

Polygon Generation and Transformation

Naci John C. Trance

Iloilo Science and Technology University, Philippines.

Email: nacijohn.trance@isatu.edu.ph

Abstract

A computer-implemented method of polygon generation and transformation wherein the generation of the polygon relates to generating an image model, exporting the image model to create an image file located outside the system, importing back the image file, and displaying it in a GUI. The transformation of the polygon relates to the combination of rotation, flipping, and scaling. The rotation of the polygon is done either in a clockwise or counter-clockwise direction. Similarly, flipping of polygon is about any vertex of the polygon, and scaling by adjusting to increase or decrease the size or area of the polygon. Within the exporting process is the classification of polygons in terms of shape.

Keywords: general polygon, generating polygon, polygon transformation, polygonal mapping

I. INTRODUCTION

The present invention relates to generating a polygon, displaying and exporting the generated image model of the polygon, and having the displayed image model transformed through rotation, flipping, and scaling. Indeed, an n -point polygon is a closed figure with n number of vertices (and n sides), with no three successive points collinear, together with the line segments joining consecutive pairs of the points (Ghosh, 2019; Gielis, 2003; Ishaque, Souvaine, & Tóth, 2012; Peterson, 2022).

A study by Aloupis (2005) and Aloupis et al. (2011) relating to “a planar polygon (or chain) with a list of edges $\{e_1, e_2, e_3, \dots, e_{n-1}, e_n\}$ ” examined the effects of several operations that permute this list, resulting in the formation of a new polygon. The reversal and transposition operations were used to explore the complexity of the given operations and in convexifying a given polygon or obtaining its mirror image. Nevertheless, instead of considering permutations of integers, the permutation of edges that form polygons or chains was considered. The generated polygon is limited such that if the orientations of the edges are retained, then not all permutations result in a

closed figure. Thus, the area of the polygon diverges.

On the other hand, a polygon generating apparatus and drawing system was introduced by Saito & Yokohama-Shi (2002) in the patent publication US2002/0167514 A1 dated November 14, 2002. The apparatus comprising of an initial connection data generator configured to generate connection data relating to connection information of vertices, sides and faces, based on vertex data of the inputted polygon; a new connection data generator configured to generate new connection data based on said connection data; a new vertex data generator configured to generate new vertex data based on said vertex data and said connection data; a data combining part configured to combine said new connection data relating to the same vertex; a connection data storing part configured to store the connection data generated by said initial connection data generator and said new connection data generator, and the connection data combined by said data combining part; and a vertex data storing part configured to store the inputted vertex data and the vertex data generated by said new vertex data generator, wherein said new connection data generator

repeatedly generates said new connection data based on the connection data stored in said connection data storing part; and said new vertex data storing part repeatedly generates said new vertex data based on the connection data stored in said connection data storing part and the vertex data stored in said vertex data storing part. This publication teaches an apparatus where connection information of the vertices, sides, and faces is stored and a basis for generating new connections. However, the present invention is related to generating polygons having their vertices defined by some reference elements comprising an origin, axis lines, and radial distances. The generated polygons are classified into different groups and transformed by rotation, flipping, and scaling to simulate the given data using the *xy*-plane.

In the patent application US20010025298 A1 entitled, "Object region data generating method, object region data generating apparatus, approximation polygon generating method, and approximation polygon generating apparatus," an object region is approximated with a polygon in each frame (Masukura, Hori, Kaneko, Mita & Yamamoto, 2000). The polygon's vertices in one frame are associated with the vertices of another polygon in an adjacent frame. This association is based on the distance between the vertices. A trajectory of the vertex is obtained on a time-series variation of the associated vertexes. The object region data is generated based on the trajectory of the vertex. Likewise, the generation of a polygon is still based on the distance between the vertices; thus, such vertices' permutation may not result in a closed figure. The same method of polygon generation was used in Azad, Ude, & Cheng (2011); Liao & Chan (2015); Kim (1996); Cieplinski & Dabis (2004); Kim (2007); Han & Kim (2000); Isao (2014); Funahashi (1995); and Vartziotis & Merger (2018).

II. THE OBJECTIVE OF THE STUDY

Given the limitations of the aforementioned prior arts, the present study aimed to create a method comprising the generation, transformation, and classification of polygons based on a predefined origin.

III. SIGNIFICANCE OF THE STUDY

The generation and transformation of polygons are helpful in anisotropy, mapping, surveying, and biomedical, especially when the area to be mapped or enclosed is not regular in shape. The present technology provides a method of generating, transforming, and classifying polygons, wherein generated polygons are closed even upon applying a combination of transformations.

