Visual Traverse: an Open Source Python Program for Traverse Computation

Yanli Zhang, Daniel Unger and Matthew McBroom, USA

Key words: compass rule, Python, survey, traverse computation

SUMMARY

Polygons are widely used to define the extent of spatial features and field data collection (surveying) is often used to establish polygon line segments. However, every measurement has error, and thus the polygon will have misclosure (angular and linear). It is necessary to assess the precision and adjust the misclosure. A Python (widely used open source and cross platform programming language) program, Visual Traverse, was developed with a graphical user interface (GUI) to cover the entire computational procedure: data entry, precision assessment, angular misclosure adjustment, and linear misclosure adjustment followed by coordinate computations and then area computation. The program can draw the corrected polygon at an appropriate scale for visualization purpose. Visual Traverse is a useful open source and independent tool to facilitate traverse computation not only for the academic and research community, but also for GIS users and surveyors. At the same time, the open source nature promises the further development of the program by the user community.

Visual Traverse: an Open Source Python Program for Traverse Computation

Yanli Zhang, Daniel Unger and Matthew McBroom, U.S.A

1. INTRODUCTION

Polygons are closed areas bounded by line segments and are widely used to define the extent of spatial feature such as forest stands, property boundaries, and research plots. One basic method to establish a polygon is field data collection (traversing in surveying), and each line that forms the polygon boundary is measured directly or indirectly for its distance (length) and angle (direction). Historically, pacing, chaining, or taping have been used for distance measurement and compass or transit were used for angle/direction measurement. When high level precision is needed or required, total station, which combines EDM (Electronic Distance Measurement) technology and electronic angular measurement component is typically used today. Currently, the most advanced total station, such as Leica ICON robot 50, can reach precision level as 1 mm ± 1.5 ppm (part per million) for length measurement and 2 seconds for angle measurement. It has been confirmed that a traditional traverse with high end instrument provides better results in densifying project control between GNSS (Global Navigation Satellite Systems) control points [Sass 2013, Ghilani 2022]. In other words, traverse is still an important method in collecting polygons or polylines in practice.

It is well known that all measurements contain errors, no matter how carefully the measurements are made [Ghilani 2022]. This is because there are many factors that impact the precision of measurements including wind, temperature, humidity, atmospheric pressure, gravity, instrument factors and limitations of the human senses of sight. For a surveyed polygon, two major measurement problems exist: angular misclosure and linear misclosure. After field data collection, traverse precision is evaluated using a non-statistical inverse ratio of linear error of closure to total traverse length. Government precision/accuracy standards require certain minimum traverse closures and precisions. U.S. Federal Geodetic Control Committee [1984] has specific recommendations for five different orders of traverse precision. The higher the order, the less the permissible misclosure value is. For example, the total interior angle of a rectangular is 360º, and the 1st and 5th order permissible angular misclosures are 3.4 seconds and 24 seconds, respectively, With respect to linear misclosure, each state has its own precision requirement. Texas Society of Professional Surveyors [2006] published allowable linear misclosure for a boundary survey in rural and urban area as 1:3,500 (misclosure equal to or less than a meter for every 3,500 meters surveyed) and 1:7,500, respectively. If the field measurements meet the federal and state requirements, it is also required by law to perform traverse computations to make the property boundary polygon mathematically correct, in other words, a truly closed geometric polygon.

There are many professional civil engineering or GIS software packages available to process surveyed polygon data, such as Carlson Survey®, Traverse PC®, and Civil 3D (AutoCAD)®. However, they are expensive and the learning curve can be challenging. An online traverse computing platform was developed by Zhou et al. in 2012. Odumosu et al. [2014] and Serwa [2016] all used Visual Basic 6.0 to develop traverse computational tools. However, Visual Basic often changes and potential users' operating system may not support the tool. For academic teaching and demonstration purpose, Xie et al. [2016] used Excel to work on traverse computation. So far, there is not a traverse computation tool developed with an open source programming language.

