the additional space created during the $l$ expansion.

$$A_{\text{total}} = A_{\text{original}} + A_{\text{expansion}} \tag{1}$$

## 4.1 Calculating the Original Area

Determining the area of a simple polygon is a well-known and documented procedure. Various methods exist, and for ease of coding the Shoelace Theorem has been chosen. This theorem is based on summing the areas of triangles composing the polygon. For a simple polygon with $N$ vertices, the area is calculated by:

$$A = \frac{1}{2}\sum_{i=1}^{N-1} \det \begin{bmatrix} x_i & x_{i+1} \\ y_i & y_{i+1} \end{bmatrix} + \frac{1}{2} \det \begin{bmatrix} x_n & x_1 \\ y_n & y_1 \end{bmatrix} \tag{2}$$

## 4.2 Area of the Expanded Region

The area of the expanded region can be calculated after breaking the expanded region up into rectangles and rhombuses, and summing the area of these regions. To help illustrate consider polygon below that has gone through the algorithm

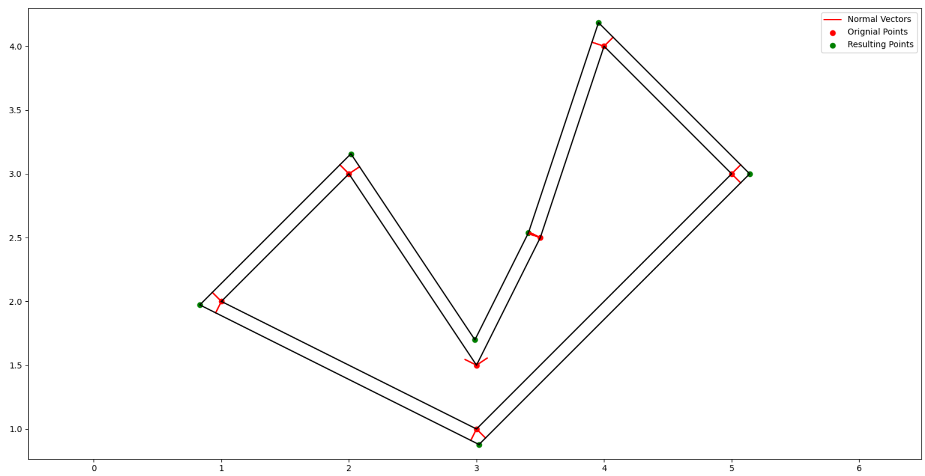


Figure 1: Outline of a concave polygon

Looking at the additional area that was added around the original area, one can convince themselves that it is constructed of a series of rectangles and rhombuses. The only exception to this is the concave region, which will be examined in detail in section 4.2.2.

### 4.2.1 Convex Region in Terms of $l$

Consider a convex region from the polygon shown in Figure 1, enlarged below:

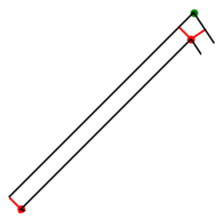

Figure 2: Convex region from the polygon in Figure 1

In this image, the area comprises a rectangle (enclosed by the two parallel normal lines and the black lines) and a rhombus (enclosed by the two non-parallel normal lines and the black lines). All red lines have the same length $l$, making the area of the rectangle $l$ times the length of the vector from red dot to red dot.

The area of the rhombus can be determined by splitting it in half, creating two triangles, as shown in Figure 3.

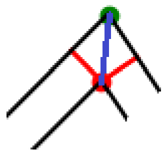

Figure 3: Rhombus from the region in Figure 2

A key realization is that the split rhombus now contains two right-angled triangles that share a hypotenuse. Since both triangles have a side length of $l$, it can be stated that these two triangles are identical. Thus, the blue line bisects the angle between the two normal vectors. Given that the angle between the two black lines meeting at the red dot is the same as the two black lines meeting at the green dot, the angle between the blue line and the red normal line is:

$$\phi = \frac{180 - \theta}{2} \tag{3}$$

Knowing this angle, the height of the triangle can be solved for using the tangent ratio:

$$h = l \tan \phi \tag{4}$$

$$h = l \tan \left( \frac{180 - \theta}{2} \right) \tag{5}$$

Where $h$ is the height of the triangle, $\theta$ is the interior angle of the polygon, and $l$ is the length of the normal vectors. With the value of $h$, the area of both triangles can be calculated:

$$A_{\text{triangle}} = \frac{1}{2} l^2 \tan \left( \frac{180 - \theta}{2} \right) \tag{6}$$

Making the area of the rhombus:

$$A_{\text{rhombus}} = l^2 \tan \left( \frac{180 - \theta}{2} \right) \tag{7}$$

The area of the concave region shown in Figure 2 can then be calculated using the following equation:

$$A_{\text{concave}} = l \cdot d + l^2 \tan \left( \frac{180 - \theta}{2} \right) \tag{8}$$

Where $d$ is the length of the vector from red dot to red dot.

### 4.2.2 Concave Region in Terms of $l$

Consider a concave region from the polygon shown in Figure 1, enlarged below:

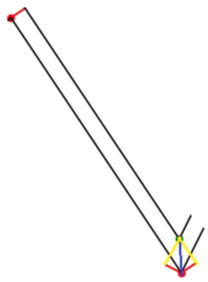

Figure 4: Concave region from the polygon in Figure 1. Where the red lines are the normal vectors, the black lines are the sides of the polygons, the blue line is the hypotenuse formed by splitting the rhombus into two right triangles, and the yellow line is the height of this right triangle.