IV. SUMMARY OF THE INVENTION

The invention relates to a computer-implemented method of polygon generation and transformation wherein the generation of the polygon relates to generating an image model, exporting the image model to create an image file located outside the system, importing back the image file, and displaying it in a GUI. The transformation of the polygon relates to the combination of rotation, flipping, and scaling. The rotation of the polygon is done either in a clockwise or counter-clockwise direction. Similarly, flipping of polygon is about any vertex of the polygon, and scaling by adjusting to increase or decrease the size or area of the polygon. Within the exporting process is the classification of polygons in terms of shape. Several rules were employed to reduce the computer processing, involving: determining the arrangements with similar results when flipping images at a given vertex, determining the number of cycle pairs, and determining the cycle pairs to be performed to complete the flipping process.

V. BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a plane view of a generated polygon and other elements based on the implemented method.

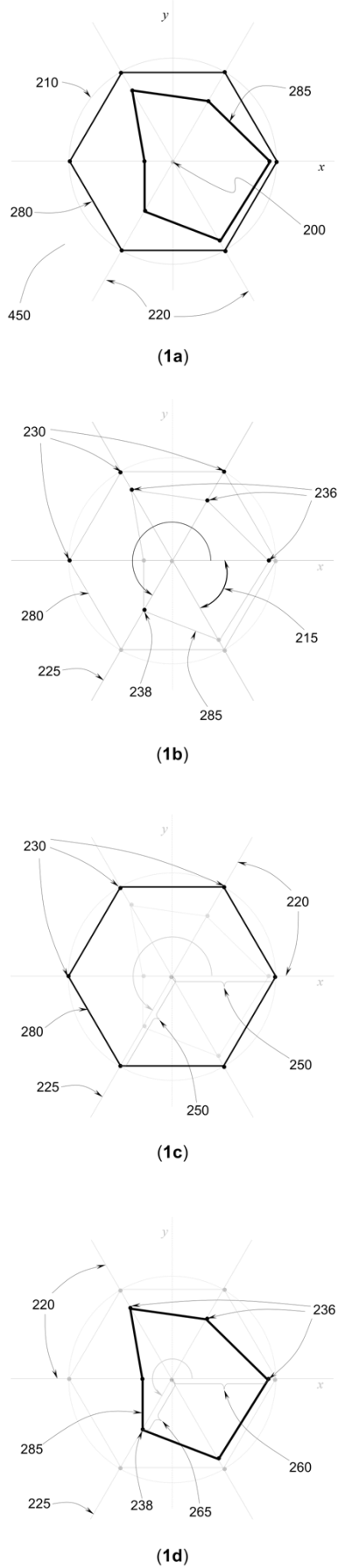


Figure. 1

Figure 2 illustrates a plane view of the transformation of the generated polygon by rotation, flipping and scaling based on the implemented method.

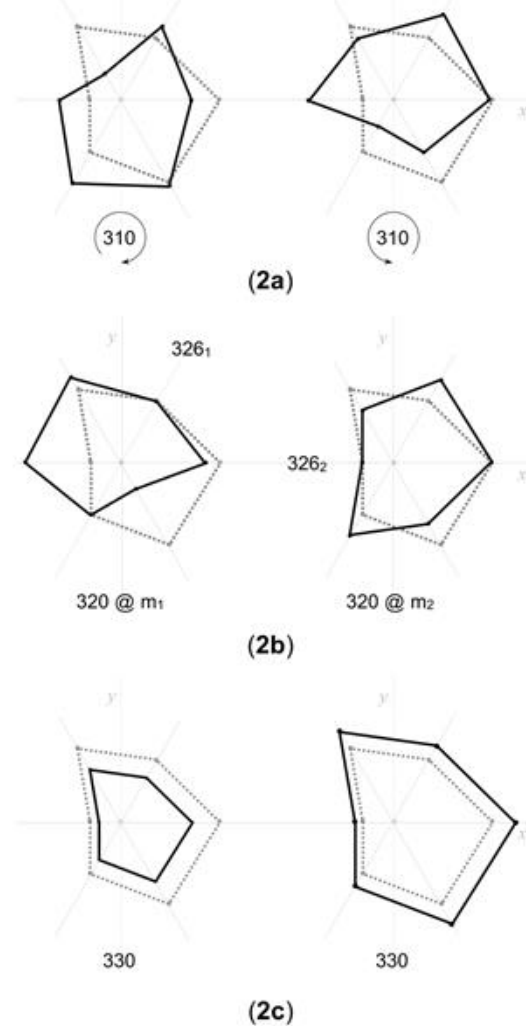


Figure. 2

VI. DETAILED DESCRIPTION OF THE INVENTION

Embodiments described herein can be understood more readily by referencing the following detailed descriptions and examples and their previous and following descriptions. However, elements, apparatus, and methods described herein are not limited to the specific embodiments presented in the detailed description and examples. It should be recognized that these embodiments are merely illustrative of the principles of the present invention. Numerous modifications and adaptations will be readily apparent to those of

skill in the art without departing from the spirit and scope of the invention.