As Lakhani and von Hippel [2000] pointed out, open source allows software to be further developed by users, and user can help each other to build the program. There are many open source programs that have been developed so professionally that are used worldwide, such as statistical software R and GIS software QGIS. However, as we know there is not an open source program for land surveying. We hope to start an open source program for the community. Visual Traverse was developed with Python, a widely used open source programming language. Data entry is easy with its user-friendly GUI (graphical user interface). It covers precision assessment, angular misclosure adjustment, linear misclosure adjustment, area computation, and draws a polygon with appropriate scale to facilitate visualization.

2. METHODS AND SOFTWARE DEVELOPMENT

To demonstrate polygon misclosure problem, student collected field data (Figure 1) are used. There are four stations (vertices), and thus four interior angles and four distances were measured. The survey was performed clockwise from station 1 to station 4. The line from station 1 to station 2 was used as the reference line to determine direction of other lines, and its bearing was precisely surveyed in the field as N25°30′40″E. The enlarged area is to demonstrate polygon misclosure. In a GIS environment, snapping (force point to snap to the other one within a certain distance tolerance) is usually used to close the polygon; however, it is not the best way to correct the polygon as it directly changes the last line's direction and length. In other words, last line segment bears all the errors that happened throughout the whole survey procedure. Polygon misclosure can be divided into angular misclosure and linear misclosure, and they need to be analyzed and adjusted systematically. The procedure is called traverse computation and one common misclosure adjustment method, compass rule, is used in the current software.

2.1 Visual Traverse development

Traverse computation is relatively complicated for beginners. To facilitate the procedure, Visual Traverse was developed with Python. As an open source programming language, Python is widely used in many areas, such as web and internet development, scientific computing, education, desktop GUI (graphical user interface) design, software development, and e-commerce system [python.org]. Python has thousands of libraries (modules) to facilitate programmers for different applications. For Visual Traverse, Python modules have been used are turtle (graphics), re (regular expression operations), math, and tkinter (interface design). The software was structured with many functions, and some functions rely on other basic functions. Figure 2 is the screen capture of the GUI. On the left side, there is a canvas to draw and visualize the polygon after traverse computation, where the top part for the right side is for data entry and the bottom part of the right side is a report area for traverse computation outputs.