A very similar approach can be taken to calculate the area of the expanded region. However, this time, if we evaluate the area of the rectangle, it can be seen that the area stretches past the edge of the polygon. Furthermore, after finishing the area calculation for this region and calculating the area of the next region to the right (not shown in Figure 4), the area enclosed by the yellow and black lines will be added again. Thus, the area of the rhombus (area enclosed by yellow and red lines) must be subtracted.

To get the area of the rhombus, one can again use the argument that the two constructed triangles will be identical, and therefore, the blue line bisects the angle between the two adjacent normal vectors (red lines).

This time, using the fact that the exterior angle, $\theta$, at the vertex at the red dot is the same as that at the green dot, and the yellow lines are simply extensions of the black lines, thus the angle between the two yellow lines must also be $\theta$. Using this fact, the angle between the red and blue lines will be:

$$\phi = \frac{180 - \theta}{2} \tag{9}$$

Making the length of the yellow line, $h$:

$$h = l \tan \phi \tag{10}$$

Making the substitution for $\phi$:

$$h = l \tan \left( \frac{180 - \theta}{2} \right) \tag{11}$$

The area of the rhombus will then be the sum of the areas of the triangles that compose it:

$$A_{\text{rhombus}} = l^2 \tan \left( \frac{180 - \theta}{2} \right) \tag{12}$$

## 4.3 Equation for $l$

Combining the cases for concave and convex, an equation for the expanded region can be formed. If a convex section is encountered, then the area that must be added is:

$$A_{\text{expansion}} = l||P_{i+1} - P_i|| + l^2 \tan \left( \frac{180 - \theta_{\angle P_i P_{i+1} P_{i+2}}}{2} \right) \tag{13}$$

If instead a concave section is encountered, then the area that must be added is:

$$A_{\text{expansion}} = l||P_{i+1} - P_i|| - l^2 \tan \left( \frac{180 - \theta_{\angle_{\text{ext}} P_i P_{i+1} P_{i+2}}}{2} \right) \tag{14}$$

Where this time $\theta$ is the exterior angle of the polygon between the line segments $\overline{P_i P_{i+1}}$ and $\overline{P_{i+1} P_{i+2}}$. This formula can be easily reformatted to be in terms of the interior angle; however, it is more natural for it to be the exterior (explained in Section 5).

Combining these two results, an equation can be formed that describes the area of the resulting polygon solely in terms of known quantities, where $N$ is the number of vertices.

$$A_{\text{new}} = l \sum_{i=1}^{N} ||P_{i+1} - P_i|| + l^2 \sum_{i=1}^{N} \pm \tan\left(\frac{180 - \theta_{\angle P_i P_{i+1} P_{i+2}}}{2}\right) + A_{\text{original}} \tag{15}$$

Note: In the case that the vertex is concave, use the minus case and the exterior angle. Otherwise, use the plus case and the interior angle.

Looking at Equation 15, it can be seen that it is quadratic in $l$. It is also worth noting that $A_{\text{new}}$ will be the same as $A_{\text{original}} * \alpha$ where $\alpha$ denotes the factor by which the area is to expand. This then leads to the final quadratic equation which can be solved for $l$:

$$l^2 \sum_{i=1}^{N} \pm \tan\left(\frac{180 - \theta_{\angle P_i P_{i+1} P_{i+2}}}{2}\right) + l \sum_{i=1}^{N} ||P_{i+1} - P_i|| + A_{\text{original}} \times (1 - \alpha) = 0 \tag{16}$$

Note: if $i + x > N$, instead take the value of the $i + x - N$ vertex

# 5 Creating the Algorithm

The mathematical derivation for calculating $l$ naturally translates into an algorithmic implementation. While Section 3 provides a high-level overview of the algorithm, the following presents a more comprehensive guide:

1. Calculate the scaling factor $l$ by iterating through the vertices using Equation 16.
2. Select a starting point.
3. Obtain the next two points in the clockwise direction.
4. Calculate the slopes of the lines connecting the first and second points and the second and third points.
5. Use the negative reciprocals of the slopes to derive equations for the perpendicular lines.
6. Obtain the unit normal vectors and multiply by $l$.
7. Translate vertices 1 and 2 by the normal vector created from the slope of the line connecting vertices 1 and 2 and get the equation for the line connecting these new points.
8. Repeat step seven for points 2 and 3.
9. Find the point of intersection of the two newly created lines and save it.
10. Repeat steps two through seven for every other point in the list, ensuring that advancement is in the clockwise direction.
11. Return the saved points.

Here are a few additional considerations to enhance the construction of the algorithm:

- Utilize the dot product to obtain the interior/exterior angle in the polygon. By using the dot product, there is no need to determine whether the angle is interior or exterior; the dot product will provide the correct angle (assuming your arccos function returns an angle between 0 and 180).
- The sign of the z-component of the cross product can be employed to discern whether the section is convex or concave, since concave section will result in a counter clockwise rotation.
- Similarly, the sign of the z-component of the cross product can ensure that the normal vector is facing outward before being scaled by $l$. An outward facing normal vector will always need to be rotated clockwise to align with the vector between the vertices.