A system installed in a data processing system such as a computer is run using a computer memory to generate polygons and manipulate the said polygons using a combination of transformations. Preferably, the system is a spreadsheet that can execute macro instructions to perform the generation and transformation of polygons.

In Figure 1, the generation of polygons involves four major processes: generating two sets of polygons A and B with image models Ao **280** and Bo **285**, respectively, from two sets of radial lengths RL; exporting said Ao **280** and Bo **285** by rendering both image models into one image thereby creating an image Io **450**; importing said Io **450** to be displayed in a graphical user interface (GUI); and classifying Io **450** according to shape and number n of vertices.

The generation process starts with selecting from a menu the number n of vertices that the polygons A and B and their image models must have. This number n of vertices can be inputted into an input box in the GUI using a computer keyboard. Ao **280** and Bo **285** are positioned (or drawn) relative to a point Po **200**, a unit circle Co **210**, a plurality of axis lines AXL **220**, and RL.

A point Po **200** is marked (0,0), indicating it as the origin of the xy axes and shall be the centre of the polygon. When the number of vertices of the polygon becomes large, the polygon resembles Co **210**. The imaginary-like Co **210** centred at Po **200** is drawn (or plotted) using a set of least dense points and serves as the boundary of Ao **280** and Bo **285**.

There are n AXL **220** to be defined for the chosen number n of vertices that is previously set and is drawn (or plotted) and displayed in the GUI using a least dense line. The 1st AXL of the AXLs **220** is concurrent with the positive x -axis. An angle θ **215** defines the angle between two consecutive AXL **220**. The measure of θ **215** is obtained using the formula $\theta = 360/n$, in degrees. Alternatively, the angular position θ_r **216** of the r^{th} AXL **225** can be computed using

the formula $\theta_r^{\text{th}} \text{ AXL} = (r - 1) * \theta$, where $1 \leq r \leq n$. Such position of the r^{th} AXL **225** is angled from the positive x -axis using the standard angle measurement. Any AXL **220** is drawn outwardly using a straight line starting from Po **200** and intersecting Co **210**. Only one AXL **220** can pass at most one of the n vertices. All AXL **220** are labelled from 1 (start) through n (end), going counter-clockwise, with a step increment of 1.

The radial length RL is the distance from Po **200** to a polygon vertex and is measured along with the AXL **220**. This RL is normalized to cater to all the supplied measures and can have any value from 0 to 1. Any vertex is always contained in that AXL **220**. Any RL is concurrent to at most one of the AXL **220**. Nevertheless, RL is not drawn (or plotted) but is measured from Po **200** to locate the polygon's vertices.

Some embodiments of polygons A and B are defined and described. Polygon A refers to the set of polygons generated as a result of the transformation by scaling T_s **330** of Ao **280**, described as Polygon A = $\{T_s(Ao)\}$. Likewise, polygon B refers to the set of polygons generated as a result of all combinations of transformation by rotation T_R **310**, flipping T_F **320**, and scaling T_s **330** of Bo **285**, which is mathematically described as Polygon B = $\{T_{R,F,s}(Bo)\}$.

Unlike polygon B, polygon A serves as a guide and is not transformed (or affected) by any combination of T_R **310** and T_F **320**. Also, polygon A is the optimum shape for polygon B. On the other hand, Ao **280** and Bo **285** will hold (to retain) the original values of the ordered sets of RLs **250** and RLs **260** for polygons A and B, respectively. This scheme ensures that the values of RLs in the ordered sets defining both polygons A and B can still be restored to default (original) values after applying any one of the combinations of T_R **310**, T_F **320**, and T_s **330**.

The image model Ao **280** has n vertices with maximum RL for polygon A, making Ao **280** the largest n -vertex polygon by area. Alternatively, the n AXL **220** extended

outwardly from Po **200** and intersected Co **210** n times form the set of vertices V_{Ao} **230** of Ao **280**. Likewise, the ordered set of RLs **250** defining Ao **280** is a set composed of “ n ” 1s since every RL is maximum (or 1). Ao **280** is generated by drawing line segments a_{LS} (using a denser line) to connect all two consecutive (succeeding) V_{Ao} **230** as previously defined. Also, Ao **280** is drawn in the display area of the GUI using a lesser dense line, with Po **200** as its centre.

For polygon B: The radial lengths RLs **260** defining all n vertices V_{Bo} **236** of Bo **285** are initially generated by random sampling between 0 and 1. These RLs **260** are set (to be fixed) accordingly by typing the values directly in the corresponding input boxes as provided in the GUI or adjusted to increase or decrease using the combination of scroll bars and spin buttons. Starting with the RL in the first AXL and going counter-clockwise towards the n^{th} RL in the n^{th} AXL, these RLs **260** constitute an ordered set defining Bo **285**.