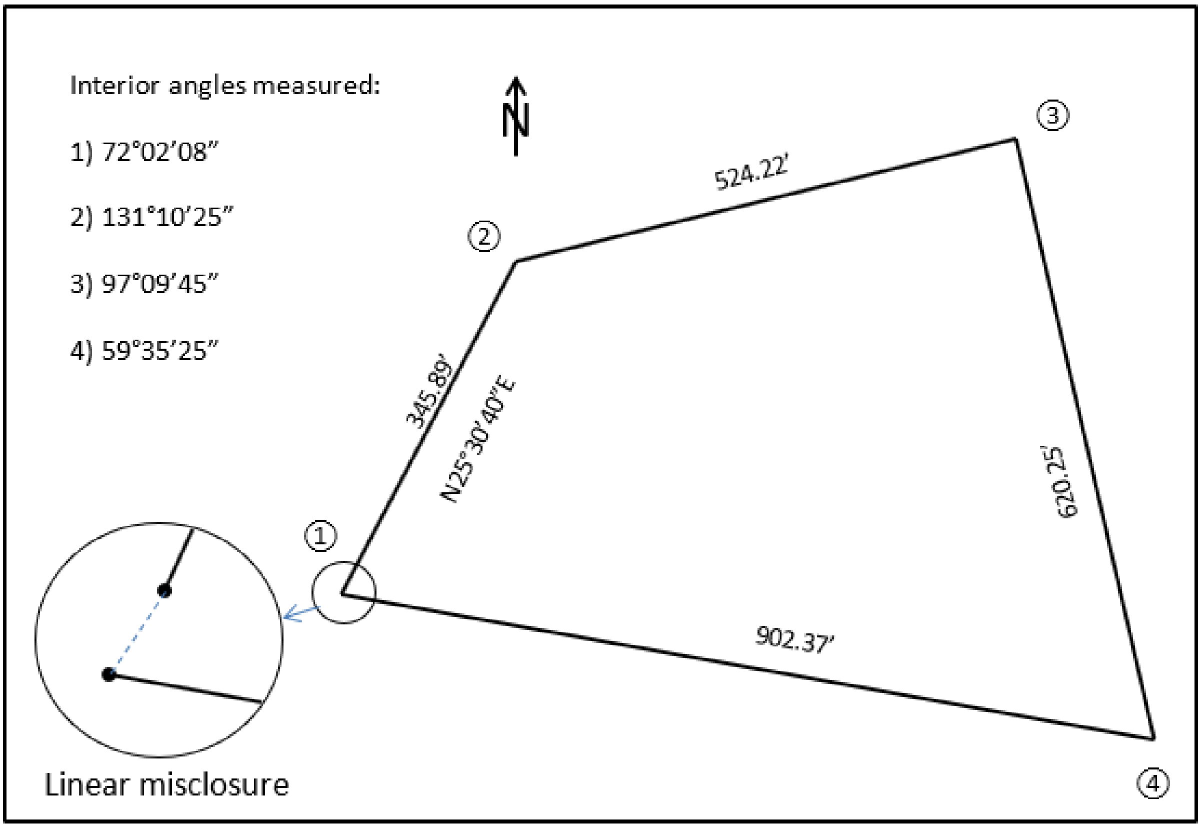


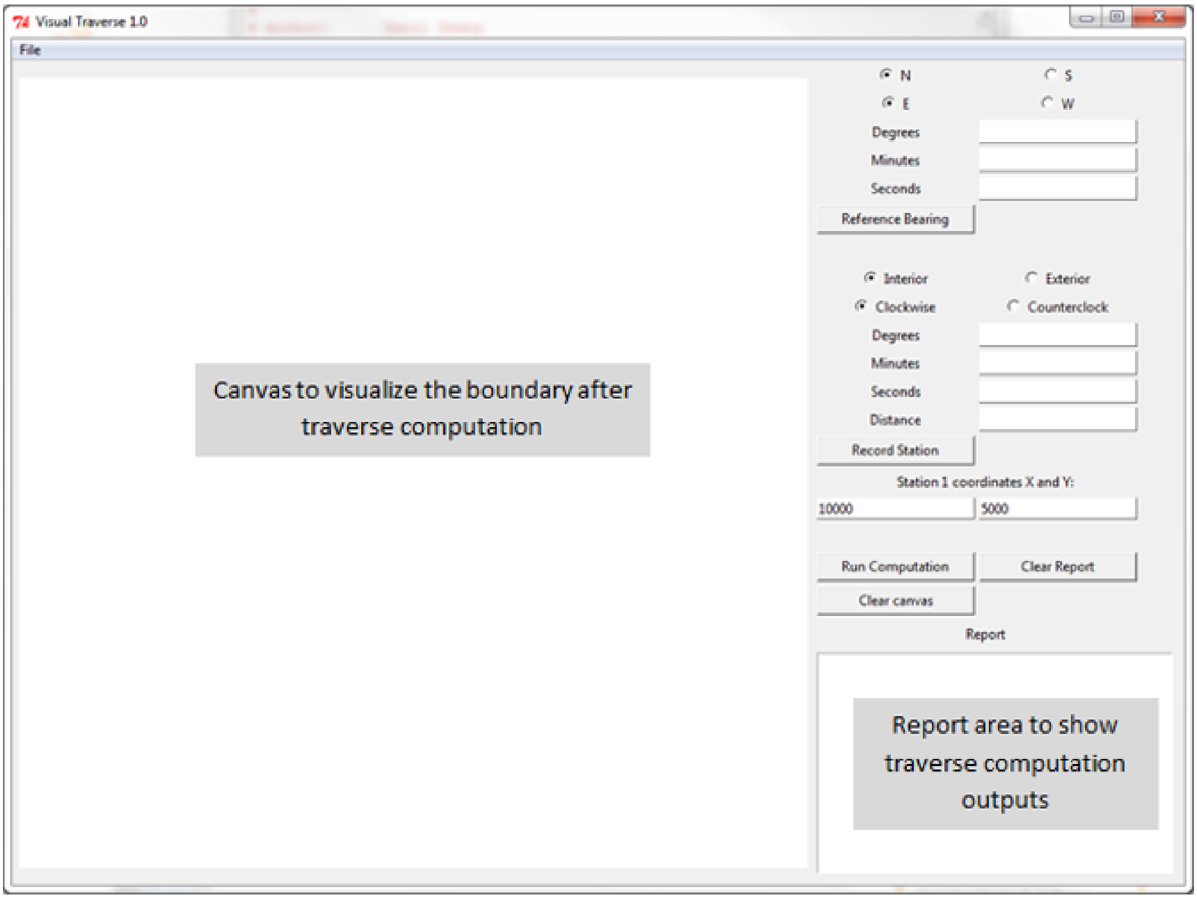
Figure 1. Sample survey data (not in real scale to represent free hand drawing in the field) and the enlarged area is to demonstrate misclosure.