The coordinates of all the vertices V_{Bo} **236** of Bo **285** are calculated using the corresponding trigonometric function and the previously set RL **260** as the triangle's hypotenuse. Specifically, the x_r - and the y_r - coordinates of the r^{th} vertex V_r **238** of Bo **285** are defined as the product of the r^{th} RL **265** and the cosine or sine function of the angle defining the position of the r^{th} AXL **225**, respectively, such that x_r -coordinate is equal to (r^{th} RL) * $\cos(\theta_r^{\text{th}}$ AXL) and y_r -coordinate is equal to (r^{th} RL) * $\sin(\theta_r^{\text{th}}$ AXL).

The computed values of x_r - and y_r -coordinates can be verified using the identity, $(r^{\text{th}} \text{ RL})^2 = (x_r)^2 + (y_r)^2$. With this identity, it is gleaned that the r^{th} RL **265** corresponds to V_r **238** of the polygon B and vice versa. In symbolic form, $r^{\text{th}} \text{ RL} \leftrightarrow V_r$. This equation means that the value of the r^{th} RL **265** is known from V_r **238**, or vice versa.

The image model Bo **285** is generated by drawing line segments b_{LS} (using a denser line) to connect all two consecutive V_{Bo} **236**. The generated Bo **285** is plotted concentrically with Ao **280**; thus, Bo **285** is also centred at Po **200**.

Besides, it should appear that the corresponding RLs **250** and RLs **260** defining both Ao **280** and Bo **285**, respectively, are concurrent to each other.

The generation of polygon B is achieved using either a manual or automatic mode. In the manual mode, the user can freely adjust, as desired, to either increase or decrease the values of RLs **260**. When the manual mode is enabled, the automatic mode is disabled, or vice versa. A dataset is imported, processed, and then enters a simulation mode in automatic mode.

The system allows a finite dataset containing n independent variables x_i and one dependent variable y to be imported into the system. The system will determine the amount of computer memory to be allocated. The system automatically calculates a regression equation with n standardized coefficients β_i using the standard procedures and formulas for regression analysis from this dataset. The regression equation shows the dependent variable as a function of n independent variables. Such equation has the form $y = \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_i x_i + \dots + \beta_n x_n$, where $1 \leq i \leq n$. The set of standardized coefficients β_i defines the set of RLs **260**. That is, $\beta_i \rightarrow \text{RL}_i$. With the set of RLs **260**, it proceeds to simulation mode. The simulation mode is a state where Bo **285** is being transformed through combinations of T_R **310**, T_F **320**, and T_S **330**. A rank ordering is used in this mode.

For simplicity, since rank is considered, two equal RLs shall have the same rank in the given ordered set (i.e., $\{[1,3,1,2]\}$, $\{[3,2,1,2]\}$, $\{[2,1,1,2]\}$). Every RL **260** is assigned to its rank, either in decreasing (H \rightarrow L) or increasing (L \rightarrow H) order and whichever is applicable, for the given set of RLs **260** defining Bo **285**. Bo \rightarrow bo, where bo represents the set of ranks of every RL **260** in the set of RLs **260** defining Bo **285**. Alternatively, bo corresponds (or is assigned) to Bo **285**. It should be noted that these ranks serve as “dummy values” to simplify the presentation of Bo **285** and should not be construed to change the values of the actual set of RLs **260**.

A list Lo of all arrangements (ordering or permutation) of ranks 1 through n is generated. Any of these arrangements in the generated list Lo can be drawn (or plotted) as a closed polygon corresponding to the image model bo whenever chosen (or selected) from the list Lo . The specific arrangement lo , its order, and the total number 180 of generated arrangements from in list Lo are shown in the GUI.

The export of the combination of Ao **280** and Bo **285** out of the system is done to generate Io **450**. Given Ao **280** and Bo **285**, Io **450** is created by rendering both Ao **280** and Bo **285** (or bo when the rank order is employed) into one and exporting the said Io **450** "outside" (not within) of the system. Also, Io **450** must be appropriately adjusted before its exportation to lessen the required computer memory during the export or import processes and display such images in the GUI. On the other hand, Ao **280** and any transformation applied to Bo **285** (or bo) using a combination of T_R **310**, T_F **320**, and T_S **330** generate a different Io **450** and is assigned a unique filename based on the given combination of transformation. With the rank order, the arrangement of RL ranks serves as the filename of Io **450**. A folder Fo (or computer path) must be specified before exporting Io **450**. This folder Fo contains all the created Io **450** upon export. An option is provided to change the folder Fo into a different computer path (or folder (not shown)). Another option is provided to change the set of ranks bo back (to reset to default) to the ordered set of actual values of RLs **260**, defining Bo **285**. This process is not tedious since an ordered one-to-one correspondence between the sets Bo **285** and bo .

Polygons are classified according to shape and the number of their vertices. The concavity or convexity of the polygon is immaterial. Likewise, T_S **330** is not used for classifying polygons because scaling per se does not essentially change the RL rank arrangement and the shape of any polygon. However, there is an increase (or a decrease) in the area. An option button is provided to start exporting and classifying all Io **450** derived from all

combinations of T_R **310** and T_F **320**. Only those polygons generated from all arrangements of RLs **260** in the set defining Bo **285** are classified. With the rank ordering, Bo **285** is converted to bo .

The process of classifying polygons utilizes rank order. It starts with the generation of the list Lo of all arrangements of RL ranks to comprise the set defining the image model bo . A copy of the list Lo is created and named lo . The first of the arrangements is the default image model of bo . This arrangement is subjected to all combinations of T_R **310** and T_F **320**. A unique Io **450** is created for each applied combination. The resulting Io **450** from the transformation of this arrangement is exported to the default path Fo and classified as one. Each classification is provided with a folder (or computer path) inside Fo and has the RL rank arrangement as the folder name. All arrangements of RL ranks resulting from said transformation are then deleted from the list lo . This process is done not to repeat the same process several times; such process is available only to unique arrangements of vertices, thereby reducing the usage of computer time and memory.

The import of an image Jo to the system is performed. The image Jo is any of the exported Io **450**. The image Io **450** has a file format (or extension filename) of GIF to minimize its size. It should be noted that Io **450** can be exported to create an image having a different file format such as JPEG and BMP instead of what is preferred. However, importation of such a file may involve different conversion processes, require a large computer memory for the operation, or result in some compatibility issues. Suppose there are no issues with importation. The imported Jo is shown in the display area of the GUI with its filename as the arrangement label LBL. This LBL is stored in duplicate, LBL_1 and LBL_2 , to the computer memory: LBL_1 is not altered, while LBL_2 changes due to performing a combination of T_R **310**, T_F **320**, and T_S **330**.

The areas of the generated Ao **280** and Bo **285**, respectively, were computed to obtain a ratio.

These areas are computed using the formula for getting the area of a given the coordinates of the n -vertex polygon. The said area can be evaluated by determinants or the method of matrices. As such, Polygon Area = $\frac{1}{2}|x_1 \ x_2 \ \dots \ x_n \ x_1|$. A $2 \times n$ table (or part of the computer memory) allocated by the system is the matrix to evaluate such a determinant. Products made by slant multiplications, i.e., x_1y_2, x_2y_3, \dots , are gathered to be added or subtracted. Half of the result represents the area of the given polygon.

The ratio (in percentage) between Bo **285** area and Ao **280** is computed. This ratio is reflected in the GUI. The ratio is calculated using the formula, Ratio = (Area of Ao) / (Area of Bo) \times 100%. The system, like a spreadsheet, allocates a range of spreadsheet cells (a part of its memory) to calculate the ratio. This ratio can be interpreted in many ways, depending on the generated images.

The transformation of an image model is achieved by any combination of T_R **310**, T_F **320**, and T_S **330**. In symbolic, the transformation is defined as $T = f(T_R, T_F, T_S)$. A part of the computer memory (or range of spreadsheet cells) is allocated by the system to hold the original values of RLs (or rank) of the set before performing the transformation. It should be noted that the transformation by T_R **310** and T_F **320** is not abelian (the order of the operation matters, and commutativity does not hold); that is, in general, the result of the operation $T_R T_F$ is not necessarily equal to that of the operation $T_F T_R$. In mathematical symbols, generally, $T_R T_F \neq T_F T_R$! Also, when performing any transformation by T_R **310**, T_F **320**, or T_S **330**, the values in the input boxes and with the group of scroll bars and spin buttons change accordingly. Another option button is provided to enable/disable the manual mode.

Specifically, the transformation of an image model by T_R **310** is achieved using shifting, as shown in Figure 2a. Shifting of RLs can be done using any one of two directions: counter-clockwise (CCW) and clockwise (CW) directions. The cycle notations can also be used

to illustrate this process. To demonstrate this process, consider a polygon with four vertices and that the ordered set $bo = \{[1, 2, 3, 4]\}$. Then, with bo , a single T_R CCW is achieved using the operation defined by the cycle notation (1234) or $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 1$ or $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{bmatrix}$. The cycle notation (1234) sends 1 to 2, 2 to 3, 3 to 4, and 4 to 1. Thus, applying a T_R single CCW to bo , (1234) $bo = \{[2, 3, 4, 1]\}$. On the hand, a single T_R CW to be applied to bo is achieved using the operation defined by the cycle notation (1432) or $1 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1$ or $\begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \end{bmatrix}$. The cycle notation (1432) sends 1 to 2, 2 to 3, 3 to 4, and 4 to 1. Thus, applying a single T_R CW to bo , (1432) $bo = \{[4, 1, 2, 3]\}$.