The program has two formats, one is in original Python file format (i.e., .py file) and the other is an executable exe file which is more convenient for non-programmers. Python is required to run the software. For GIS users, computers with ArcGIS Desktop or ArcGIS Pro have Python installed in default setting. For users without ArcGIS, Python package is available free at https://www.Python.org. When Python is available, users can right click the file in File Explorer, choose "open" to run the program, or choose "Edit with IDLE" to view the code and run the program.

2.2 How to use Visual Traverse

Data shown in Figure 1 are used to demonstrate the use of Visual Traverse and the steps are:

1. Input the reference line’s bearing. Pick up N and E, type in degrees, minutes and seconds, respectively, then click the Reference Bearing button to record the information in the Report panel.
2. Select Interior for the angles measured at each station.
3. Select Clockwise for station sequence.
4. Record each station’s angle and forward distance. For example, for station 1, the interior angle is 72°02’08” and the distance is 345.89 units. After typing in one station’s information, just click the Record Station button to save the inputs. Type in all the stations in sequence.
5. Set up starting X and Y coordinates. The default values for X and Y are 10,000 and 5,000, respectively. Users can have different values, such as GNSS receiver collected state plane coordinates (SPC) to compute SPCs for all other stations.
6. Click the Run Computation button to perform traverse computation. All the results will be shown in the Report panel.



● N ○ S

● E ○ W

● Interior ○ Exterior

● Clockwise ○ Counterclock

Station 1 coordinates X and Y: 10000 5000

Canvas to visualize the boundary after traverse computation

Report area to show traverse computation outputs

Figure 2. Graphical user interface (GUI) of Visual Traverse.

The method to calculate directions of each line covered by Ghilani [2022] and Kavanagh [2010] was used in Visual Traverse. As sines and cosines of azimuth provide correct algebraic sign that facilitates further calculation, azimuths are used for all courses. The flow chart of Figure 3 explains the logic procedure executed within Visual Traverse to calculate azimuth for each line.

One thing worth mentioning is that degree, minutes, and seconds are the common way to represent angles in land survey. However, in Python, they need to be converted to decimal degrees. Further, decimal degrees will be converted to radians to be used in sine and cosine function. The example code would be *math.cos(math.radians(azimuth))* in Python. Traverse computation covers angle misclosure adjustment and linear misclosure adjustment. Also coordinates for each station and total area are calculated. The unit for area is dependent on the input unit. If meters are used, then the area is in square meters. If foot is used, then the area is in square feet.

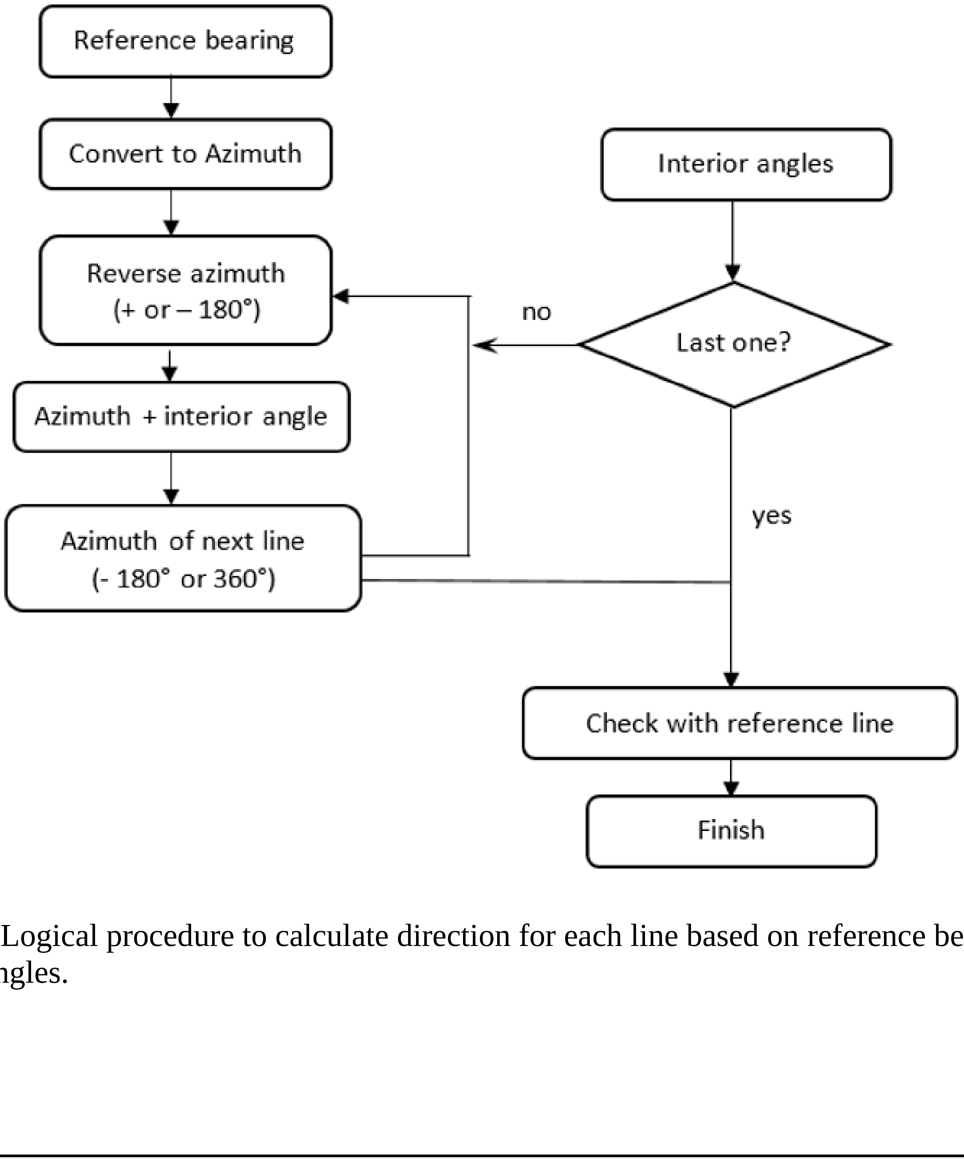
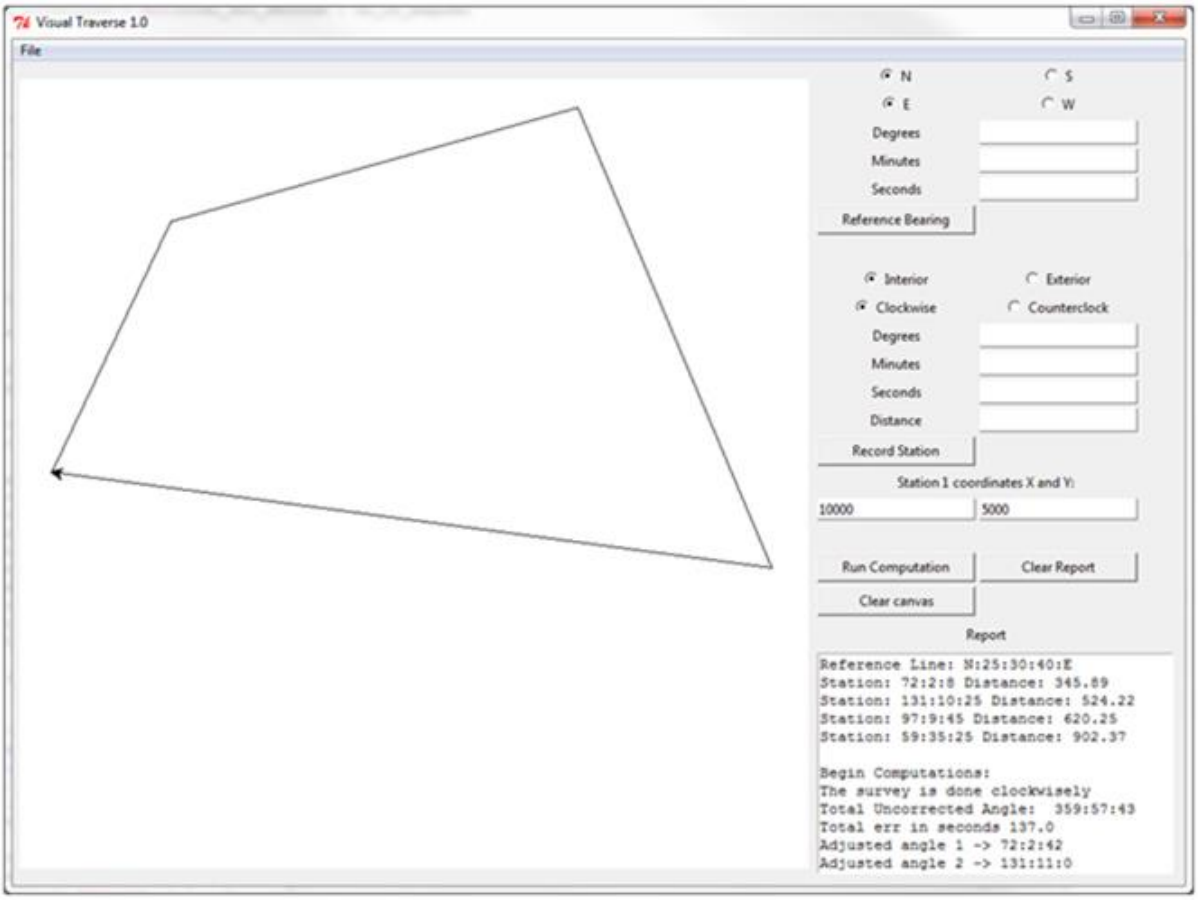