Figure 2b shows that the transformation of an image model by T_F **320** is achieved using a mirroring technique. The mirroring technique works by first defining V_r **238** of the polygon where the T_F **320** is applied. Then, a mirror line M_L **326** is defined to traverse P_o **200** and V_r **238**. By mirror effect, the vertices on the right must appear on the left and vice versa; the vertices above must appear below and vice versa. However, the vertex V_r **238**, where M_L **326** traverses, retains its position.

To obtain I_o **450** after the application of T_F **320**, several rules are employed. Three of these rules were given as:

$$(1) T_{Fn}(i) = T_{Fn}(i + n/2), \text{ for } n = \text{even and } 1 \leq i \leq n$$

$$T_{Fn}(i) \neq T_{Fn}(j), \text{ for } n = \text{odd, } i \neq j, \text{ and } 1 \leq i, j \leq n$$

$$\text{i.e., } T_{F8}(2) = T_{F8}(2 + 8/2) = T_{F8}(6)$$

$$\text{and } T_{F5}(2) \neq T_{F5}(4)$$

$$(2) \text{nop}_n = \begin{cases} \frac{n}{2} - 1 & , \quad n = \text{even} \\ \frac{n-1}{2} & , \quad n = \text{odd} \end{cases} \text{ pairs}$$

$$\text{i.e., } \text{nop}_8 = (8/2) - 1 = 3 \quad \text{nop}_5 = (5 - 1)/2 = 2.$$

$$(3) \text{ With the clock arithmetic, } T_{Fn}(i) = \Pi(i \pm k), \text{ where } 1 \leq i \leq n \text{ and } k \text{ is defined as } 1 \leq k < \frac{n}{2} \text{ for } n = \text{even} \text{ and } 1 \leq k \leq \frac{n-1}{2} \text{ for } n = \text{odd.}$$

Here, $(i \pm k) = (p \ q)$ such that
 $p = \min(i - k, i + k)$ and
 $q = \max(i - k, i + k)$.
 Nevertheless, $(p \ q) = (q \ p)$.

i.e.,

$$\begin{aligned} \mathbf{T}_{F8}(2) &= \Pi(2 \pm k), 1 \leq k < 4 \text{ (since } n = 8) \\ &= (13)(84)(75) = (13)(48)(57) \end{aligned}$$

and

$$\begin{aligned} \mathbf{T}_{F5}(2) &= \Pi(2 \pm k), 1 \leq k \leq 2 \text{ (since } n = 5) \\ &= (13)(54) = (13)(45). \end{aligned}$$

Rule 1 discusses the similarity of the results of $\mathbf{T}_{Fn}(i)$ when applied to the i^{th} and $(i + n/2)^{\text{th}}$ vertices of the polygon when n is even; however, the result of $\mathbf{T}_{Fn}(i)$ is unique when n is odd. The notation $\mathbf{T}_{Fn}(i)$ is written as \mathbf{T}_{Fi} when n is given. Rule 2 shows that the number of cycle pairs to be performed is determined for odd or even n so that none of the pairs will be missed out. Rule 3 identifies the cycles to be performed when performing \mathbf{T}_F 320, with the M_L 326 traversing the i^{th} vertex of the polygon. However, applying the same \mathbf{T}_F 320 twice or for multiples of two retains bo.

The transformation of an image model by \mathbf{T}_S 330 is the simplest of all the three transformation schemes presented. The transformation \mathbf{T}_S 330 is shown in Figure 2c. The operation \mathbf{T}_S 330 shows a change in magnitudes of RLs and neither the shape nor the orientation of the polygon. An input box is provided to accept the scale α of choice. For instance,

$$\text{Polygon A} = \{\mathbf{T}_S(\text{Ao})\} = \{\alpha \text{Ao}\}, 0 < |\alpha| < N$$

$$\begin{aligned} \text{Polygon B} &= \{\mathbf{T}_{R,F,S}(\text{Bo})\} \\ &= \{\alpha \mathbf{T}_{R,F}(\text{Bo})\}, 0 < |\alpha| < N, \end{aligned}$$

where N is not a very large number.

Several options for rendering for export or showing I_o 450 in the display area of the GUI are made available using option buttons. Four of which pertain to the combination to show the (a) x- and y- axes; (b) ranks; (c) actual values of RLs 260; (d) C_o 210 and AXL 220 in the display area of the GUI. Another option is provided where a particular arrangement of RL ranks is manually chosen from the list, L_o . An exit button is provided to close the system.