Figure 3. Logical procedure to calculate direction for each line based on reference bearing and interior angles.

2.3 Visual Traverse outputs

After traverse computation, the program is ready to draw the corrected (adjusted) polygon boundary in the canvas area. By using the mouse to click anywhere (the click point is the first station or vertex) on the canvas, the boundary will be drawn automatically at a scale that fits the canvas area (700 by 700 dots) (Figure 4). If the polygon is not well located on the canvas, such as one station is out of the canvas, users can click the Clear Canvas button to erase the drawing, and then redefine the first station to have the polygon drawn again. The clear report button is used to clear all saved information for current traverse computation. Users can copy the outputs of traverse computation and the polygon into other applications, such as Microsoft Word or PowerPoint.



● N ○ S
● E ○ W
● Interior ○ Exterior
● Clockwise ○ Counterclock

Station 1 coordinates X and Y: 10000 5000

```
Reference Line: N:25:30:40:E
Station: 72:2:8 Distance: 345.89
Station: 131:10:25 Distance: 524.22
Station: 97:9:45 Distance: 620.25
Station: 59:35:25 Distance: 902.37

Begin Computations:
The survey is done clockwisely
Total Uncorrected Angle:  359:57:43
Total err in seconds 137.0
Adjusted angle 1 -> 72:2:42
Adjusted angle 2 -> 131:11:0
```

Figure 4. Screen capture of traverse computation outputs for sample input data.

3. DISCUSSION AND FUTURE WORK

The outputs of Visual Traverse have been verified with manual calculation and related examples from the textbook of “Elementary Survey” [Ghilani 2022] have tested with the software, resulted same accurate values. For the example data giving in Figure 1, the relative precision is 1/2864 and the area of the land surveyed is 28,773.8 m^2. Coordinates of points and line distances and directions are shown in Table 1.