VII. RESULTS

What is claimed is:

1. A computer-implemented method of polygon generation and transformation comprising the steps of: *using* one or more computer processors of the computer; *opening* a spreadsheet in the computer comprising of macroinstructions wherein said spreadsheet becomes an executable file; *creating* a graphical user interface (GUI) that is output to a computer monitor; *receiving* a manual input from an input device, the input relating to one or more GUI elements of the GUI, the input relating to the number of vertices of the polygon to be generated; *defining* a point serving as the origin or reference of all the elements of the polygon to be generated; *defining* an imaginary-like unit circle serving as the boundary of the polygon to be generated, the circle having centered at said point; *defining* a plurality of axis lines whose plurality is equal to said number of the vertices, the axis line being drawn outwardly from the reference point and intersecting the unit circle; *computing* an angle using the said number of vertices, the angle defining the angular separation between two consecutive axes lines, the said angle defining the angular position of any axis line; *defining* a plurality of radial lengths whose plurality is equal to said number of the vertices, the radial length being measured along the axis line; *defining* a plurality of vertices of the polygon, the vertices being defined by the radial length from the reference point; *defining* a plurality of line segments whose plurality is equal to said number of vertices, the line segment connecting two consecutive vertices to form the polygon; *defining* image models of the polygon with specified number of vertices, the image models being drawn relative to the reference point, the unit circle, the axes lines, and the radial lengths; *generating* a polygon image or group of images from

- said image models, the polygon image being generated by rendering the image models and exported to a computer path not from within the spreadsheet; *importing* the polygon image into the spreadsheet to be displayed in the GUI, said polygon image being any of the rendered images of the image models; *classifying* polygon images according to shape, the polygon images being generated as a result of combinations of transformation; *transforming* polygon images by means of rotation, the rotation being either a clockwise or a counter-clockwise direction; *transforming* polygon images by means of flipping, the flipping having a mirror line passing through the reference point and the reference vertex of the flipping; *transforming* a polygon image by means of scaling, the scale having a positive number that is not very large; *calculating* an area of the resulting image model of the polygon; *calculating* a ratio of the area of the image models, the ratio being displayed in the GUI; *determining* a total number of arrangements of vertices defined by radial lengths, such number being displayed in the GUI; *displaying* any of the generated polygon images in the GUI, such images resulting from any combination of transformation by rotation, flipping, and scaling; and *providing* a plurality of options for displaying and exporting images, the options comprising of the enabling/disabling display of the *xy*-axes, the labels, the graduation measurements, and the combination of the axis lines, the unit circle, and the reference point.
2. The method in claim 1, wherein the spreadsheet can operate either manually or automatically.
 3. The method in claims 1 or 2, wherein the spreadsheet operating in the automatic mode allows the import of a dataset prior to the generation of the polygon image.
 4. The method in any previous claims, wherein a standardized regression equation is obtained from the imported dataset.
 5. The method in any of the previous claims, wherein the coefficients of the standardized regression equation form the set of radial lengths of one of the image models of the polygon.
 6. The method in claims 1 or 2, wherein the spreadsheet utilizes a rank ordering during the automatic mode.
 7. The method in any previous claims, wherein the spreadsheet converts the set of actual values of the radial lengths derived from the standard regression equation into a set of ranks using the rank order.
 8. The method in any of the claims 1 to 3 and 5 to 7, wherein the spreadsheet enters a simulation mode after employing the rank ordering during the automatic mode.
 9. The method in any of the claims 1, 5, and 7, wherein a list and its duplicate are created to contain all arrangements of vertices of the polygon as defined by the radial lengths.
 10. The method in claims 1 or 9, wherein the exported images are classified into a different folder or computer path for every classification.
 11. The method in claims 1 or 2, wherein the set of radial lengths is adjusted to increase or decrease when operating in manual mode.
 12. The method in claims 1 or 9, wherein the flipping of images specifying several rules to reduce the process available only to unique arrangements of vertices comprising the steps of *determining* the arrangements with similar results when performing flipping of images at a given vertex, such arrangements are deleted from the duplicate list; *determining* the number of cycle pairs to perform to predict the amount of processing time needed to complete the flipping process; and *determining* the cycle pairs to be performed to complete the flipping process.

13. The method in claims 1, 9, or 12, wherein the arrangements with similar results when flipping is done, is determined using the formula $T_n(i) = T_n(i + n/2)$ for an even number of vertices.
14. The method in any of the claims 1, 9, and 12, wherein the number of cycle pairs for polygons with an even number of vertices equals one less than half of the number of vertices, the number of cycle pairs for polygons with an odd number of vertices equals half of one less than the number of vertices.

ACKNOWLEDGEMENT

The polygon generation and transformation method has been applied for its patent with application number PH12018000429. Profound gratitude is given to ISAT University President Dr Raul F. Muyong and VP for Research and Extension, Dr Carmelo V. Ambut.