Table 1. Traverse computation outputs for sample input data.

STATION	X	Y	ADJUSTED DISTANCE	ADJUSTED BEARING
1	10,000	5,000		
			345.7833567	N25°31’14”E
2	10,148.97523	5,312.04568		
			524.1782438	N74°20’50”E
3	10,653.71413	5453.47230		
			620.4530302	S22°50’14”E
4	10,894.52071	4,881.65576		
			902.3151704	N82°27’49”W
1	10,000	5,000		

Visual Traverse is useful for academic, teaching, and professional applications in the areas such as surveying, forestry, civil engineering, and GIS. Its open source nature also provides users the opportunity to view equations used and its programming algorithms, thus to customize the program for their specific purposes. Currently, Visual Traverse is available at http://www.faculty.sfasu.edu/zhangy2/download.htm (right click and choose “Save Link As”. Python code may not open correctly in web browser).

Python language was selected for two considerations. First, Python is commonly thought as a script language and not very good for GUI applications by GIS user community. Visual Traverse is a successful application example to demonstrate GUI development with Python. Secondly, similar functions can be developed as an extension or add-in to other packages, such as ArcGIS®. However, extensions are usually not compatible when the package moves to new versions. With Python, Visual Traverse is a standalone and cross platform program, and users do not need to worry about software compatibility or version change.

Future work should cover topics like curve calculation, using least square adjustment method, leveling procedures, and more graphic functions like zoom and pan, thus to make it a package more like a commercial one. As it is said, we hope the open source nature would allow its future development by user community.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge an anonymous senior programmer’s significant contribution and the research is supported by USDA McIntire-Stennis fund.

REFERENCES

FGCS (Federal Geodetic Control Subcommittee), 1984. FGCS Standards and Specifications for Geodetic Control Networks, [Online]. Available: https://www.ngs.noaa.gov/FGCS/tech_pub/1984-stds-specs-geodetic-control-networks.pdf [Accessed 2/15/2023].

Ghilani D.C. Elementary Surveying, an Introduction to Geomatics, 16th Edition, Pearson, 2022.

Kavanagh B.F. 2010. Surveying with construction applications. 7th edition, Pearson.

Lakhani K. and von Hippel E. 2000. How open source software works: "Free" user-to-user assistance. Working Paper 4117, MIT Sloan School of Management, Cambridge, MA.

Odumosu, J.O.; Ajayi, O.G.; Ibrahim, P.; Okorocha, V.C. and Idowu, F.F. 2014. Development of an Object Oriented Program for Traverse Computation. International Journal of Scientific Engineering and Technology, 3(7): 967-973.

Python.org. Applications for Python, [Online], https://www.python.org/about/apps/, [Accessed 2/15/2023].

Sass, J., 2013. GNSS or Total Station? Selecting the right tool for the job. Machine Control magazine, 3(1): 46-48.

Serwa, A. and El-Semary, H.H. 2016. Development of soft computational simulator for traversing. Surveying and Land Information Science, 75(1): 7-16.

Texas Society of Professional Surveyors (TSPS), 2006. Manual of Practice for land surveying in the state of Texas.

Xie, X.; Fang, W.; Tian L.; Fan Z.; Luo F. 2016. Design of teaching system for surveying traverse calculation based on EXCEL VSTO. Mine Surveying, 4, 25.

Zhou, Q.S., Yang, J.L., Cao, X.G., Si, H.Y. and Liu, Y. 2012. Design of Surveying and Mapping Network Computing Platform. Applied Mechanics and Materials, 170: 2827-2830

BIOGRAPHICAL NOTES

Dr. Zhang, Dr. Unger and Dr. McBroom are faculty at Arthur Temple College of Forestry and Agriculture at Stephen F. Austin State University in Nacogdoches, TX of USA.

CONTACTS

Yanli Zhang
Stephen F. Austin State University
419 E. College St.
Nacogdoches, TX
U.S.A
Tel. +1-936-468-2157
Email: zhangy2@sfasu.edu
Web site: http://www.faculty.sfasu.edu/zhangy2/download.htm