REFERENCES

1. Aloupis, G. (2005). Reconfigurations of Polygonal Structures. Date Accessed: November 21, 2018. <http://cgm.cs.mcgill.ca/~athens/thesis/final-thesis.pdf>
2. Aloupis, G., Bose, P., Demaine, E. D., Langerman, S., Meijer, H., Overmars, M., & Toussaint, G. T. (2011). Computing Signed Permutations of Polygons. *International Journal of Computational Geometry & Applications*, 21(01), 87–100. doi.org/10.1142/S0218195911003561
3. Azad, P., Ude, A., & Cheng, G. (2011). Edge-Tracking Method And Computer Program Thereof (JP2006285358A; JP4836065B2). Japan Patent Office. <https://worldwide.espacenet.com/patent/search/family/037407260/publication/JP4836065B2?q=pn%3DJP4836065B2>
4. Cieplinski, L., & Dabis, H. (2004). Method and apparatus for representing moving objects in a sequence of images (US20040165751A1; EP1418546A1; JP4550398B2; US7400747B2). US, European, and Japan Patent Offices. <https://worldwide.espacenet.com/patent/search/family/032104010/publication/US2004165751A1?q=pn%3DUS2004165751A1>
5. Funahashi, T. (1995). Device for calculating polygon area (JPH07210688A). Japan Patent Office. Worldwide. [espacenet.com/patent/search?q=pn%3DJPH07210688A](https://worldwide.espacenet.com/patent/search?q=pn%3DJPH07210688A)
6. Ghosh, P. (2019). Polygon and its Classification | Parts of a Polygon | Diagonal of a Polygon. Polygon and Its Classification. <https://www.math-only-math.com/polygon-and-its-classification.html>
7. Gielis, J. (2003). A generic geometric transformation that unifies a wide range of natural and abstract shapes. *American journal of botany*, 90(3), 333-338.
8. Han, S., & Kim, J. (2000). Method and apparatus for binary shape encoding (CN1174628C;CN1205598A;DE19739266A1;DE19739266B4;GB2327310A;GB2327310B;IN192097B;JPH1155664A;KR100295798B1;KR19990009714A;US6128041A). China, Germany, Indonesian, Japan, Korean, US Patent Offices. <https://worldwide.espacenet.com/patent/search/family/019514157/publication/US6128041A?q=pn%3DUS6128041A>
9. Isao, Y. (2014). Pattern Data Conversion Method and Pattern Data Conversion Device (JP2011076146A; JP5423286B2). Japan Patent Office. <https://worldwide.espacenet.com/patent/search/family/044020105/publication/JP5423286B2?q=pn%3DJP5423286B2>
10. Ishaque, M., Souvaine, D. L., & Tóth, C. D.(2012). Disjoint Compatible Geometric Matchings. *Discrete & Computational Geometry*, 49(1), 89–131. <https://doi.org/10.1007/s00454-012-9466-9>
11. Kim, J. (1996). Method for encoding a contour of an object in a video signal by using a contour motion estimation technique (US5635986A; CN1124042C; CN1162237A; EP0801504A2; EP0801504A3; EP0801504B1; IN189234B; JP3725250B2; JPH1093972A; KR100203

- 656B1; KR970073155A). US, China, European, Indonesian, Japan, and Korean Patent Offices. <https://worldwide.espacenet.com/patent/search/family/019455335/publication/US5635986A?q=pn%3DUS5635986A>
12. Kim, J. (2007). Method and apparatus for adaptively coding a contour of an object (CN1147156C;CN1196642A;DE69738016T2;EP0871331A2;EP0871331A3;EP0871331B1;JP3924048B2;JPH10290466A;KR100229544B1;KR19980076589A;US5929917A). China, Germany, European, Japan, Korean, US Patent Offices. <https://worldwide.espacenet.com/patent/search/family/019502525/publication/EP0871331B1?q=pn%3DEP0871331B1>
 13. Liao, C., & CHAN, K. (2015). Electronic device and video object tracking method thereof (US20150138345A1; TWI570666B). US and Taiwan Patent Offices. worldwide.espacenet.com/patent/search/family/053172903/publication/US2015138345A1?q=pn%3DUS2015138345A1
 14. Masukura, K., Hori, O., Kaneko, T., Mita, T., & Yamamoto, K. (2000). Object region data generating method, object region data generating apparatus, approximation polygon generating method, and approximation polygon generating apparatus (US20010025298 A1). US Patent Office. <https://appft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&p=1&u=/netahtml/PTO/srchnum.html&r=1&f=G&l=50&d=PG01&s1=20010025298.PG NR>.
 15. Peterson, D. (2022, March 11). Fine Points on Polygons and Polyhedra. Fine Points on Polygons and Polyhedra. Retrieved April 15, 2022, from <https://www.themathdoctors.org/fine-points-on-polygons-and-polyhedra/>
 16. Saito, T., & Yokohama-Shi. (2002). Polygon Generating Apparatus and Drawing System (US2002/0167514 A1). U.S. Patent Office. <https://patentimages.storage.googleapis.com/bb/b4/0e/23eeec982235a2/US20020167514A1.pdf>
 17. Vartziotis, D., & Merger, J. (2018). GETMe. anis: On geometric polygon transformations leading to anisotropy. arXiv preprint arXiv:1805.01767.